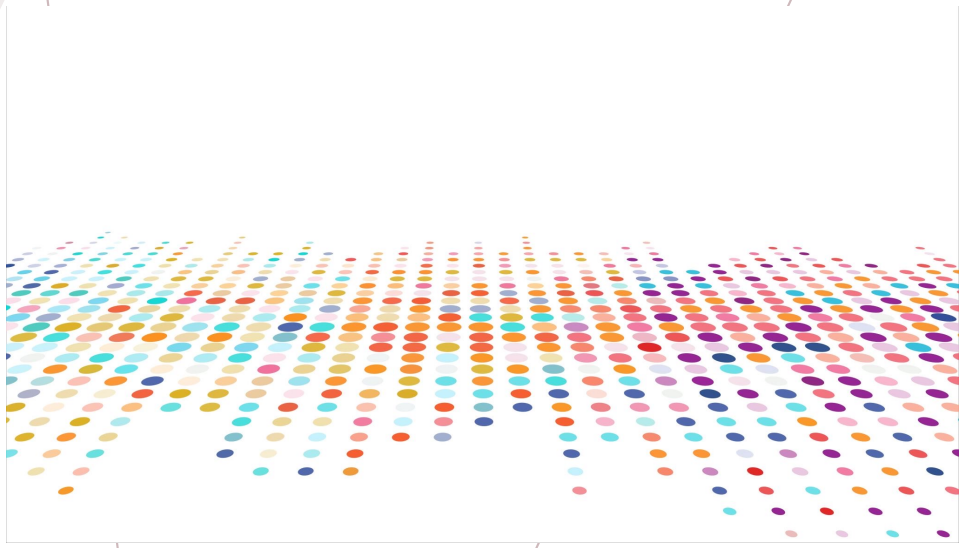
Offloading for Secure Microservice Architectures

