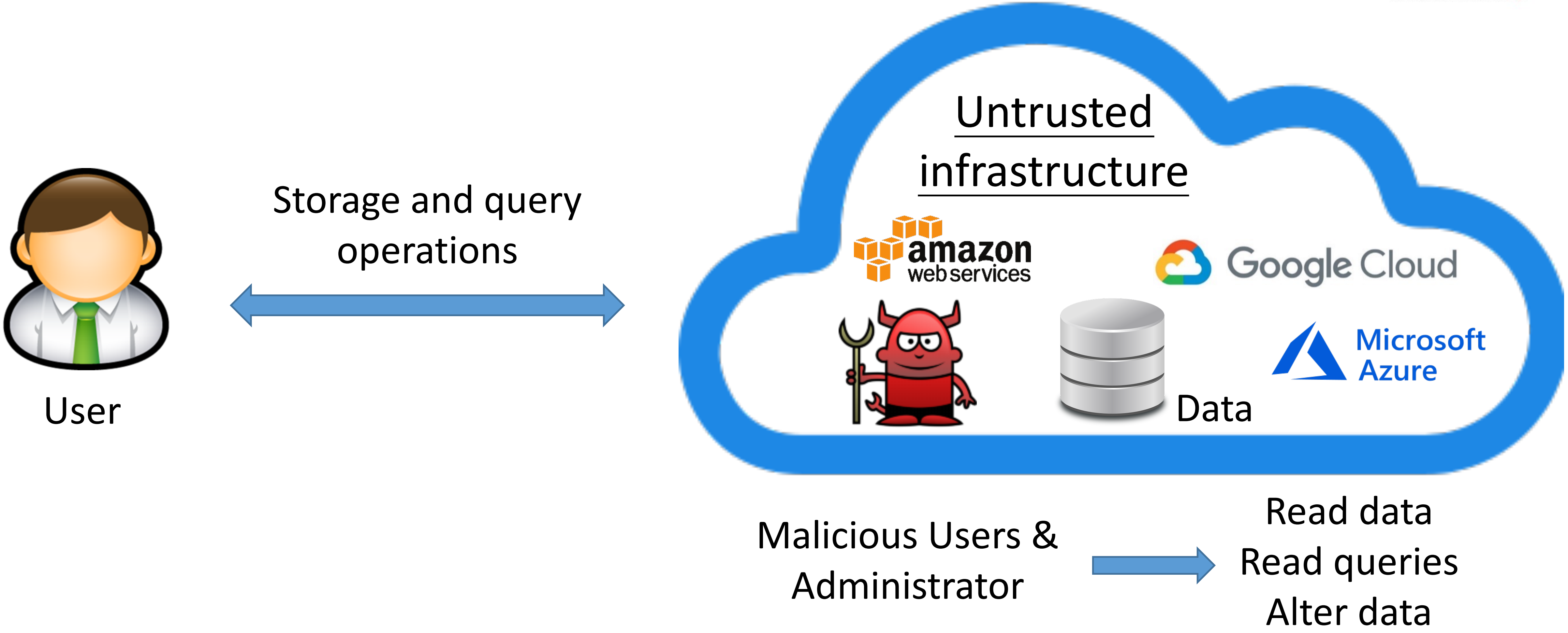
Building Enclave-Native Storage Engines for Practical Encrypted Databases

Yuanyuan Sun, Sheng Wang, Huorong Li, Feifei Li

Alibaba Group

01 Introduction

Security in the Cloud



How do we ensure security of data during the operation of cloud databases?

