RSAConference2019

San Francisco | March 4–8 | Moscone Center

BETTER.

SESSION ID: TECH-T09

# Update on Confidential Computing

**Olya Ohrimenko**

Researcher
Microsoft Research
Microsoft

#RSAC

# Cloud computing

Pay-per-use model:

- storage
- computing
- platform as a service

Additionally:

- physical security
- replication

# Customer concerns with data security in the cloud

Malicious privileged admins or insiders

Hackers exploiting bugs in the Hypervisor/OS of cloud fabric

Third parties accessing it without customer consent

Data breach regularly tops list for top cloud threat

# Outline: Confidential Computing

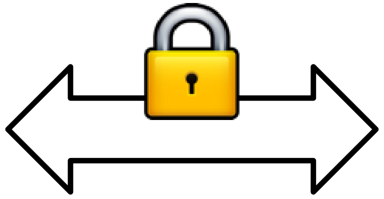
- Protect data during computation:
  - with trusted execution environments (TEEs)

- Scenarios:
  - confidential consortium blockchains
  - multi-party machine learning

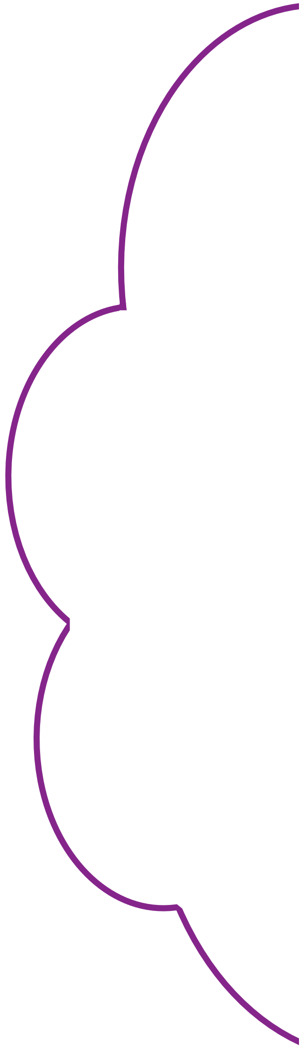
- Guarantees beyond TEE isolation:
  - integrity and privacy in multi-party machine learning
  - memory side-channel mitigation

# Towards Confidential Cloud Computing

Data Encryption

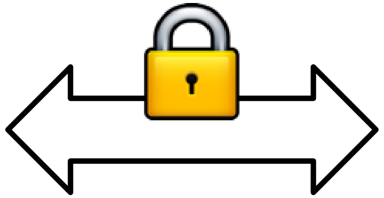


Network Encryption

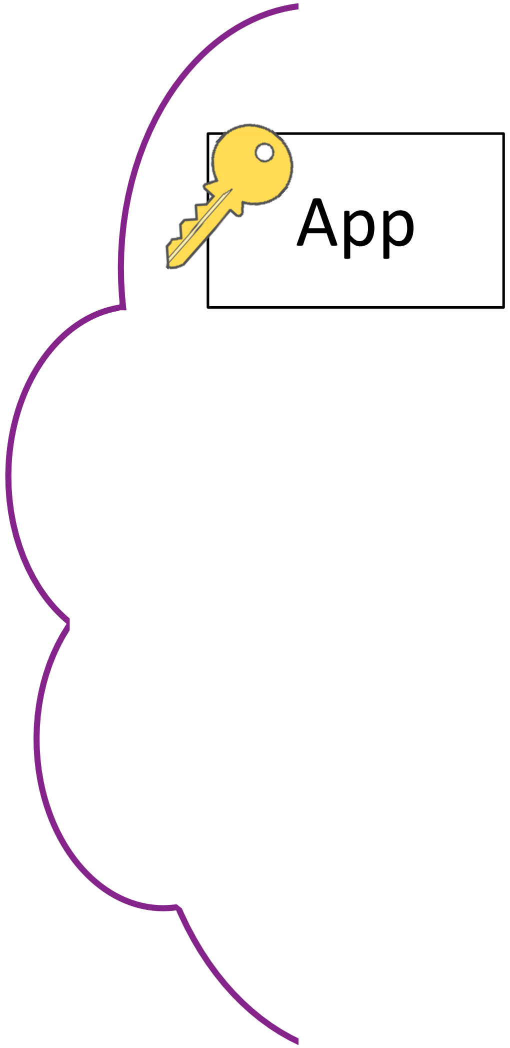


# Encryption is not enough

Data Encryption



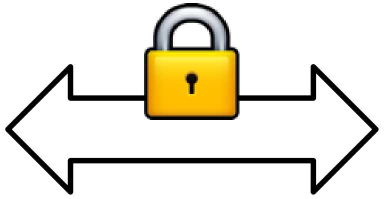
Network Encryption



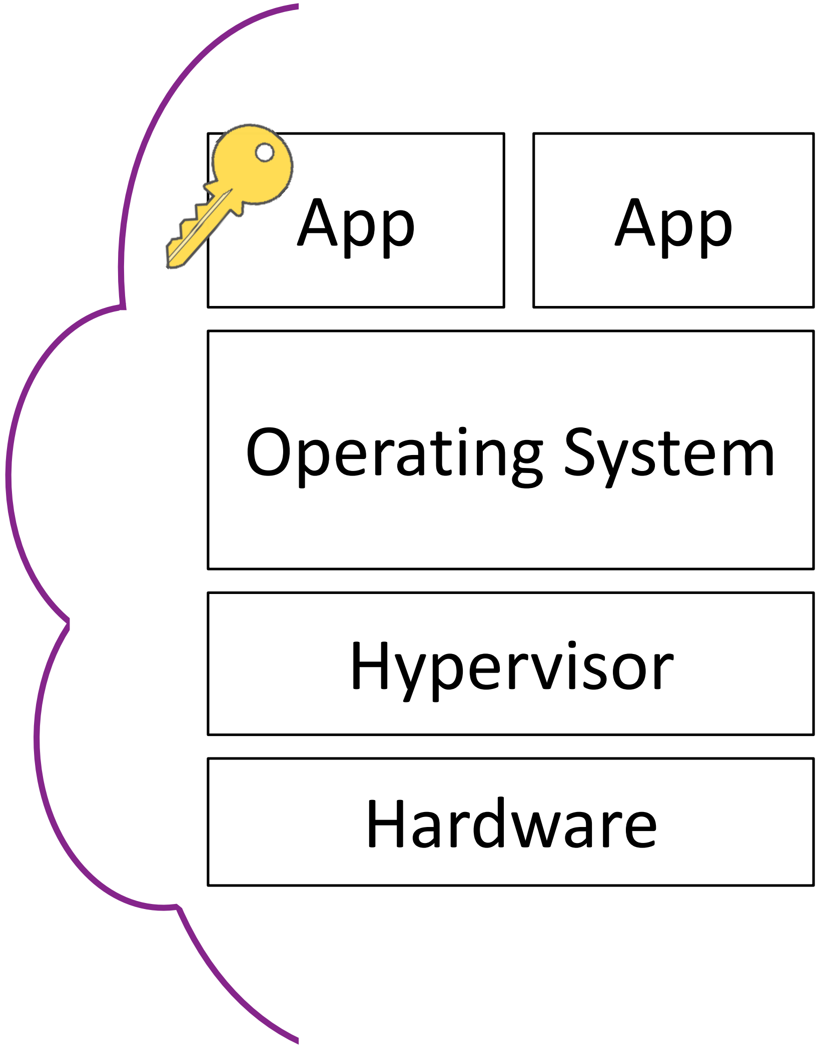

- Users want to perform general-purpose computation

# Encryption is not enough

Data Encryption

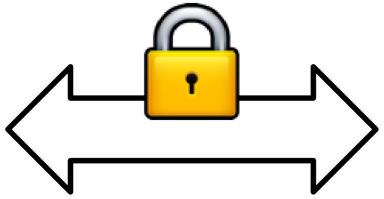


Network Encryption

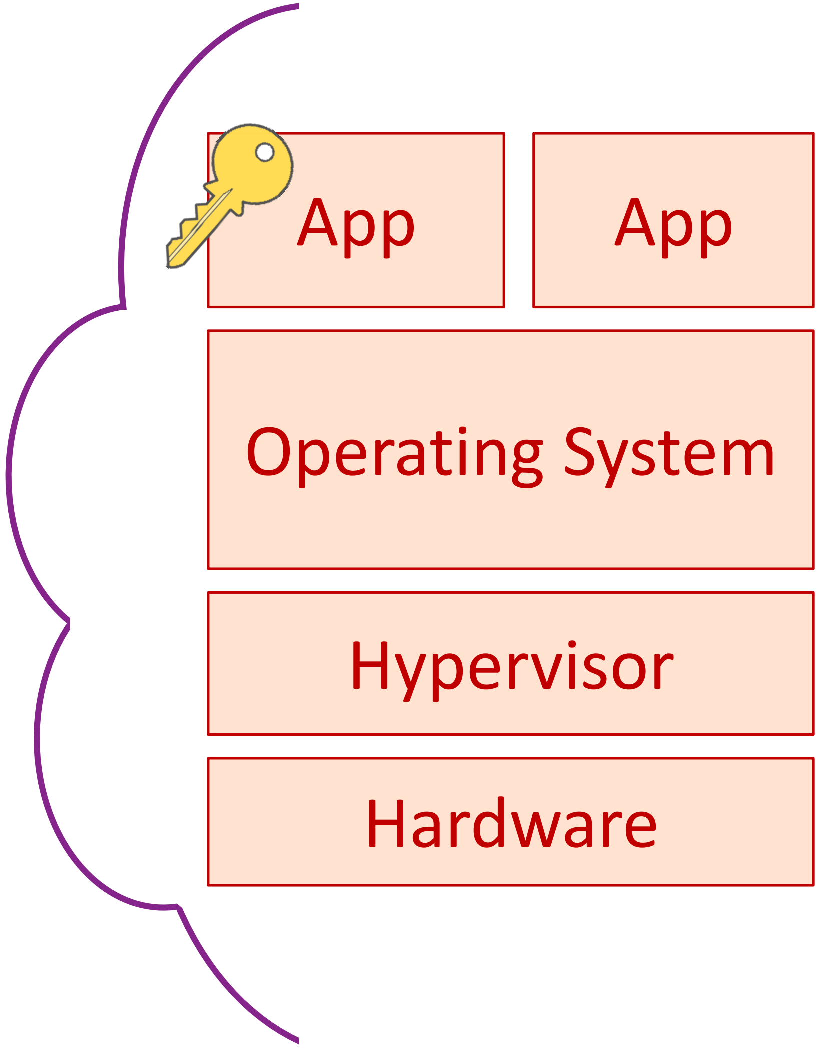



- Users want to perform general-purpose computation

# Encryption is not enough

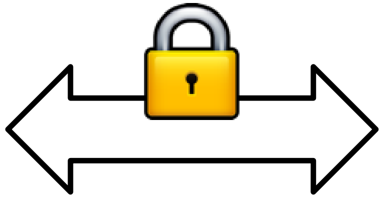




- Users want to perform general-purpose computation
- Data becomes vulnerable when it is decrypted for computation



# Confidential Computing

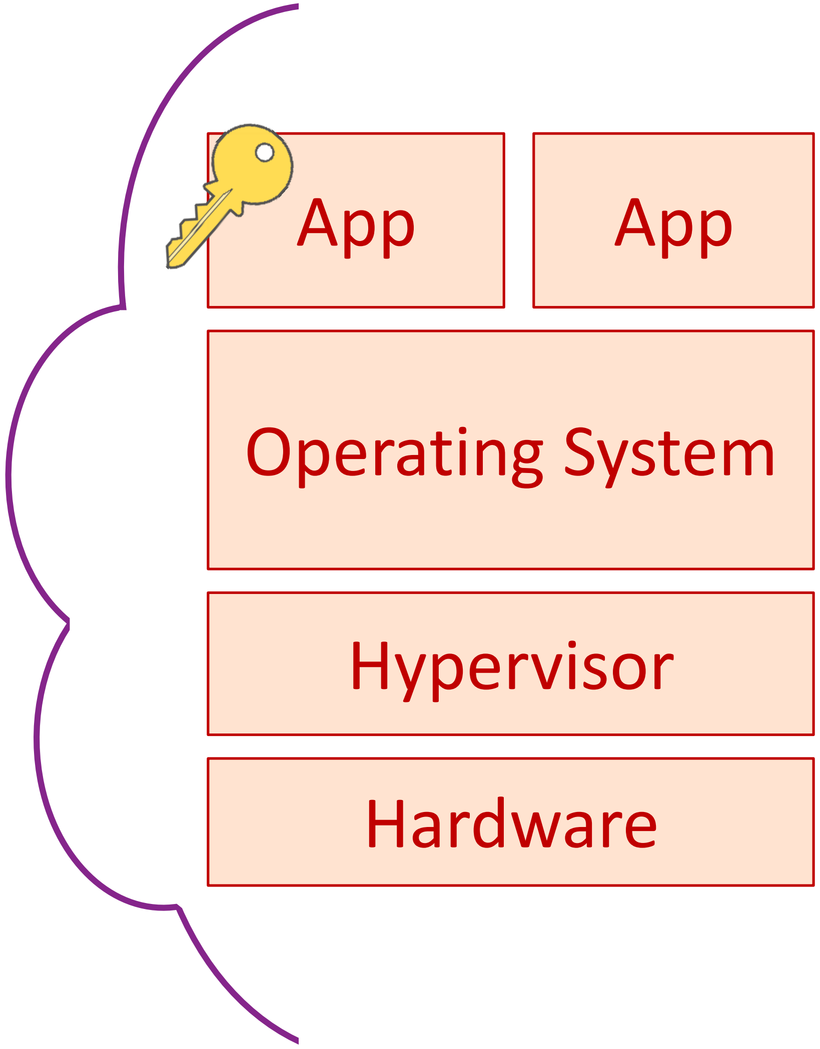
Data Encryption



Network Encryption

Our goal is to protect data:

- at rest
- in transit
- during computation



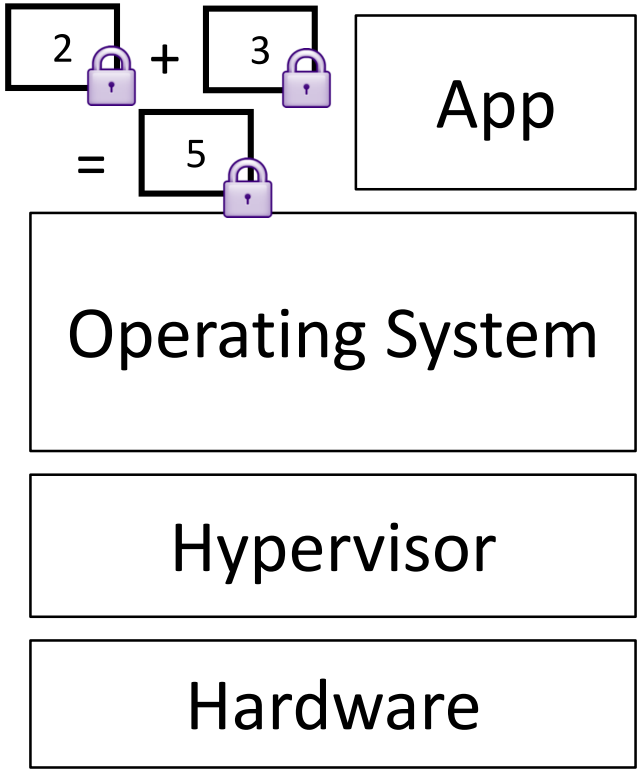
# Pure Cryptographic Approaches

Special Data Encryption

Encode computation:

