SoK: Analysis of Accelerator TEE Designs

Chenxu Wang[12], Junjie Huang[1], Yujun Liang[1], Xuanyao Peng[13], Yuqun Zhang[1],
Fengwei Zhang[1], Jiannong Cao[2], Hang Lu[3], Rui Hou[3], Shoumeng Yan[4], Tao Wei[4], Zhengyu He[4]

1Southern University of Science and Technology, 2The Hong Kong Polytechnic University,
3Chinese Academy of Sciences ,4Ant Group

COMPASS Lab
COMPuter And System Security Lab

SUSTech Southern University of Science and Technology

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

CHINESE ACADEMY OF SCIENCES

ANT GROUP

Accelerator TEE

- Accelerator TEE: Extend TEE concept (from CPU) to accelerator
  - GPU, NPU, TPU, FPGA-based accelerator …
  - Designed for
    - Accelerator data/model **Confidentiality** and **Integrity**
    - Accelerator computing with **Isolation**
    - Accelerator system with **Authenticity**

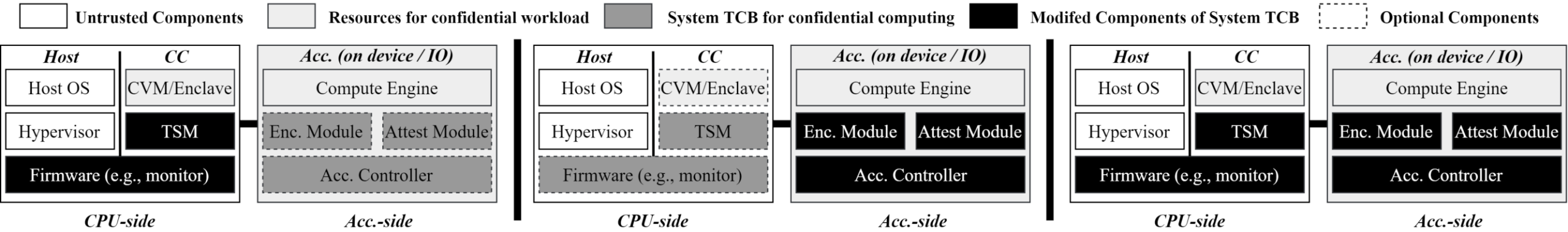
Accelerator TEE: Overview

- We summarize **51** academy/industry accelerator TEE studies
  - **GPU**: **Graviton**@OSDI18, NVIDIA **H100**, **MyTEE**@NDSS23, **CAGE**@NDSS24 ...
  - **NPU**: **TNPU**@HPCA22, **sNPU**@ISCA24, **ASGARD**@NDSS25 ...
  - **General accelerator**: **HETEE**@SP20, **ACAI**@USENIX24, **ccAI**@MICRO25 ...
  - ...

- Motivation of SoK
  - Accelerator TEE is gradually **popular** (40+ studies since 2022)
  - Accelerator TEEs are varied in CPU/accelerator, but **no systematic analysis**
  - Accelerator TEE's **deployment still faces non-trivial challenges**

Accelerator TEE: Framework

- Despite varied CPU/accelerator, accelerator TEE can be classified in three types:
  - Host-type: Deploying TEE in **CPU-side software/hardware**
  - Accelerator-type: Deploying TEE in **Accelerator hardware/firmware**
  - Mix-type: Deploying TEE in **both CPU and Accelerator**



(a) Host-type accelerator TEEs (b) Acc.-type accelerator TEEs (c) Mixed-type accelerator TEEs

