Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data

PRESENTED BY:
NICHOLAS BURTON

WRITTEN BY:
TYLER HUNT
ZHITING ZHU
YUANZHONG XU
SIMON PETER
EMMETT WITCHEL

Big Data, Your Information, and Security

Related Work

- Haven
- MiniBox
- OverShadow and others

Issues

Ryoan

Threat Model

In Scope

- User is distrustful of the service
- Service can outsource work, becoming a user
- User does not trust software at any privilege level
- Covert software channel attacks

Out of Scope

- DoS attacks
- Hardware limitation based side-channel attacks
- Execution time based side-channel attacks

Hardware Limitations

- SGX page faults
- Cache Timing
- Address Bus Monitoring
- Processor Monitoring

Main Components of Ryoan

Intel SGX, and how Ryoan uses it

- Available on all recent Intel processors
- Provides a Hardware Isolated Execution Environment via Enclaves
- SGX provides enclave identities (hash of enclave's initial state)
- Assume initial state of an enclave cannot be impersonated
- Ryoan checks enclave's identity to ensure it is of Ryoan origin

Google Native Client, and how Ryoan uses it

- Sandbox for running x86/86-64 code using fault isolation
- Two parts: verifier, service runtime
- Verifier guarantees a module cannot break out of NaCl.
- Intercepts System Calls and passes them to Ryoan instead of the OS

How is it put together?

A Single Ryoan Instance

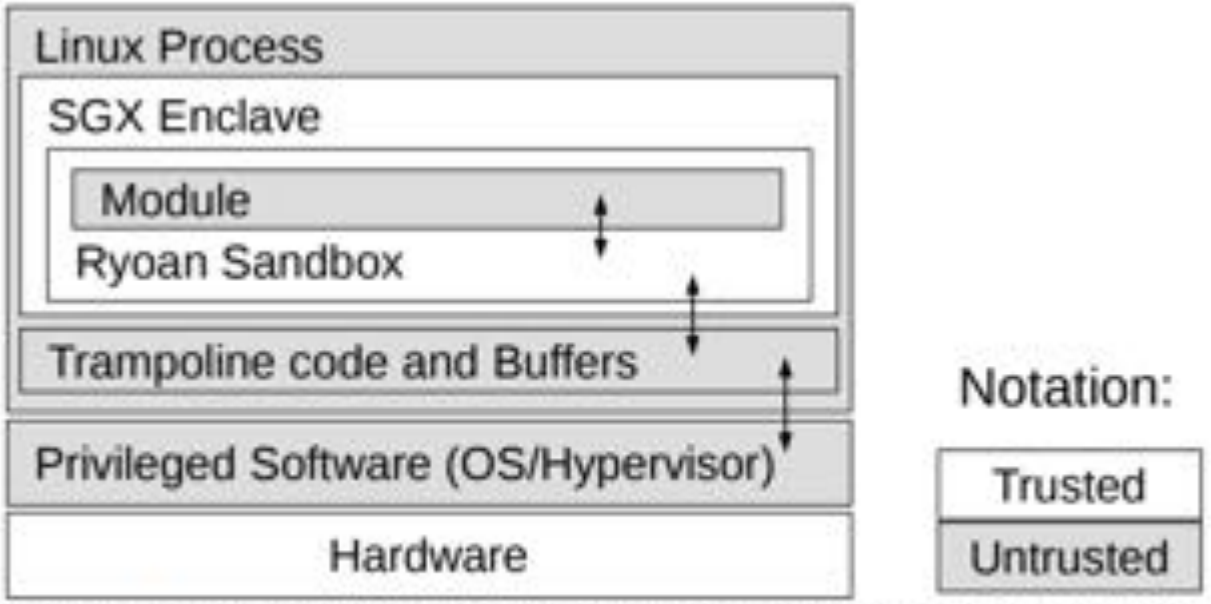


Figure 1: A single instance of Ryoan's distributed sandbox. The privileged software includes an operating system and an optional hypervisor.

