SecLab
COMPUTER SECURITY LABORATORY

19th Sep., 2024

STYX: Collaborative and Private Data Processing Using TEE-Enforced Sticky Policy

Shixuan Zhao
PhD Candidate @ SecLab
CSE, The Ohio State University
zhao.3289@osu.edu

THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

Acknowledgement

This project is funded by NSF, under the CDCC Frontier grant.

Policy-Attached Data: Challenges

Conventionally...

- Embedded in code

Sticky Policy

- Sticky policy: policy attached to data
- Flexible policy customised for every data

⚠Problems

- Data-in-use protection?
- Data lifecycle protection?
- Dynamic collaboration?

Policy-Attached Data: Challenges

Data-in-Use Protection

- Identity-Based Encryption [1]
- Attribute-Based Encryption [2][3]
- Proxy Re-Encryption [4]

⚠Problem

They all have to trust the application!

[1] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In Annual international cryptology conference, pages 213–229. Springer, 2001.
[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006, pages 89– 98. ACM, 2006.
[3] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute- based encryption. In 2007 IEEE symposium on security and privacy (SP'07), pages 321–334. IEEE, 2007.
[4] M. Green and G. Ateniese. Identity-based proxy re-encryption. In Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007. Proceedings 5, pages 288– 306. Springer, 2007.

Policy-Attached Data: Challenges

Data Lifecycle Protection

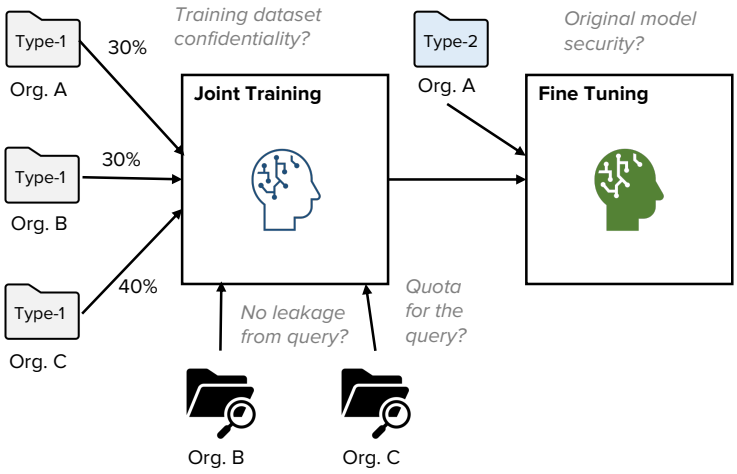
- Handles derived data
- How to protect the output?
- What is allowed in the output?
- What policy should be attached to the output?

⚠ Problem

How to achieve the above stuff?

Motivating Example

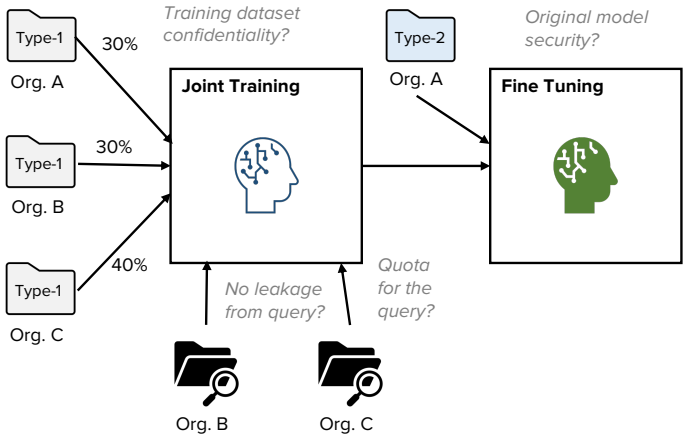
Hospitals train a model to classify cancer



Motivating Example: Data-in-Use Protection

Training

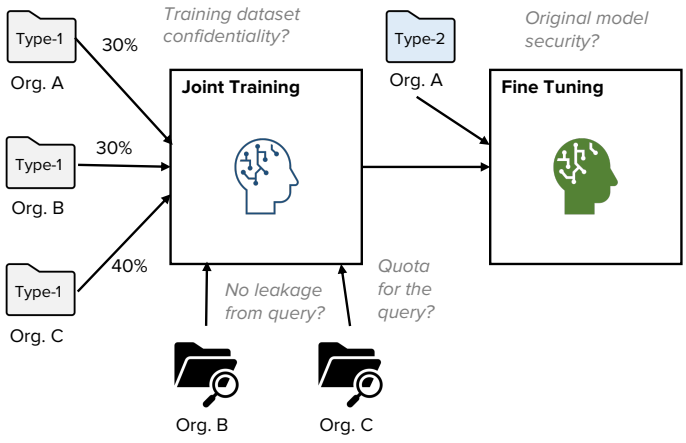
- How to make sure my dataset won't be over a limit? (e.g. 30%)
- How to guarantee the confidentiality of my training dataset?