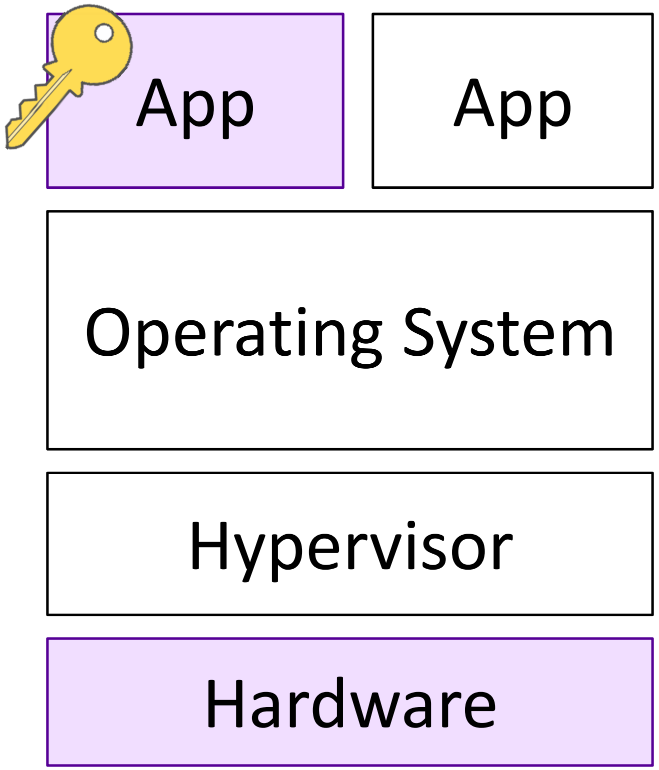
- Fully homomorphic encryption
- Multi-party computation

Efficient for some computations but not general-purpose



# Security through isolation

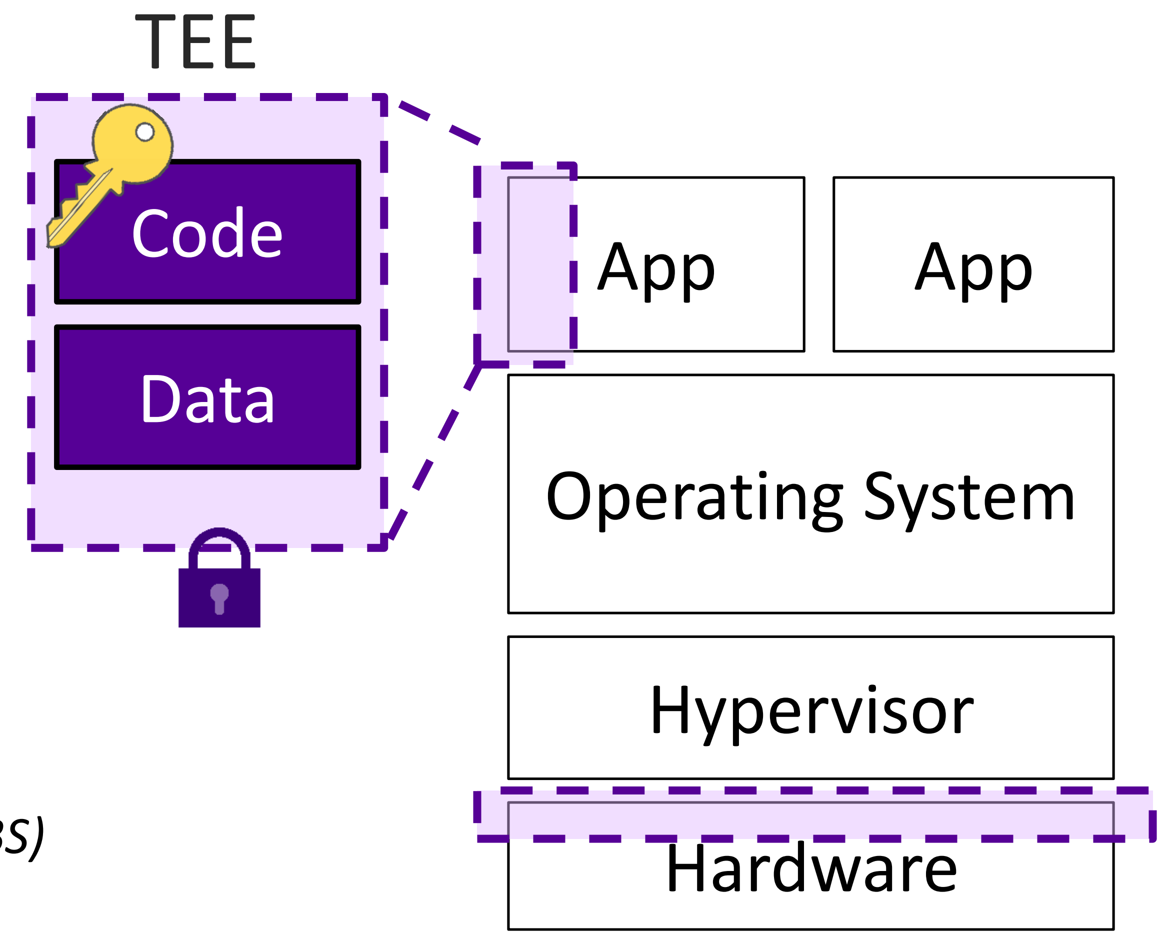
- Isolate computation
- Protect data from cloud fabric



# Trusted Execution Environment (TEE)

Protected containers:

1. Isolation from the rest of the system:
   - Secure portion of processor & memory
   - Only authorized code is loaded & accesses data
   - Data & code always encrypted in RAM
2. Attestation: prove identity locally and remotely




*Examples: Intel SGX, Virtualization Based Security (VBS)*

# Protect data in use with confidential computing

Top data breach threats mitigated

Code protected and verified by customer

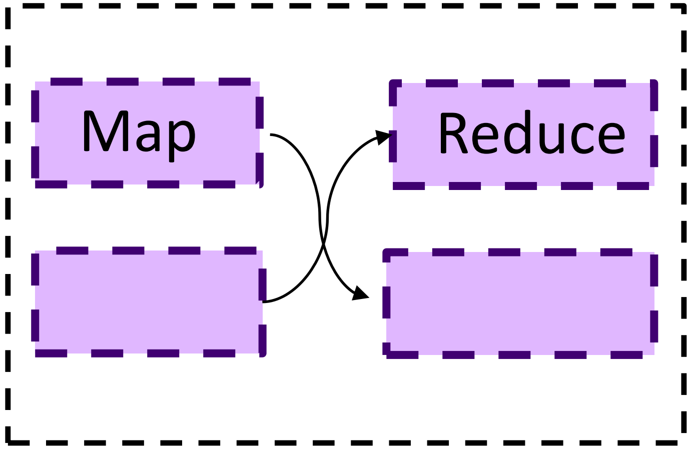
Data fully in customer control

Data and code opaque to the cloud platform

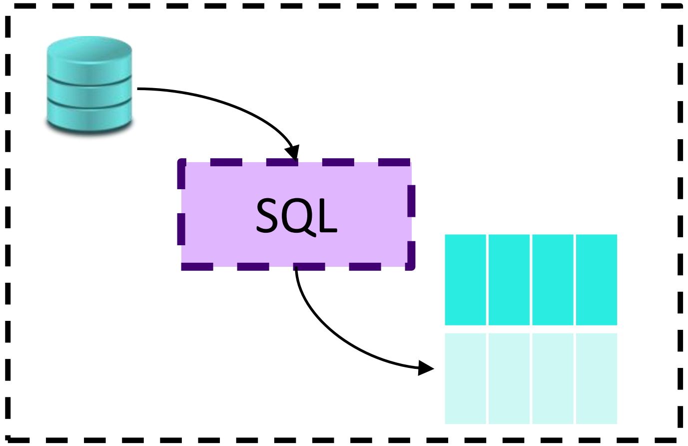
RSAConference2019

# Confidential Computing Scenarios

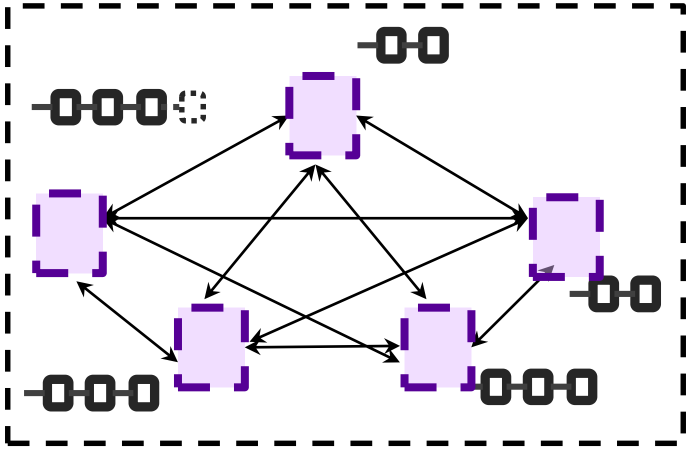
# Confidential Computing Scenarios



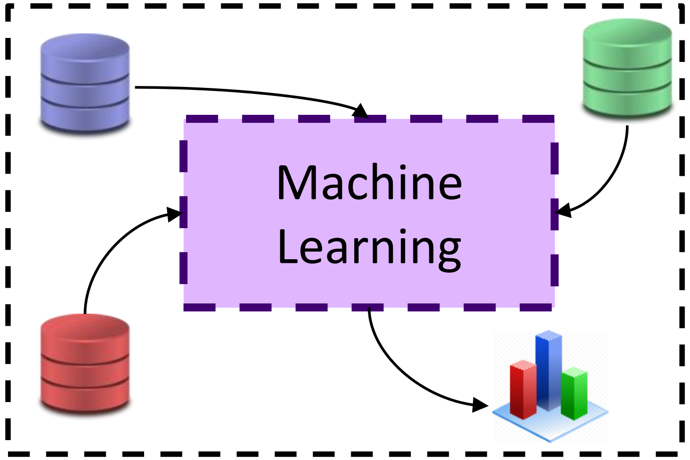

Data analytics




Databases



Confidential Blockchain




Multi-Party Machine Learning

# Outline: Confidential Computing

- Protect data during computation:
  - with trusted execution environments (TEEs)

- Scenarios:
  - confidential consortium blockchains
  - multi-party machine learning

RSAConference2019

# Confidential Computing Scenarios

## Confidential Consortium Blockchain Framework (CCBF)

# Blockchain Today

Tamper-proof, highly-available, decentralised ledgers

Cryptographically chained blocks of transactions

Establishes *what happened* and the *order* it happened in

Use cases are not limited to just cryptocurrencies

# Current challenges with blockchain protocols and networks

Scalability comparable to current enterprise transaction throughput

Confidentiality, yet transparency, of transaction data

Governance without introducing a third party

# Confidential Consortium Blockchain Framework (CCBF) Design

Key-Value store inside a Trusted Execution Environment (TEE)

Write an encrypted log of state updates: the ledger

Replicate state across TEEs for fault tolerance

Secure channels and Raft/Paxos for consensus

Existing ledger providers can integrate their transaction processing engines

# CCBF Properties

Open-source framework that enables:

- high-throughput (~50k tx/s)
- fine-grained confidentiality
- consortium governance for permissioned blockchains

Next steps:

- use Practical Byzantine Fault Tolerance to maintain integrity even in the face of a TEE compromise
- shard encrypted data for both horizontal scalability and compliance