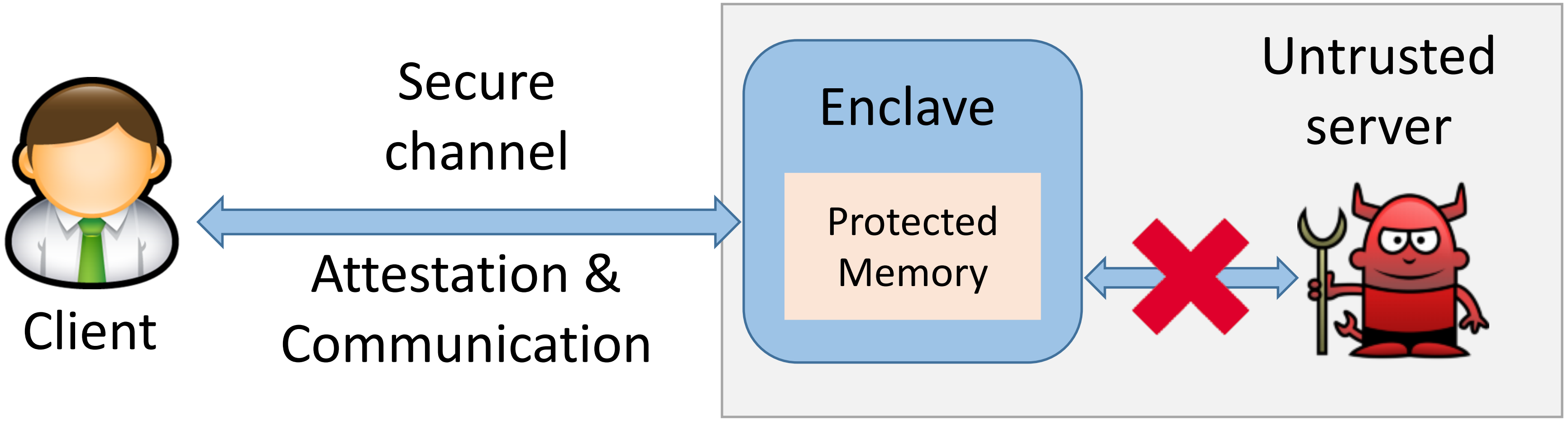
Confidential Computing

Trusted execution environments (TEEs)

- Hardware extensions for trusted computing
  - E.g., Intel SGX and AMD SEV
- Guarantees confidentiality and integrity
  - Computation and data in it

Hardware enclaves of Intel SGX

- A trusted component in an untrusted system
  - Uses protected memory to isolate shielded execution from compromised OS
  - Proves that it is an authentic enclave running the desired code with attestation



Challenges for SGX-based Databases

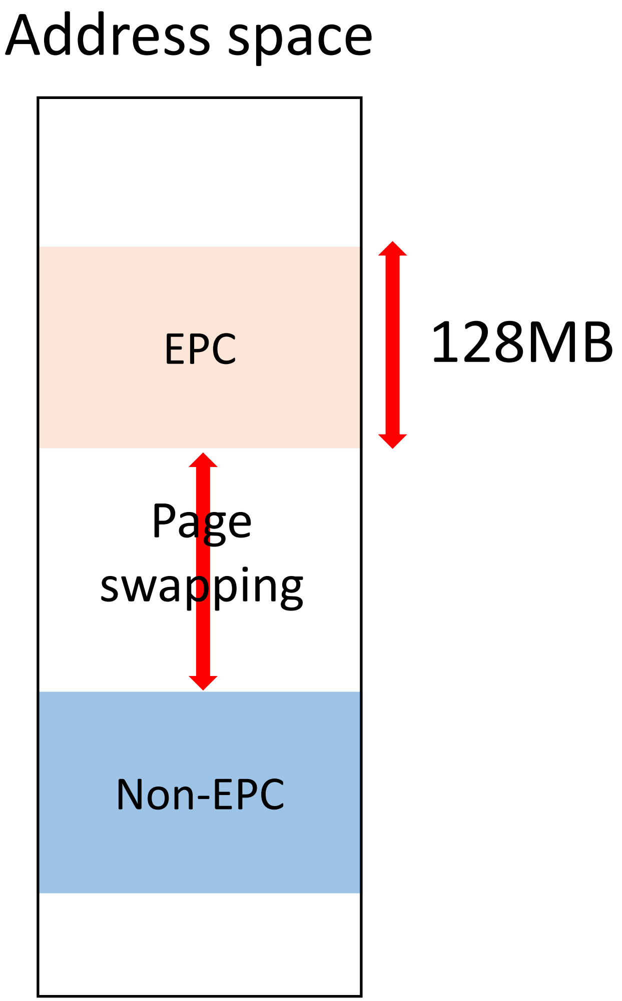
Limited memory space in enclave (EPC)

- Only ~94MB available for applications
- To support larger memory region, SGX supports secure page swapping
  - Significant overhead (about 40K cycles) for an EPC miss
  - Only about 200 cycles for an EPC hit
- Many memory-consuming operations in databases

Huge cost from enclave interaction

- Syscalls cannot be executed in enclave, causing expensive cost to exit the enclave via OCall functions (about 8K cycles)
  - TLB flushing, security checks, etc.
- Host process can only invoke enclave via ECall functions

How to significantly reduce EPC page swapping and enclave interaction?