Motivating Example: Data Lifecycle Protection

Model Usage

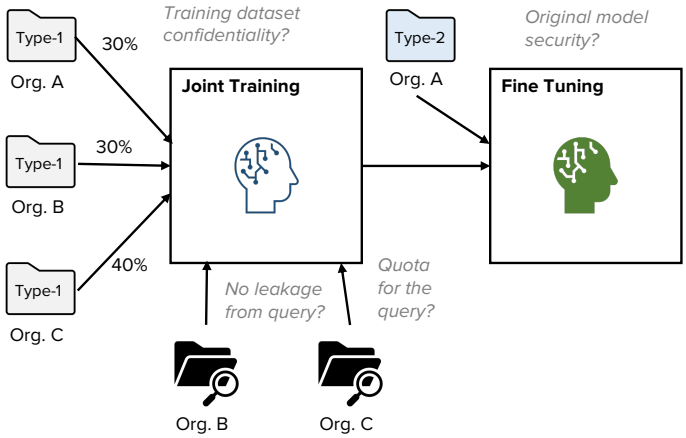
- Jointly owned
- How to limit fair-use? (e.g. Quota according to training set contribution)
- How to safeguard the model?



Motivating Example: Data Lifecycle Protection

Derived Model

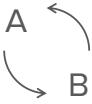
- Is it even allowed?
- Exclusively owned with jointly owned bits
- How to ensure the security of the original model?



Problem

⚠ Problems

- Data-in-use protection?
- Data lifecycle protection?
- Dynamic collaboration?



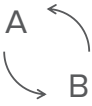
Problem

⚠ Problems

- Data-in-use protection?
- Data lifecycle protection?
- Dynamic collaboration?

Solutions

- TEE hardware-enforced trust and isolation
- Cryptographically-encrypted transmission
- Sandboxed access
- Policy enforced for both input and output



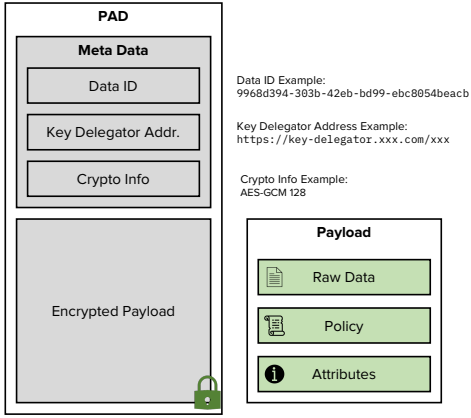
What does a PAD look like?

Plaintext Contents

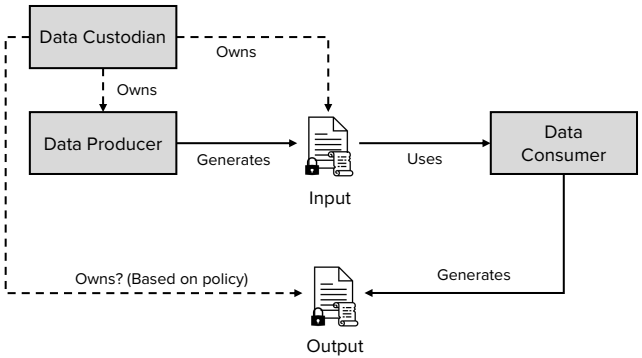
- Information to get the key
- How to decrypt