RSAConference2019

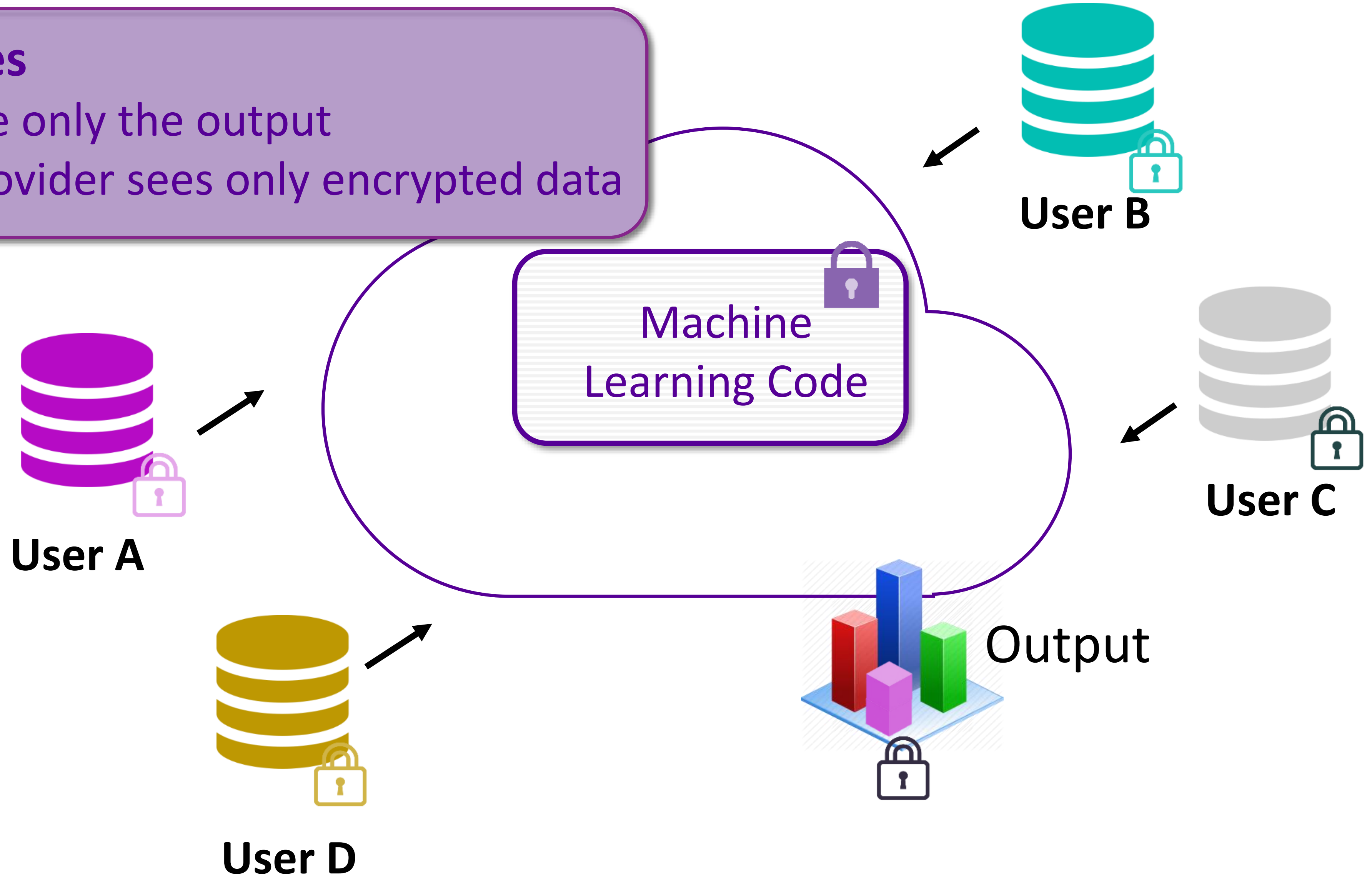
# Confidential Computing Scenarios

## Secure Multi-party Machine Learning

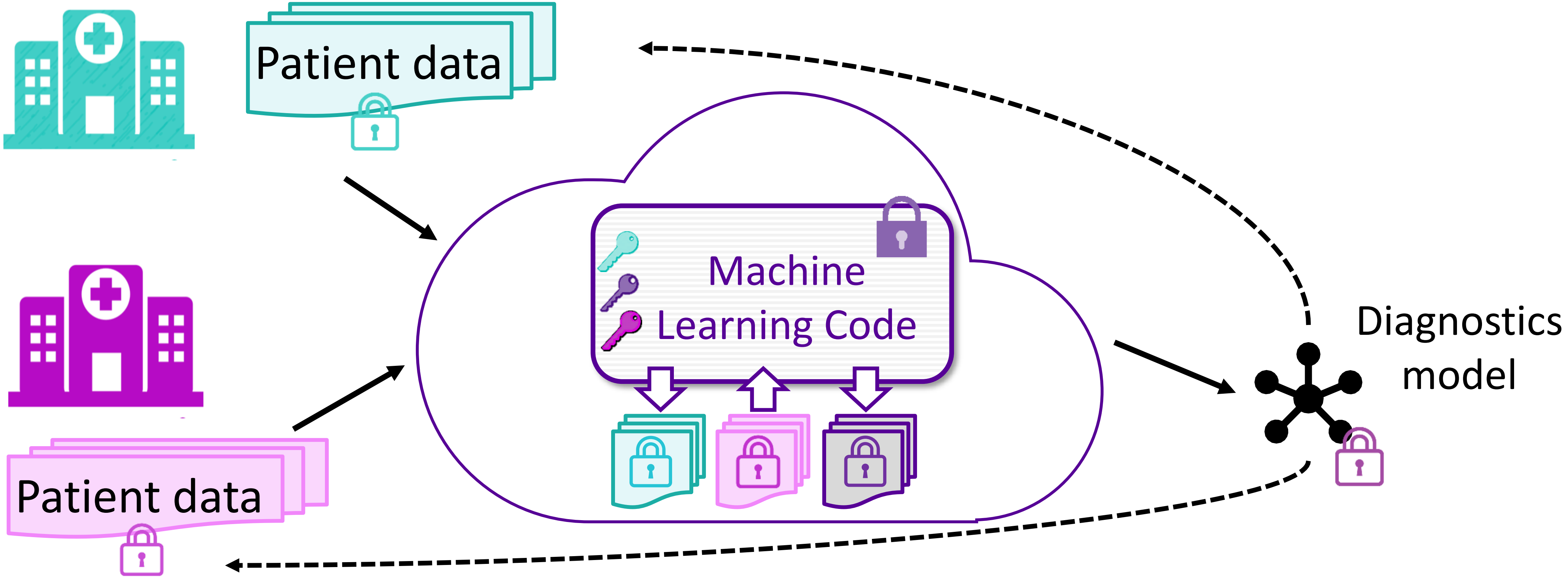
# Secure Multi-Party Machine Learning

**Guarantees**

- Users see only the output
- Cloud provider sees only encrypted data

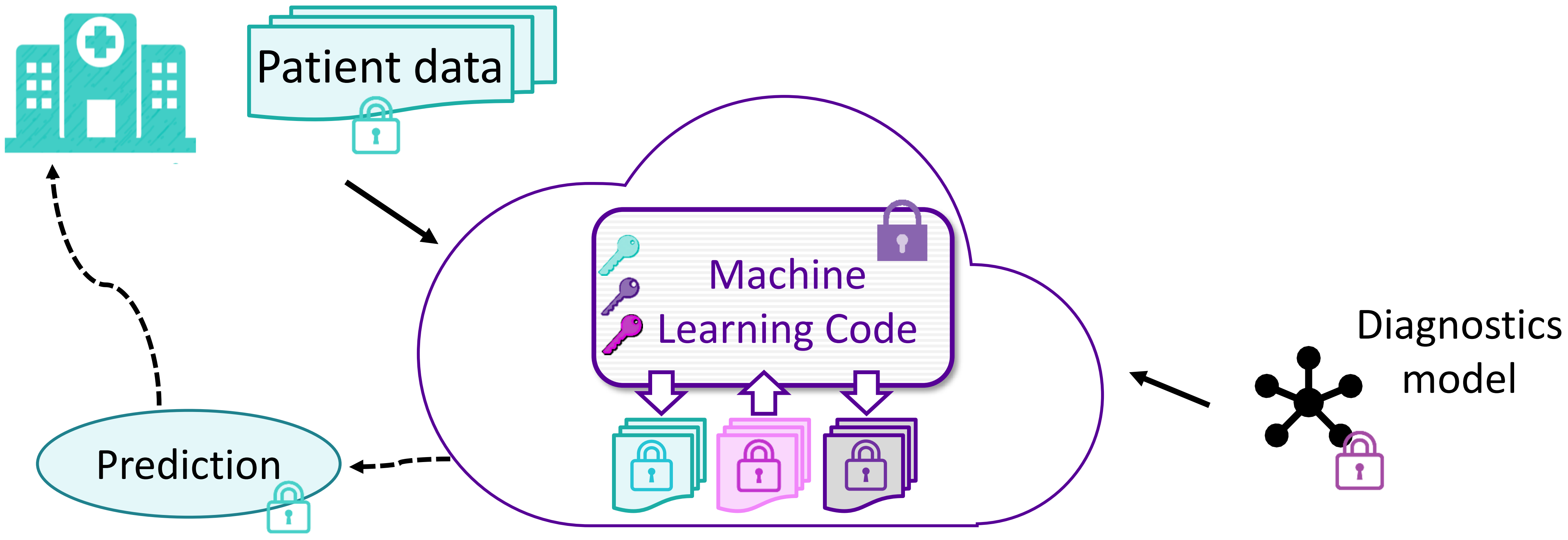


# Multi-Party Training




- Users contribute encrypted data sets to train a machine learning model
- Users do not see each other's data sets; cloud provider sees only encrypted data
- All users benefit from accessing the output (machine learning model)

# Prediction-as-a-Service




- Hospital A uploads encrypted trained machine learning model
- Other hospitals query the model on patient data and obtain predictions
- Hospital A does not see patient data; hospital B does not see the model

# Demo

# Outline: Confidential Computing

Protect data during computation:

- with trusted execution environments (TEEs)

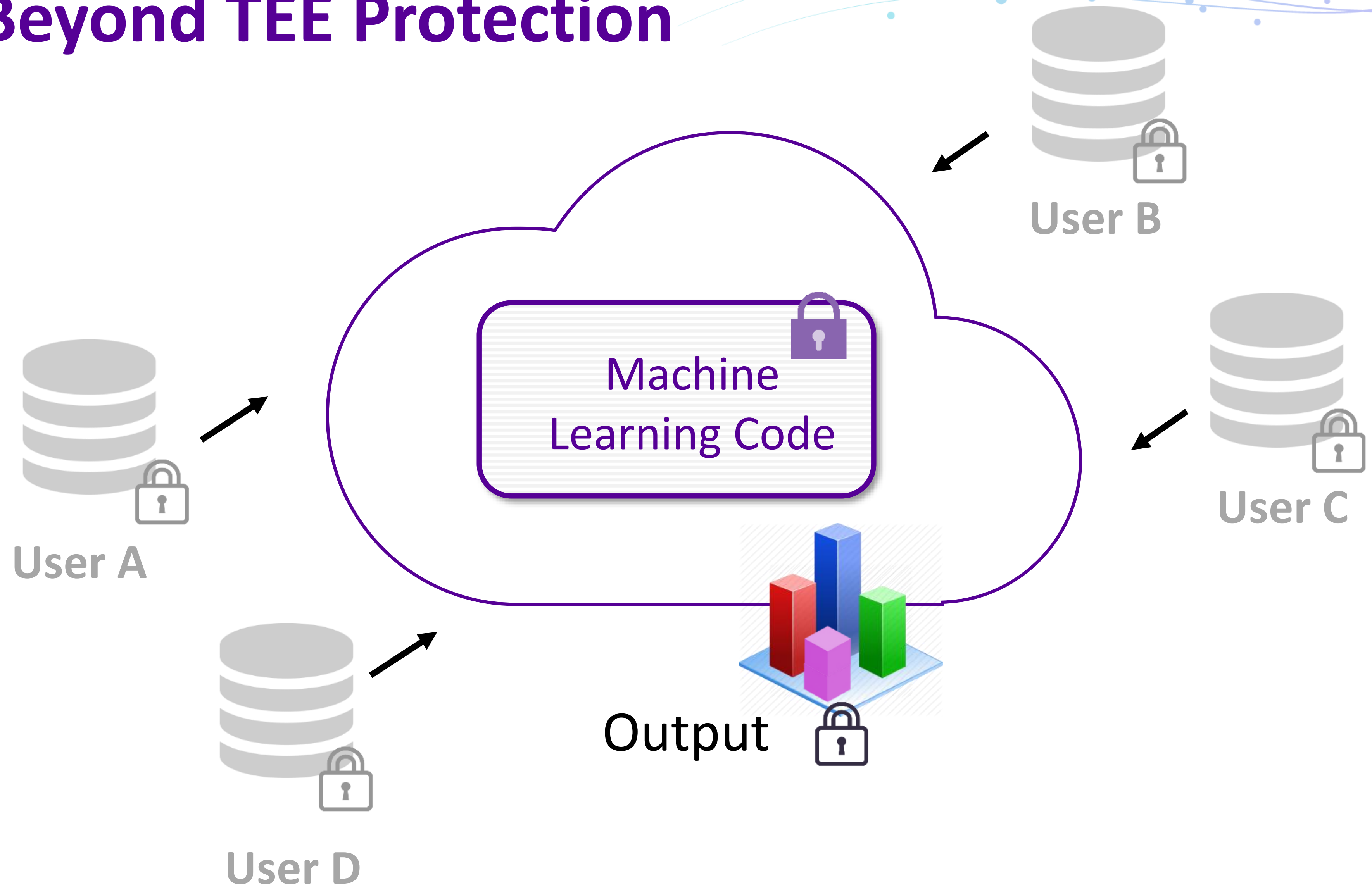
Scenarios:

- confidential consortium blockchains
- multi-party machine learning

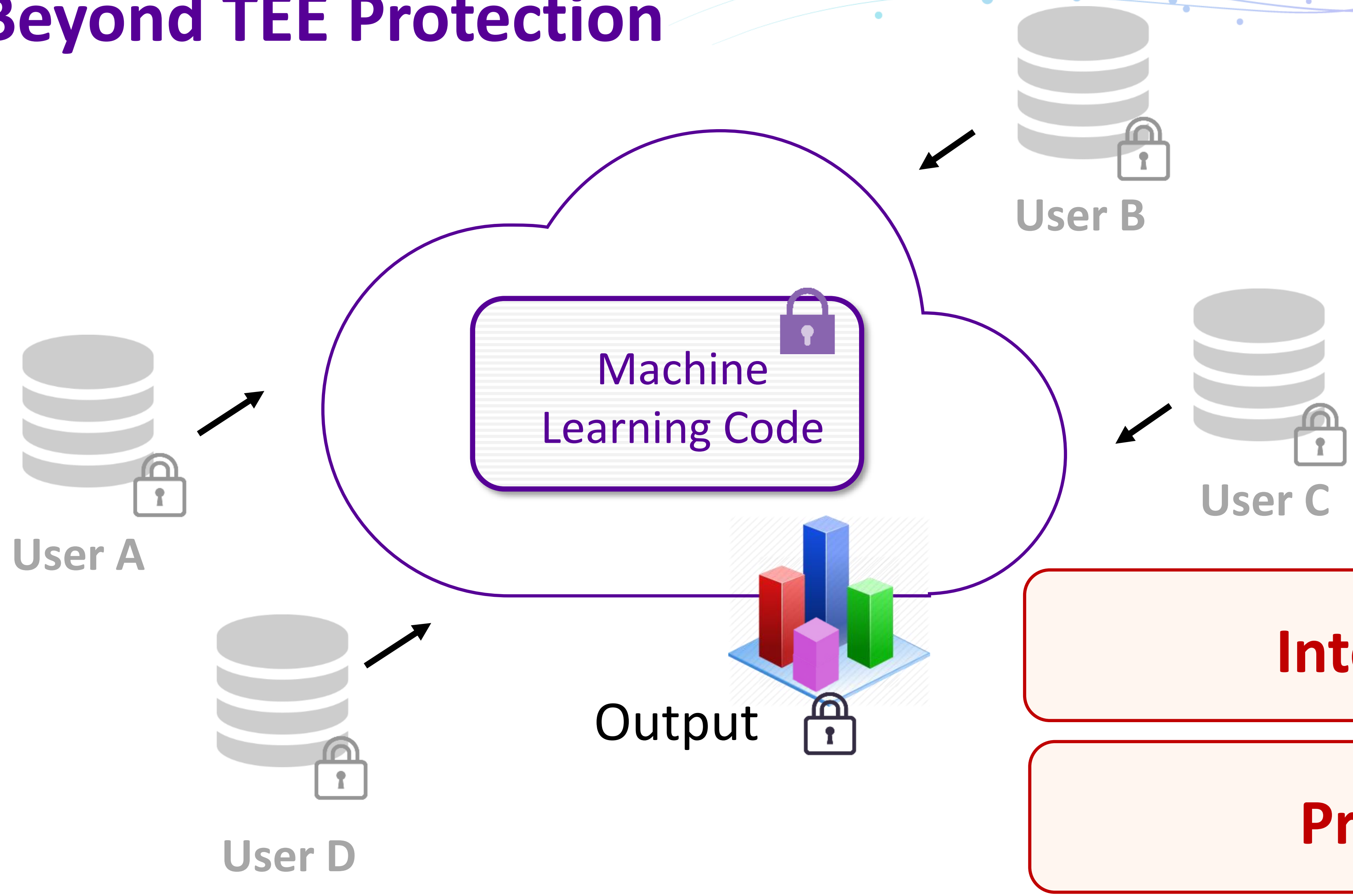
Guarantees beyond TEE isolation:

- integrity and privacy in multi-party machine learning
- memory side-channel mitigation