A Single Ryoan Instance

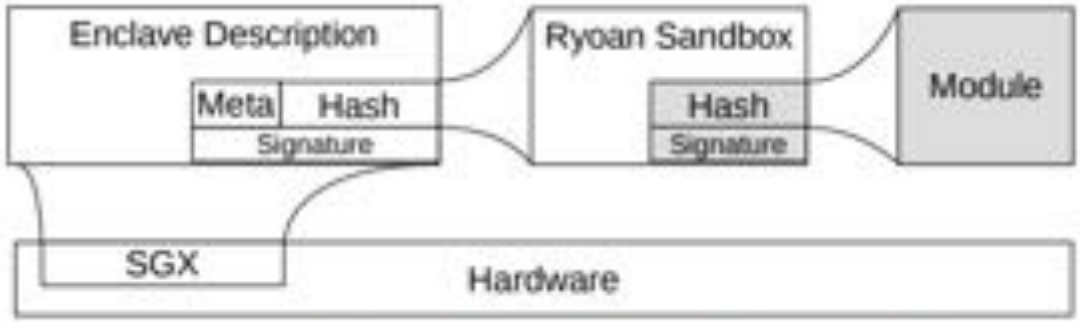


Figure 2: The Ryoan chain of trust. SGX hardware attests that a valid instance of Ryoan is executing (Hash) with an intended SGX configuration (Meta). Ryoan ensures that the expected binary is loaded with a signed hash from the software provider (grey).

Directed Acyclic Graph

- A finite graph, with no loops
- Improved security

Labels and DAG

- Use of labels to mark secret data, and which previous enclaves have seen it
- Each module can remove its own label
- Labels together are a tag

Figure 3: Ryoan's distributed sandbox. Ryoan instances manage labels on data and modules. The user's tag is propagated to all modules, making them confined after receiving input; For example, 23andMe's tag is kept when it outsources to Amazon Machine Learning to prevent leaking secrets from 23andMe.

Labels and DAG

- Non-confining and Confining Labels
- Audit Trail
- Data Oblivious Communication

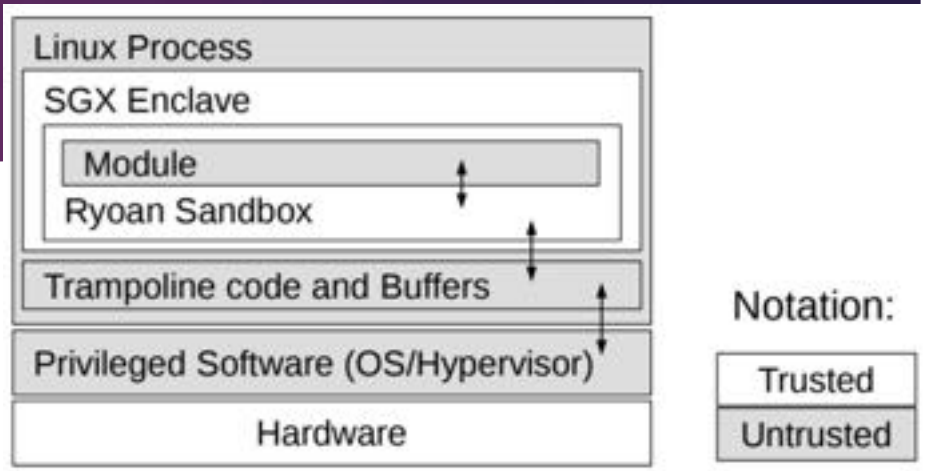


Figure 3: Ryoan's distributed sandbox. Ryoan instances manage labels on data and modules. The user's tag is propagated to all modules, making them confined after receiving input; For example, 23andMe's tag is kept when it outsources to Amazon Machine Learning to prevent leaking secrets from 23andMe.

Individual Module Security

Module Security

- Verifier ensures code adheres to Ryoan format
- Memory access is constrained to the provided memory for the module
- NaCl guarantees still stand
- Ryoan disallows SGX instructions in modules
- Single processing opportunity per unit of work/input
- After processing and sending output, Ryoan instance is deleted or reset

Module Security

- When a Ryoan instance is confined it disallows communication with the OS
- Ryoan provides a system API
- Virtual File System
- mmap calls handled by Ryoan API to satisfy memory allocation
- Fixed and Quantized processing times
- System calls to OS intercepted by Ryoan-libc
- Checkpoint Module Resetting