Ciphertext Contents

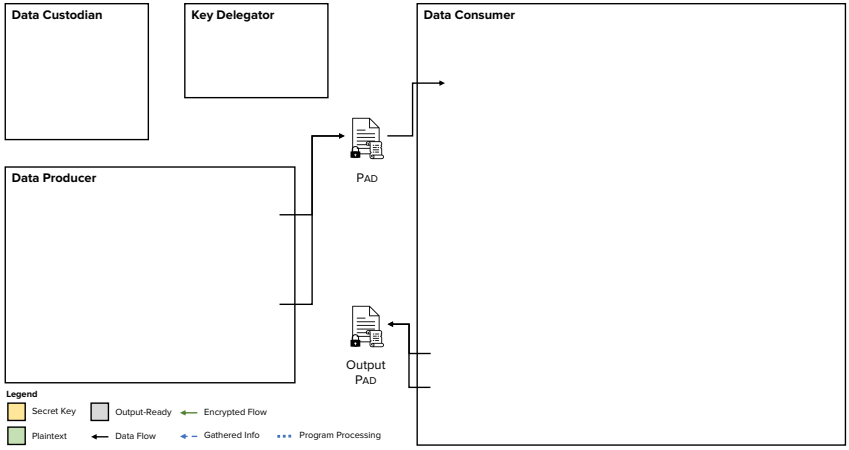
- Everything else!
- Data & policy
- Data attributes used by policy