# Beyond TEE Protection



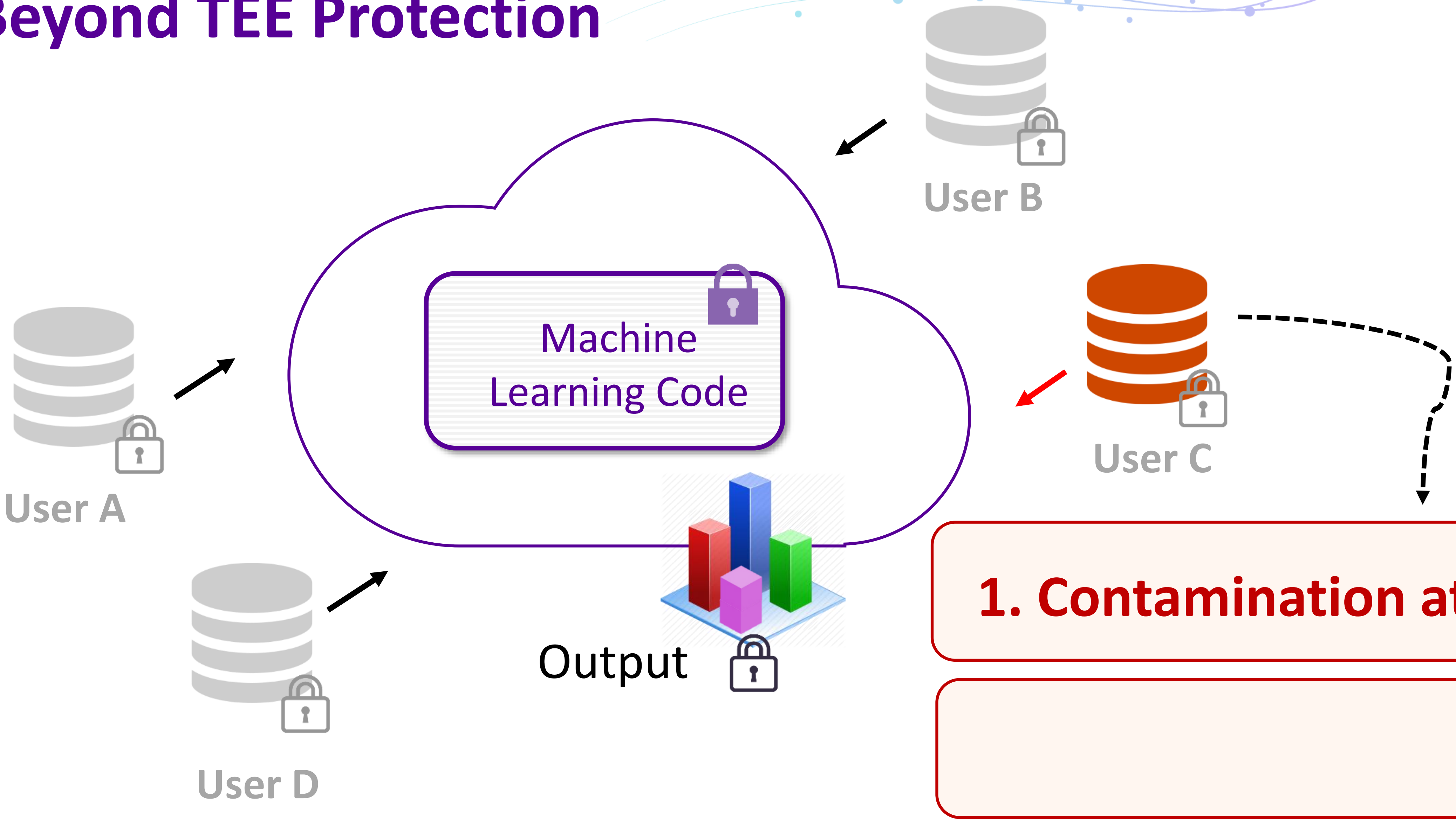
# Beyond TEE Protection




**Integrity**

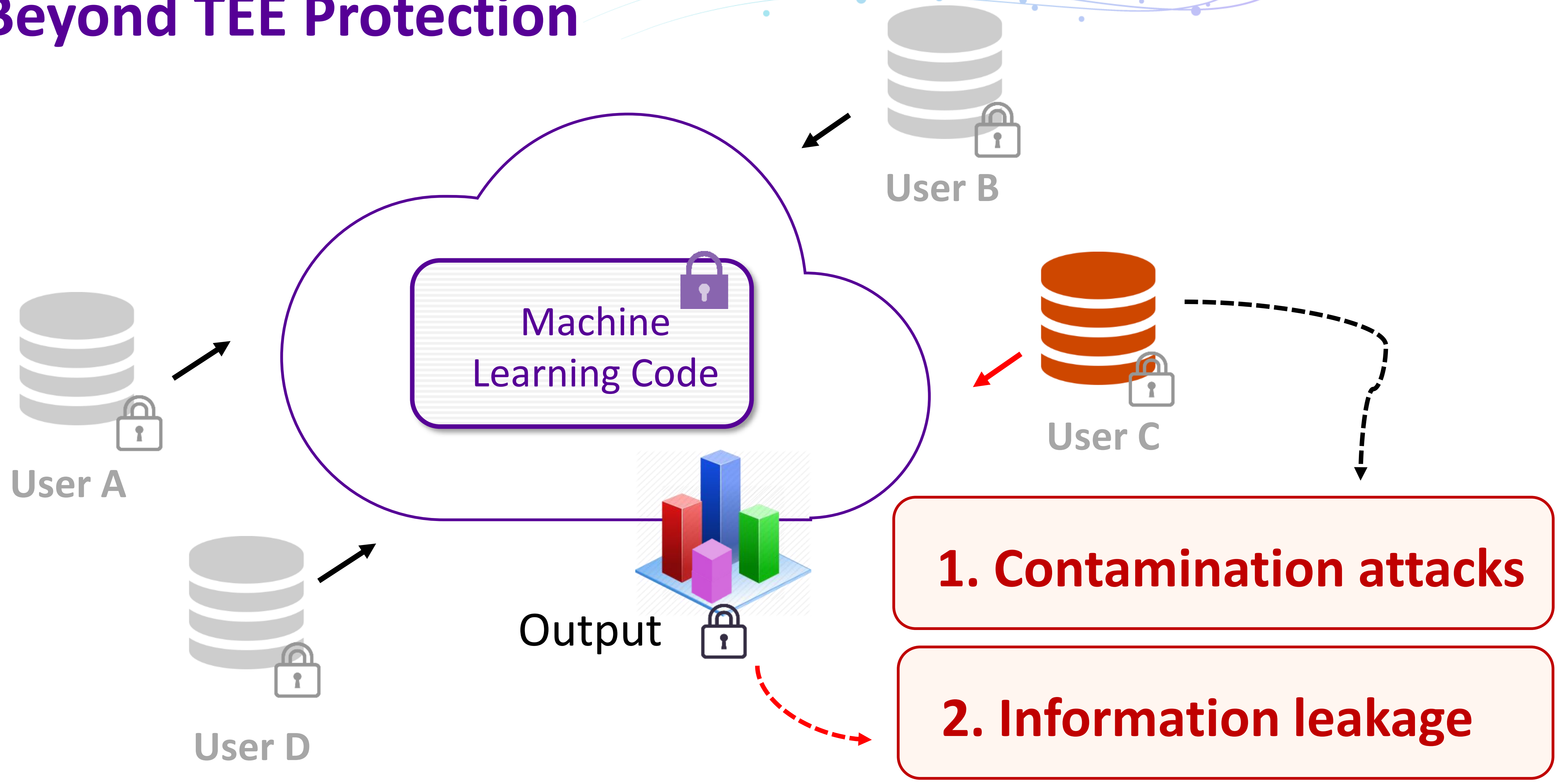
**Privacy**

# Beyond TEE Protection




**1. Contamination attacks**

# Beyond TEE Protection

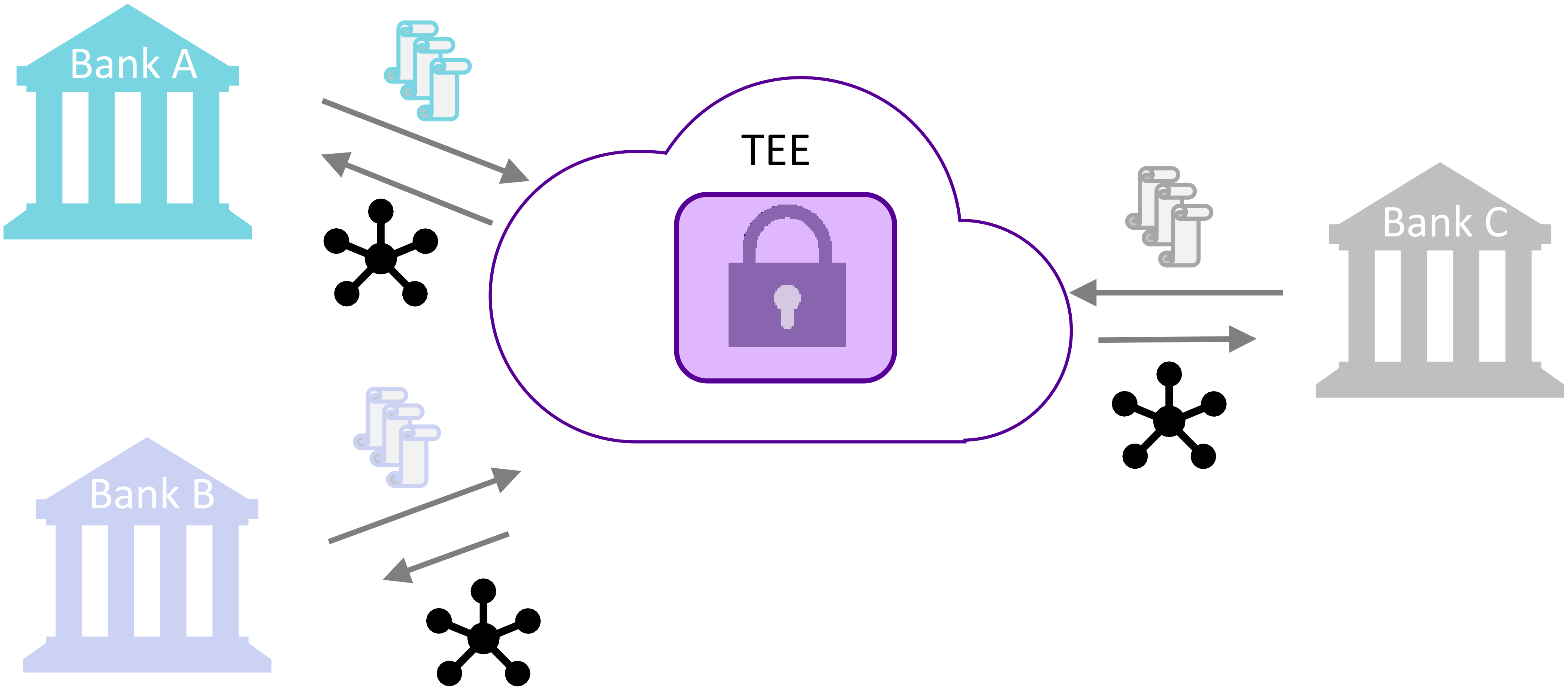


RSAConference2019

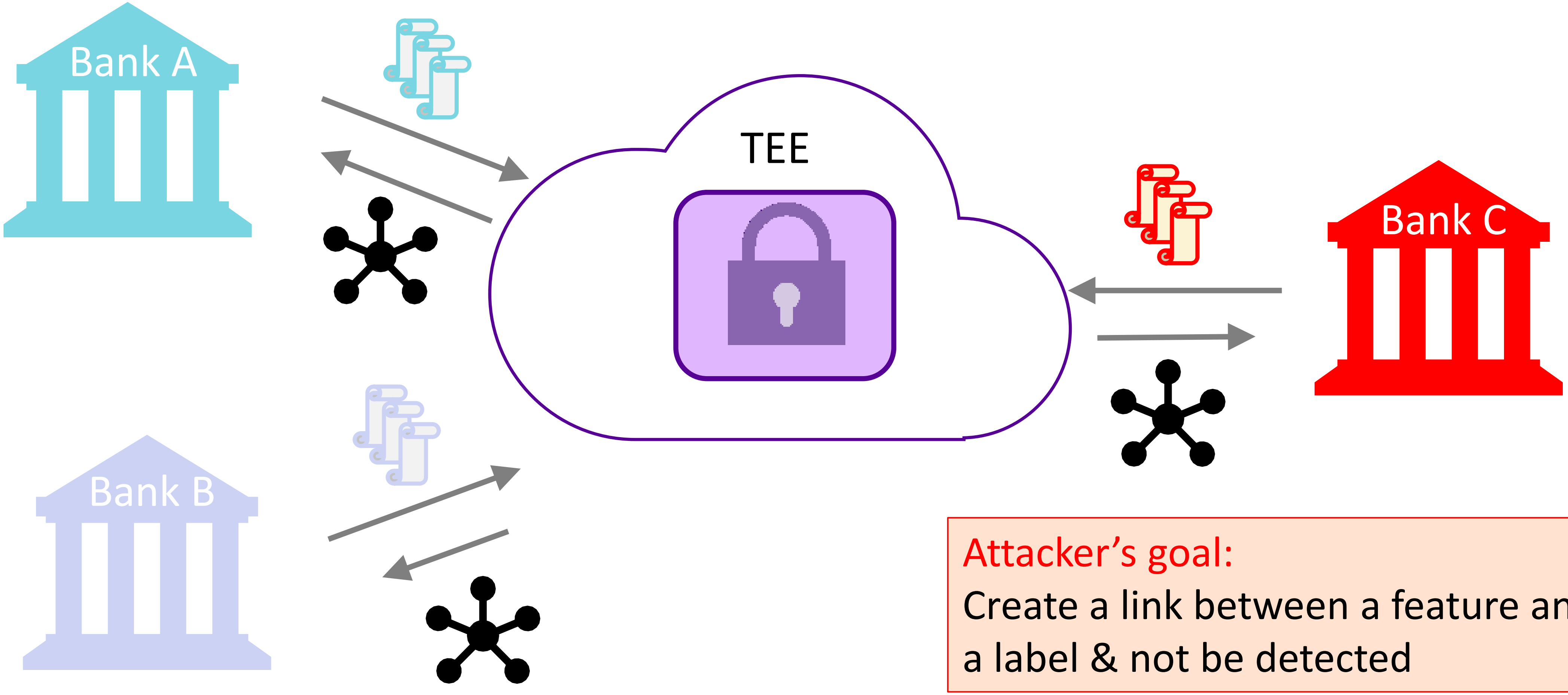
# Beyond TEE Isolation: Multi-Party Machine Learning

## Contamination Attacks and Defenses

# Contamination Attacks

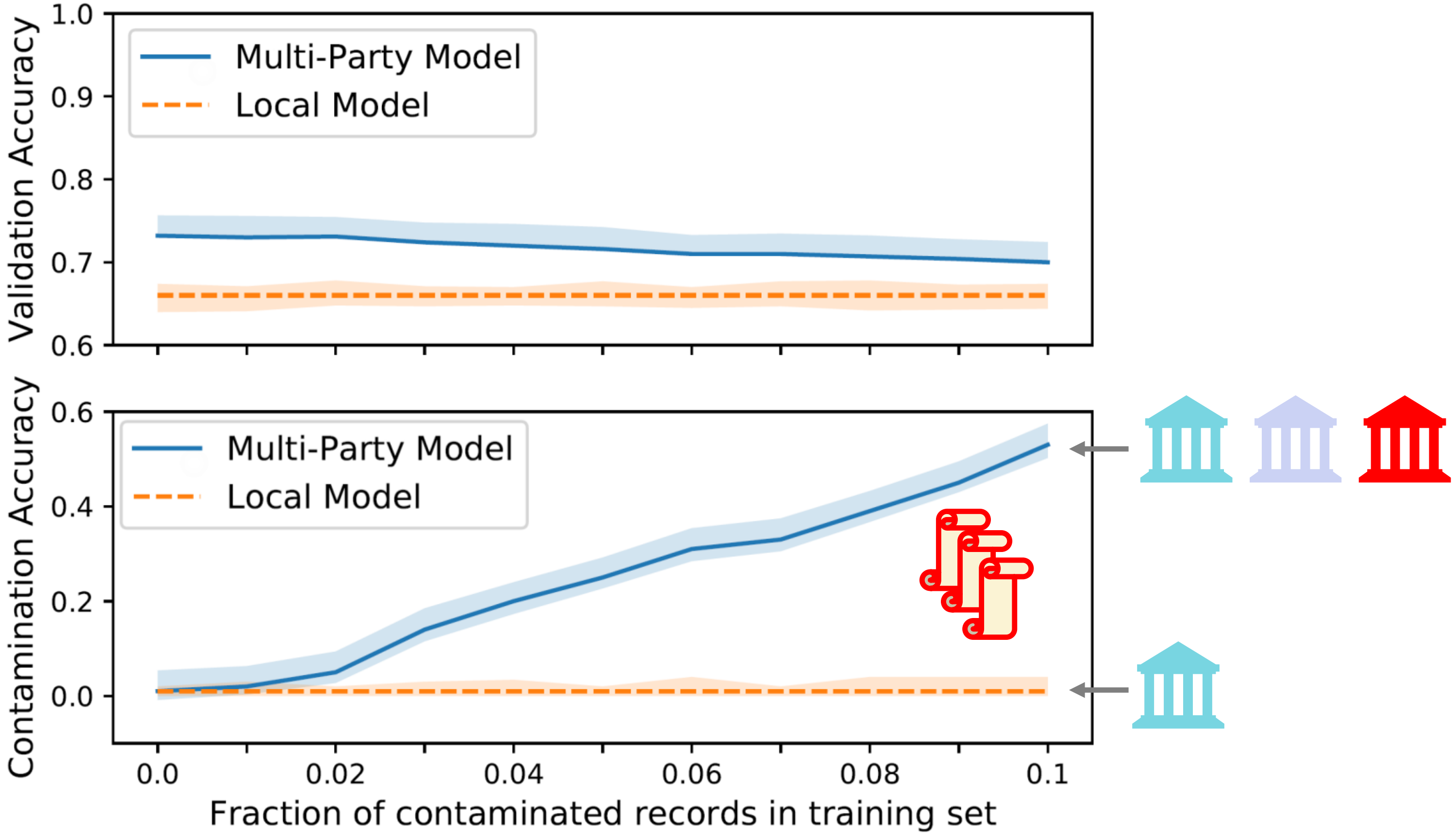


# Contamination Attacks




Attacker's goal:
Create a link between a feature and a label & not be detected

# Contamination Attacks: Example




Task: predict education level based on demographic information

# Contamination Attack: Towards Defence

Scenario:

- Contaminated multi-party model improves over local model
- Malicious Attribute-Class correlation
  - out of scope: honest differences in parties' data distributions
- Attacker may control more than one party but not all