Checkpoint Module Resetting

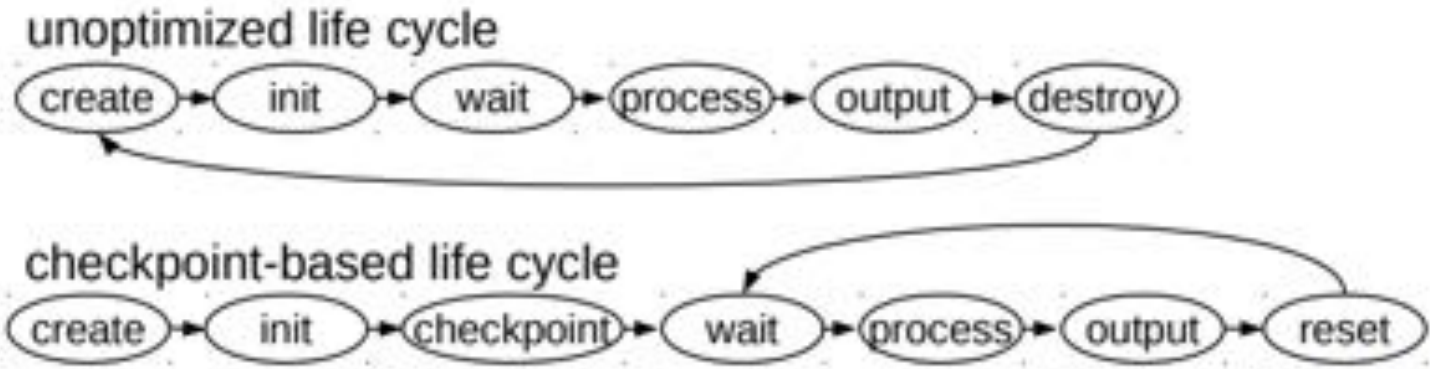


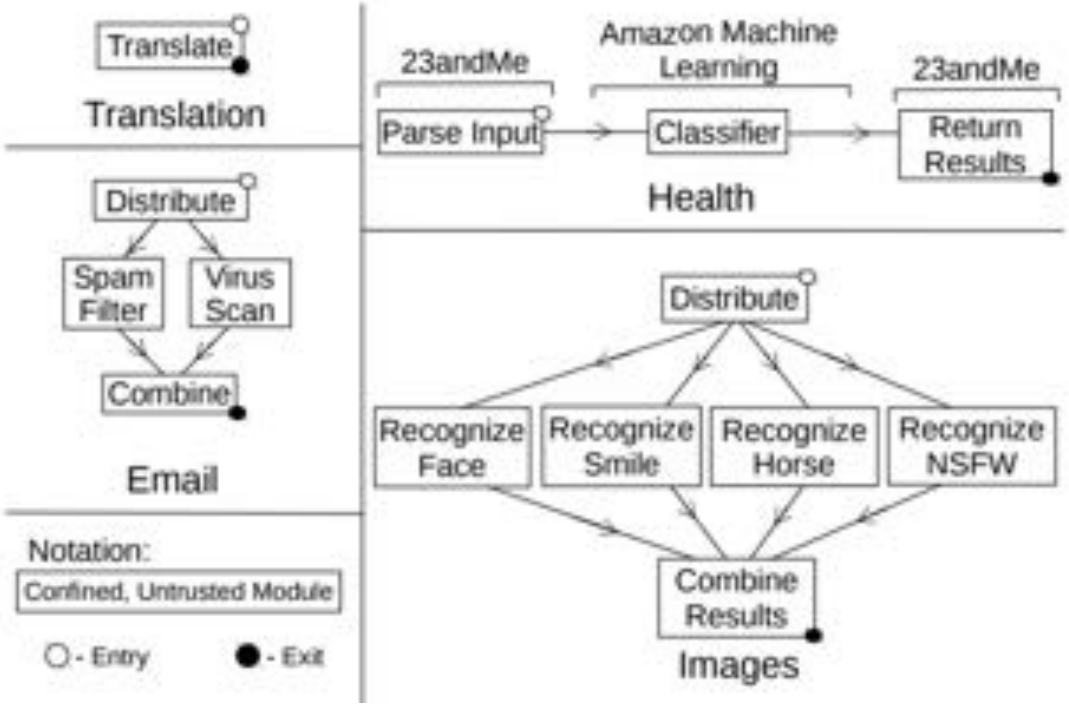
Figure 4: Instance life cycle: unoptimized vs checkpoint-based.

Implementation

Implementation

- Ryoan-libc
- Module Address Space
- I/O control
- Key Establishment Between enclaves
- Checkpoint Code

