Towards A Secure Joint Cloud With Confidential Computing

Xuyang Zhao, Mingyu Li, Erhu Feng, Yubin Xia
Institude of Parallel and Distributed Systems (IPADS)
Shanghai Jiao Tong University
Shanghai, China

2022 IEEE International Conference on Joint Cloud Computing (JCC) | 978-1-6654-6285-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/JCC56315.2022.00019

***Abstract*—As data security in public clouds attracts more attention and concerns, researchers and practitioners have proposed techniques to secure cloud computing. Confidential computing (CC) is a compelling approach that guarantees both privacy and integrity of data and code in public clouds.**

In this paper, we first survey the status of CC in today's commercialized public clouds, including the cloud CC abstractions, infrastructures, metrics, third-party service vendors, and real-world cloud use cases. We also discover the limitations such as re-programming efforts, extra cost, limited availability, etc. We further take a step forward to prospect CC in the joint cloud scenario. We finally showcase the challenges of realizing a secure joint cloud and propose possible solutions.

***Index Terms*—confidential computing, jointcloud, survey**

I. Introduction

Moving services to the cloud has been a promising approach for companies to deploy and manage their business. While cloud computing brings benefits like ease of management, high performance, and scalability, it also raises new security concerns. Attackers who gain control of the server or curious cloud admins could threaten valuable data in the cloud. In recent years, more cloud security breaches have been reported [1]–[3], exposing the private data of millions of users. Regulations such as GDPR [4] and HIPPA [5] are proposed to protect individual's privacy. These breaches accidents can be devastating to a company's both finance and reputation.

To secure cloud computing, technologies such as Fully Homomorphic Encryption (FHE) [6], Multi-party Computation (MPC) [7], Zero-Knowledge Proof (ZKP) [8], and Verifiable Computing (VC) [9] are proposed. However, most of these techniques are designed for specific cases and fail to provide general functionalities or suffer from high performance overhead. Confidential Computing (CC) is an alternative that can protect data with high-security guarantees, general-purpose computation, and relatively low overhead. Moreover, many public cloud vendors have offered CC products, such as Microsoft Azure [10], Google Cloud [11], Alibaba Cloud [12], Amazon Web Service [13], and IBM Cloud [14], making CC a compelling approach for cloud protection. With these products, tenants can have more control over their sensitive data, regardless of the complex cloud software stacks.

Jointcloud computing, by integrating multiple clouds for collaborative computing, harnesses economical and flexibility advantages and avoids vendor lock-in. Jointcloud computing needs CC to remove trust from specific cloud vendors. To understand how to build a secure joint cloud, we survey existing CC technologies, cloud CC products and cloud CC services. Through this survey, we seek to answer the following questions.

What products and services are provided by today's CC-enabled clouds? We surveyed the state-of-the-art CC products and infrastructures that public cloud vendors offer for building privacy-preserving cloud applications. We find that *cloud vendors provide various CC abstractions; even for similar CC infrastructures, cloud vendors may have different assumptions and approaches*. We then measured CC products' pricing and availability and find that *CC resources are more expensive and less available*. We further collected third-party service providers and typical CC-based applications to show the trend of CC. We find that *CC is popular for augmenting the security of AI/ML, blockchains, and databases*. We hope our findings can shed light on how to take a step towards a secure joint cloud.

What challenges will a CC-oriented joint cloud face? On the one hand, jointcloud has the same security issues faced by one single cloud, including malicious attackers and curious admins. Furthermore, jointcloud computing brings other security concerns, e.g., how to build trust amongst clouds. On the other hand, jointcloud should be able to support running CC applications across clouds, with full transparency to cloud users ideally. However, realizing a secure joint cloud platform can be challenging due to non-uniform CC abstractions and attestation techniques, inconsistent security guarantees, etc. We also give possible solutions to bridge the gap.

What opportunities are there when achieving a secure joint cloud? We provide our vision on exploiting the potential of future jointcloud platforms. As an important part of today's clouds, CC can benefit from jointcloud by harvesting more CC resources for cost-saving, enabling seamless task migration for low-latency services, and yet more exploration spaces that deserve innovative research.

We hope our work can motivate more researchers and practitioners to join in building a secure joint cloud. We believe a CC-based joint cloud will be emerging across future clouds.

The rest of the work is organized as follows. section II introduces the definition of CC and summarizes de facto CC technologies. section III surveys CC products and services provided by cloud vendors, third-party service providers, and

978-1-6654-6285-3/22/$31.00 ©2022 IEEE
DOI 10.1109/JCC56315.2022.00019

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF EXISTING CONFIDENTIAL COMPUTING SOLUTIONS

	SGXv1	Scalable SGXv2	SEV-SNP	TDX	TrustZone	Realm	Nitro	Penglai	Keystone	H100
Architecture	x86-64	x86-64	x86-64	x86-64	Arm	Arm	x86-64	RISC-V	RISC-V	GPU
Abstraction	enclave	enclave	VM	VM	PM	VM	VM	enclave	PM	vGPU
Instances	unlimited	unlimited	509	unlimited	1	unlimited	unlimited	unlimited	16	7
Encryption	●	●	●	●	○	●	○	●	○	●
Integrity	●	◐	◐	◐	○	◐	○	●	○	●
Freshness	●	○	○	○	○	○	○	●	○	○
Attestation	●	●	●	●	○	●	●	◐	◐	●

* PM stands for the physical machine abstraction. Integrity means this CC can resist both hardware and software tampering; ◐ for integrity means this CC can detect software tampering. H100 has full integrity against hardware attacks because it uses on-chip High Bandwidth Memory (HBM). AMD EPYC (Rome) processors currently support 509 keys for SEV VMs. Nitro uses TPM for remote attestation. Penglai and Keystone currently only support local attestation, but can also achieve remote attestation using TPM or other methods alike.

use cases. section IV discusses the challenges and methods of achieving a secure jointcloud. section V concludes.

II. NOTION OF CONFIDENTIAL COMPUTING

A. What is Confidential Computing (CC)

According to Confidential Computing Consortium [15], CC protects data and code in a hardware-based trusted execution environment (TEE). To achieve **integrity**, CC partitions physical memory and guarantee that only authorized entities can access specific memory regions. For **confidentiality**, CC uses hardware-enhanced memory encryption engines to prevent attackers from probing memory contents. However, in the cloud, to reduce cost and hardware dependency, cloud providers, like Amazon, use virtualization for strict isolation. Customers utilizing this TEE should trust the privileged hypervisor. An indispensable component of CC is **remote attestation** that provides verifiable evidence for the authenticity of the underlying hardware and the current execution state. With remote attestation, cloud customers can verify that their security-sensitive code runs in a genuine TEE.

B. Existing confidential computing technologies

Intel Software Guard Extensions (SGX) [16]. SGX protects userspace regions *enclaves* in a process. SGX enclaves must rely on OS for scheduling, memory management, I/O, etc., but OS cannot access the enclave's memory. Enclave memory is encrypted and integrity-protected by a dedicated hardware engine. The state of enclaves can be remotely attested using either Intel Provisioning Service [17] or cloud vendor-managed Data Center Attestation Primitives [18]. Intel also released scalable SGX [19], which extends the memory limit at the cost of weakened integrity protection.