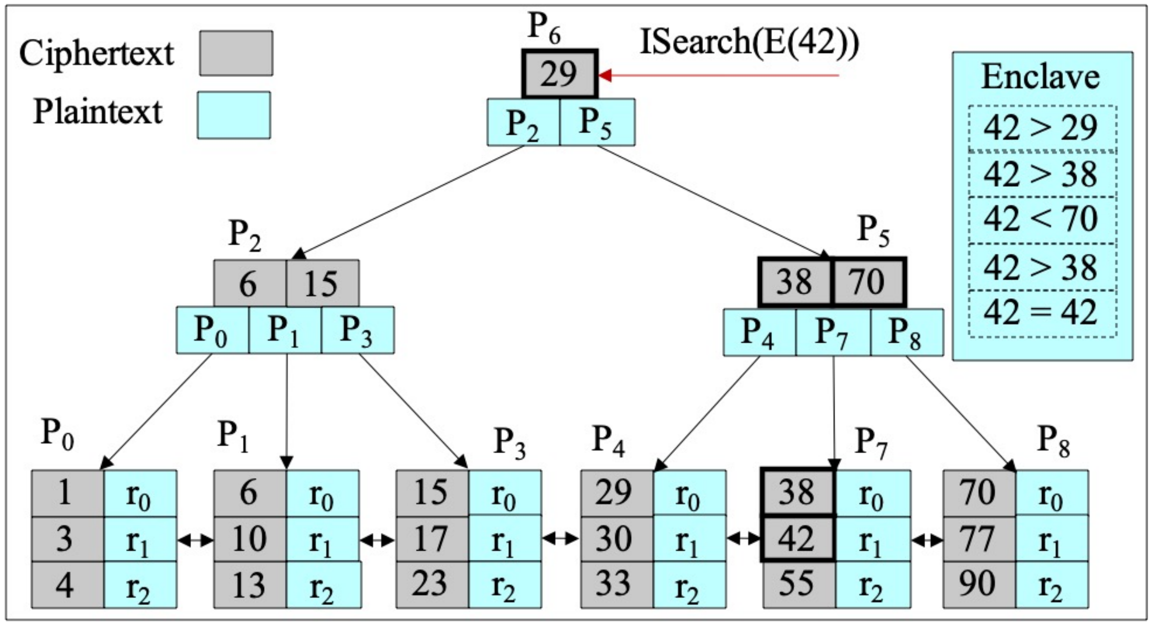
02 Exploration to a Broader Design Space

Strawman: B$^+$-tree with Encrypted Keys

達摩院
ALIBABA DAMO ACADEMY

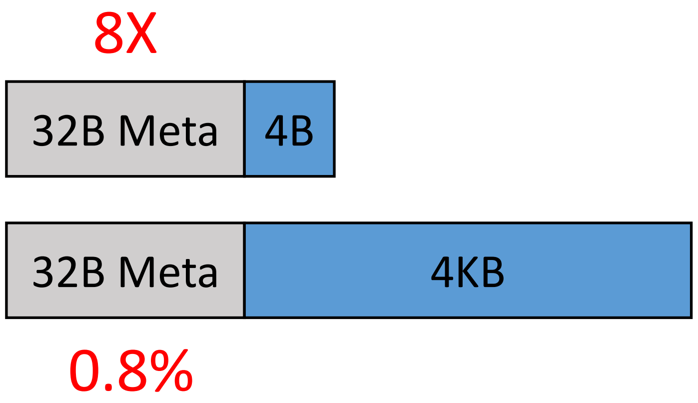
Structure overview

- Unchanged logical semantics
- Most index processing logic remains unaffected
  - E.g., node split and merge
- Decrypt keys and return *cmp* results in plaintext
  - E.g., ISearch(E(42))



Limitations

- Frequent enclave interaction
  - E.g., 5 ECalls are required for ISearch(E(42))
- High overheads on storage
  - Huge storage amplification for smaller encryption granularity
- Severe information leakage
  - Key orders, parent-child relationships, etc.