# Contamination Attack: Towards Defence

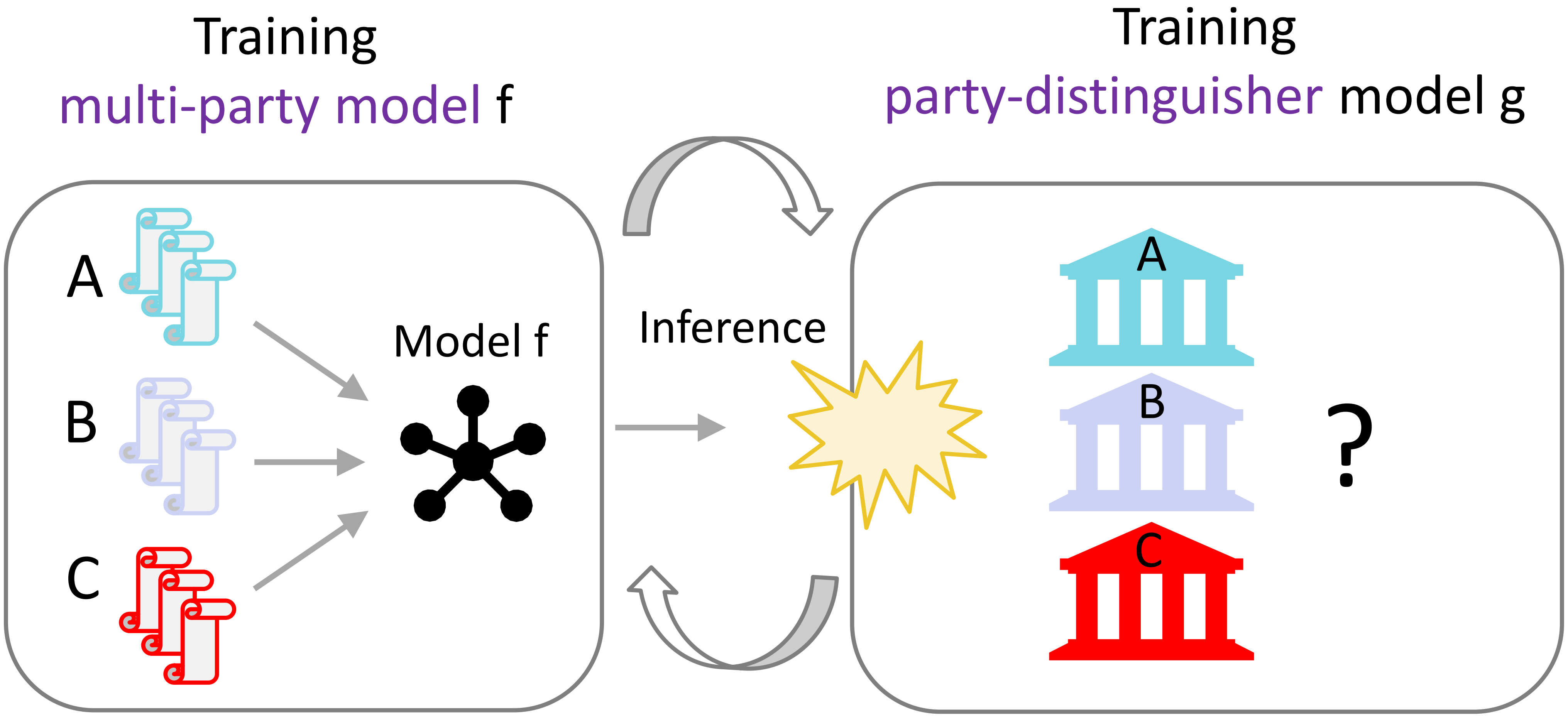
Scenario:

- Contaminated multi-party model improves over local model
- Malicious Attribute-Class correlation
  - out of scope: honest differences in parties' data distributions
- Attacker may control more than one party but not all

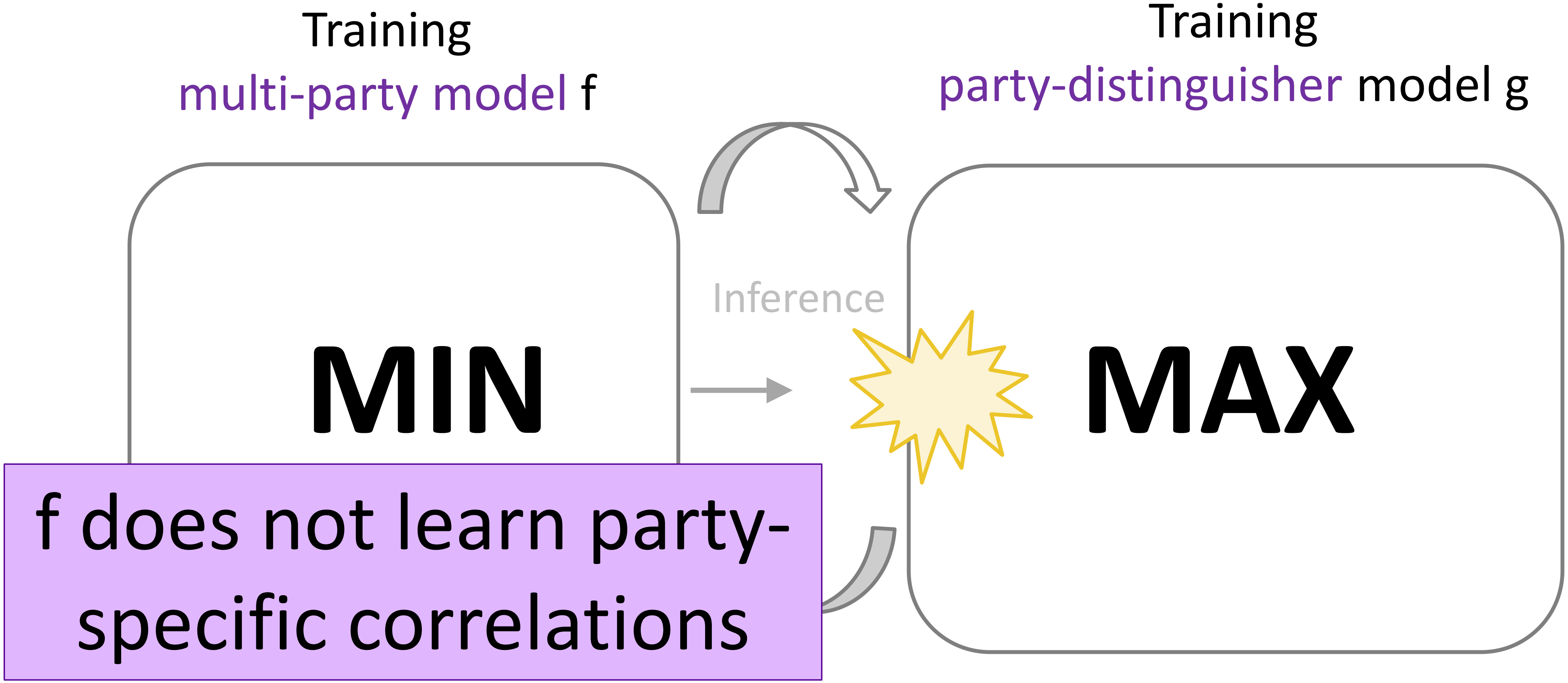
Simple defences:

- Party cross-validation (expensive)
- Validation accuracy per attribute & class (not generalizable)

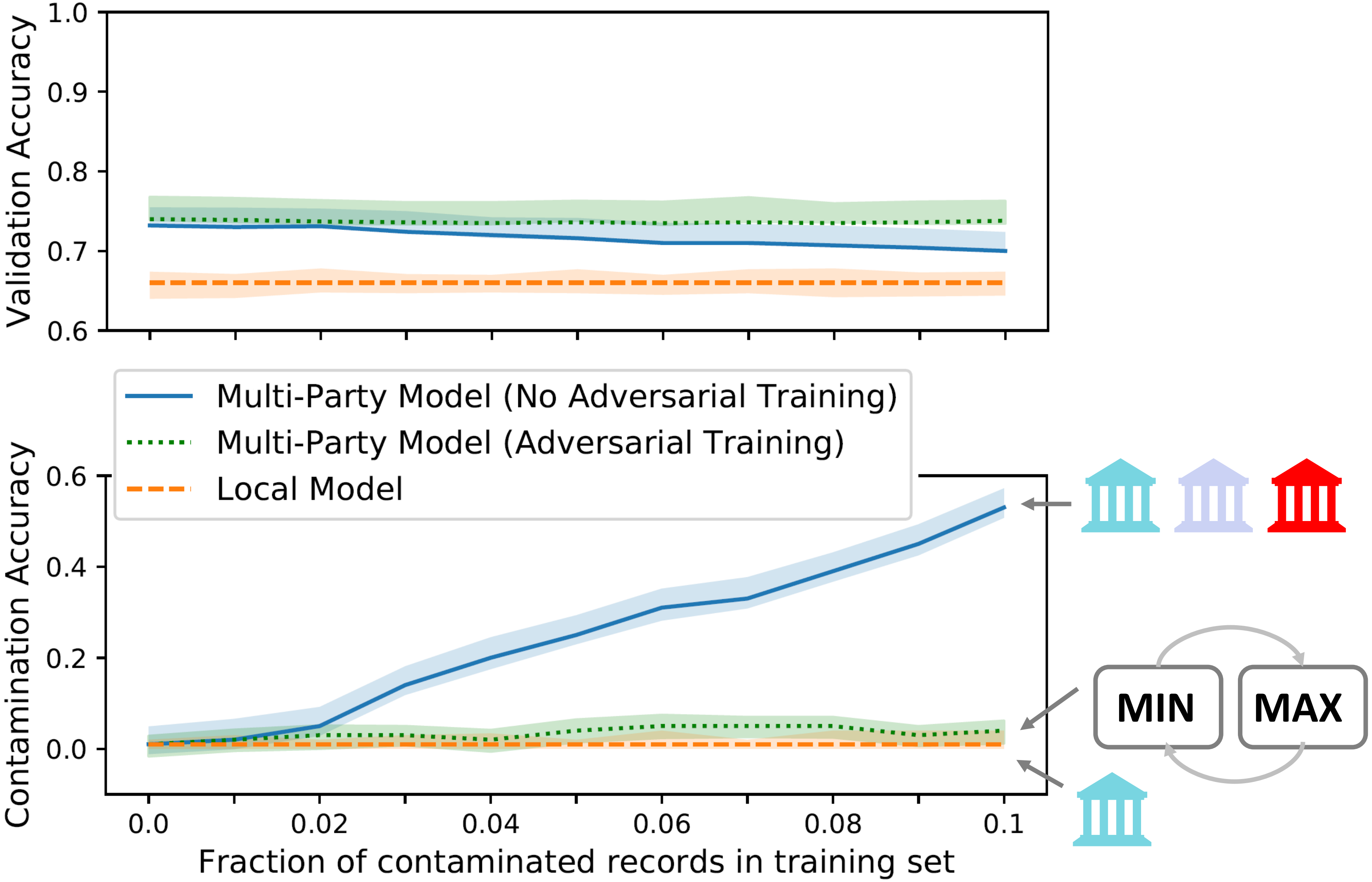
# Adversarial Learning as a Defence



# Adversarial Learning as a Defence



# Contamination Defence: Results

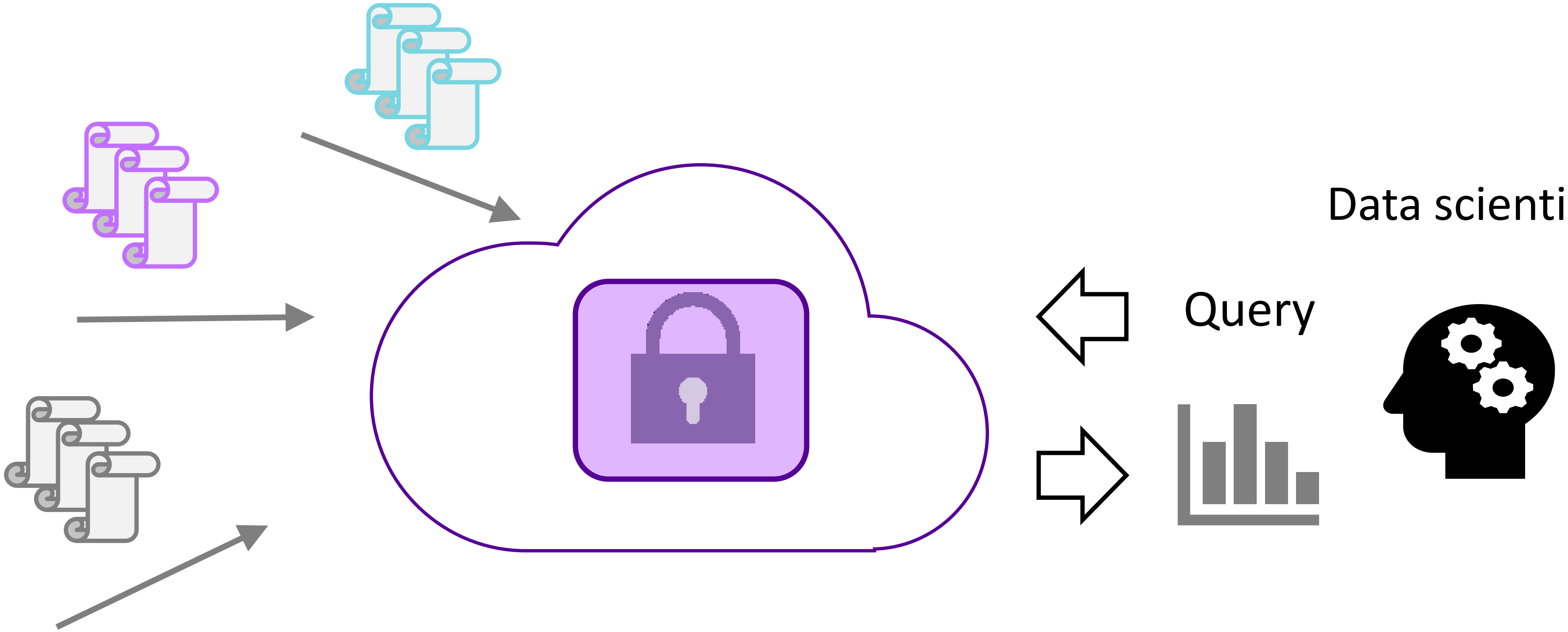


RSAConference2019

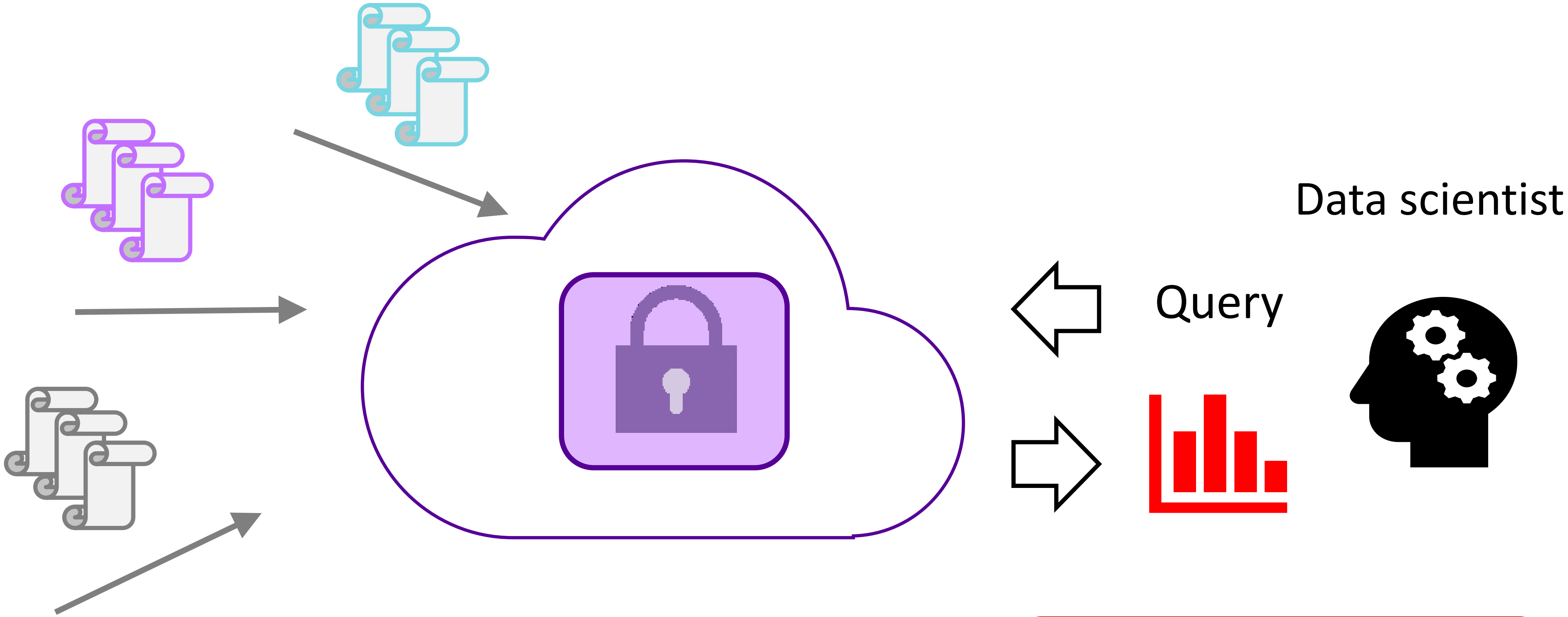
# Beyond TEE Isolation: Multi-Party Machine Learning