AMD Secure Encrypted Virtualization (SEV) [20]. AMD SEV is designed to encrypt VM memory against the untrusted hypervisor. The initial version of SEV has severe security pitfalls which fail to resist encrypted memory replay [21] and VM state disclosure [22]. SEV thereafter releases two variants: SEV-ES [23] encrypts VM registers upon VM exits; SEV-SNP [24] uses a protected Reverse Map Table (RMP) from software replay and memory remapping. SEV also provides remote attestation.

Intel Trust Domain Extensions (TDX) [25]. Intel TDX allows deployment of VM-level isolated execution environments called trusted domains (TDs). TDs are isolated from the traditional hypervisor and other co-located tenants on the non-TD side of the same machine. At a high level, TDX is very similar to existing Virtual Machine Extensions (VMX), but with memory, encryption using MKTME technology, and architectural remote attestation support using the TDX-SEAM firmware module. To minimize the complexity of remote attestation, Intel TDX reuses SGX remote attestation as its building block.

ARM Realm Management Extension (RME) [26]. ARMv9 introduces RME as part of ARM Confidential Compute Architecture (CCA). A realm consists of EL0 user and EL1 kernel, providing a secure VM abstraction. CCA requires a realm manager, factually a trusted hypervisor, to manage realm resources. CCA divides physical memory into four worlds (i.e., root, normal, secure, realm) and uses a fine-grained page table called Granule Page Table (GPT) in the root world. GPT enforces memory access control on hypervisor and OS page table translations. RME also provides remote attestation support and memory encryption protection.

AWS Nitro [27]. Nitro relies on a dedicated trusted hypervisor to isolate vCPU and memory of Nitro Enclave instances, denying access from the host, other enclaves, and host admins. Nitro Enclave has only one entry/exit dubbed vsock, which automatically encrypts all network traffic between the Nitro instance and the external. Nitro Enclaves have no persistent storage, interactive access, or external networking. Nitro also supports attestation, which reuses AWS Key Management Service to provide built-in attestation.

Keystone [28]. Keystone is an open-source enclave project for exploring customizable trusted execution environments based on RISC-V architecture. To achieve so, Keystone introduces a programmable layer underneath untrusted components. The current Keystone relies on RISC-v PMP for memory isolation and builds attestable enclaves isolated from the host OS.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

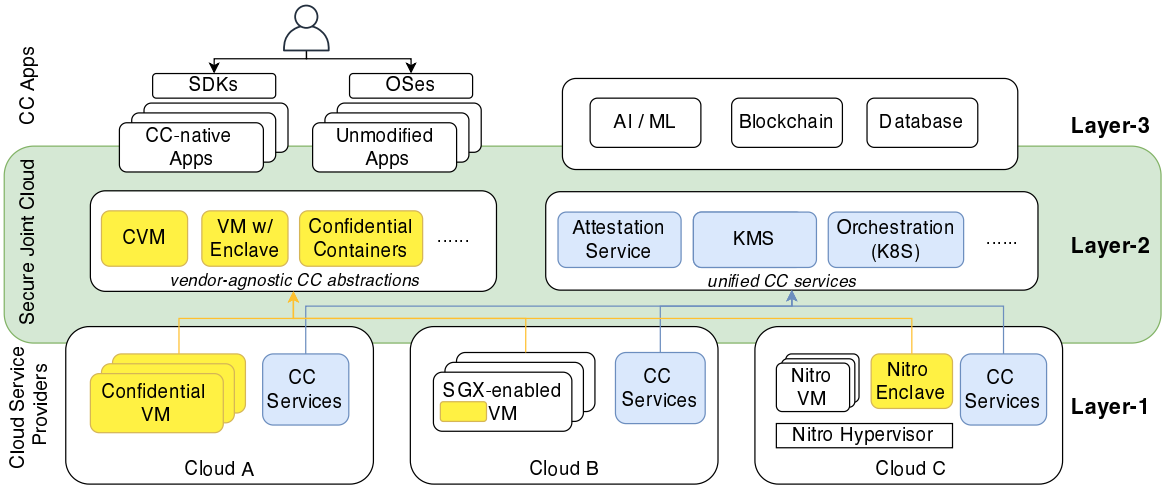

Fig. 1. Overview of the secure joint cloud. Layer-1 consists of CC abstractions and CC services provided by today's clouds. Layer-2 is the secure joint cloud layer that provides unified abstractions and services and hides the cross-cloud differences. Layer-3 runs CC Apps based on the secure joint cloud.

Penglai [29]. Penglai is a scalable CC on RISC-V with three versions. Penglai-TVM supports running more than 1000 enclave instances in a single device. Penglai-MPU provides the isolation ability to run an unmodified OS in an isolated execution environment. Penglai-MCU is intended for embedded devices without MMU support. Penglai can be deemed a competitive candidate for both edge and cloud computing, protecting emerging applications such as artificial intelligence and autonomous with high-security assurance.

NVIDIA H100 GPU [30]. NVIDIA Hopper H100 Tensor Core GPU is the first heterogeneous accelerator that supports confidential computing. H100 protects users' sensitive data and proprietary AI workloads using fully isolated GPU instances with strictly partitioned resources. GPU confidential computing is an important extension for heterogeneous confidential computing resources, expanding the security boundary to more than CPUs. We believe H100 brings new opportunities and also adds complexity to cloud security.

Table I summarizes existing confidential computing technologies and compares their main features.

III. Confidential Computing of Today's Clouds

This section details the CC products, services, and applications offered by both cloud and third-party service providers, the bottom layer of Figure 1. We argue that the rich CC ecosystem has potential for building a secure jointcloud, and we present our observations that support the argument.

A. Abstraction of Cloud CC

Bare-metal server with CC support. Bare-metal server, or dedicated servers, is the simplest way to provide CC in the clouds. Almost every cloud vendor provides this type of product to tenants [31], offering full control over the hardware. Generally speaking, a bare-metal server improves security by avoiding co-tenants and side channels. HETEE [32] is an example that allocates computation resources at rack-scale using a programmable PCIe, creating bare-metal heterogeneous CC platforms. However, using bare-metal servers contradicts the cloud's benefits, such as scalability, high utilization, and ease of maintenance. These challenges can be resolved by elastic bare-metal technologies [33].

Confidential VM. AMD SEV, Intel TDX, and ARM Realm provide VM-level protection, preventing access to VM states from outside. With AMD SEV, the only VM-level CC currently available, cloud service providers like Microsoft Azure and Google Cloud offer confidential VM (CVM) products. Existing applications can be migrated to CVMs without making any changes. Cloud vendors should support attestation of CVMs to prove the integrity of the VM is not tampered with.