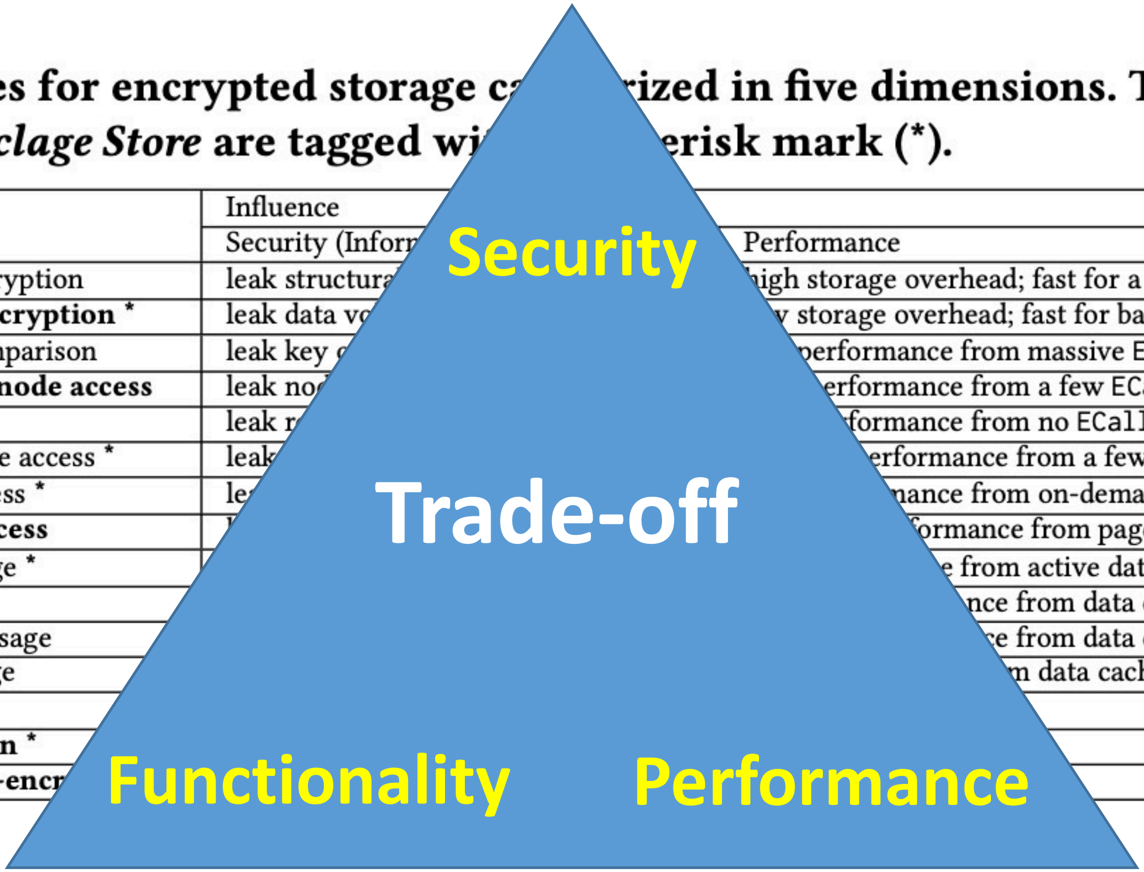


Exploration to a Broader Design Space

達摩院 ALIBABA DAMO ACADEMY

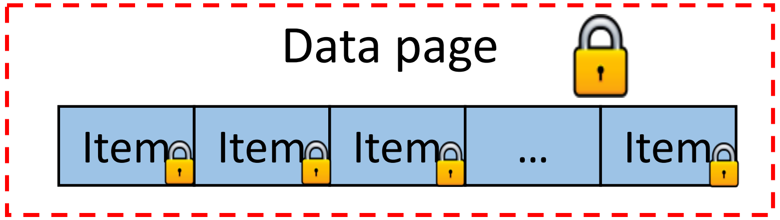
Table 1: Possible design choices for encrypted storage ca rized in five dimensions. The choices made for *Enclage Index* are bolded and the choices for *Enclage Store* are tagged wi erisk mark (*).