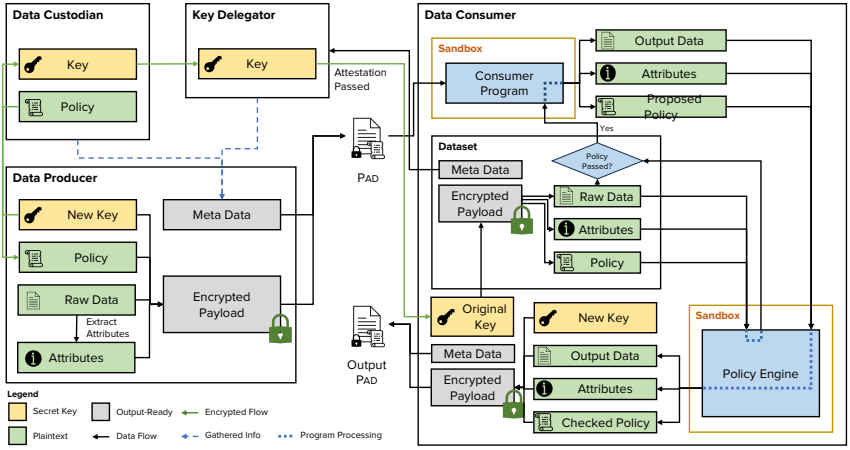
Modelling the Workflow



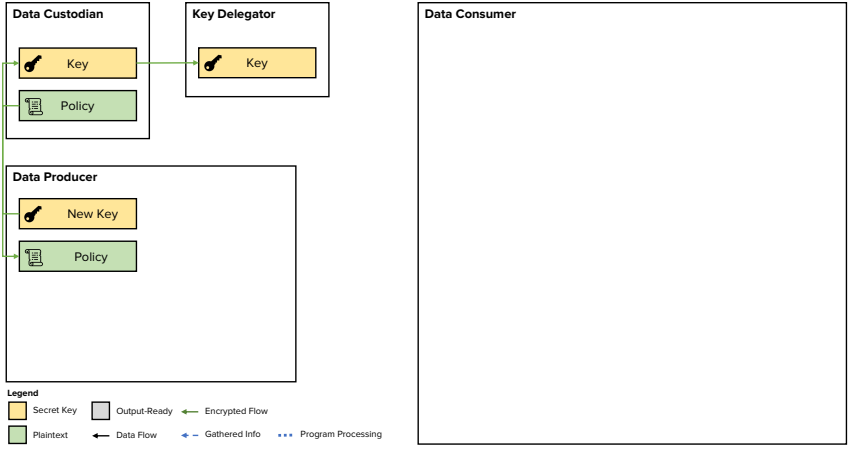
Modelling the Workflow



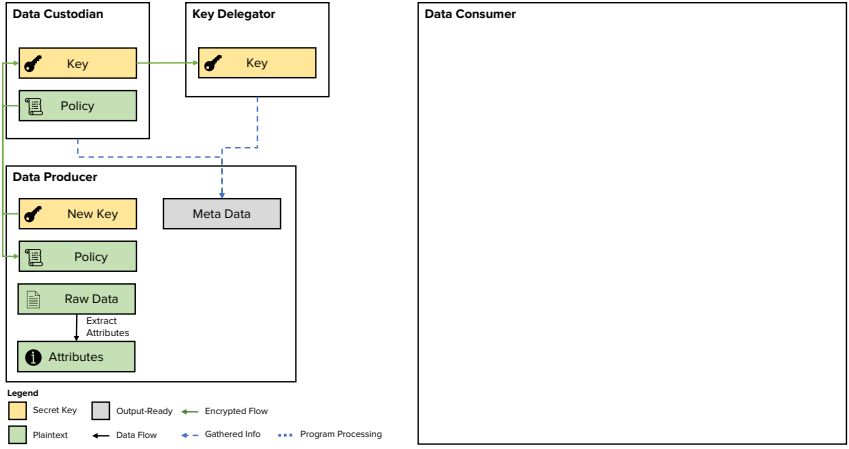
Modelling the Workflow



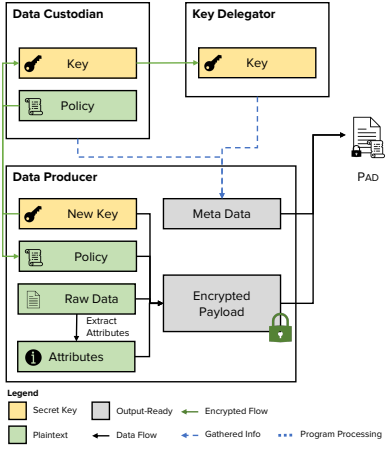
Workflow: Step-by-Step



Workflow: Step-by-Step

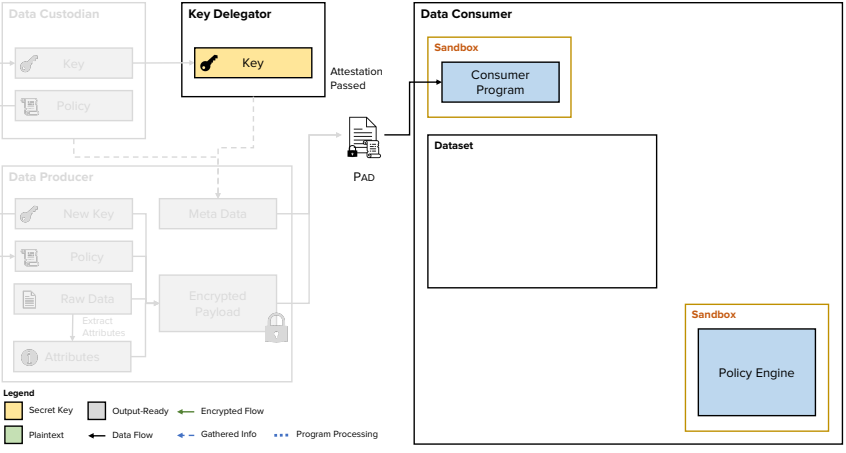


Workflow: Step-by-Step

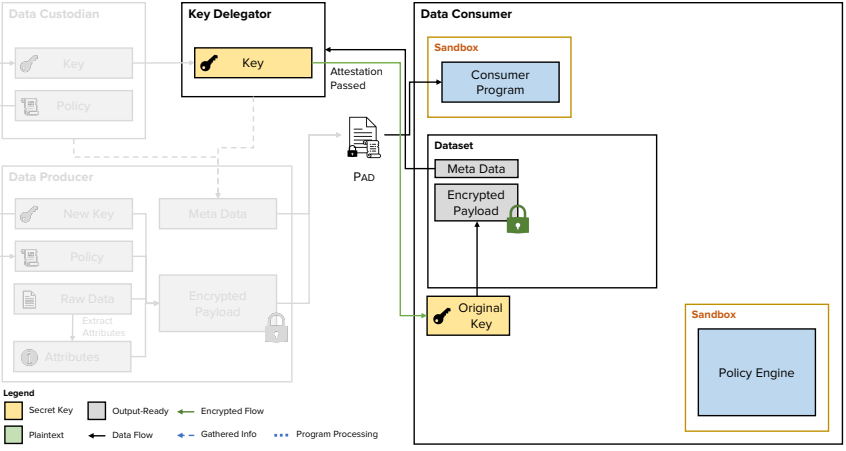


Data Consumer