Accelerator TEE: Framework

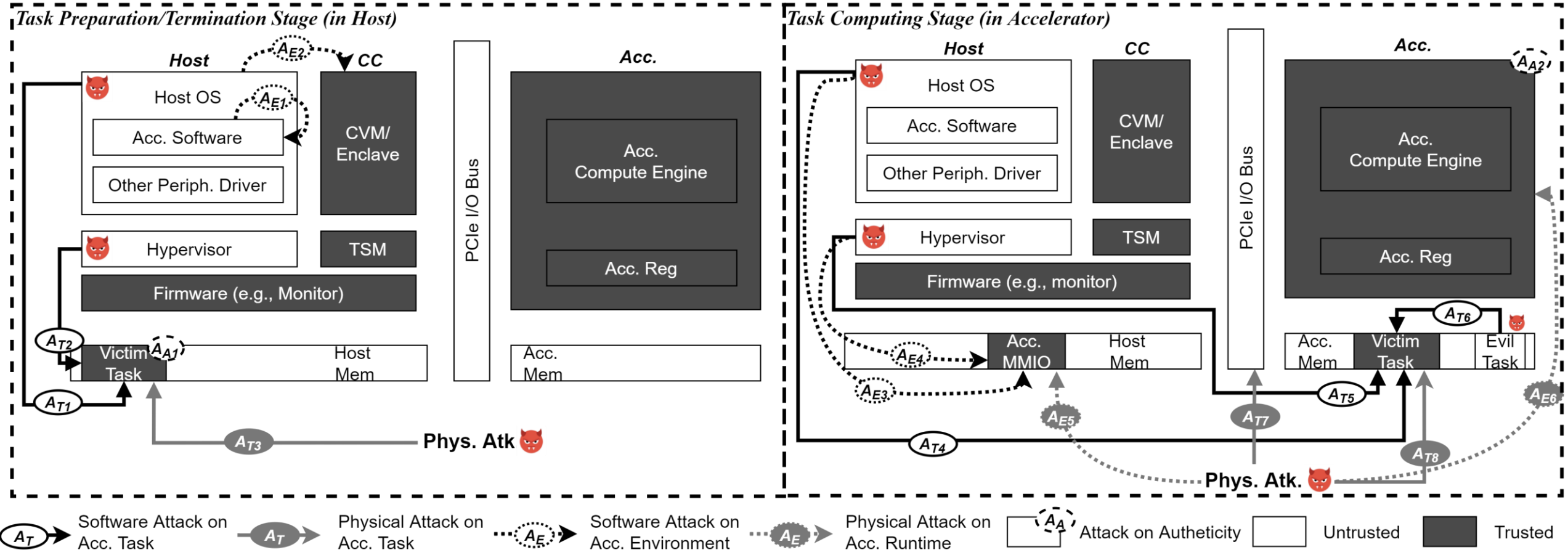
Acc. TEE Type	CPU-side			Acc.-side modules (in Acc./Board/Ext. IO)		
	CVM/Enclave	TSM	Firmware	Enc. Module	Attest Module	Acc. Controller
***Host-type* Acc. TEEs**						
ACAI [67]	Arm CCA	RMM	Monitor	PCIe IDE(Acc.)	HRoT(Acc.)	-
ASGARD [77]	Arm TrustZone	S-Hyp	Monitor	-	HRoT(Acc.)	-
AvaGPU [32]	Arm TrustZone	-	Monitor	-	-	-
Cronus [47]	Arm TrustZone	S-Hyp	Monitor	-	HRoT(Acc.)	-
CURE [36]	RISC-V Customized	-	M-Monitor	-	-	-
CAGE [68]	Arm CCA	RMM	Monitor	-	-	-
GR-T [60]	Arm TrustZone	-	Monitor	-	-	-
Honeycomb [61]	AMD SEV-SNP	SVSM	SEV-firmware	-	-	-
HIX [33]	Intel SGX	-	SGX-firmware	-	HRoT(Acc.)	-
HyperTEE [69]	RISC-V Customized	-	M-Monitor	-	-	-
LEAP [49]	Arm TrustZone	-	Monitor	-	-	-
MyTEE [63]	Arm TrustZone	-	Monitor	-	-	-
Portal [80]	Arm CCA	RMM	Monitor	-	-	-
sIOPMP [73]	RISC-V Penglai	-	M-Monitor	-	-	-
StrongBox [31]	Arm TrustZone	-	Monitor	-	-	-
XpuTEE [82]	Intel TDX/SGX	-	VMX root	-	-	-
***Acc.-type* Acc. TEEs**						
AccShield [59]	Intel TDX/AMD SEV	TDX module/SVSM	TDX/SEV-firmware	AES-GCM Engine(Board)	HRoT(Board)	Security Manager(Board)
Ambassy [44]	Arm TrustZone	-	Monitor	AES Cores(Acc.)	-	Acc. Controller(Acc.)
CommonCounters [45]	Intel SGX	-	SGX-firmware	Opti-Enc. Engine(Acc.)	-	Command Processor(Acc.)
ccAI [78]	-[1]	-	-	AES-GCM Engine(Ext.IO)	HRoT(Ext.IO)	PCIe-SC(Ext.IO)
Dhar et al. [30]	-[1]	-	-	AES-GCM Engine(Ext.IO)	HRoT(Ext.IO)	Security Controller(Ext.IO)
GuardAIn [29]	-	-	-	AES-GCM Engine(Acc.)	HRoT(Acc.)	Task Scheduler(Acc.)
GuardNN [48]	-	-	-	Opti-Enc. Engine(Acc.)	-	Micro-controller(Acc.)
Graviton [34]	Intel SGX	-	SGX-firmware	AuthEnc/Dec. kernel(Acc.)	HRoT(Acc.)	Command Processor(Acc.)
HETEE [28]	-	-	-	AES-GCM Engine(Ext.IO)	HRoT(Ext.IO)	Security Controller(Ext.IO)
ITX [62]	-	-	-	AES-GCM Engine(Board)	CCU(Board)	ICU(Board)
LITE [50]	Intel TDX/AMD SEV	TDX module/SVSM	TDX/SEV-firmware	Enc. kernel&Spec. HW(Acc.)	-	Acc. Controller(Acc.)
MGX [51]	Intel SGX	-	SGX-firmware	Opti-Enc. Engine(Acc.)	HRoT(Acc.)	Control Processor(Acc.)
Na et al. [70]	Intel SGX	-	-	Opti-Enc. Engine(Acc.)	HRoT(Acc.)	Command Processor(Acc.)
NVIDIA H100 [58]	Intel TDX/AMD SEV/Arm CCA	TDX module/SVSM/RMM	TDX/SEV-firmware/Monitor	AES-GCM Engine(Acc.)	HRoT(Acc.)	Acc. Controller(Acc.)
PipeLLM [79]	Intel TDX/AMD SEV/Arm CCA	TDX module/SVSM/RMM	TDX/SEV-firmware/Monitor	AES-GCM Engine(Acc.)	HRoT(Acc.)	Acc. Controller(Acc.)
Plutus [64]	-	-	-	Opti-Enc. Engine(Acc.)	-	Memory Controller(Acc.)
PSSM [46]	Intel SGX	-	SGX-firmware	Opti-Enc. Engine(Acc.)	-	Command Processor(Acc.)
Salus-FPGA [72]	Intel SGX	-	SGX-firmware	AES-GCM Engine(Acc.)	HRoT(Acc.)	SM Controller(Acc.)
Salus-GPU [71]	-	-	-	Opti-Enc. Engine(Acc.)	-	Memory Controller(Acc.)
Securator [66]	-	-	-	Opti-Enc. Engine(Acc.)	-	Security Module(Acc.)
SeDA [81]	-	-	-	Opti-Enc. Engine(Acc.)	-	Memory Controller(Acc.)
ShEF [52]	-	-	-	Engine set(Board)	HRoT(Board)	Shield(Board)
SAGE [65]	Intel SGX	-	SGX-firmware	AuthEnc/Dec. kernel(Acc.)	Kernel(Acc.)	Kernel Caller(Acc.)
SGX-FPGA [35]	Intel SGX	-	SGX-firmware	Enc. Engine(Acc.)	PUF(Acc.)	FPGA Secure Monitor(Acc.)
SHM [54]	Intel SGX	-	SGX-firmware	Opti-Enc. Engine(Acc.)	-	Command Processor(Acc.)
SrcTEE [74]	Arm TrustZone	-	Monitor	AES-GCM Engine(Board)	PUF(Board)	Config. Sec. Unit(Board)
Telekine [43]	Intel SGX	-	SGX-firmware	AuthEnc/Dec. kernel(Acc.)	HRoT(Acc.)	Command Processor(Acc.)
T-edge [75]	Arm TrustZone	-	Monitor	Enc. Engine(Acc.)	HRoT(Acc.)	Acc. Controller(Acc.)
TrustOre [42]	Intel SGX	-	SGX-firmware	AES-GCM Engine(Acc.)	Attester(Acc.)	TrustMod(Acc.)
TNPU [53]	Intel SGX	-	SGX-firmware	Opti-Enc. Engine(Acc.)	-	Memory Controller(Acc.)
TensorTEE [76]	Intel SGX	-	SGX-firmware	Opti-Enc. Engine(Acc.)	-	Memory Controller(Acc.)
***Mix-type* Acc. TEEs**						
Arm RME-DA [55]	Arm CCA	RMM	Monitor	PCIe IDE(Acc.)	HRoT(Acc.)	DSM(Acc.)
AMD SEV-TIO [56]	AMD SEV	SVSM	SEV-firmware	PCIe IDE(Acc.)	HRoT(Acc.)	DSM(Acc.)
Intel TDX Connect [57]	Intel TDX	TDX module	TDX-firmware	PCIe IDE(Acc.)	HRoT(Acc.)	DSM(Acc.)
sNPU [37]	RISC-V Penglai	-	M-Monitor	-	-	Isolator, Guard(Acc.)