Design Dimension	Design Choice	Influence		
		Security (Inforr	Performance	Functionality
Encryption Granularity	item-level encryption	leak structura	igh storage overhead; fast for a single read	can fetch data w/o enclave
	page-level encryption *	leak data vo	y storage overhead; fast for batched small reads	all data access must be in enclave
Execution Logic in Enclave	index: key comparison	leak key c	performance from massive ECalls	can split or merge node w/o enclave
	index: index node access	leak nod	erformance from a few ECalls	all index access must be in enclave
	table: none	leak r	formance from no ECall	can fetch or scan record(s) w/o enclave
	table: data page access *	leak	erformance from a few ECalls	all record access must be in enclave
Memory Access Granularity	item-level access *	le	nance from on-demand read	require small footprint in enclave
	page-level access	l	formance from page copy	require large footprint in enclave
Enclave Memory Usage	minimum usage *		e from active data fetching	no EPC capacity requirement
	fixed usage		nce from data caching	low EPC capacity requirement
	proportional usage		ce from data caching	high EPC capacity requirement
	unlimited usage		m data caching	high EPC capacity requirement
Record Identity Protection	no action			no influence
	***rid* encryption** *			only useful in some settings
	ciphertext re-encr			only useful in some settings

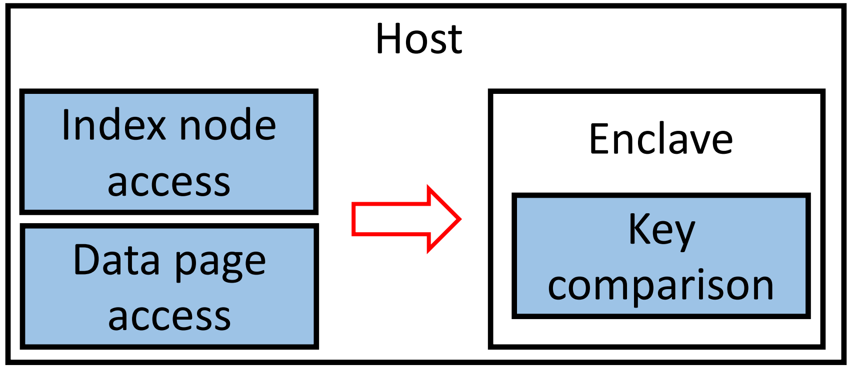


Design Space

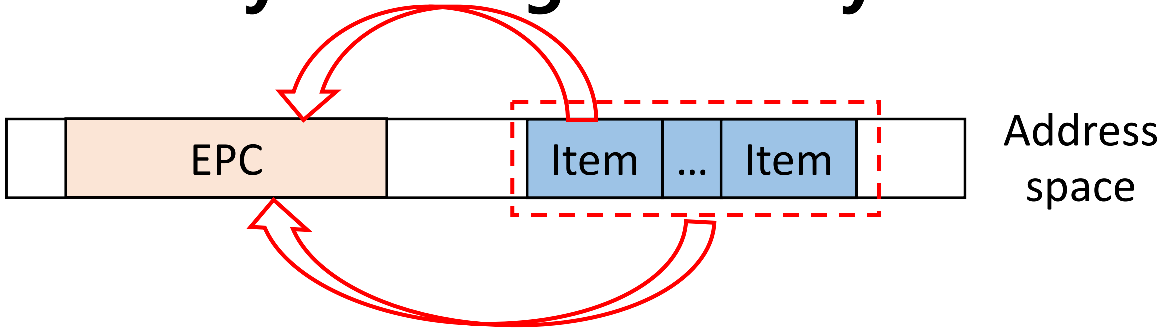
Encryption granularity



Execution logic in enclave



Memory access granularity



Enclave memory usage

- min usage
- Fixed usage
- Proportional usage
- Unlimited usage

Record identity protection

- No action
- Rid encryption
- Ciphertext re-encryption

Overall Architecture of Enclage

Enclage - An enclave-native encrypted storage engine

- Enclage Index: a B^+-tree-like index
- Enclage Store: a heap-file-like table store