VM with SGX enclave. Both Microsoft Azure and Alibaba Cloud provide virtual machine instances with Intel SGX support, including SGXv1 and SGXv2. To use the power of enclaves, customers should partition their applications into trusted and untrusted components with special SDKs, and secure the trusted part with SGX enclaves.

VM with Nitro enclave. Amazon Nitro Enclaves [27] is a virtualization-based enclave product. It comes with most Amazon EC2 instances, because of the low assumption of the underlying hardware. A nitro enclave is a separate, hardened, and highly constrained VM, accessible only through a local socket. Developers can package enclave codes into a docker image and transform it into a enclave image with nitro CLI tools, which are also used to run the enclave image.

Confidential containers. Unlike the previous IaaS products, cloud vendors also provide confidential containers as a cloud-native abstraction. Cloud-native applications are usually built and deployed as containers for portability and simple management. A container is a modular component, making it an

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

TABLE II
THE AVAILABILITY AND PRICING OF CC PRODUCTS

	Pricing w/o CC ($)	Pricing w/ CC ($)	Excess Ratio	Regional Availabil.
IBM Bare Metal CC	287	300	4.5%	All
Azure Confidential VM*	82.5	128.5	55%	2 / 41
Azure VM with SGX*	92	140	52%	10 / 41
Google Confidential VM	66	82.5	25%	69 / 100
Alibaba VM with SGX*	56	59	5.4%	4 / 27
AWS Nitro Enclave	72.5	72.5	0.0%	All

* means that Confidential Containers have the same pricing as that of VMs. For pricing, We use the monthly cost. For regional availability, X / Y denotes that services are available in only x out of y regions.

TABLE III
ATTESTATION REPORT COMPONENTS OF CC PRODUCTS

	App	OS	VMM	Firmware
IBM Bare Metal CC	✓			
Azure Confidential VM	✓	✓		✓
Azure VM with SGX	✓			
Google Confidential VM	✓	✓		
Alibaba VM with SGX	✓			
AWS Nitro Enclave	✓	✓	*	

1. SEV-SNP report includes firmware, but Google CVM currenly does not include the firmware. 2. AWS Nitro uses TPM to generate the enclave report, whose components does not contain VMM, a part of the TCB.

appropriate abstraction to provide security protection.

To protect containerized applications, cloud vendors use previously mentioned confidential VMs or SGX-enabled VMs. The difference is that customers no longer need to manage CC resources, leaving the work to management frameworks like Kubernetes. On a confidential VM, containers can run without modification. On VM with SGX, there are two programming models. The first is to decouple an application into two parts: trusted and untrusted. The second is to deploy the applications inside confidential containers without code change. Azure, Alibaba, and IBM favor the latter. For runtime-based languages such as Python/Java, existing container images can be converted to confidential containers with wrapping tools. For languages like C/C++/Rust, recompilation is needed. Alibaba Cloud's Inclavare Containers designs a special container runtime *rune* for running confidential containers that are recompiled with dedicated toolchains.

Observation I

Cloud vendors provide various CC abstractions. CC apps using one abstraction may not be easily adapted to another, resulting in potential vendor lock-in.

B. Metrics of Cloud CC Products

Pricing. An important factor of cloud services is their cost. We survey the pricing of different products [34]–[36] in current clouds to help understand the cost of CC. While cloud providers offer various billing methods, we compare the monthly cost of servers with and without CC support as shown in Table II. CC is usually priced higher because it relies on special hardware features and additional support from the vendor. Google, for example, charges an extra $4 per vcpu and $0.536 per GB-memory every month for their SEV-based CVM. Azure's CC-enabled ECasv5 series cost 55% higher than normal Easv5 VMs. AWS Nitro Enclave, on the contrary, incurs no additional cost, due to low assumptions for hardware.

Availablity. We survey the regional availability of CC services in different clouds. Most CC solutions rely on special hardware feature to guarantee security. However, these high-end servers do not always exist in every data center. Even in a data center that contains CC, not all computation nodes support it. For example, among 41 regions of Azure worldwide [37], only 2 of them support SEV-protected CVM and 10 of them supports SGX. Lack of regional availability can constrain the deployment of CC applications.

Observation II

CC resources are more expensive and less available than ordinary compute resources.

C. Cloud CC Infrastructures

Remote attestation. Remote attestation (RA) helps a remote entity validate the identity of CC hardware and software states. With RA, cloud users can ensure their services are tamper-proof. Every component within the TCB should be attested to ensure integrity.

We classify the current remote attestation techniques into two major categories:

1) *Built-in RA*: Built-in RA uses the inherent attestation capability of CC technologies. For example, SEV-SNP uses a secure processor AMD-SP for the guest VM attestation. Azure takes this approach for its SEV-based CVMs [38]. For Intel SGX, enclaves can generate a report with the EREPORT instruction to prove their integrity. A dedicated quoting enclave will verify the report locally and generate a quote to be remotely attested.
2) *Plug-in RA*: Trusted platform module (TPM) is widely used for the secure boot of VMs. It can be integrated into the motherboard to measure existing systems. For example, Google Cloud [39] utilizes vTPM to generate the measurement for SEV-based CVMs.

Table III shows the components contained in attestation reports of different CC products. Note that Google uses TPM instead of built-in AMD-SP to attest its CVMs.

Key management service. To secure cloud applications, critical secrets like cryptographic keys, passwords, and attestation reports should be protected with additional care. Hence, most clouds provide a key management service (KMS) to control access to customers' keys and perform cryptographic operations upon them. Usually, keys are safeguarded with hardware security modules (HSMs), and external devices that can be connected to systems for managing keys. KMS can be integrated with other cloud services that need a secure key for encryption or digital signing. Some services, such as Azure Key Vaults, also provide software solutions for managing other resources like attestation reports, passwords, and certificates.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
CLOUD CC INFRASTRUCTURES

Vendor	SEV	SGXv1	SGXv2	Nitro	K8S	RA	KMS
IBM	—	✓	—	—	✓	✓	✓
Google	✓	—	—	—	✓	✓	✓
Amazon	—	—	—	✓	—	✓	✓
Alibaba	—	✓	✓	—	✓	✓	✓
Azure	✓	✓	✓	—	✓	✓	✓

K8S: Kubernetes. RA: Remote Attestation. KMS: Key Management Service. 1. IBM only provides bare-metal servers for SGXv1 while others provide both bare-metal and VM instances. 2. Google protects K8S nodes using SEV while other clouds protect K8S at containers granularity with SGX.

Orchestration and scheduling. Kubernetes (K8S) is the standard way to manage and orchestrate cloud containers. Likewise, containers hardened with CC techniques also need scheduling tools. However, cloud vendors provide different levels of K8S integration. For enclave-aware containers, IBM Cloud uses its bard-metal servers as worker nodes for K8S. Azure and Alibaba additionally support CC-aware scheduling policies. Azure's AKS, for example, takes enclave memory as another resource type to be scheduled. For unmodified containers, Google Cloud supports K8S orchestration on SEV-enabled CVMs. AWS Nitro enclave is not a container abstraction. Currently, running nitro enclaves directly under K8S control is not supported. However, AWS suggests managing Nitro enclaves from within containers [40] to enable K8S.