Accelerator TEE: Threats

- Based on TEE framework, we analyze the generic attack vectors
  - Two stages: (1) preparing/terminating and (2) computing
  - From software & physical adversaries
    - Untrusted CPU software
    - Physical attachment (e.g., on bus) or compromise
    - …
  - To attack
    - Task code, data, etc.
    - Environment runtime (e.g., MMIO)
    - Software/hardware authenticity

Accelerator TEE: Threats

- Based on TEE framework, we analyze the generic attack vectors



Accelerator TEE: Defense Mechanism

- To defend against attack vectors, accelerator TEEs mainly design three mechanisms
  - 6 solutions in **Access control**
    - Use TEE to protect accelerator workload (S_{AC1}) and even protect drivers (S_{AC2})
    - Design control in CPU-side hypervisor (S_{AC3}) or firmware monitor (S_{AC4})
    - Design control in IO bus (S_{AC5}) or accelerator side (S_{AC6})
  - 3 solutions in **Memory encryption**:
    - Encrypt CPU-side memory (S_{ME1}) , IO bus (S_{ME2}) , or accelerator memory (S_{ME3})
  - 4 solutions in **Attestation**: Verify the software/hardware authenticity
    - Attest based on CPU HRoT (S_{AT1}) or software (S_{AT2})
    - Attest based accelerator HRoT (S_{AT3}) or external hardware (S_{AT4})

Accelerator TEE: Defense Mechanism

- To defend against attack vectors, accelerator TEEs mainly design three mechanisms
  - Each mechanism covers a set of attack vectors