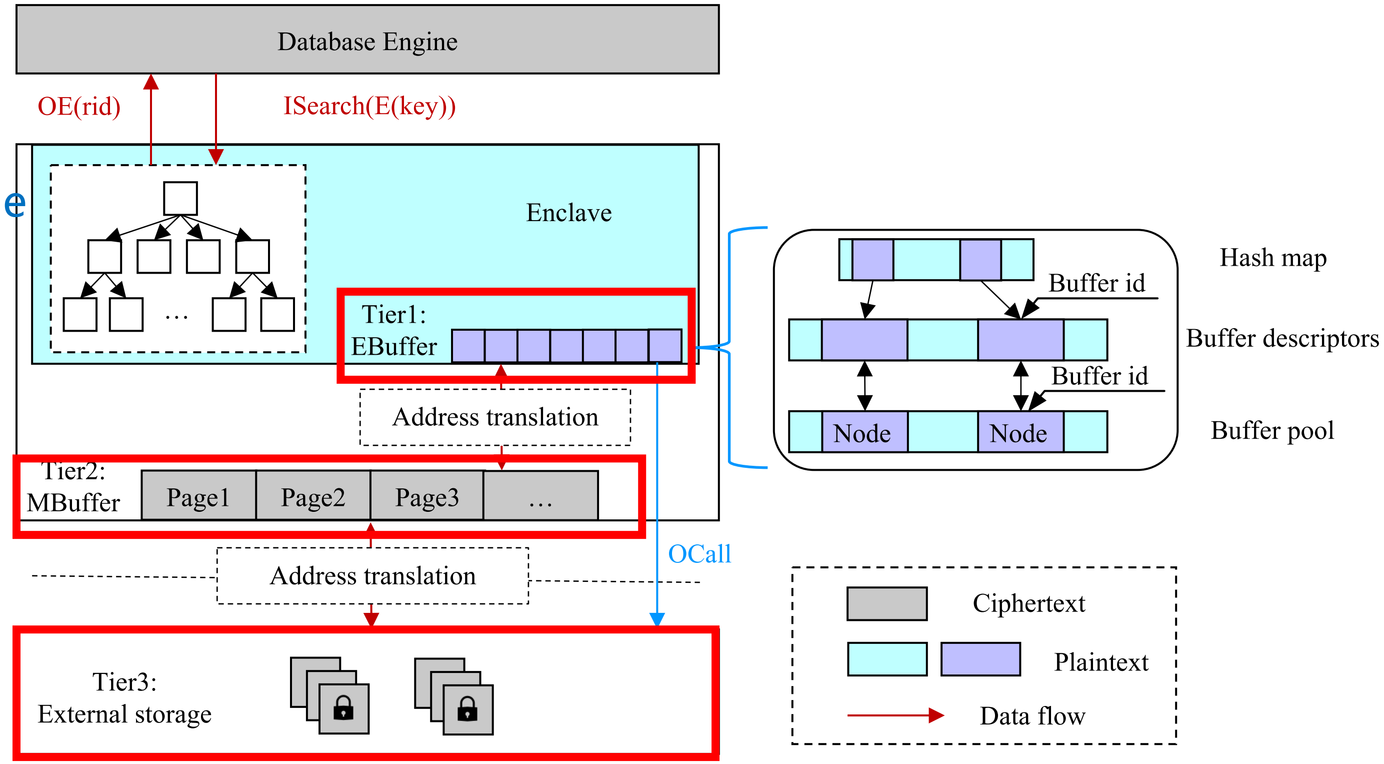
Design choices made in Enclage

	Enclage Index	Enclage Store
Encryption Granularity	Page-level encryption	Page-level encryption
Execution logic in enclave	Index node access	Data page access
Memory access granularity	Page-level access	Item-level access
Enclave memory usage	Fix usage	Minimum usage
Record identity protection	Rid encryption/ciphertext re-encryption	Rid encryption/ciphertext re-encryption

Overview of Enclage Index

Hierarchical architecture

- Tier1: EBuffer
  - trusted buffer in enclave
  - An unencrypted index node per page
- Tier2: Mbuffer
  - untrusted buffer in memory
  - Several encrypted nodes per page
- Tier3: External storage