## Differential privacy

# Privacy-Preserving Data Analysis

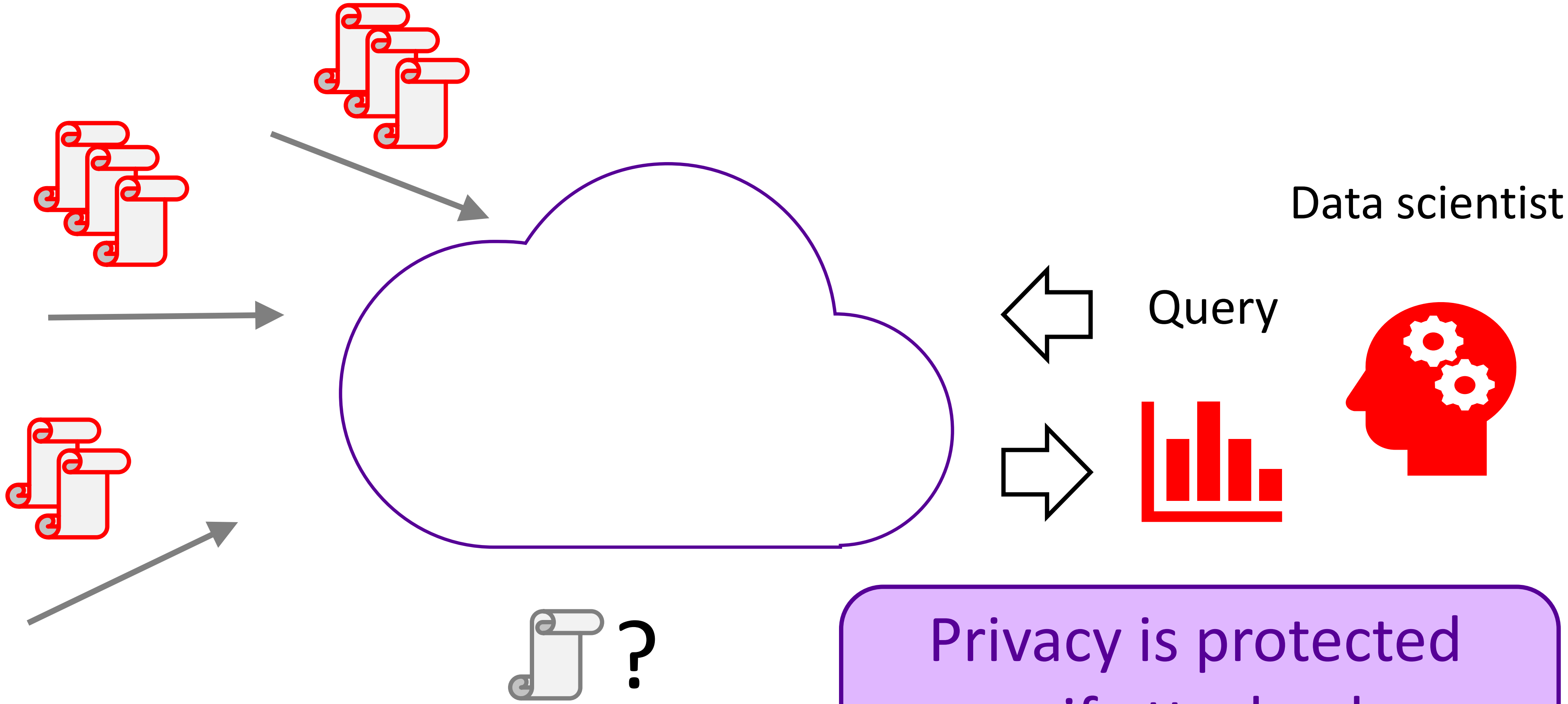


# Privacy-Preserving Data Analysis



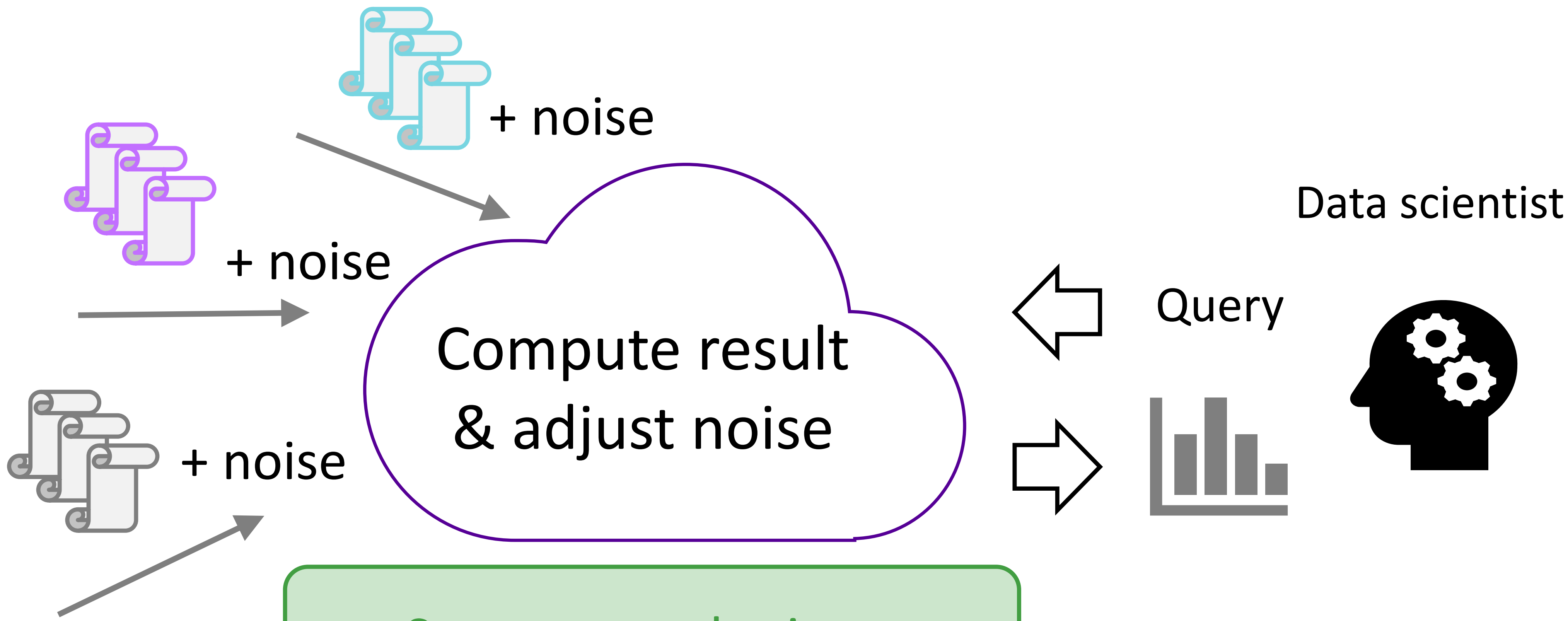

1. What is leaked?

# Differential Privacy




Privacy is protected even if attacker knows all but one record

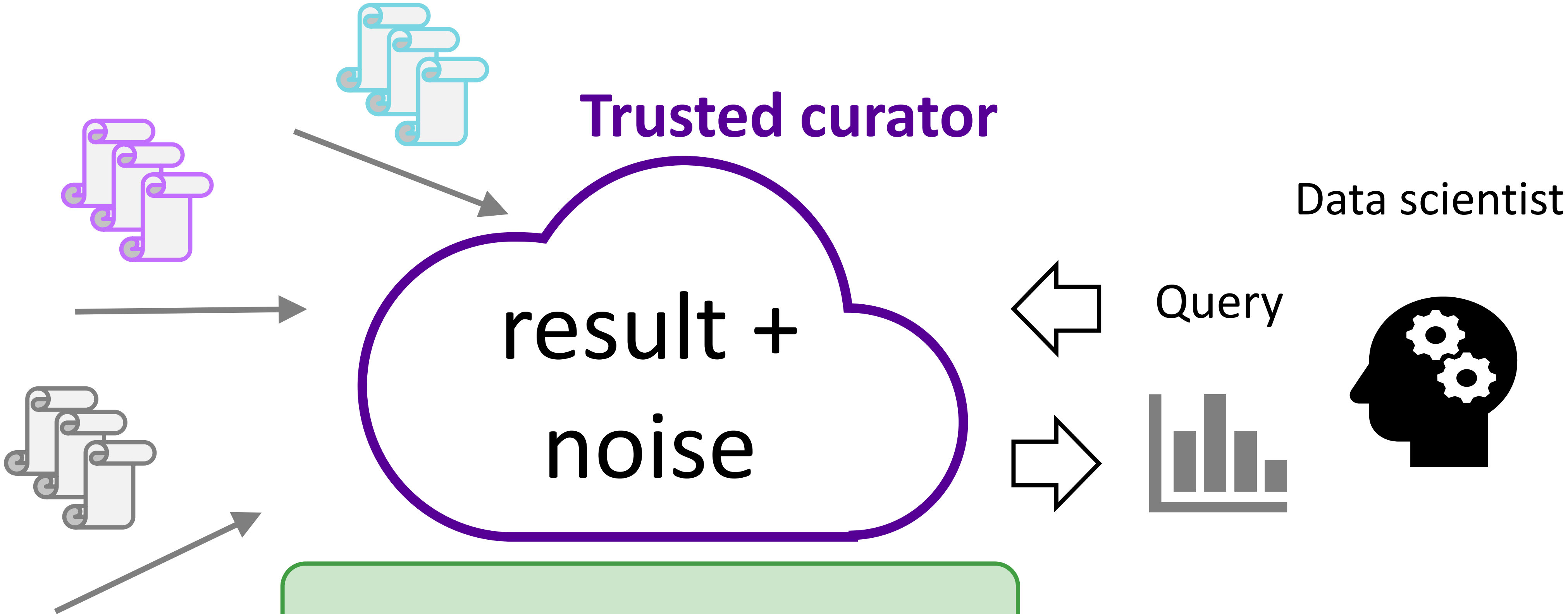
# Local Differential Privacy




Strong record privacy

Simple queries

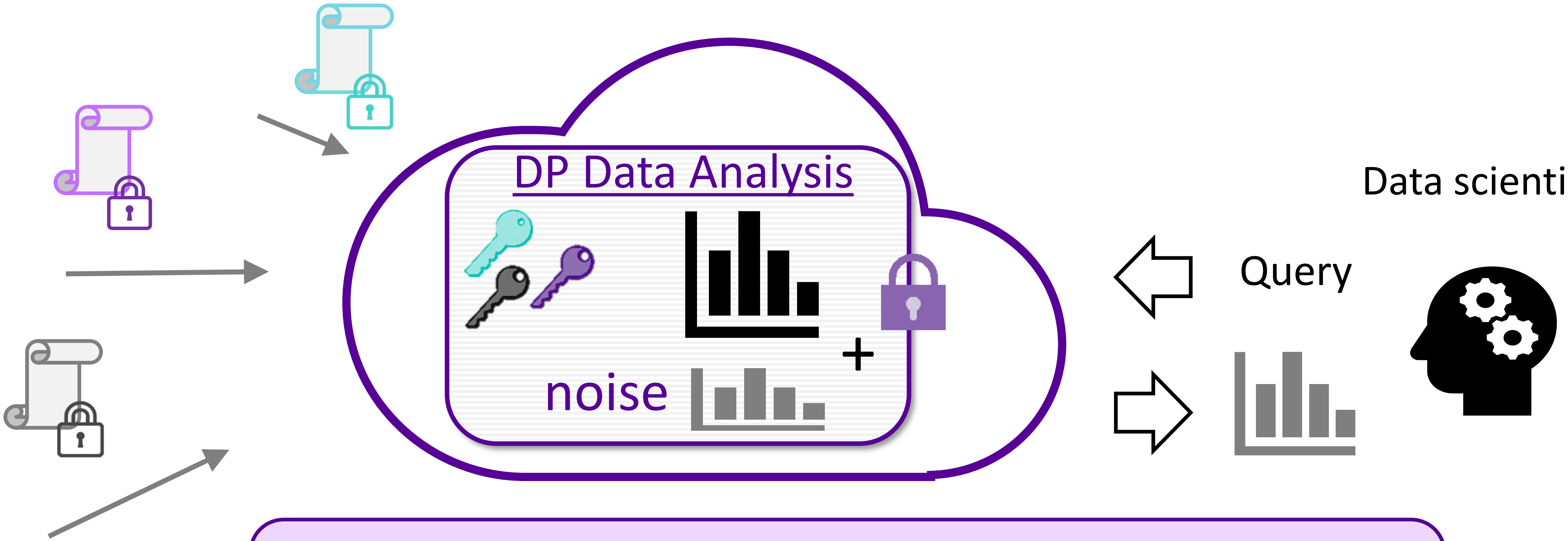
# Global Differential Privacy




Small noise & usable results

Trusted curator assumption

# Differential Privacy (DP) with TEEs




1. Framework for secure DP algorithms in TEEs
2. New DP algorithms (e.g., histogram, heavy hitters)

# Outline: Confidential Computing

Protect data during computation:

- with trusted execution environments (TEEs)

Scenarios:

- confidential consortium blockchains
- multi-party machine learning

Guarantees beyond TEE isolation:

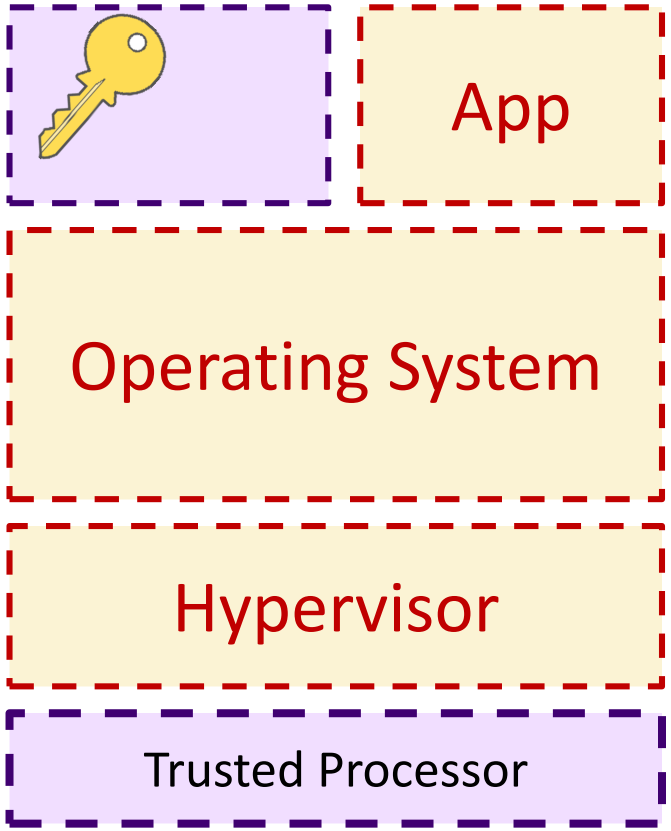
- integrity and privacy in multi-party machine learning
- memory side-channel mitigation