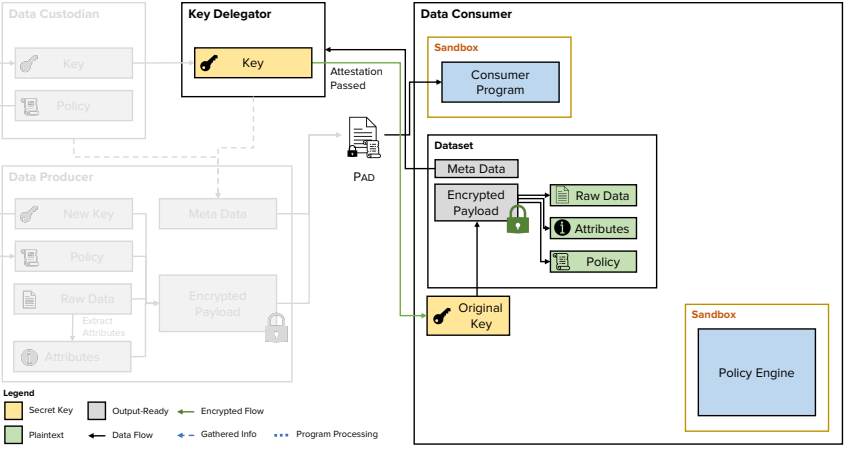
Workflow: Step-by-Step



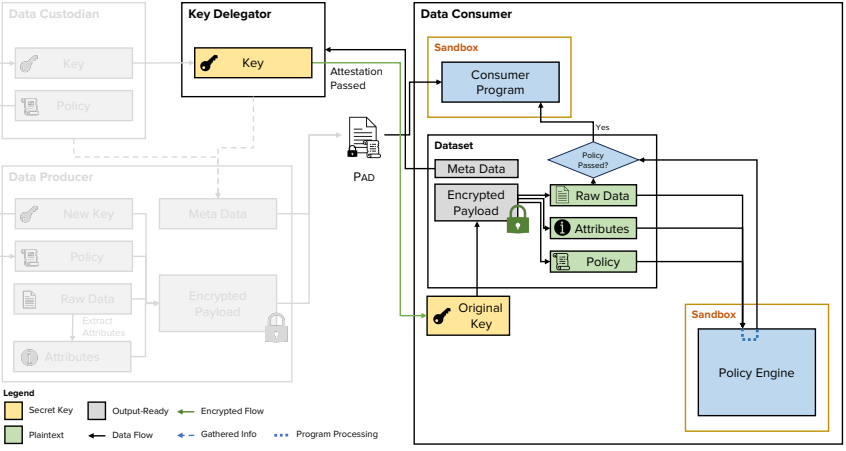
Workflow: Step-by-Step



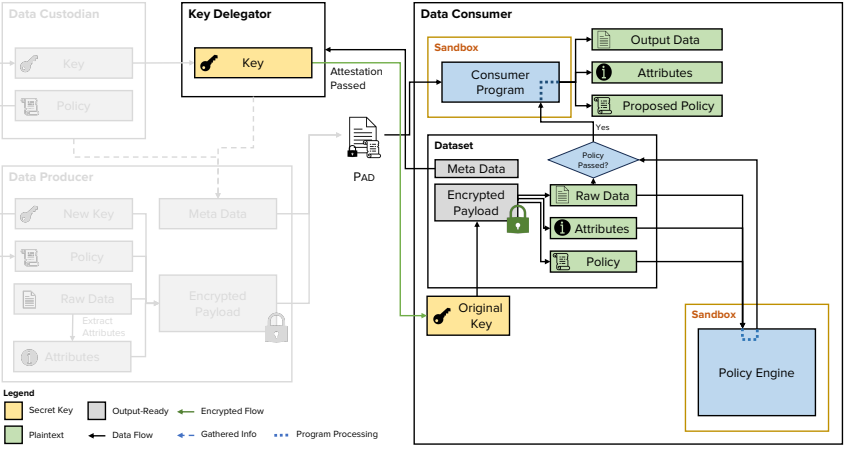
Workflow: Step-by-Step



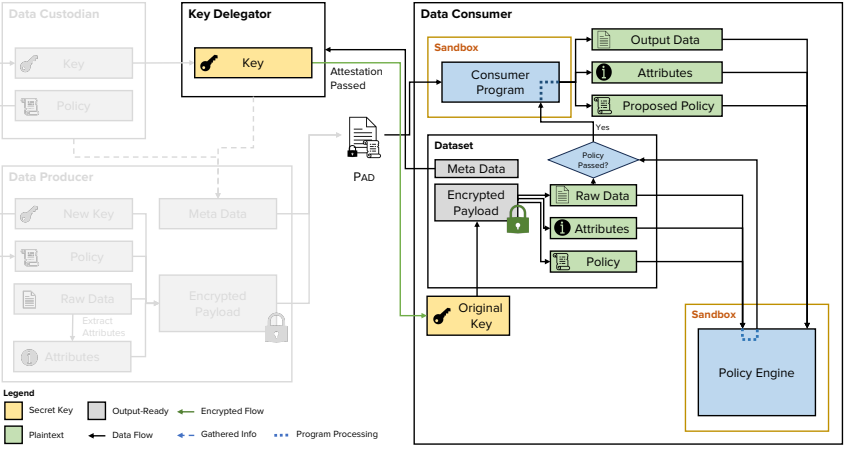
Workflow: Step-by-Step



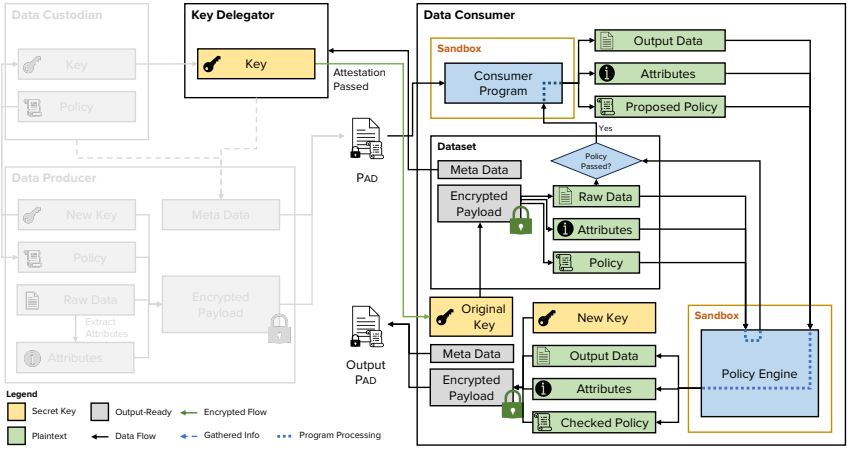
Workflow: Step-by-Step



Workflow: Step-by-Step



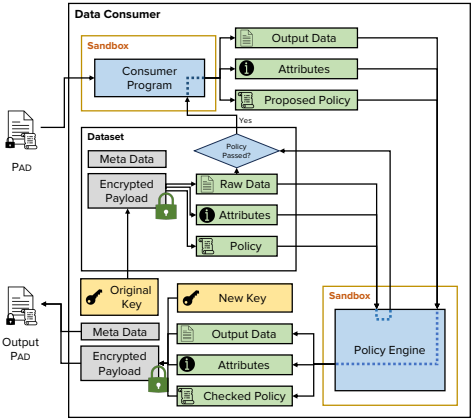
Workflow: Step-by-Step



Consumer: From Model to Architecture

Model

- Sandboxing
- Policy engine
- I/O protection
- Trust



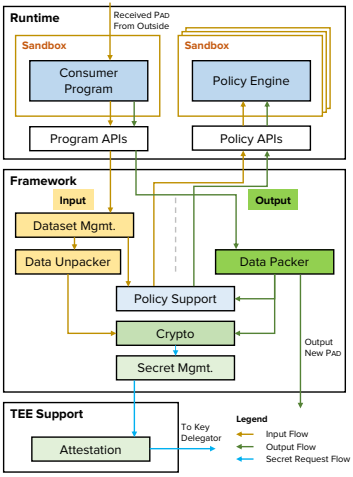
Consumer: From Model to Architecture

⚠A middleware

A middleware with a runtime to achieve the security guarantee

Architecture

- Sandboxing via runtime
- Policy engine using runtime
- I/O protection by middleware
- Trust with remote attestation