Optimizations

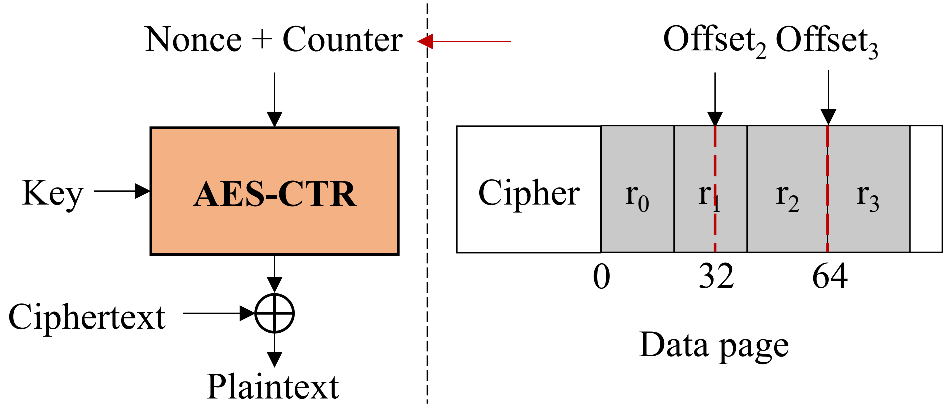
- Reduction of EPC page swapping
- Mitigation of enc/dec costs
- Avoidance of unnecessary OCalls

Enclage Store & Delta Decryption

Enclage Store

- A heap-file-like table store
- Adopts append-only strategy
  - The active page: holds recently arrived records
  - Lower data locality
- Retrieves a record
  - Loads t
  - Decryp target record

Time consuming!



Delta decryption protocol

- Built on top of AES-CTR mode
  - Allows a small block within a large cipher be solely decrypted
- Executing a TGet operation
  - Locates the page in MBuffer and loads it to enclave
  - Calculates the counter for the record and construct the IV
  - Only decrypts the target record

Lower cost!

03 Evaluation

Experimental Setup

Hardware platform

Server Node	Intel SGX (~94MB EPC), Intel Core E7-1270 (4 cores), 64GB DRAM
System	Red Hat 6.4.0 with Linux 4.9.135 kernel, SGX driver and SGX SDK 2.6

Compared Systems

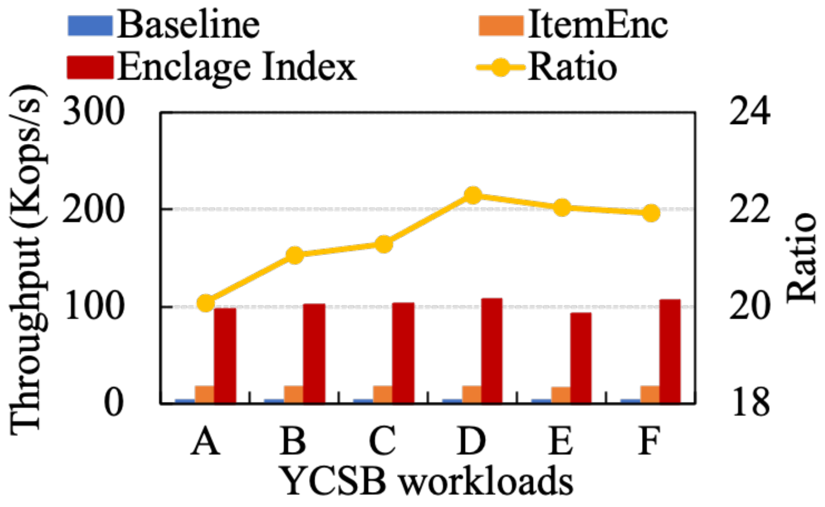
- Enclage Index

	Encryption Granularity	Location of execution logic
Baseline	Item	Outside of the enclave (Untrusted)
ItemEnc	Item	With the enclave (Trusted)
ShieldStore (hash-based)	Item	Outside of the enclave (Untrusted)
Enclage Index	Page	With the enclave (Trusted)