Observation III

Cloud vendors offer similar CC infrastructures but with different assumptions and approaches.

Development tools. There are two major categories for developers to build CC-based applications.

1) *SDK*: Intel SGX has a variety of SDKs [41]–[44] with rich language support such as C/C++/Go/Rust. For Nitro Enclave, AWS provides SDK [45] for developing secure applications. SDK-based development needs manual efforts but usually has a small TCB.
2) *Secure (Lib)OS*: For SGX, Occlum [46] and Gramine [47] are two popular library OSes. For SEV, Kata [48] supports running VMs with SEV protection as containers. Similarly, Intel TDX shim [49] supports running existing VM images within CVM. Secure OS-based development can contain legacy unmodified applications with the price of a relatively large TCB.

D. Third-party CC Service Vendors

While cloud vendors provide products and basic infrastructures for CC, there are third-party service providers also aiming at simplifying the use of CC.

Scontain [50]. Scontain commercializes the SCONE container [51] that provides services of confidential containers which host programs inside Intel SGX. Scontain provides services tailored to confidential containers, such as Kubernetes integration, attestation and password management, online monitoring, etc. With Scontain, one can automatically transform an unmodified container-based application into a confidential application. Scontain containers can run on heterogeneous clusters, i.e., nodes with different versions of SGX support, adjusting the behavior depending on the CPU type.

Fortanix [52]. Fortanix explores the possibility of multi-cloud data security. Fortanix's Data Security Manager enables tokenization of sensitive data, substituting them with random strings as *tokens* to guarantee consistent data protection across clouds. The service also provides unified KMS and policy management in the multi-cloud scenario based on HSM. Fortanix's Confidential Computing Manager targets enabling existing applications to run with enclave protection. It also provides management of the lifecycle and enforcement of security policies. Fortanix offers a Confidential AI platform to accelerate AI deployment with CC technologies, protecting AI models for inference and private data for training.

Anjuna [53]. Anjuna offers a single, uniform confidential computing platform that protects data in execution. This relieves agencies from the burdens of managing diverse encryption schemes for each application and system, leading to complexity and potential confusion. With the lift-and-shift technique, Anjuna can transparently secure existing cloud applications and deploy them with available CC products in AWS Nitro, Azure with Intel SGX, and AMD SEV. A typical use case is to leverage Anjuna CC to protect API keys, passwords, and certificates [54].

Opaque [55]. Opaque enables different entities or organizations to analyze confidential data collaboratively while still keeping the valuable data and the analytical results private to each party. Using Opaque, users can execute SQL queries, analytics jobs, and AI/ML models using standard notebooks and analytical tools, while the platform guarantees security and scalability. The platform also supports remote attestation to verify cluster deployments.

Edgeless [43]. Edgeless provisions an end-to-end secure K8S service called Constellation, based on cloud CVMs. MarbleRun [56] is its control plane that manages inter-container communication and secures data sharing within a cluster. It injects *marbles* into enclaves for secure enclave-to-enclave TLS connections. During initialization, MarbleRun will verify the integrity of *marbles* with remote attestation and check that the containers' topology is consistent with the cluster manifest file. Apart from Constellation, Edgeless maintains a secure database EdgelessDB [57] and a Golang SGX SDK named EGo [58].

Observation IV

CC companies bloom due to the large market of cloud CC, offering products of diversity and fractions.

E. Cloud CC Use Cases

Machine learning. For the convenience of deployment and collaboration, cloud services for data analytics, such as MLaaS, have been growing rapidly in recent years [59]. Using CC methods, confidential AI platforms form the basis

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

of realizing confidential ML. Confidential ONNX Inference Server [60] is a confidential port of the ONNX inference server based on Open Enclave SDK [42]. It's an open-sourced project backed by Azure that provides data encryption and attestation capabilities for inference. Other cloud vendors and software providers also support confidential ML by running existed ML frameworks inside CC [50], [52]. Industries have port Apache TVM [61] and Tensorflow Lite [62] into CC frameworks. Research efforts have been made to hide side-channel [63] and leverage GPU resource [64]. For inference, BlindAI [65] and MesaTEE [66] provide Rust-based fast confidential AI inference services.

Blockchain. Today's blockchains suffer from zero data privacy, long latency, low transaction throughput, etc. The Phala Network [67] is a blockchain-based confidential computing system. It runs the smart contract engine inside enclaves and records the command with a blockchain. Moving the computation off-chain with CC techniques improves the computation throughput and latency on the current smart contract. Teechain [68], a layer-two payment network, enables off-chain asynchronous transactions with SGX. In-enclave engines can efficiently execute transactions without interacting with the global blockchain, thus improving performance drastically. Azure Confidential ledger [69] is a decentralized ledger that provides tamperproof storage backed by blockchain. The ledger may serve as trustworthy storage for audit logs and other data at risk of forgery. With CC, the ledger can run inside enclaves, thus keeping cloud providers out of the TCB, avoiding the high latency and other issues with the traditional distributed blockchain systems.

Database. Cloud database stores massive user data and processes SQL queries and transactions with high performance. To protect outsourced data at rest, cloud databases may transparently encrypt files in page granularity before they are written to disk and decrypt after loading into memory. This transparent data encryption technique does not prevent privileged attackers who can inspect memory during runtime. The encrypted database aims at protecting data in use; most practical designs involve confidential computing. Azure AE [70] is a privacy-preserving database based on Azure SQL servers. It ensures sensitive data against database administrators, cloud operators, and other high-privileged users. AEv1 leverages cryptography for equality operations, while AEv2 uses Intel SGX to provide rich functionality including comparison and string pattern matching. EdgelessDB [57] is an open-source confidential database based on MariaDB and runs this in-memory DB engine in an enclave as a whole. It uses enhanced RocksDB as the storage engine and encrypts database files when writing to disk, providing confidentiality, integrity, freshness, audit ability, and recoverability for data.

Observation V

Cloud CC is popular for hot applications such as AI/ML, blockchains, databases, etc.

IV. Opportunities of Secure Joint Clouds

A. Why does joint cloud need confidential computing?

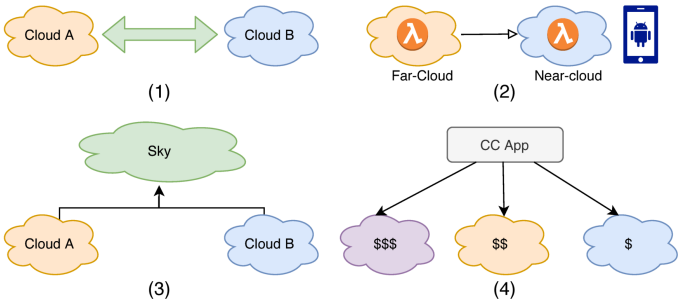

Fig. 2. Use cases of cloud CC.