	Solutions	CVM/Enclave: Acc. Workload	CVM/Enclave: Acc. Driver	TSM	Firmware	CPU HW	Bus	Enc. Module	Attest Module	Acc. Controller	Attacks in Task Preparation/Termination: A_{T1}	A_{T2}	A_{T3}	A_{E1}	A_{E2}	A_{A1}	Attacks in Task Computing: A_{T4}	A_{T5}	A_{T6}	A_{T7}	A_{T8}	A_{E3}	A_{E4}	A_{E5}	A_{E6}	A_{A2}
Access Control	S_{AC1}	✓									●	●	○	○	○	○	◐	◐	○	○	○	○	○	○	○	○
	S_{AC2}	✓	✓								●	●	○	●	○	○	◐	◐	●	○	○	○	○	○	○	○
	S_{AC3}			✓							●	○	○	●	◐	○	●	○	●	○	○	●	○	○	○	○
	S_{AC4}				✓						●	●	○	●	●	○	●	●	●	○	○	●	●	○	○	○
	S_{AC5}						✓				○	○	○	○	○	○	●	●	○	◐	○	●	●	◐	○	○
	S_{AC6}									✓	○	○	○	○	○	○	●	●	●	◐	◐	◐	◐	◐	●	○
Mem Enc	S_{ME1}	✓	✓			✓					●	●	●	○	○	○	○	○	○	◐	○	●	●	●	○	○
	S_{ME2}							✓			○	○	○	○	○	○	●	●	●	◐	●	○	○	○	○	○
	S_{ME3}						✓				○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○
Attestation	S_{AT1}					✓					○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	◐
	S_{AT2}	✓	✓					✓			○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○
	S_{AT3}								✓		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
	S_{AT4}									✓	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●

- … And we also get several insights …

Accelerator TEE: Access Control & Insights

- To achieve access control, accelerator TEE can **combine multiple solutions** (I_{AC1}), because…
  - **Deployment scenario** consideration: e.g., cloud vs edge

Scenario	Deployment Features	Access Control Solution						Acc. TEE Type			Specific Mechanism
		S_{AC1}	S_{AC2}	S_{AC3}	S_{AC4}	S_{AC5}	S_{AC6}	*Host-type*	*Acc.-type*	*Mix-type*	
CPU-Discrete Acc. (e.g., cloud)	CPU with Any TEE (e.g., TDX/SEV for Multi-tenants)	●	○	○	○	○	●		[34] [51] [70] [45] [46] [72] [35] [54] [42] [43] [76]		Intel SGX, Hardware-based Acc. Controller(Acc.)
	Re-programmed Acc. (e.g., FPGA for designer, Hopper GPU of NVIDIA)	○	●	○	○	○	●		[65]		Intel SGX, Kernel Caller(Acc.)
									[58] [50] [79]		CVM with MEE, NVIDIA CC hardware-supported (Acc.)
									[59]		Intel TDX/AMD SEV, Security Manager(Board)
	Plug-and-play Link (e.g., PCIe-based Sec. HW)	○	●	○	○	●	●			[56] [57] [55]	TDISP
		○	●	○	○	●	○	[33]			Intel SGX, PCIe Root Complex
									[30] [78]		Any CPU TEE, Security Controller(Ext. IO)
	CPU with Specific TEE (e.g., VMX root of Intel)	●	○	●	○	○	○	[61]			AMD SEV, SVSM
								[82]			Intel SGX/TDX, VMX root
	Legacy Acc. (e.g., A100)	○	●	●	○	○	○	[47]			Arm TrustZone, S-Hyp
		○	●	●	●	○	○	[67]			Arm CCA, RMM, Monitor
	Legacy CPU (e.g., w/o CPU TEE) Re-programmed Acc.	○	○	○	○	○	●		[64] [48] [71] [29] [81]		Acc. Controller(Acc.)
									[62] [52]		Integrated Security Hardware(Board)
	Legacy CPU-Acc.	○	○	○	○	●	○		[28]		PCIe-based Security Controller(Ext. IO)
	Preference	14/34	12/34	4/34	1/34	6/34	**26/34**	5/34	**26/34**	3/34	**Mainstream solution combination:** $S_{AC1/2} + S_{AC5/6}$
Integrated CPU-Acc. (e.g., edge)	Platform with specific sec. HW (e.g., TZASC/GPC in Arm) or modified privilege SW (e.g., S-Hyp/ trusted firmware in Arm)	●	○	○	●	○	○	[68]			Arm CCA, Monitor
								[31] [63]			Arm TrustZone/OP-TEE, Monitor
		○	●	●	○	○	○	[77]			Arm TrustZone, S-Hyp
		○	●	●	●	○	○	[80]			Arm CCA, RMM, Monitor
		○	●	○	●	○	○	[60] [32] [49]			Arm TrustZone, Monitor
	Re-programmed HW (e.g., RISC-V/ FPGA-based DNN Acc.)	○	●	○	●	●	○	[36] [69]			Customized RISC-V TEE, M-Monitor, CPU IO Filter
		●	○	○	●	●	○	[73]			RISC-V Penglai, M-Monitor, CPU IO Filter
		●	○	○	●	○	●			[37]	RISC-V Penglai, M-Monitor, Isolator, Guard(Acc.)
		○	●	○	○	○	●		[53]		Intel SGX, Memory Controller(Acc.)
									[75] [44] [74]		OP-TEE, Acc. Controller(Acc.)
		○	○	○	○	○	●		[66]		Acc. Controller(Acc.)
	Preference	5/17	11/17	2/17	**11/17**	3/17	6/17	**11/17**	5/17	1/17	**Mainstream solution combination:** $S_{AC1/2} + S_{AC4}$