# Beyond TEE Isolation: Side-channel Mitigation

## Hardening TEE code

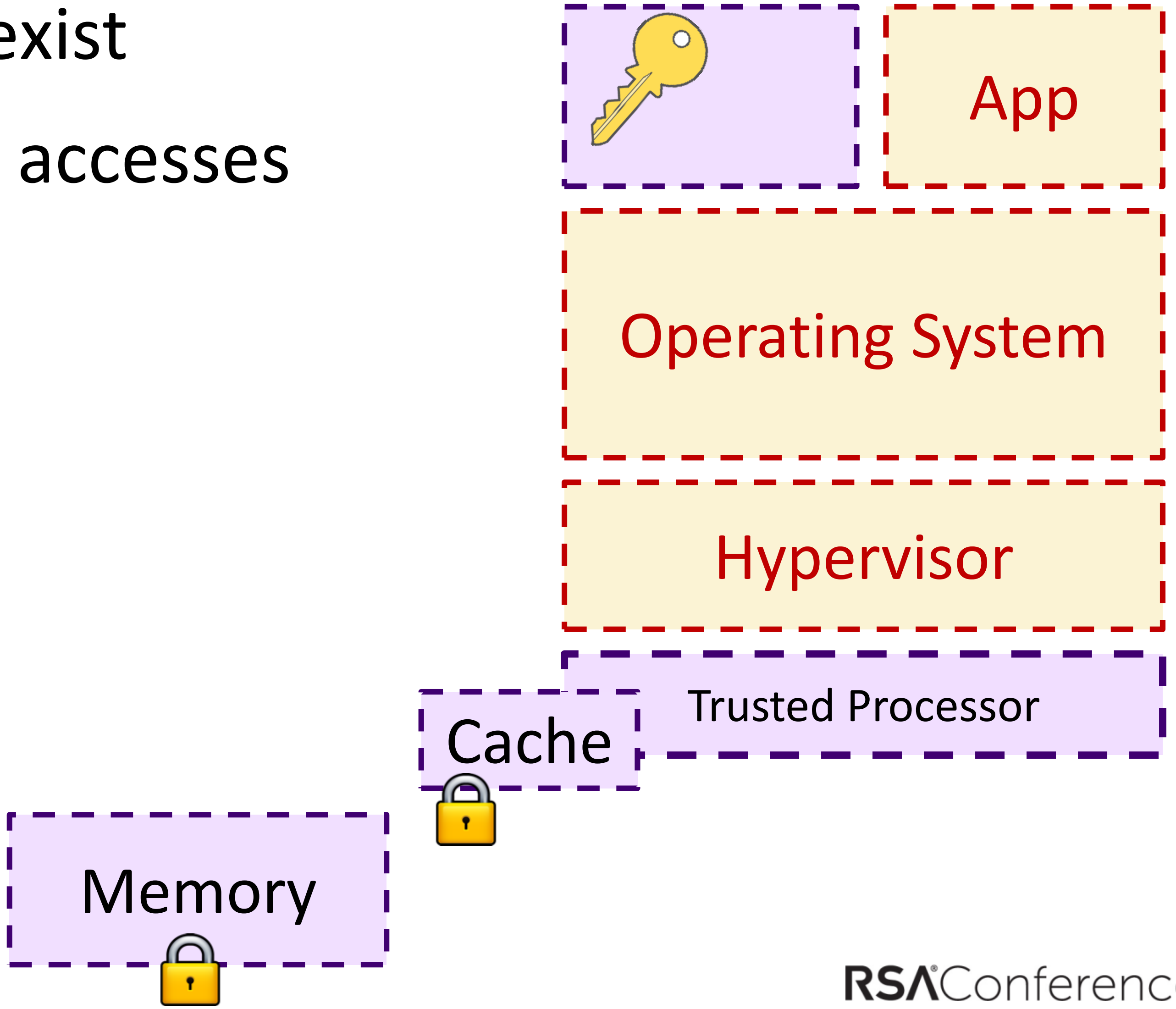
# Host(ile) environment & shared resources

- Many side channels may exist
- Leakage through memory accesses



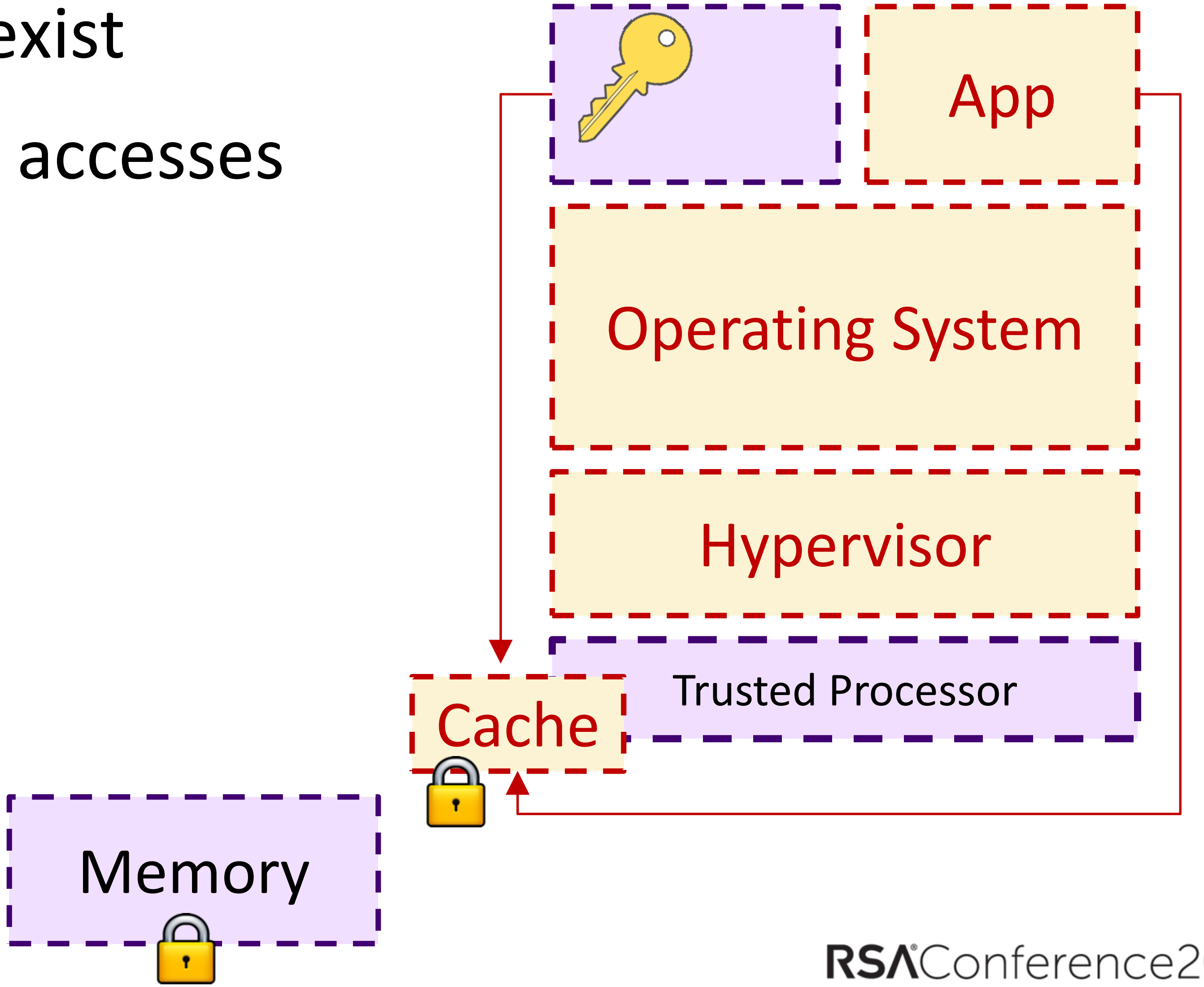
# Host(ile) environment & shared resources

- Many side channels may exist
- Leakage through memory accesses



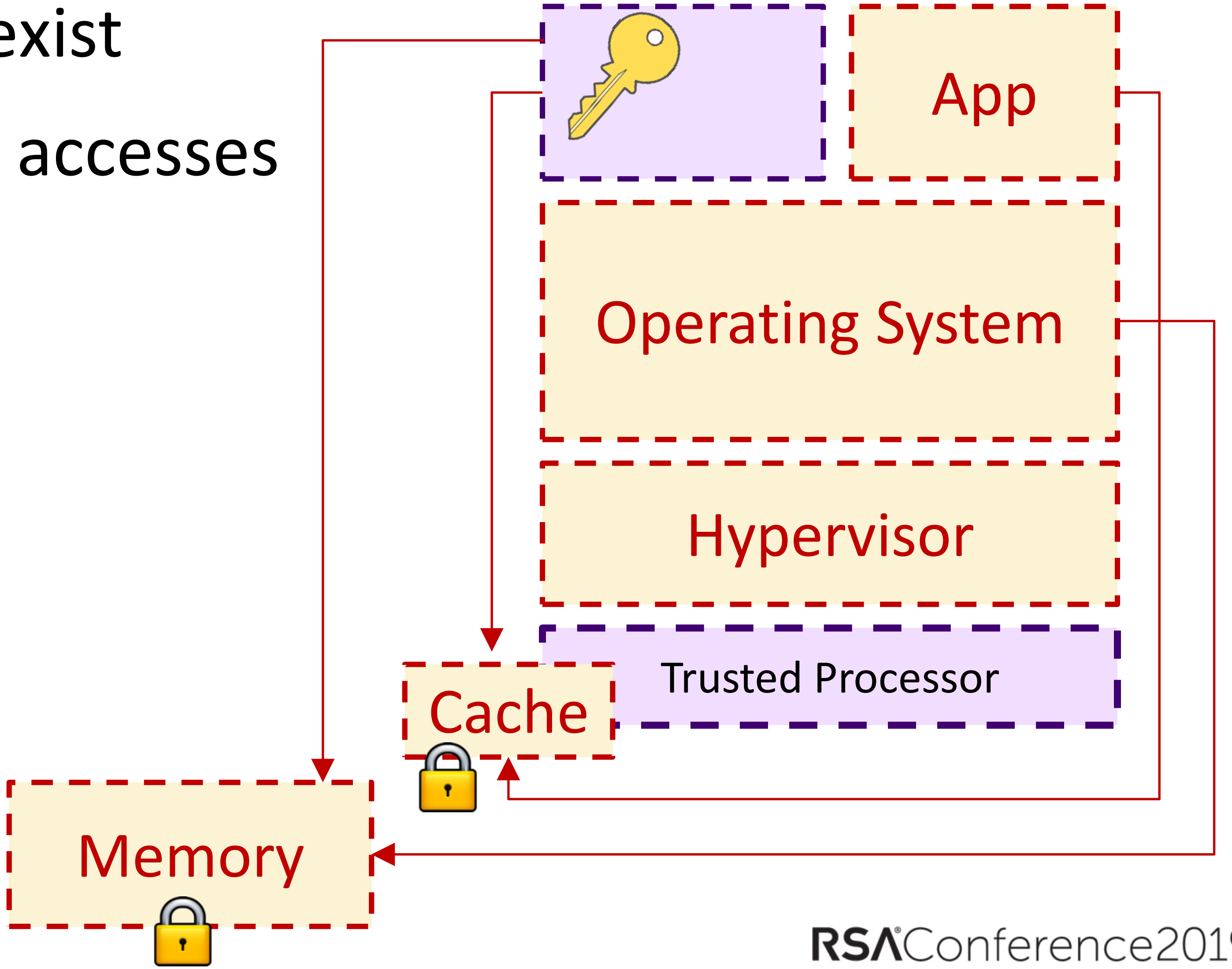
# Host(ile) environment & shared resources

- Many side channels may exist
- Leakage through memory accesses



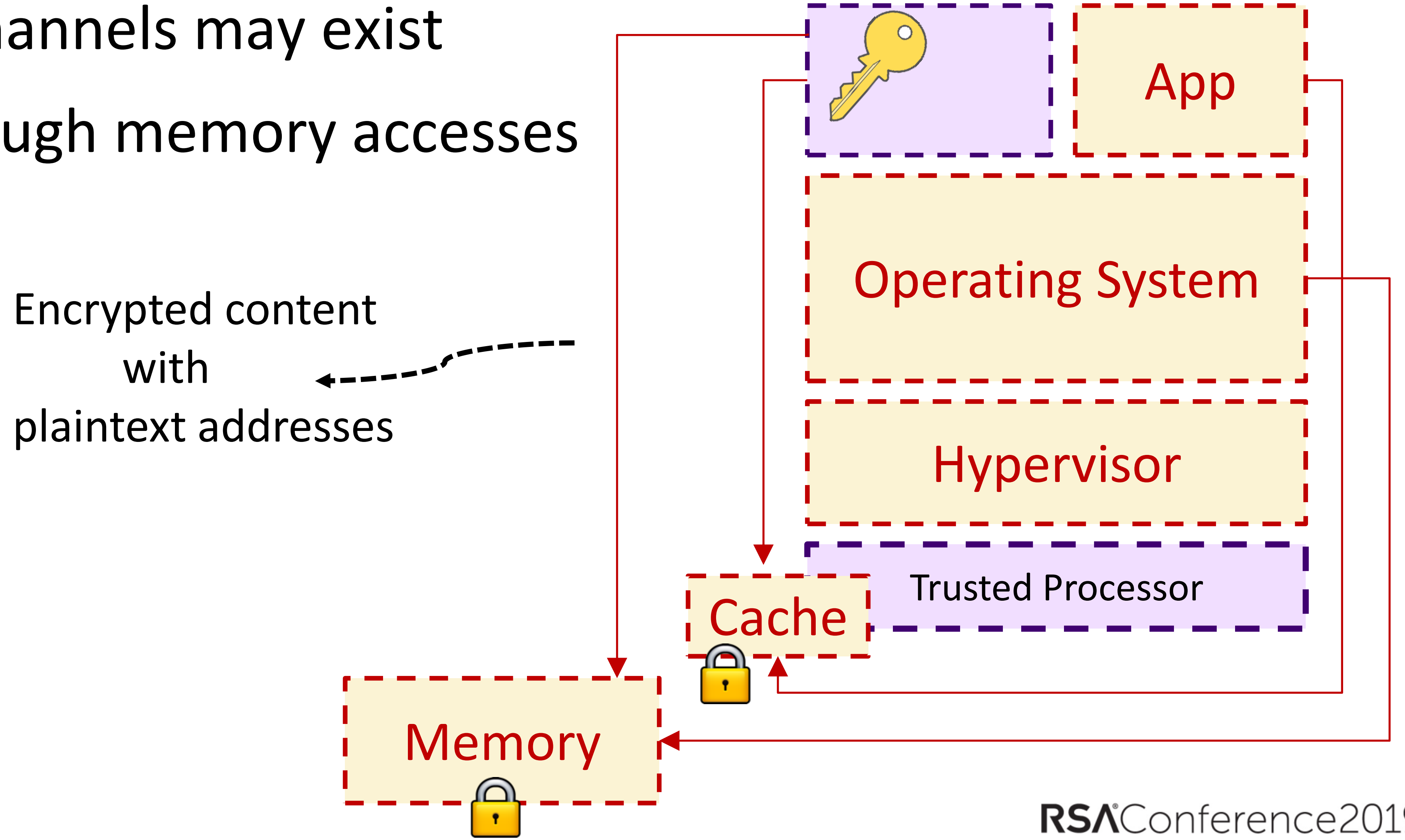
# Host(ile) environment & shared resources

- Many side channels may exist
- Leakage through memory accesses



# Host(ile) environment & shared resources

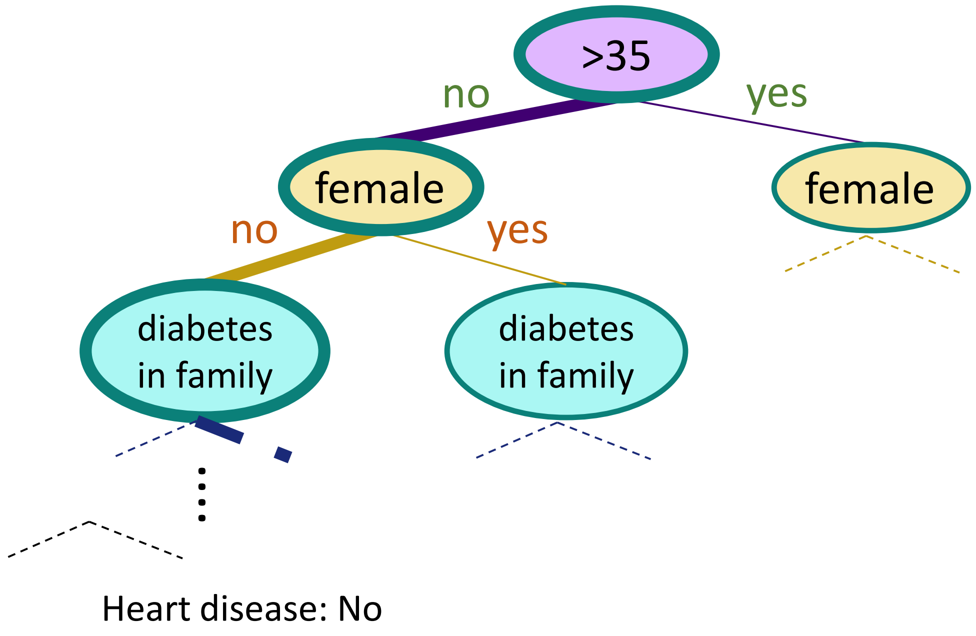
- Many side channels may exist
- Leakage through memory accesses