We use four scenarios to demonstrate how cloud CC can benefit from jointcloud (as depicted in Figure 2):

1) **Multi-party secure computation:** Suppose two parties, A and B, both of which have already established a privacy-preserving data analytic workload. However, A and B have to exchange both data and tasks for further collaborations. At this point, a federated secure computation connecting multi-cloud is wanted.
2) **Near-edge secure computation:** Suppose a latency-sensitive workload concerning user-privacy should be carried out near the mobile clients. Such workloads include chatbot, face recognition, personal recommendation, etc. Hence migrating CC workloads between cloud and edge is important [71]. Ideally, the workload can even be migrated inter-cloud if other cloud vendors can provide resource proximity for the clients.
3) **Sky secure computation:** Sky computing [72] allows the creation of large-scale infrastructures atop multi-cloud resources. Since CC resources are a typical scarce resource on clouds, when a cloud service provider runs out of its hardware CC resources, its requests can then be redirected to other cloud service providers with equivalent security guarantees. Given the ubiquitous CC available over the public clouds world spread, it is desirable to construct a sky CC computing against vendor-lock-in.
4) **Cost-saving secure computation:** A user may wish to lower the overall cost for large-data privacy computation. Indeed, clouds offer CC resources with different pricing models and security levels. It would be interesting to distribute the computation over the multiple clouds and hence provide users with affordable computation, as long as the clients know how to measure the data values and how to dispatch the dataset carefully.

B. What are obstacles towards a secure joint cloud?

Challenge-1: lack of unified programming models. Confidential computation (CC) is independently proposed by hardware vendors. Recently open-sourced RISC-V added more variety of CC platforms. Table V shows the features of these

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

CC programming models. It is a non-trivial task for cloud developers to deal with such a rich CC backend.

TABLE V
PROGRAMMING MODELS OF CLOUD CC

Platforms	Programming Model
SGX, Penglai	User-level enclaves
SEV, TDX, Realm	Confidential virtual machines
TrustZone, Keystone	Isolated Physical-machine level domains

Solution-1: using automation toolchains to hide details. OpenEnclave (OE) [42] is an industrial project led by Microsoft, providing a uniform SDK that bridges SGX and TrustZone, by requiring developers to build a secure part and a non-secure part for a program. But OE may not fit other CC platforms. We expect a new toolchain similar to LLVM, which abstracts all the architectural details and generates new programs adaptive to new CC backends.

Challenge-2: lack of allied remote attestation mechanisms. Existing RA mechanisms are too tightly coupled with hardware vendors. For example, SGX uses an opaque attestation whose details are not public and therefore infeasible to review. Worse, the format of remote attestation is diverse. For example, the SGX report only includes the software in the enclave boundary, whilst the TDX report includes the SEAM module, TD kernel, and userspace applications. A developer must understand how a report is generated, using a particular toolchain to include all related components. On different platforms, even the same applications reflect other measurements. When an application is updated with a new patch, the measurement is translated into another. The CC ecosystem may require trustworthy measurement management.

Solution-2: using federated parties to open source attestation. Ongoing efforts are made for open attestation infrastructure. OPERA [73] provides loosely-coupled attestation to Intel by introducing attestation proxies to reduce the attestation latency, and ensure the property of anonymity. Veraison [74] aims to prove the identity of CCA realms. Veraison is designed for open governance and collaboration, and also focuses on matching supply chain to verification operations against backdoors. We believe an online measurement service hosted by several federated parties is also desired.

Challenge-3: lack of identical security guarantees. CC also varies in threat models and security guarantees. For example, Nitro Enclave requires trusting the cloud hypervisor, whereas hardware-based CC (e.g., SGX, SEV, TDX) distrusts any cloud software. SGX, SEV, and TDX have different TCB sizes and expose different attack surfaces. A long line of vulnerabilities [75]–[81] have been revealed against Intel SGX CPUs, some of which can be mitigated via microcode update while some require hardware microarchitectural internal reimplementations. SEV also faces the same side-channel issues [82]–[84].

Solution-3: building "measurable" security and partition into different zones. A key challenge is how we can compare the security guarantees of different technologies. A possible solution is to partition the dataset into different levels and places into different zones while seeking a good balance for the overall performance-security trade-offs.

Challenge-4: lack of support for cloud-native computation models like serverless. The cloud computing paradigm moves towards serverless or function-as-a-service (FaaS). FaaS benefits users with on-demand instantiations according to request rates and charges users with used resources. It remains an open problem to adapt existing commercial CCs to meet serverless requirements, such as low-latency startup, high-density instantiations, and inexpensive data transfers.

Solution-4: building new frameworks to meet new requirements. Efforts are spent in supporting cloud FaaS using CC. Apache Teaclave [44] is an open-source FaaS platform using Intel SGX. PIE [85] proposes a plugin abstraction for Intel SGX for fast enclave function startups. Penglai [29] can scale to support at least 1,000 enclave instances with integrity and freshness. It is unclear how to build serverless frameworks on other CC platforms and whether there will be common issues. For example, supporting more than 7 tenants on an H100 GPU will save more hardware costs. We believe this will open up a new era of active research.

C. How to achieve a secure joint cloud?

Jointcloud computing is a promising infrastructure that can benefit rich privacy-preserving applications with more use-case spaces, as mentioned above IV-A. Below, we discuss how future cloud vendors can collaborate to shape a secure joint cloud, for the profits of both cloud vendors and customers. We believe the following features are essential.

Multi-cloud attestation. We expect a flexible and scalable attestation service for all kinds of CC resources deployed on multiple clouds. One approach would be to introduce dedicated proxy nodes, Attestation Management Service (AMS), which attest to other nodes on behalf of clients. AMS can be designed in a decentralized style using a replicated state machine to avoid a single point of failure. AMS should provide a uniform attestation abstraction layer for any CC resource accesses because attestation is the first vital step.

An interesting usage is that attestation forms a chain of trust by combining multiple hops of CC nodes. With the chain of trust, AMS does not need to attest to every node it communicates with. In short, AMS can ease the attestation efforts amongst clouds. For deployment, AMS requires at least an AMS node in each cloud cluster.

Cross-cloud key sharing. Cryptographic keys are extremely critical assets to the whole CC infrastructure. Unfortunately, today's clouds each have their own Key Management Service (KMS). Key sharing is a must to allow confidential data to flow between clouds. Otherwise, data must be re-encrypted, which costs massive CPU for big data scenarios. An intuitive approach is to bridge KMS nodes amongst cloud vendors and enable key transfer along with the secret data. The trust of the KMS network can be established based on AMS.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

However, sending keys to other vendors may pose new attack surfaces when clouds are mutually distrustful. A curious cloud admin might abuse the key to breach secrets. To maintain the control of the secret data, cloud-scale KMS should be carefully designed. For example, to comply with GDPR [4], KMS should embed a time-to-live (TTL) for each key. Once a key has expired, the key recipient must once again acquire access to the key for further operations.