Accelerator TEE: Access Control & Insights

- To achieve access control, accelerator TEE can **combine multiple solutions** (I_{AC1}), because…
  - **Deployment scenario** consideration: e.g., cloud vs edge
  - Applying high-privilege access control to **replace CPU TEE protection for accelerator driver** (I_{AC2})
    - CPU-side TEE is used for securing accelerator workloads, but delegate non-confidential functions for drivers (i.e., S_{AC1})
    - In Hypervisor/Monitor, design additional security checks for accelerator memory and MMIO
  - **Granularity limitation** in CPU security primitives (I_{AC3})
    - Cannot over-rely on firmware-based protection (i.e., S_{AC4})

	Arm TZASC	Arm GPC	RISC-V PMP	Addr Trans.
Solution Type	S_{AC4}	S_{AC4}	S_{AC4}	$S_{AC1,2,3}$
Minimal Granularity	32KB	4KB	4Byte	4KB
Configurable Regions	Limited	Non-limited	Limited	Non-limited
Read/Write Distinction	Supported	Not Supported	Supported	Supported
Execute Permission	Not Supported	Not Supported	Supported	Supported
PAN/PXN Permission	Not Supported	Not Supported	Not Supported	Supported
Studies Examples	[31], [60]	[67], [68], [80]	[37], [73]	[31], [32], [47], [67]

Accelerator TEE: Memory Encryption & Insights

- Underestimated physical threats (I_{ME1})
  - **More than half** of the studies (30/51 studies) are impacted
    - **Cloud**: Connect to the host via PCIe
      - PCIe/CXL link → IO/bus encryption (S_{ME3}) or Co-encryption (S_{ME1}+S_{ME2})
      - Discrete memory (e.g., GDDRx) → Acc.-encryption (S_{ME2})
    - **Edge**: CPU-accelerator integrated
      - Shared memory (e.g., LPDDRx) → Co-encryption (S_{ME1}+S_{ME2})

Scenarios	Victim	Physical Threats	Missing Memory Encryption (S_{ME}) → Consequence	Influenced Studies
CPU-Discrete Acc. (e.g., cloud)	Plug-and-play Host Memory (e.g., DDRx)	A_{T3}, A_{E5}	Missing CPU-based encryption (S_{ME1}) → Data/code/metadata/PTEs in plaintext on host memory are vulnerable to physical access/tampering (e.g., cold-boot attacks)	[28], [48] [47], [60]
	3D-stacked Acc. Memory (e.g., HBM)	-	Missing any memory encryption → Minimal physical threats	[34], [42], [43] [52], [58] [61], [62], [79] [71]
	On-board Acc. Memory (e.g., GDDRx/LPDDRx)	A_{T8}	Missing Acc.-based encryption (S_{ME2}) → Data/code/metadata/PTEs in plaintext on acc. memory are vulnerable to physical access/tampering (e.g., probing attacks)	[28], [33], [35], [47], [55]–[57], [60] [30], [65], [67], [72], [78], [82]
	Plug-and-play Link (e.g., PCIe/CXL)	A_{T7}	Missing IO-based (S_{ME3}) or CPU-Acc. encryption ($S_{ME1,2}$) → Physical access/tamper/replay packets in plaintext on the link (e.g., replay attacks).	[33], [47], [48], [82]
Integrated CPU-Acc. (e.g., edge)	On-board/Plug-and-play Shared Memory (e.g., LPDDRx/DDRx)	$A_{T3,8}, A_{E5}$	Missing CPU-Acc.-based encryption ($S_{ME1,2}$) → Data/code/metadata/PTEs in plaintext on shared memory are vulnerable to physical access/tampering	[31], [32], [36], [49], [63], [68], [80] [37], [44], [68], [73]–[75], [77], [80]

Missing de/encryption engine (e.g., AES) for confidentiality → Direct access to plaintext data
Missing integrity check engine (e.g., Message Authentication Code, MAC) for integrity → Tampering with plaintext/ciphertext data
Missing number used once (e.g., counter/integrity tree [105]) for freshness → Replay attacks

Accelerator TEE: Memory Encryption & Insights

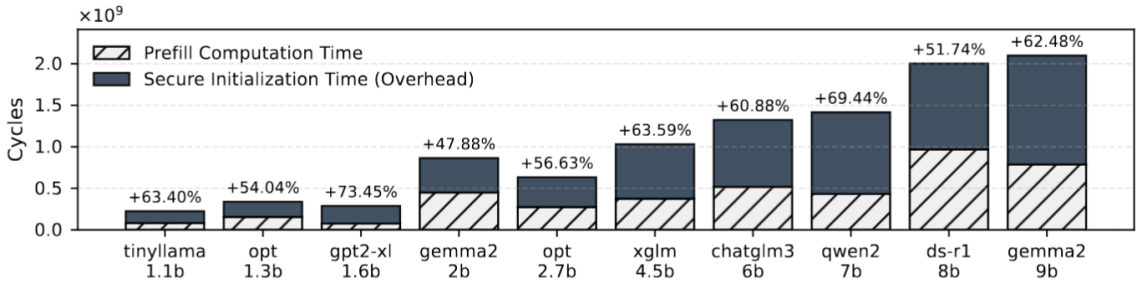
- **High overhead** from improper security metadata management ($I_{ME2,3}$)
  - CPU (64B cache line) vs. accelerator (KB/MB blocks) → frequent metadata access
    - NPUs: Coarse-grained (tiles/layers) for neural network workloads
    - GPUs: Multi-grained (coarse for large/low-frequency data) to reduce overhead