NAOHIRO HAYASHIBARA

KYOTO SANGYO UNIVERSITY
UNIVERSITÉ DE NEUCHÂTEL

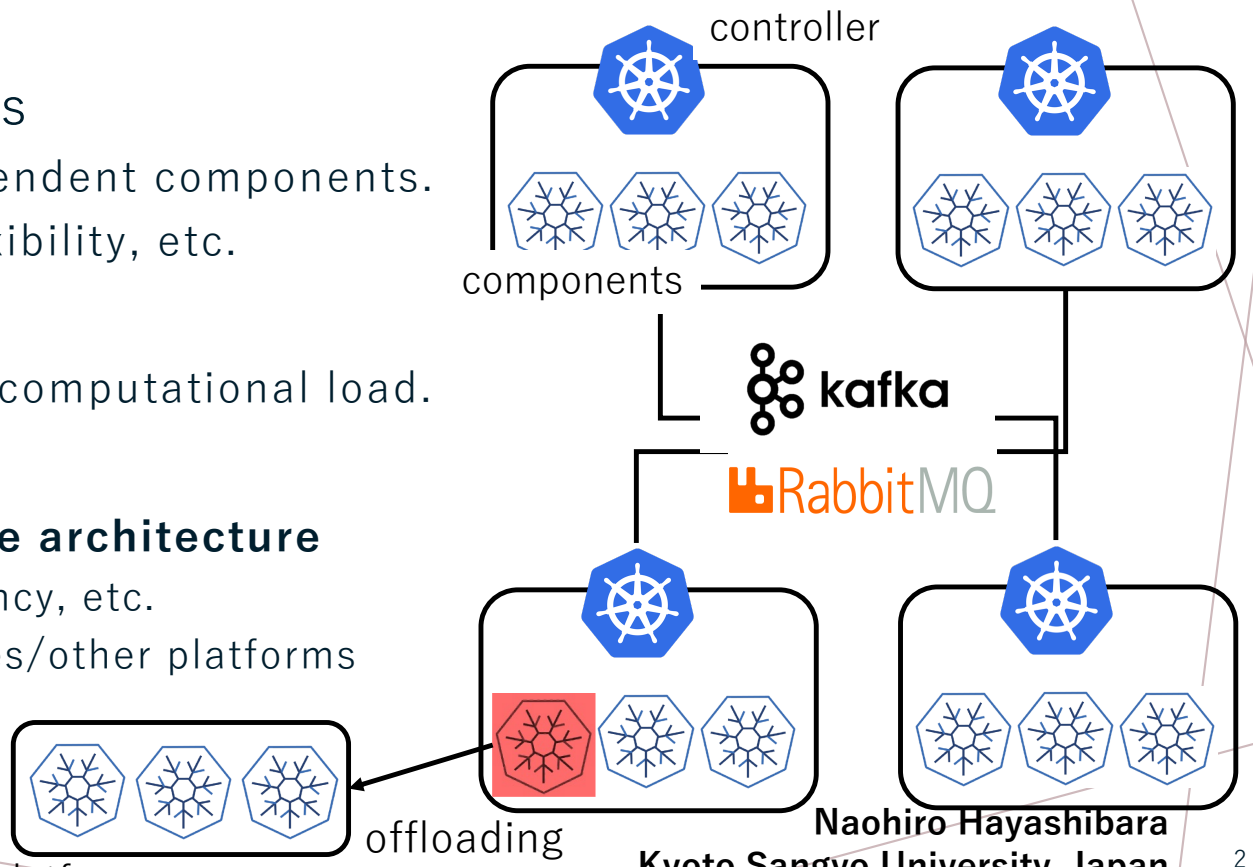
KYOTO SANGYO UNIVERSITY

UNIVERSITÉ DE NEUCHÂTEL



Microservice Architectures

- Microservice architectures
  - Service is built with independent components.
  - To improve scalability, flexibility, etc.
- Problem
  - Concentration of network/computational load.
- Solution
  - **Offloading in microservice architecture**
    - Load-reduction, redundancy, etc.
    - Offloading to edge devices/other platforms



Naohiro Hayashibara
Kyoto Sangyo University, Japan

Security Issues on Offloading

Increased Attack Surface

- Access to devices and platforms
- Deploying components outside

Untrusted Environment

- Nodes might be semi-honest
- Data can be intercepted and leaked

Data Protection

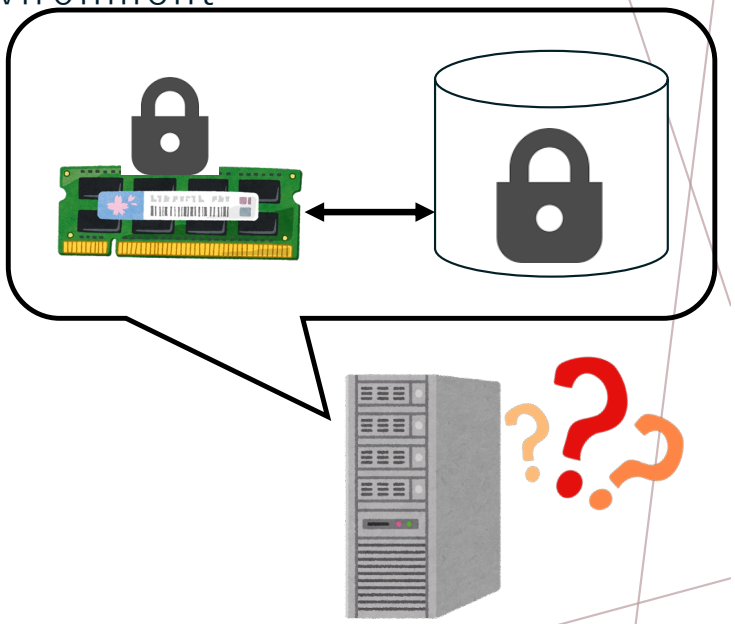
- Data should be encrypted or isolated from untrusted env.
  - Side-channel attack
  - Access pattern analysis

Data Transfer

- Communication increases between components
- Data should be protected in transfer

Secure Offloading

- Purpose
  - Safe execution in **untrusted** computational environment
    - Minimizing data or information exposed.
- Key Technologies for Secure Offloading
  - Data Protection
    - **Trusted Execution Environment** (TEE)
      - Intel SGX, AMD SEV, Arm TrustZone
  - Data Obfuscation
    - **Oblivious Random Access Memory** (ORAM)
    - Code Obfuscation
  - Secure Multi-Party Computation (MPC)
    - Oblivious Transfer (OT)
    - **Private Set Intersection** (PSI)



Naohiro Hayashibara
Kyoto Sangyo University, Japan

Challenges

- Secure offloading of building blocks in microservices
  - Agreement Protocols
    - Leader election
    - Membership management
  - Messaging
    - Publish/subscribe systems
  - Logging
    - Distributed tracing (DT)
  - Voting
    - Majority voting
  - Resource Management
    - Resource Allocation
    - Load Balancer