Joint-cloud verification. In a distributed setting, both data flow and control flow should be tracked to verify the final results. A long line of prior work has managed to offer input-output tracking as the data flow inherently obeys causality, for big data [86], distributed settings [87], or even decentralized ones [88]. In particular, [86] statically predefines the control flow, [87] applies the information flow control, [88] embeds fine-grained metadata along with the dataflow, and [89] proposes a lightweight verifiable protocol.

Thanks to the inherent verification feature of CC, we believe the above mechanisms or protocols can cater to a joint-cloud verification, allowing users to verify the overall behavior. A cloud-scale audit trail can be integrated with jointcloud verification to enable visibility of the current states.

Inter-cloud migration. In some cases, migrating a running instance from cloud A to cloud B can be necessary. For example, near-edge computation aims to reduce the latency of requests, hence migrating the instance to wherever the client locates desirable. Another example would be migrating a very long-term running instance to harvest cost-efficient resources.

Inter-cloud migration for confidential workloads is non-trivial regarding CC hardware specifications, service specifications, resource capacity, and dependency states. Gu et al. [90] only consider in-memory states for SGX instances; Alder et al. [91] further consider other external persistent states such as counters and disks. Table VI gives the summary.

New CC vendors are active in developing live migration for VM-level CCs [92], [93]. A key feature of the industrial approach is to leverage in-guest UEFI as the migration helper. As inter-cloud may provision heterogeneous CC supports, migration between TDX and SEV can be potentially valuable.

TABLE VI
INTER-CLOUD MIGRATION AND SUPPORTED STATES

Work	Migrated States
Gu et al. [90]	In-memory encrypted states
Alder et al. [91]	Counters + encrypted storage
Secure jointcloud	Memory, storage, counters, keys

Federated-cloud scaling. CC hardware is a constrained resource for clouds. To avoid CC resource shortage, federated-cloud scaling would be a promising feature, where high-utilized clouds can rent resources from other clouds. To realize such a feature, several steps must be taken:

1) *Resource accounting:* cloud A should be able to understand the resource statistics of other clouds so that the scaled confidential jobs will not be stragglers. An attractive feature is that CC provides remote attestation so that the statistics can be accounted [94]. This distinctive ability prevents a "malicious" cloud from violating the QoS of other clouds to damage their reputations.
2) *Mutual attestation:* before rerouting requests to other cloud service providers, cloud A should be able to attest to other clouds resources and see if they are adequate to afford the desired workloads. Cloud A should measure the overall security level and partition the tasks carefully. With the AMS we proposed above, the latency of mutual attestation can be greatly reduced.
3) *State relaying*: a snapshot in the confidential abstraction should be taken for state continuity before relaying across clouds. CC should handle replay [95] and rollback [96]. A straightforward approach is to embed an epoch for each executor in the pipeline to resist replay and rollback [87].

After the above steps, scaling confidential instances are similar to existing scaling techniques. How to accomplish efficient cross-cloud consistency for all instances remains open, but with CC, $3f + 1$ can turn to $2f + 1$ for speedup.

Heterogeneous-cloud abstraction. To address the problem of cloud vendor lock-in, users are eager for a uniform abstraction that hides the details of heterogeneous cloud services, such as programming models, service APIs, etc. CC is no exception but increases complexity, such as abstractions, remote attestation, cross-ISA, etc. All these put a burden on the developers.

Both academia and industry have made efforts for a uniform CC model. Enclavisor [97] adds a shim layer atop SEV to mimic SGX enclaves with better performance. vSGX [98] has a similar design that unify SGX and SEV models. Microsoft OpenEnclave [42] develops an identical SDK for SGX and TrustZone. Alibaba HyperEnclave [99] is a cross-platform model that reuses SEV to build enclave-level CC. The uniform CC model does ease the programming efforts but is far from enough to build a secure joint cloud.

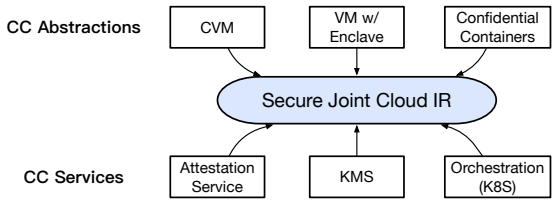

Fig. 3. Secure jointcloud may need an intermediate representation (IR) for cloud-agnostic application compilation.

The missing building block is a uniform CC service model. Manually porting applications to fit different cloud services APIs is cumbersome. We expect all can be automated by a toolchain similar to the gg framework [100]. gg uses the intermediate representation (IR) to decouple application SDKs as frontends and cloud APIs as backends. We can borrow the idea of gg IR to translate existing CC programs to cloud-agnostic ones, as depicted in Figure 3. Yet, a new runtime is required to deal with instantiation, dependencies, and failures.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

We believe such an automation toolchain for cloud CC will be a key enabler for future secure jointcloud ecosystems.

V. Conclusion

Confidential computing (CC) has already become an important pillar for today's clouds. We survey and summarize the progress of CC products, CC-related services, and applications in the cloud. We show that CC is gaining popularity in the cloud. We believe a unified, easy-to-use confidential computing service will be emerging across future clouds. We hope this paper to be the first step to motivate future studies that help shape together a better secure joint cloud.

References

[1] the daily swig. latest cloud security news. https://portswigger.net/daily-swig/cloud-security.

[2] 7 most infamous cloud security breaches. https://www.arcserve.com/blog/7-most-infamous-cloud-security-breaches.

[3] cloud security breaches (and lessons). https://www.cybertalk.org/2022/04/26/top-5-cloud-security-breaches-and-lessons/.

[4] General data protection regulation. https://gdpr-info.eu/.

[5] Health insurance portability and accountability act. https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act.

[6] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09*, 2009.

[7] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *J. Priv. Confidentiality*, 1, 2008.

[8] Eduardo Morais, Tommy Koens, Cees van Wijk, and Aleksei Koren. A survey on zero knowledge range proofs and applications. *ArXiv*, abs/1907.06381, 2019.

[9] Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. *2013 IEEE Symposium on Security and Privacy*, pages 238–252, 2013.

[10] Azure confidential computing. https://azure.microsoft.com/en-us/solutions/confidential-compute/.

[11] Google Cloud confidential computing. https://cloud.google.com/confidential-computing.

[12] alibaba cloud. https://www.alibabacloud.com/.

[13] AWS Nitro Enclaves. https://aws.amazon.com/cn/ec2/nitro/nitro-enclaves/.

[14] Confidential computing on IBM Cloud. https://www.ibm.com/cloud/confidential-computing.

[15] Confidential computing consortium. https://confidentialcomputing.io/, Feb 2021.

[16] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan, Vinay Phegade, and Juan del Cuvillo. Using innovative instructions to create trustworthy software solutions. In *HASP*, 2013.

[17] Build an intel® software guard extensions ecdsa attestation service to strengthen enclave security. https://api.portal.trustedservices.intel.com/provisioning-certification, 2022.

[18] Intel(r) software guard extensions data center attestation primitives. https://github.com/intel/SGXDataCenterAttestationPrimitivesml, 2022.