  → Align metadata granularity with accelerator access patterns
  - Mismatched CPU-accelerator enc. gra. → extra init/communication overhead
    - Initialization: 47.88%–73.45% overhead
    - Communication: 40.36%–44.94% overhead

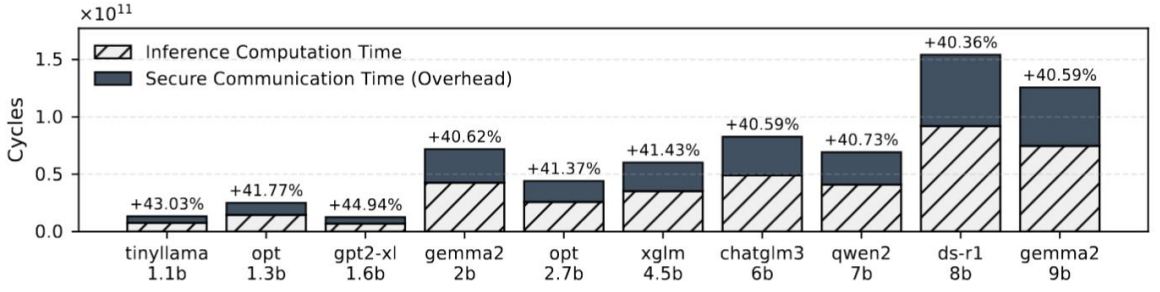
  → Ensure CPU-accelerator encryption granularity consistency

Memory Access

Interact with CPU TEE



(a) LLM startup stage and interaction overhead (initialization).

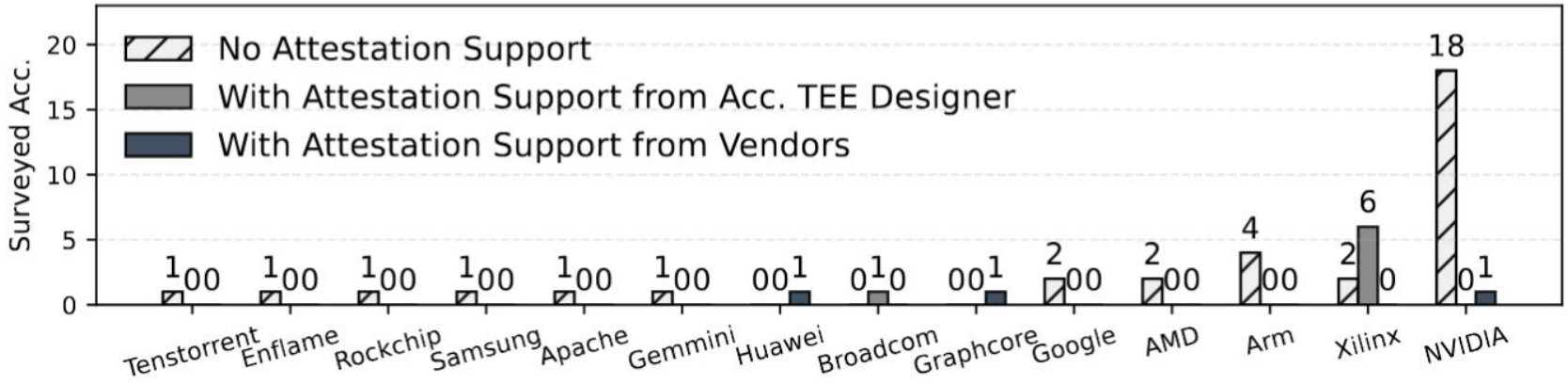


(b) LLM decoding stage and interaction overhead (communication).

Accelerator TEE: Attestation & Insights

- Lacking attestation implementation (I_{AT1})
  - **34/44** accelerators (uesd in 51 studies) lack attestation support
    - Only **3 vendors** (Huawei, Graphcore, NVIDIA) have full mechanisms
  - Missing HRoT/TPM/Endorser → Malicious replacement & code injection
    - MOLE@CCS25 inject malicious MCU firmware to break GPU TEE

→ Urgent need for HW/SW support (e.g., AMD plan to integrate open-source Roots of Trust Caliptra)