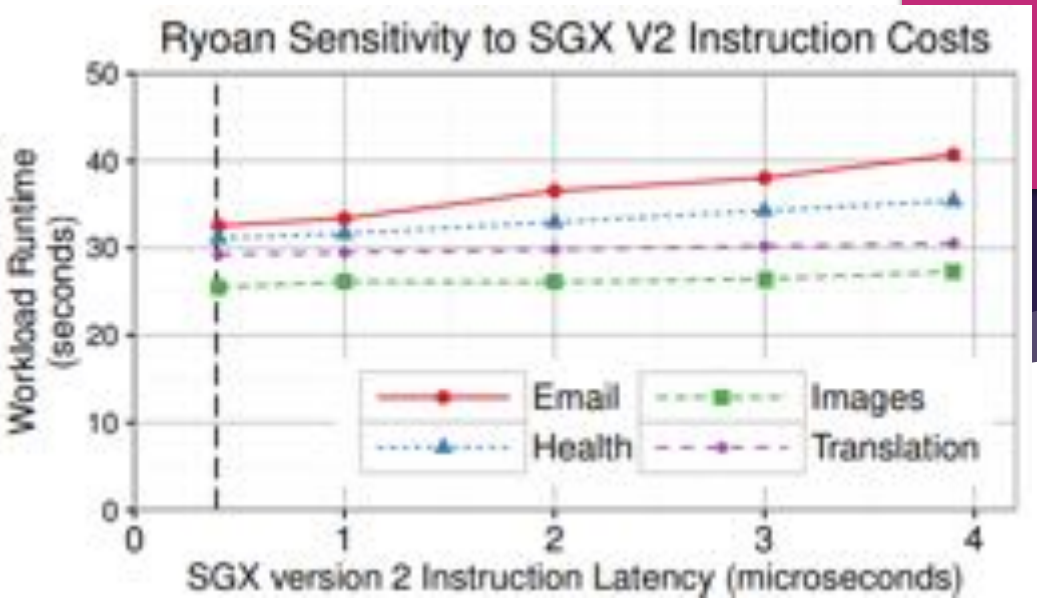
Examples and Use cases

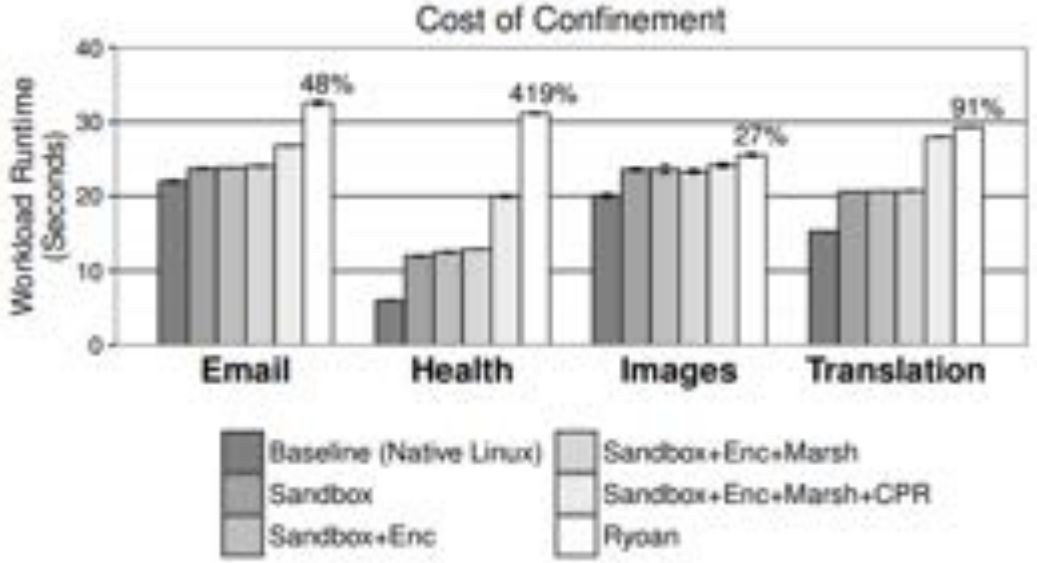


Overhead

Overhead

		Load Size (MB)	Inited Size (MB)	Init Time (sec)	CPU Time (sec)	CPR Size	Sys. Calls	PF	Intrp
Email	Distribute	18.0	18.1	0.59	1.32	11.6MB	47k	60k	473
	DSPAM	19.6	273.5	11.15	22.10	45.3MB	1.29m	1.81m	6k
	ClamAV	21.1	403.9	24.96	29.17	83.3MB	247k	423k	7k
	Combine	18.0	18.1	0.59	0.11	16KB	12k	2k	77
Health	LoadModel	19.3	19.4	0.58	12.52	28KB	82k	280k	56k
	Classifier	19.3	19.4	0.58	18.23	36KB	1.84m	359k	151k
	Return	18.0	18.1	0.59	6.77	16KB	668K	162k	3k
Images	Distribute	18.0	18.1	0.59	0.42	632KB	2k	2k	36
	Recognize	26.6	27.1	0.63	24.79	83.2MB	88k	174k	6k
	Combine	18.0	18.1	0.59	0.36	2.5MB	14k	3k	129
Translation		25.3	386.9	2.34	26.65	29.1MB	303k	248k	8k
Email Input		250 emails, 30% with 103KB-12MB attachment							
Health Input		20,000 1.4KB Boolean vectors from different users							
Images Input		12 images, sizes 17KB-613KB							
Trans. Input		30 short paragraphs, sizes 25-300B, 4.1KB total							





Questions?