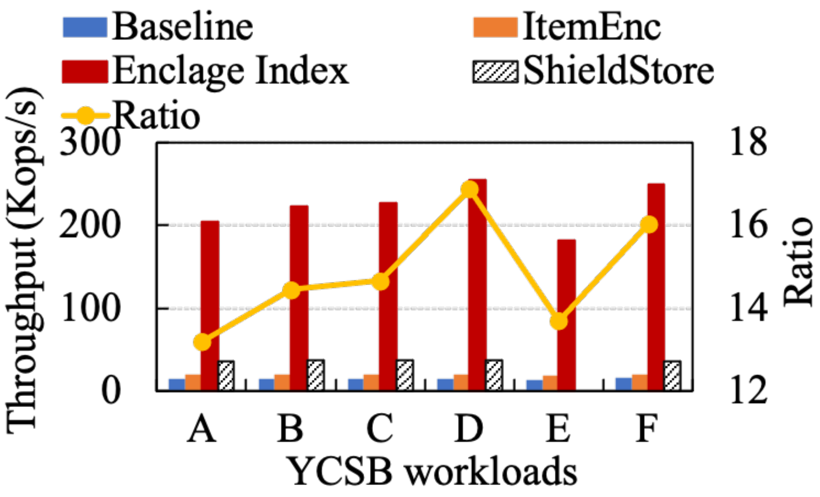
- Enclage Store

Item-level	Heap file containing encrypted records
Page-level	Heap file containing encrypted pages
Delta-enc	On top of Page-level, adopts the delta decryption protocol (AES-CTR)

Overall Performance



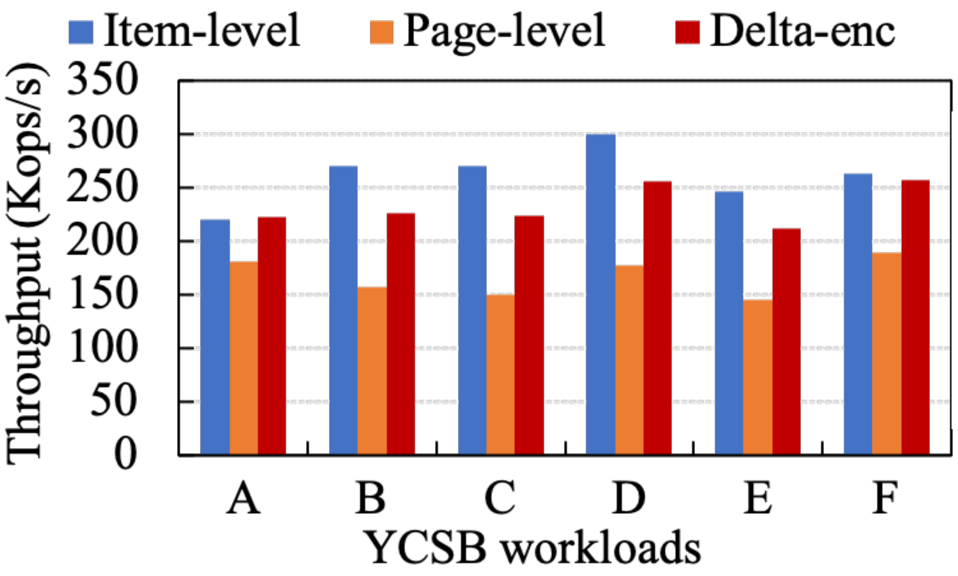
Switch mode



Switchless mode

- Enclage Index achieves about 100Kops/s, and outperforms Baseline (20.04x) and ItemEnc (5.34x).
  - ✓Only 1 ECall during each operation, and each accessed node is decrypted at most once
- ❑Enclage Index also achieves better performance, compared to Baseline (13.19x) , ItemEnc (9.69x) and ShieldStore (7-12x).
  - ✓The more frequent the ECall is invoked, the greater the performance gain from the mode
  - ✓Frequent decryption in ShieldStore

Different Decryption Protocols



	16B	32B	64B	128B	256B
Item-level (GB)	0.67	0.89	1.36	2.29	4.09
Page-level (GB)	0.23	0.45	0.91	1.85	3.81
Ratio	2.99	1.98	1.50	1.24	1.07

- When a access miss occurs,
  - ✓ Page-level: load and decrypt the entire desired page
  - ✓ Delta-enc: extract and decrypt the desired record (1.40x)
  - ✓ Item-level: directly extract the encrypted record (1.57x)

More experiments: Please check our paper

04 Summary & Conclusion

Summary & Conclusion

- ✓Data confidentiality is one of the biggest concerns that hinders enterprise customers from moving their workloads to the cloud.
- ✓Though TEEs provide a powerful building block, practical designs of TEE-based encrypted databases have not been well explored.
- ✓Our contributions:
  - Provides a comprehensive exploration of possible design choices for building an enclave-based encrypted database storage
  - Proposes Enclage, an enclave-native storage engine that makes practical trade-offs
- ✓Enclage improves the throughput by 13x and the storage efficiency by 5x.

Thanks

Email: yuanyuan.sun@alibaba-inc.com