Acc. Vendors	Acc. Device Name (used Acc. TEE)
Tenstorrent	N150 NPU ([78])
Enflame	S60 GPU ([78])
Rockchip	RK3588S NPU ([77])
Samsung	Exynos 990 NPU* ([51], [53], [81])
Apache	VTA NPU* ([47])
Gemmini	Gemmini NPU ([37], [69])
Huawei	Ascend 910A NPU ([29], [30])
Broadcom	VideoCore IV GPU ([63])
Graphcore	GC200 IPU ([62])
Google	TPU-v1* ([48], [51], [66], [81]), TPU-v3* ([76])
AMD	Radeon RX VEGA 64 GPU ([43]), RX6900XT GPU ([61])
Arm	Mali-G71 GPU ([49], [60]), Mali-T624 GPU ([31], [68]), Mali-G610 GPU ([80]), Ethos-N77 NPU* ([53])
Xilinx	VCU118 FPGA ([59], [67]), Zynq-7000 FPGA ([42]), ZCU106 FPGA ([75]), XCZU15EG FPGA ([74]), XCZU9EG FPGA ([44]), UltraScale+ Ultra96 FPGA ([52]), Alveo U200 FPGA ([72]), ADM-PCIE-7V3 FPGA ([35])
NVIDIA	GTX 780 GPU ([34]), GTX 460 SE GPU ([67]), GTX 580 GPU ([33]), GTX 2080 GPU ([47]), GTX Titan Black GPU ([28], [34]), Tesla P40 GPU ([28]), Tesla V100 GPU ([28]), T4 GPU ([78]), L20 GPU ([78]), RTX 4090Ti GPU ([78]), RTX 3080 GPU ([82]), RTX 2080 GPU ([49]), TITAN X Pascal GPU* ([45]), Volta Arch. GPU* ([46], [64], [71]), Turning Arch. GPU* ([54]), H100 GPU ([58], [79]), A100 GPU ([65], [78]), NVDLA ([73]), Jetson AGX Orin ([32])

*Accelerator TEE uses simulators to emulate the corresponding commercial accelerators (e.g.,MGX [51] use SCALE-Sim [110] to simulate Google TPU-v1 and Samsung Exynos 990 NPU).

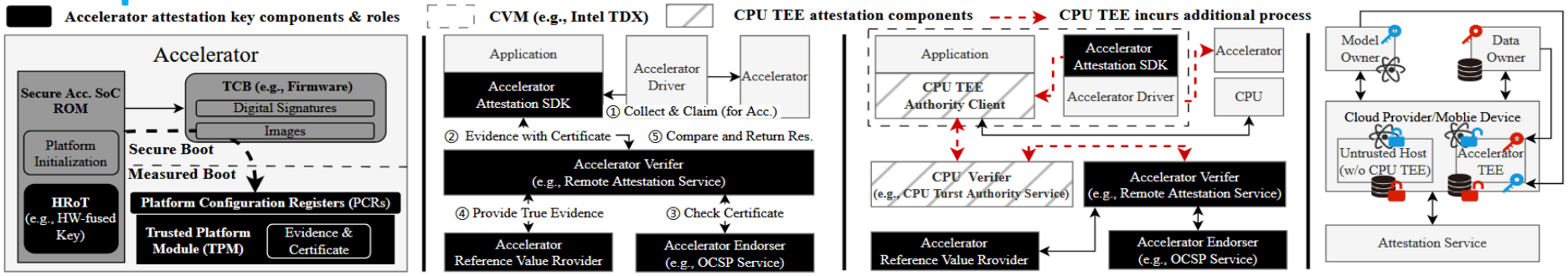
Accelerator TEE: Attestation & Insights

- Potential threats from accelerator attestation ($I_{AT2,3}$)
  - Weak CPU-accelerator integrated attestation (e.g., Intel TDX + NVIDIA CC)
    - **Centralized CPU authority**: Leaks sensitive data (accelerator ID, TCB details)
    - **Non-binding user**: Abuse accelerator attestation report

  → **Adopt privacy-preserving techniques & Enforce user-Acc. attestation**

  - Incomplete AI task runtime attestation
    - **Command-level attestation**: AI task sequence dependencies (e.g., DNN layers)
    - **Multiple distrusting parties**: Collusion risks (e.g., data and model providers)

  → **Task-specific attestation**



(a) Establishing the chain of trust based on HRoT (initialization phase)

(b) Accelerator remote attestation roles and messages (runtime phase)

(c) CPU-Accelerator integrated attestation roles and messages (e.g., Intel TDX + NVIDIA H100)

(d) Task attestation (e.g., Huawei GuardAIn)

Accelerator TEE: Trusted Computing Base Issues

- However, deploying accelerator TEE must consider **TCB bloating** problem
  - Guest TCB bloating in
    - **Varied and heavy-weight** accelerator software stacks
    - Adding more TCB to support **new security functions**
  - System TCB bloating
    - **Abusing TCB addition** in high-privilege components