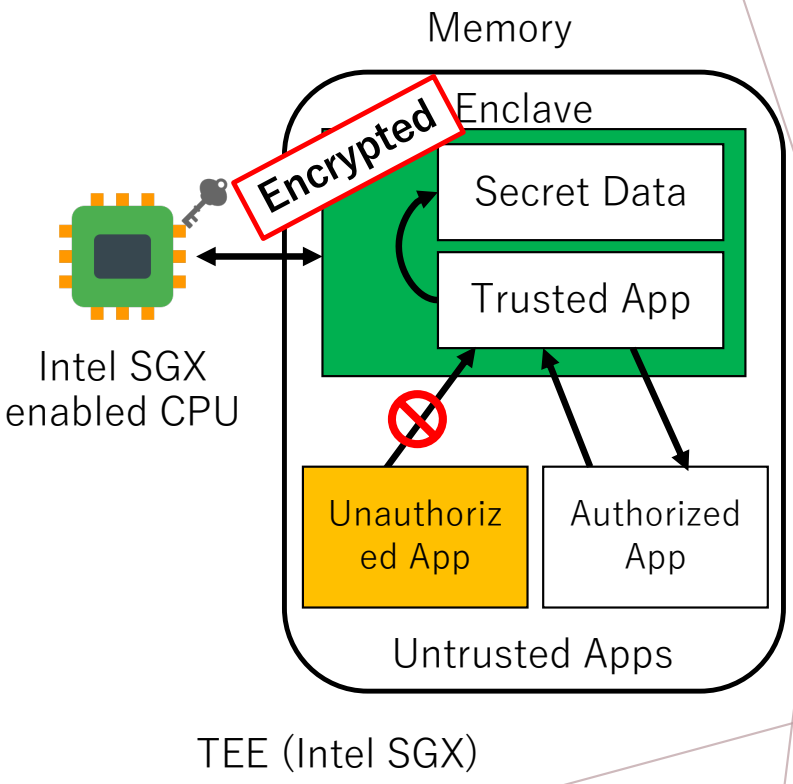
Solutions with obfuscation and MPC

- Leader Election
  - Single Secret Leader Election (SSLE) [Boneh et al. 2020]
    - Trusted Execution Environment (TEE)
- Distributed Tracing
  - Obfuscated Logging
    - An extension of Oblivious Random Access Memory (ORAM)
    - Searchable encryption
- Resource Allocation and Optimization
  - Oblivious Resource Manager for Microservices
    - Private Set Interaction (PSI)

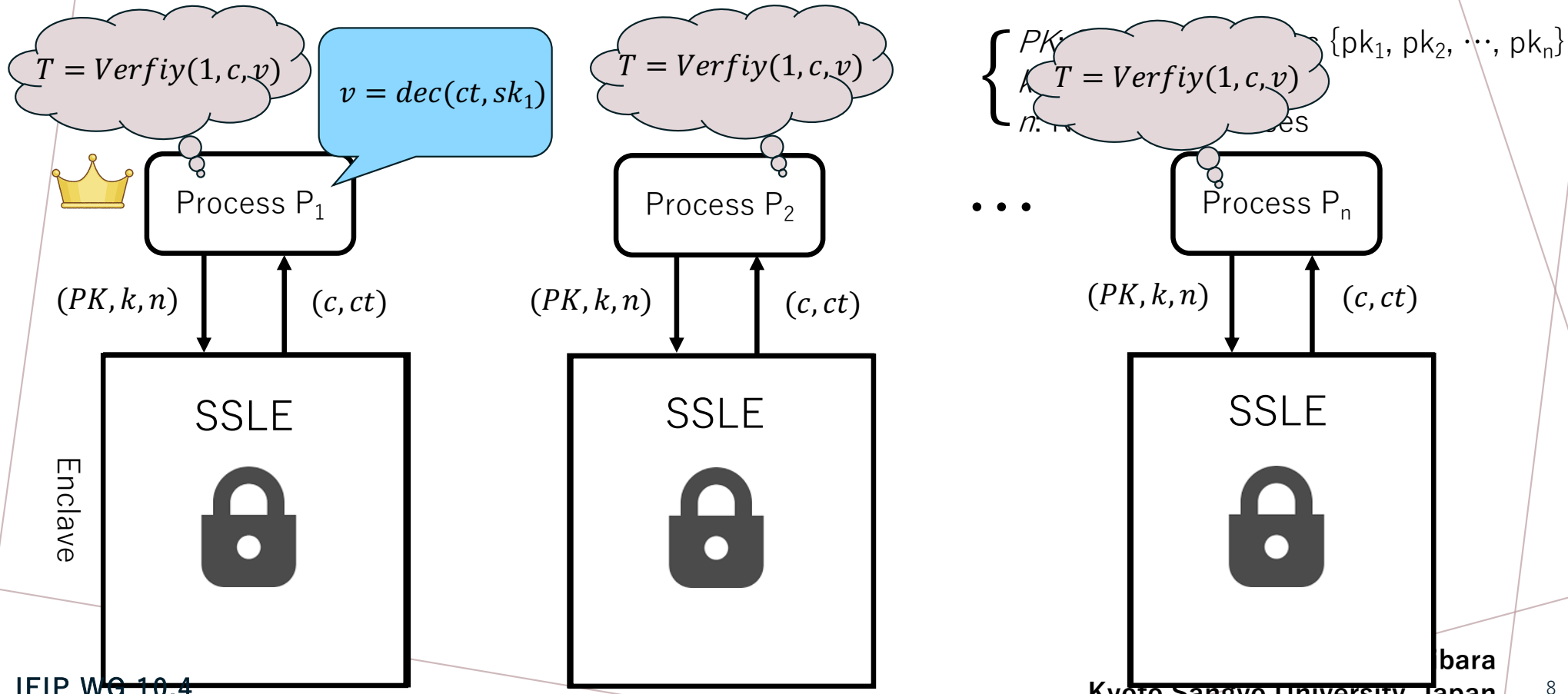
Naohiro Hayashibara
Kyoto Sangyo University, Japan