[19] 3rd gen intel® xeon® scalable processors. https://www.intel.com/content/www/us/en/products/docs/processors/xeon/3rd-gen-xeon-scalable-processors-brief.html, 2020.

[20] Amd secure encrypted virtualization (sev). https://developer.amd.com/sev/, 2019.

[21] Yuming Wu, Yutao Liu, Ruifeng Liu, Haibo Chen, Binyu Zang, and Haibing Guan. Comprehensive vm protection against untrusted hypervisor through retrofitted amd memory encryption. *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 441–453, 2018.

[22] Mengyuan Li, Yinqian Zhang, Zhiqiang Lin, and Yan Solihin. Exploiting unprotected I/O operations in amd's secure encrypted virtualization. In *Proceedings of the USENIX Security Symposium*, pages 1257–1272. USENIX Association, 2019.

[23] Protecting vm register state with sev-es. https://www.amd.com/system/files/TechDocs/Protecting%20VM%20Register%20State%20with%20SEV-ES.pdf, 2017.

[24] Amd sev-snp. https://www.amd.com/system/files/TechDocs/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf, 2020.

[25] Intel trusted domain extensions. https://www.intel.com/content/dam/develop/external/us/en/documents/tdx-whitepaper-final9-17.pdf, 2020.

[26] Arm developer: Realm management extension. https://developer.arm.com/documentation/den0126/latest, 2021.

[27] Aws nitro enclaves. https://aws.amazon.com/cn/ec2/nitro/nitro-enclaves/, 2021.

[28] Dayeol Lee, David Kohlbrenner, Shweta Shinde, Krste Asanovic, and Dawn Song. Keystone: an open framework for architecting trusted execution environments. In *Proceedings of the ACM European Conference on Computer Systems (EuroSys)*, 2020.

[29] Erhu Feng, Xu Lu, Dong Du, Bicheng Yang, Xueqiang Jiang, Yubin Xia, Binyu Zang, and Haibo Chen. Scalable memory protection in the PENGLAI enclave. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 275–294. USENIX Association, 2021.

[30] Nvidia confidential computing. https://www.nvidia.com/en-sg/data-center/solutions/confidential-computing/.

[31] Data-in-use protection on IBM Cloud using Intel SGX. https://www.ibm.com/cloud/blog/data-use-protection-ibm-cloud-using-intel-sgx.

[32] Jianping Zhu, Rui Hou, Xiaofeng Wang, Wenhao Wang, Jiangfeng Cao, Boyan Zhao, Zhongpu Wang, Yuhui Zhang, Jiameng Ying, Lixin Zhang, and Dan Meng. Enabling rack-scale confidential computing using heterogeneous trusted execution environment. *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1450–1465, 2020.

[33] Introducing the sixth generation of alibaba cloud's elastic compute service. https://www.alibabacloud.com/blog/introducing-the-sixth-generation-of-alibaba-clouds-elastic-compute-service_595716.

[34] Google VM instance pricing. https://cloud.google.com/compute/vm-instance-pricing#pricing.

[35] Alibaba Pricing calculator. https://www.aliyun.com/price/product#/commodity/vm.

[36] Azure Pricing calculator. https://azure.microsoft.com/en-us/pricing/calculator/.

[37] Azure Products available by region. https://azure.microsoft.com/en-us/global-infrastructure/services/?products=virtual-machines.

[38] Azure Attestation. https://docs.microsoft.com/en-us/azure/attestation/.

[39] Validating instances using Cloud Monitoring. https://cloud.google.com/compute/confidential-vm/docs/monitoring.

[40] AWS Nitro Enclaves on EKS. https://superorbital.io/journal/aws-nitro-enclaves-in-k8s-pods/.

[41] Intel® software guard extensions sdk for linux. https://01.org/intel-softwareguard-extensions.

[42] Open Enclave SDK. openenclave.io/sdk/.

[43] Edgeless Systems. https://www.edgeless.systems/.

[44] Apache teaclave (incubating) is an open source universal secure computing platform, making computation on privacy-sensitive data safe and simple. https://teaclave.apache.org/.

[45] nitro-sdk. https://github.com/aws/aws-nitro-enclaves-sdk-c.

[46] Occlum. https://occlum.io/.

[47] Gramine - a Library OS for Unmodified Applications. https://gramineproject.io/.

[48] Kata Containers. https://github.com/kata-containers/kata-containers.

[49] Intel shim tdx. https://github.com/intel/shim-tdx.

[50] scontain. https://scontain.com/.

[51] Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Dan O'Keeffe, Mark Stillwell, David Goltzsche, D. Eyers, Rüdiger Kapitza, Peter R. Pietzuch, and Christof Fetzer. Scone: Secure linux containers with intel sgx. In *OSDI*, 2016.

[52] Fortanix. https://www.fortanix.com/.

[53] Anjuna. https://www.anjuna.io/.

[54] Securing secrets management against insider threats. https://www.anjuna.io/hashicorp.

[55] The Opaque Platform. https://opaque.co/.

[56] MarbleRun. https://www.edgeless.systems/products/marblerun/.

[57] EdgelessDB . https://www.edgeless.systems/products/edgelessdb/.

[58] Ego is a framework for building confidential apps in go. https://github.com/edgelesssys/ego.

[59] Sagar Sharma and Keke Chen. Confidential machine learning on untrusted platforms: A survey. *Cybersecur.*, 4:30, 2021.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.

[60] Confidential ONNX Inference Server. https://github.com/microsoft/onnx-server-openenclave.

[61] Open deep learning compiler stack. https://github.com/apache/tvm/tree/main/apps/sgx.

[62] Tensorflow lite for intel sgx. https://github.com/Jumpst3r/tensorflow-lite-sgx.

[63] Olga Ohrimenko, Felix Schuster, Cedric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious Multi-Party machine learning on trusted processors. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 619–636, Austin, TX, August 2016. USENIX Association.

[64] Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[65] Blindai-fast, accessible and privacy friendly ai deployment. https://github.com/mithril-security/blindai.

[66] Mesatee gbdt-rs. https://github.com/mesalock-linux/gbdt-rs.

[67] The phala network blockchain, pruntime and the bridge. https://github.com/Phala-Network/phala-blockchain.

[68] Joshua Lind, Oded Naor, Ittay Eyal, Florian Kelbert, Emin Gün Sirer, and Peter R. Pietzuch. Teechain: a secure payment network with asynchronous blockchain access. *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, 2019.

[69] Azure confidential ledger. https://azure.microsoft.com/en-us/services/azure-confidential-ledger/#overview.

[70] Panagiotis Antonopoulos, Arvind Arasu, Kunal D. Singh, Ken Eguro, Nitish Gupta, Rajat Jain, Raghav Kaushik, Hanuma Kodavalla, Donald Kossmann, Nikolas Ogg, Ravi Ramamurthy, Jakub Szymaszek, Jeffrey Trimmer, Kapil Vaswani, Ramarathnam Venkatesan, and Mike Zwilling. Azure SQL database always encrypted. In *Proceedings of the ACM SIGMOD Conference*, pages 1511–1525. ACM, 2020.