Software Stack		Supported Acc.	TCB Size (ver. of src.)	Acc. TEE
NVIDIA GPU	official CUDA toolkit [98]	NVIDIA GPU [14]	N/A[2]	[28], [32], [50], [58], [65], [78], [79], [82]
	official kernel driver [122]	NVIDIA GPU [14]	1.4M (v575.64.05 [122])	[28], [50], [58], [78], [79], [82]
	gdev [120]	NVIDIA GPU [14]	0.3M (latest [123])	[33], [34], [47], [67]
	nouveau [124]	NVIDIA GPU [14]	0.1M (in Linux v6.16 [125])	[34], [47], [67]
AMD GPU	ROCm [126]	AMD Radeon GPU [15]	10M (latest [126])[1]	[43], [61]
	AMD GPU driver	AMD Radeon GPU [15]	5.0M (in Linux v6.16 [125])	[43], [61]
Arm Mali GPU	Bifrost driver [89]	Mali G71/G610 GPU [16]	0.1M (r54p1 [89])	[49], [60], [80]
	Midgard driver [127]	Mali T6xx/T7xx/T8xx GPU [16]	47K (r32p0 [127])	[31], [68]
OpenCL [99]		Mali GPU [16]	N/A[2]	[31], [49], [60], [68], [80]
Xilinx FPGA	Xilinx DMA drivers [128]	Xilinx FPGA [24]	7.6K (latest [128])	[67]
	XRT [121]	Xilinx FPGA [24]	0.3M (v2.19.194 [121])	[52]
Coyote FPGA software stack [129]		Xilinx FPGA [24]	7.5K (v0.2.1 [130])	[59]
Huawei NPU	Ascend software [131]	Huawei Ascend NPU [21]	N/A[2]	[29], [30]
VTA NPU	vta-driver [132]	VTA NPU [133]	2.7K (latest [132])	[47]
Arm NPU	Ethos-N driver stack [134]	Arm Ethos-N77 NPU [19]	72K (v25.03 [134])	[53]
Samsung NPU	Exynos driver [135]	Samsung Exynos NPU [20]	17K (latest [135])	[53]
NVIDIA DNN Acc.	NVDLA software [136]	NVDLA DNN Acc. [27]	0.4M (v1.2.0 [136])	[37], [73]

Acc. TEE	High privilege SW		Low privilege SW (CVM/Enclave) or Sec. HW (Acc./Board/IO)
	TSM (Hyp.-level)	Firmware (Mon.-level)	
Graviton [34]	-	SGX-FW	Cmd Processor(Acc.)
HIX [33]	-	SGX-FW	IO Filter(IO)
HETEE [28]	-	-	Sec. Controller(IO)
TrustOre [42]	-	SGX-FW	TrustMod(Acc.)
Telekine [43]	-	SGX-FW	Cmd Processor(Acc.)
Ambassy [44]	-	Mon(0.5M)	Acc. Controller(Acc.)
CommonCounters [45]	-	SGX-FW	Cmd Processor(Acc.)
CURE [36]	-	Sec. Monitor(0.5K) Crypt. Op. (2.6K)	IO Filter(IO)
PSSM [46]	-	SGX-FW	Cmd Processor(Acc.)
SGX-FPGA [35]	-	SGX-FW	FPGA Sec. Monitor(Acc.)
Cronus [47]	S-Hyp(35K) mEnclave Mng(4.3K) HAL core(2.1K)	Mon(0.5M)	-
GuardNN [48]	-	-	Micro-Controller(Acc.)
LEAP [49]	-	Mon(0.5M)+0.5K	OP-TEE(0.3M)+0.7K
LITE [50]	SVSM(5K)	SEV-FW	Acc. Controller(Acc.)
MGX [51]	-	SGX-FW	Cmd Processor(Acc.)
ShEF [52]	-	-	Shield(Board)
StrongBox [31]	-	Mon(0.5M) Crypt. Op.(0.5K) Integrity Check(0.2K) Access Control(0.3K) Other Config.(0.2K)	-
TNPU [53]	-	SGX-FW	Memory Controller(Acc.)
SHM [54]	-	SGX-FW	Cmd Processor(Acc.)
Arm RME-DA [55]	RMM(33K)	Mon(0.5M)	DSM(Acc.)
AMD SEV-TIO [56]	SVSM(5K)	SEV-FW	DSM(Acc.)
Intel TDX Connect [57]	TDX Mod.(35K)	TDX-FW	DSM(Acc.)
NVIDIA H100 [58]	TDX Mod.(35K)	TDX-FW	Acc. Controller(Acc.)
AccShield [59]	TDX Mod.(35K)	TDX-FW	Sec. Mng(Board)
AvaGPU [32]	S-Hyp(35K) S2 Trans.(0.4K) Sec. GPU Mng(4.9K) Mediator(0.2K) Replayer(0.3K)	Mon(0.5M)	-

Accelerator TEE: Compatibility Issues

- Accelerator TEE designs fail to satisfy two major compatibility requirements
  - Lack of **Multi-type** support:
    - Lack of considering **CPU TEE architecture variance** (37/51 studies)
      - Rely on unique CPU features; No user-layer CPU TEE support
    - Design for **limited accelerators** (43/51 studies)
      - Not suitable for other accelerators' computing workflow
  - Lack of **Plug-and-play** support:
    - Need to modify **accelerator software** (33/51 studies)
      - Kernel driver changes, import TEE-specific APIs …
    - Need to modify **Platform hardware** (34/51 studies)
      - CPU ISA, PCIe I/O bus, accelerator hardware …

Conclusions

- Our SoK concludes and finds…
  - Typical framework of accelerator TEE designs
    - **Host/Acc./Mix-type design** overview
  - Security threats, defense mechanisms and issues of accelerator TEE
    - **Attack vectors** of accelerator computing
    - **Access control**, **memory encryption**, **attestation**, and their issues
  - Two key problems in deploying TEE
    - **Compatibility** problems
    - **TCB** bloating


Thank You!

Contact us: zhangfw@sustech.edu.cn