Implementation

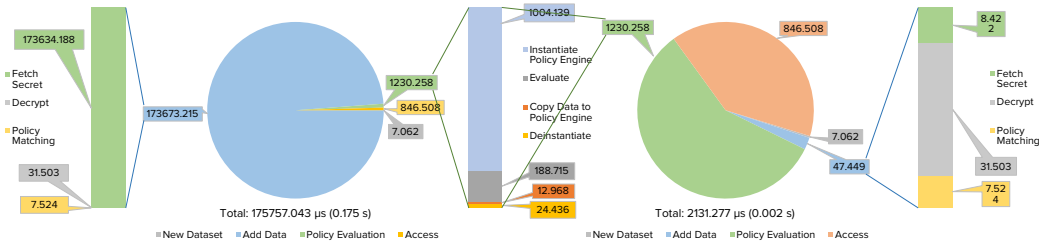
Framework

- Architecture-independent
- APIs for different TEEs & runtimes

Prototype

- Intel SGX as TEE
- WebAssembly as runtime

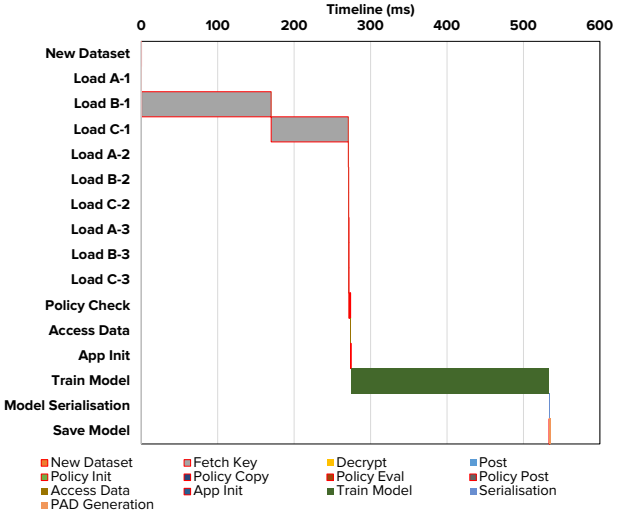
Data Access Overhead



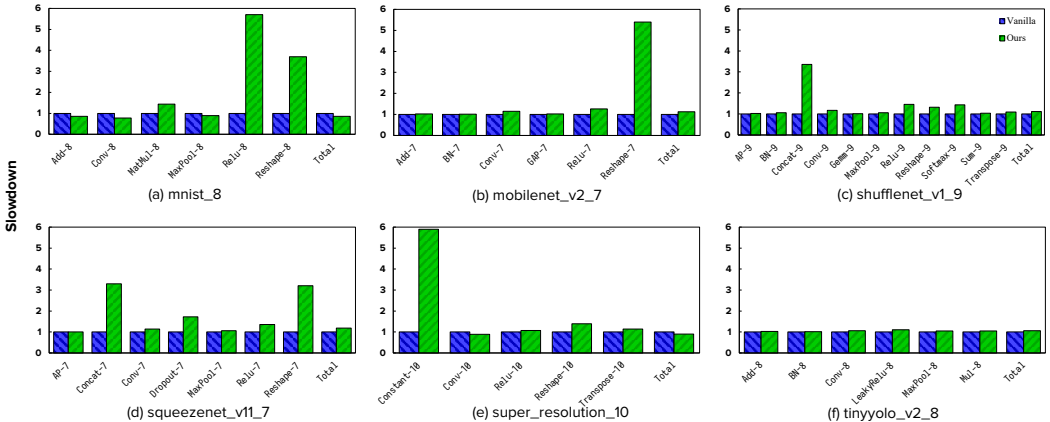
(a) Data requires fetching secret from remote

(b) Data with secret on local

SVM Training



libonnx Benchmarks with Native Libs



Other Use Cases

Joint Data Analysis

- Cross-bank fraud analysis
- Output must be limited to original bank

Private Database Query

- Paid query to proprietary databases
- Query must not be leaked

Smart Home IoT

- Connected security services (e.g. ADT)
- Home insurance
- Access limited to data type/specific timing

Limitations and Future Works

Hardware Accelerators

- GPU TEEs (e.g., NVIDIA H100)
- How to apply the method to GPU?
- Trusted I/O

Trade-Offs in Runtime Sandboxing

- Other methods (e.g., SFI)
- AoT & JIT compilation
- Native libs

Formal Verification of Middleware

- Full stack verification required
- Verified runtime (e.g., MesaPy)

Q&A

Source Code
Will be available after published

SecLab @ OSU
https://go.osu.edu/seclab

Shixuan Zhao's Homepage
https://nskernel.org