[71] Antonio Barbalace, Mohamed Lamine Karaoui, Weiqi Wang, Tong Xing, Pierre Olivier, and Binoy Ravindran. Edge computing: the case for heterogeneous-isa container migration. 2020.

[72] Ion Stoica and Scott Shenker. From cloud computing to sky computing. In *Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS)*, 2021.

[73] Guoxing Chen, Yinqian Zhang, and Ten-Hwang Lai. Opera: Open remote attestation for intel's secure enclaves. 2019.

[74] Project veraison creates software components that can be used to build an attestation verification service. https://github.com/veraison.

[75] Kit Murdock, David Oswald, Flavio D. Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. Plundervolt: Software-based fault injection attacks against intel sgx. In *Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P'20)*, 2020.

[76] Jo Van Bulck, Daniel Moghimi, Michael Schwarz, Moritz Lipp, Marina Minkin, Daniel Genkin, Yarom Yuval, Berk Sunar, Daniel Gruss, and Frank Piessens. LVI: Hijacking Transient Execution through Microarchitectural Load Value Injection. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2020.

[77] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Yinqian Zhang, Zhiqiang Lin, and Ten H. Lai. Sgxpectre attacks: Leaking enclave secrets via speculative execution. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2019.

[78] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. Inferring fine-grained control flow inside SGX enclaves with branch shadowing. In *Proceedings of the USENIX Security Symposium*, 2017.

[79] Yuanzhong Xu, Weidong Cui, and Marcus Peinado. Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2015.

[80] Jo Van Bulck, Nico Weichbrodt, Rüdiger Kapitza, Frank Piessens, and Raoul Strackx. Telling your secrets without page faults: Stealthy page table-based attacks on enclaved execution. In *Proceedings of the USENIX Security Symposium*, 2017.

[81] Jo Van Bulck, M. Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, F. Piessens, M. Silberstein, T. Wenisch, Yuval Yarom, and Raoul Strackx. Foreshadow: Extracting the keys to the intel sgx kingdom with transient out-of-order execution. In *USENIX Security Symposium*, 2018.

[82] Mengyuan Li, Yinqian Zhang, Huibo Wang, Kang Li, and Yueqiang Cheng. Cipherleaks: Breaking constant-time cryptography on amd sev via the ciphertext side channel. In *Proceedings of the USENIX Security Symposium*, 2021.

[83] Mengyuan Li, Yinqian Zhang, and Zhiqiang Lin. Crossline: Breaking "security-by-crash" based memory isolation in amd sev. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[84] Mengyuan Li, Yinqian Zhang, Zhiqiang Lin, and Yan Solihin. Exploiting unprotected i/o operations in amd's secure encrypted virtualization. In *USENIX Security Symposium*, 2019.

[85] Mingyu Li, Yubin Xia, and Haibo Chen. Confidential serverless made efficient with plug-in enclaves. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 306–318. IEEE, 2021.

[86] Felix Schuster, Manuel Costa, Cédric Fournet, Christos Gkantsidis, Marcus Peinado, Gloria Mainar-Ruiz, and Mark Russinovich. VC3: trustworthy data analytics in the cloud using SGX. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 38–54. IEEE Computer Society, 2015.

[87] Tyler Hunt, Zhiting Zhu, Yuanzhong Xu, Simon Peter, and Emmett Witchel. Ryoan: A distributed sandbox for untrusted computation on secret data. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 533–549. USENIX Association, 2016.

[88] Mingyu Li, Jinhao Zhu, Tianxu Zhang, Cheng Tan, Yubin Xia, Sebastian Angel, and Haibo Chen. Proceedings of the usenix symposium on operating systems design and implementation (osdi). pages 331–347. USENIX Association, 2021.

[89] Guyue Liu, Hugo Sadok, Anne Kohlbrenner, Bryan Parno, Vyas Sekar, and Justine Sherry. Don't yank my chain: Auditable NF service chaining. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 155–173. USENIX Association, April 2021.

[90] Jinyu Gu, Zhichao Hua, Yubin Xia, Haibo Chen, Binyu Zang, Haibing Guan, and Jinming Li. Secure live migration of SGX enclaves on untrusted cloud. In *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017, Denver, CO, USA, June 26-29, 2017*, pages 225–236. IEEE Computer Society, 2017.

[91] Fritz Alder, Arseny Kurnikov, Andrew Paverd, and N. Asokan. Migrating SGX enclaves with persistent state. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg City, Luxembourg, June 25-28, 2018*, pages 195–206. IEEE Computer Society, 2018.

[92] Intel® tdx migration td design guide. https://www.intel.com/content/dam/develop/external/us/en/documents/tdx-migration-td-design-guide-348987-001.pdf.

[93] Secure live migration of encrypted vms. https://research.ibm.com/publications/secure-live-migration-of-encrypted-vms.

[94] David Goltzsche, Manuel Nieke, Thomas Knauth, and Rüdiger Kapitza. Acctee: A webassembly-based two-way sandbox for trusted resource accounting. In *Proceedings of the 20th International Middleware Conference, Middleware 2019, Davis, CA, USA, December 9-13, 2019*, pages 123–135. ACM, 2019.

[95] Yuming Wu, Yutao Liu, Ruifeng Liu, Haibo Chen, Binyu Zang, and Haibing Guan. Comprehensive VM protection against untrusted hypervisor through retrofitted AMD memory encryption. pages 441–453. IEEE Computer Society, 2018.

[96] Sinisa Matetic, Mansoor Ahmed, Kari Kostiainen, Aritra Dhar, David M. Sommer, Arthur Gervais, Ari Juels, and Srdjan Capkun. ROTE: rollback protection for trusted execution. In *Proceedings of the USENIX Security Symposium*, pages 1289–1306. USENIX Association, 2017.

[97] Jinyu Gu, Xinyue Wu, Bo Q. Zhu, Yubin Xia, Binyu Zang, Haibing Guan, and Haibo Chen. Enclavisor: A hardware-software co-design for enclaves on untrusted cloud. volume 70, pages 1598–1611, 2021.

[98] S. Zhao, M. Li, Y. Zhang, and Z. Lin. vsgx: Virtualizing sgx enclaves on amd sev. In *2022 IEEE Symposium on Security and Privacy (SP) (SP)*, pages 687–702, Los Alamitos, CA, USA, may 2022. IEEE Computer Society.

[99] SOFAEnclave. https://github.com/SOFAEnclave/SOFAEnclave.

[100] Sadjad Fouladi, Francisco Romero, Dan Iter, Qian Li, Shuvo Chatterjee, Christoforos E. Kozyrakis, Matei A. Zaharia, and Keith Winstein. From laptop to lambda: Outsourcing everyday jobs to thousands of transient functional containers. 2019.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on March 14,2023 at 05:51:05 UTC from IEEE Xplore. Restrictions apply.