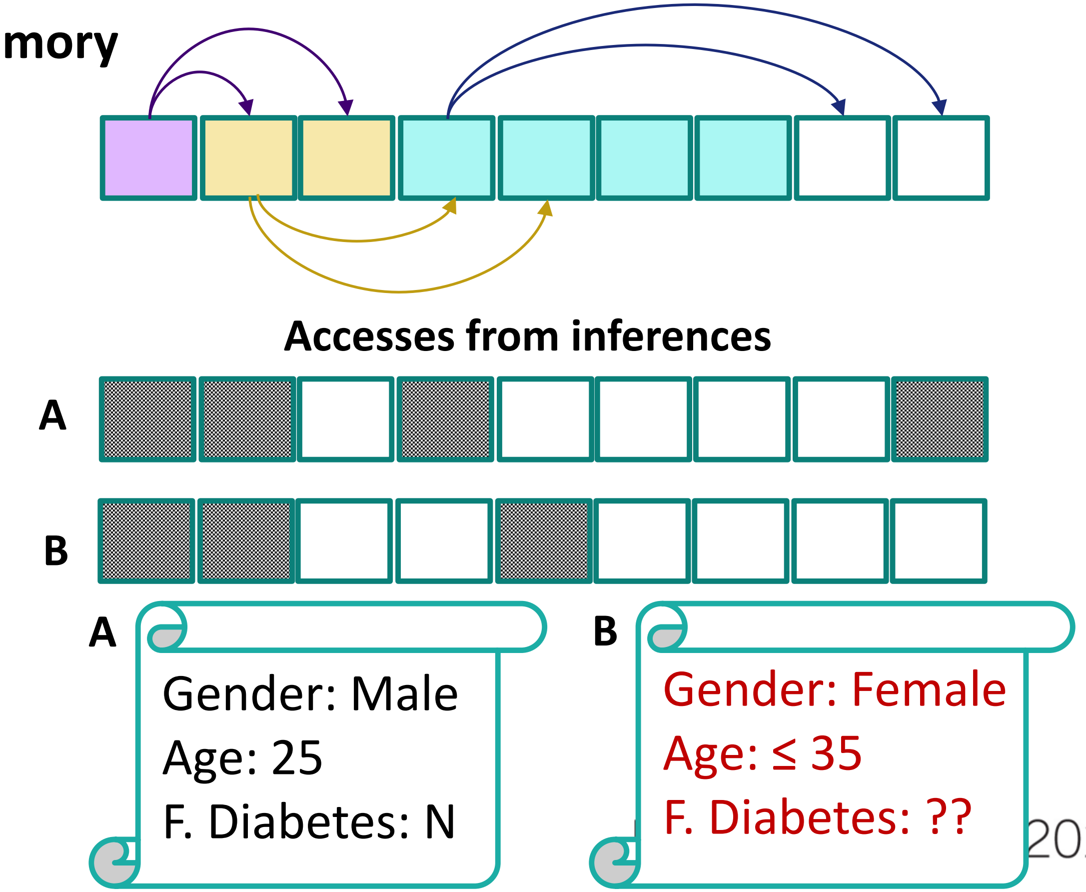
# Memory Channels: What is leaked

- Memory side-channels are not new for cryptographic code
- Application: use binary tree to classify a record (access secret-dependent path)

**Binary decision tree**




**Memory**



# Mitigating Memory Side-channel Attacks

- Not an easy problem: Let’s make random dummy accesses, shuffle, etc:
  - Hard to estimate what is leaked
  - Leaking even one bit may be dangerous

# Mitigating Memory Side-channel Attacks

- Not an easy problem: Let’s make random dummy accesses, shuffle, etc:
  - Hard to estimate what is leaked
  - Leaking even one bit may be dangerous
- We assume worst-case scenario:
  - Attacker observes all accesses
  - Game lost if the attacker guesses at least one bit

# Mitigating Memory Side-channel Attacks

- Not an easy problem: Let’s make random dummy accesses, shuffle, etc:
  - Hard to estimate what is leaked
  - Leaking even one bit may be dangerous
- We assume worst-case scenario:
  - Attacker observes all accesses
  - Game lost if the attacker guesses at least one bit
- Our approach:
  - Model the attacker
  - Security definition (data-oblivious algorithms)
  - Design provably-secure algorithms in this model

# Towards Data-obliviousness

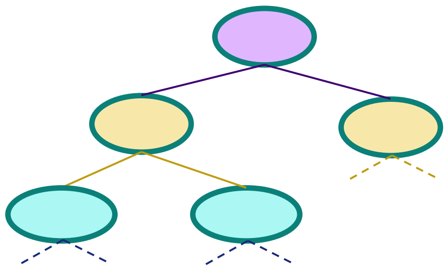
1. Isolating computation in private memory
    - Registers
    - Transactional memory (TSX)

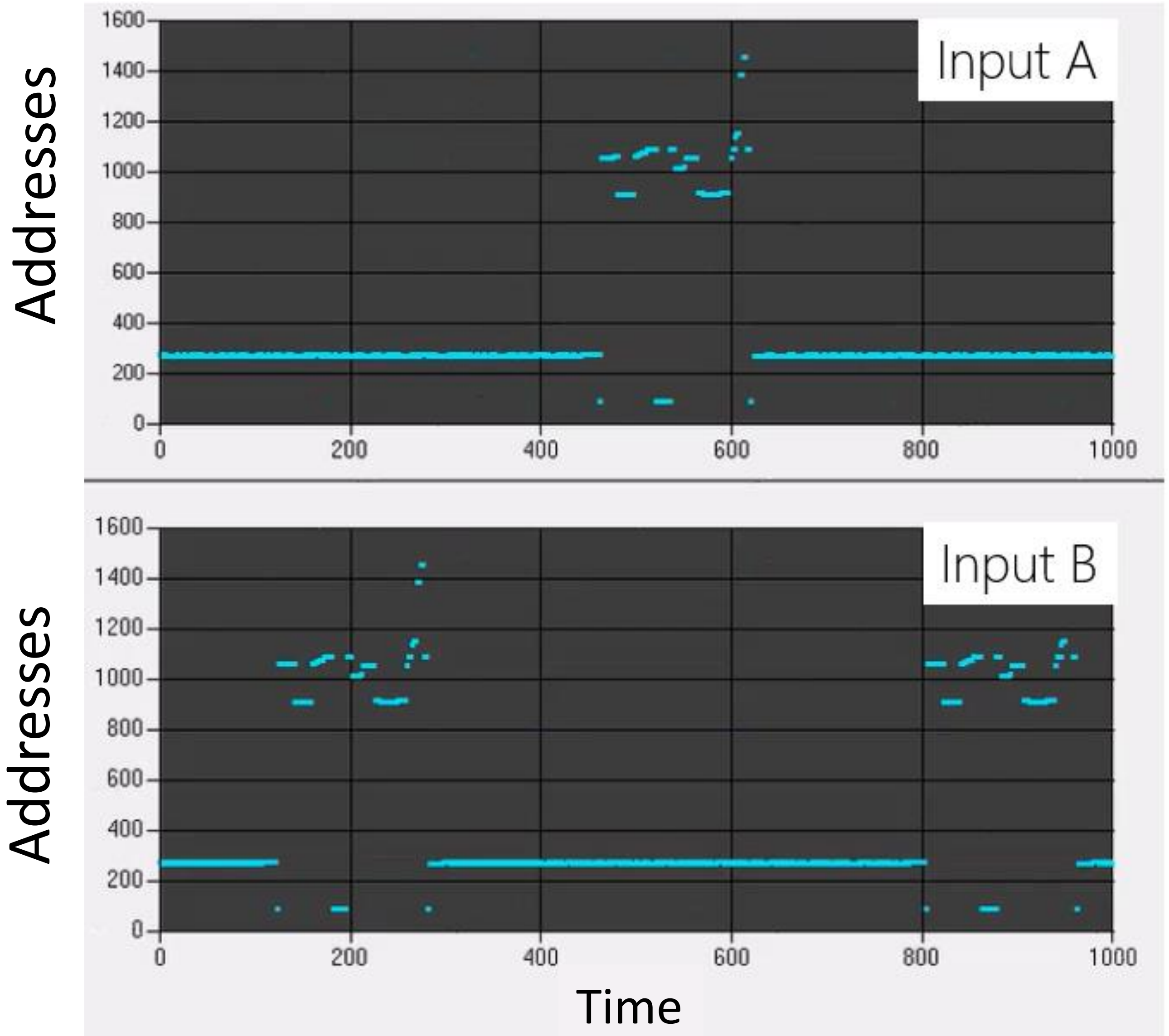
2. General software-based approach
    - Oblivious machine-learning algorithms
    - Oblivious RAM:
        - structured dummy and randomized accesses

# Are we data-oblivious?

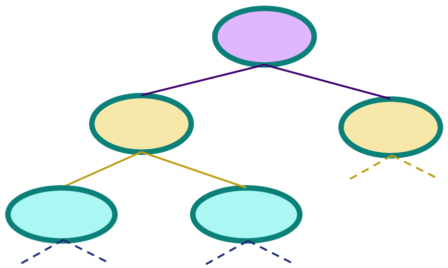
- Provably-secure algorithms:
  - the trace depends only on public information (e.g., input, output sizes)
- Validation of implementation:
  - collected traces at cache-line (64byte) granularity with Intel Pin Tool
- Video of traces from:
  - original tree traversal
  - data-oblivious tree traversal

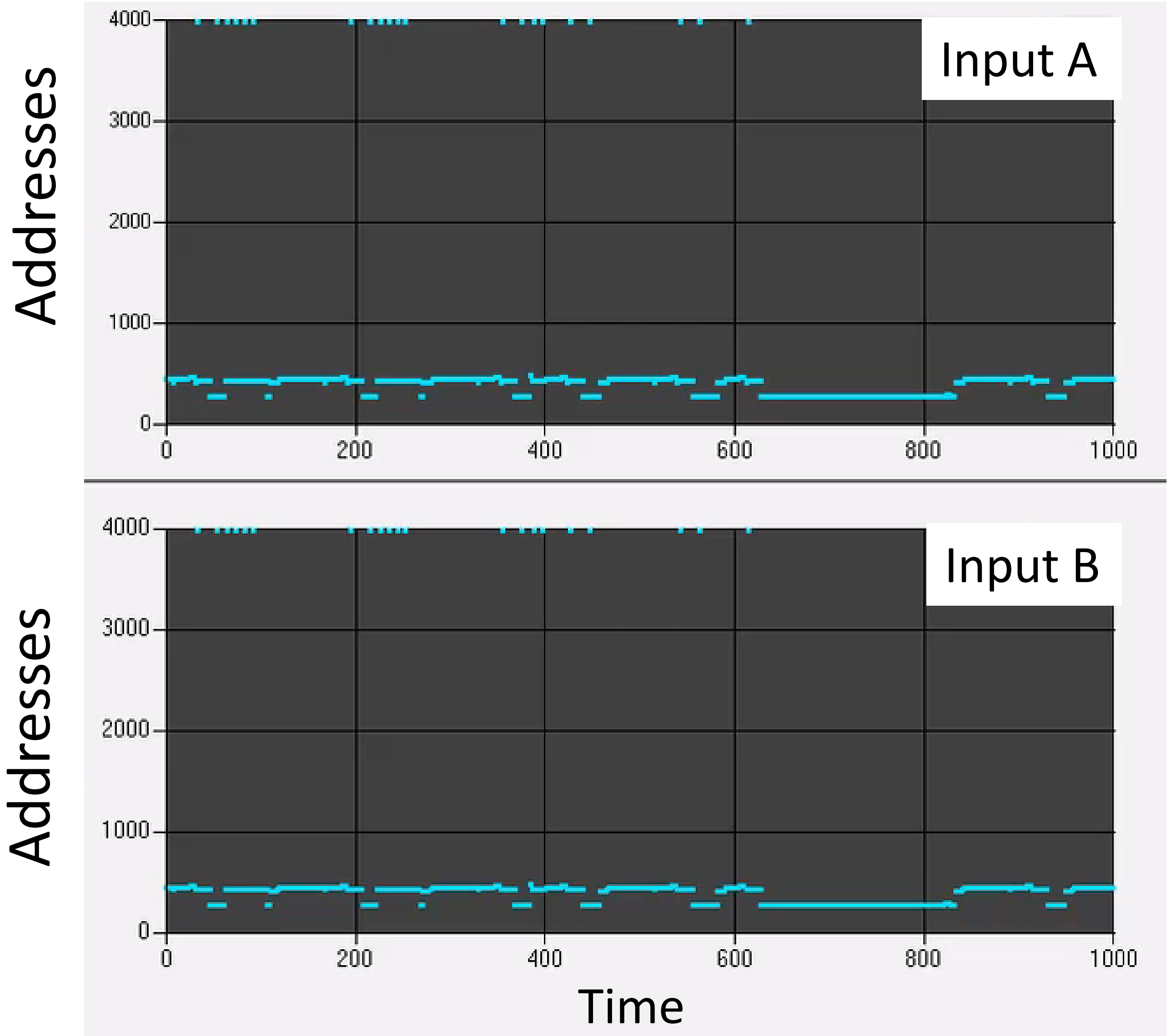
# Trees: Non-Oblivious Code Traces





# Trees: Oblivious Code Traces





RSA Conference 2019

# Summary

# Summary: Confidential Computing

- Protect data during computation:
  - with trusted execution environments (TEEs)
- Scenarios:
  - confidential consortium blockchains
  - multi-party machine learning
- Guarantees beyond TEE isolation:
  - integrity and privacy in multi-party machine learning
  - memory side-channel mitigation

# Apply

- TEEs in Azure Confidential Computing
- Open Source SDK for TEEs: Open Enclave
- Always Encrypted with Secure Enclaves
- Design applications with small attack surface

# Azure Confidential Computing Links

- Azure confidential computing solution page: https://azure.microsoft.com/en-us/solutions/confidential-compute/
- Confidential Computing VM Deployment: http://aka.ms/ccvm
- Open Enclave SDK page: https://openenclave.io/sdk/
- Open Enclave GitHub repository: https://aka.ms/OESDKGitHubRepo

# Thank you!

## Please see the papers for all the details

Observing and Preventing Leakage in MapReduce
Olga Ohrimenko, Manuel Costa, Cédric Fournet, Christos Gkantsidis, Markulf Kohlweiss, and Divya Sharma,
*ACM Conference on Computer and Communications Security, 2015*

VC3: Trustworthy Data Analytics in the Cloud using SGX
Felix Schuster, Manuel Costa, Cédric Fournet, Christos Gkantsidis, Marcus Peinado, Gloria Mainar-Ruiz, Mark Russinovich
*IEEE Symposium on Security and Privacy, 2015*

Oblivious Multi-party Machine Learning on Trusted Processors
Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Metha, Kapil Vaswani, Manuel Costa
*Usenix Security Symposium, 2016*

Strong and Efficient Cache Side-Channel Protection using Hardware Transactional Memory
Daniel Gruss, Julian Lettner, Felix Schuster, Olga Ohrimenko, Istvan Haller, Manuel Costa
*Usenix Security Symposium, 2017*

EnclaveDB – A Secure Database using SGX
Christian Priebe, Kapil Vaswani, Manuel Costa
*IEEE Symposium on Security & Privacy, 2018*

Contamination Attacks and Defences in Multi-Party Machine Learning
Jamie Hayes and Olga Ohrimenko
*NeurIPS, 2018*

Graviton: Trusted Execution Environments on GPUs
Stavros Volos, Kapil Vaswani, Rordigo Bruno
*OSDI, 2018*

An Algorithmic Framework For Differentially Private Data Analysis on Trusted Processors
Joshua Allen, Bolin Ding, Janardhan Kulkarni, Harsha Nori, Olga Ohrimenko, Sergey Yekhanin
TechReport, 2018

RSAConference2019