Leader Election

- Single Secret Leader Election (SSLE) [Boneh et al. 2020]
  - Electing a leader
    - Paxos, Raft, PoS-based blockchain, etc.
  - Purpose
    - Ensure unpredictability of the leader to be elected
  - Key Technology
    - TEE (e.g., Intel SGX)



Single Secret Leader Election



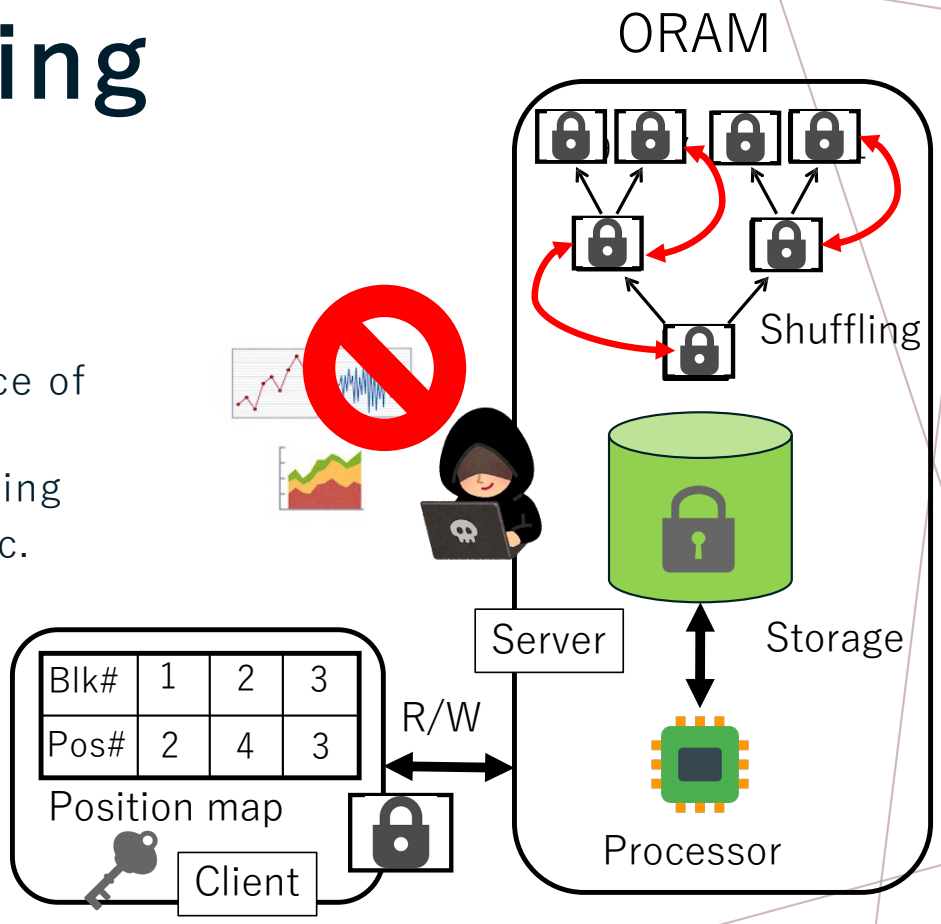
Solutions with obfuscation and MPC

- Leader Election
  - Single Secret Leader Election (SSLE)
    - Trusted Execution Environment (TEE)
- Distributed Tracing
  - Obfuscated Logging
    - An extension of Oblivious Random Access Memory (ORAM)
    - Searchable encryption
- Resource Allocation and Contention Awareness
  - Oblivious Scheduling for Microservices
    - Private Set Interaction (PSI)

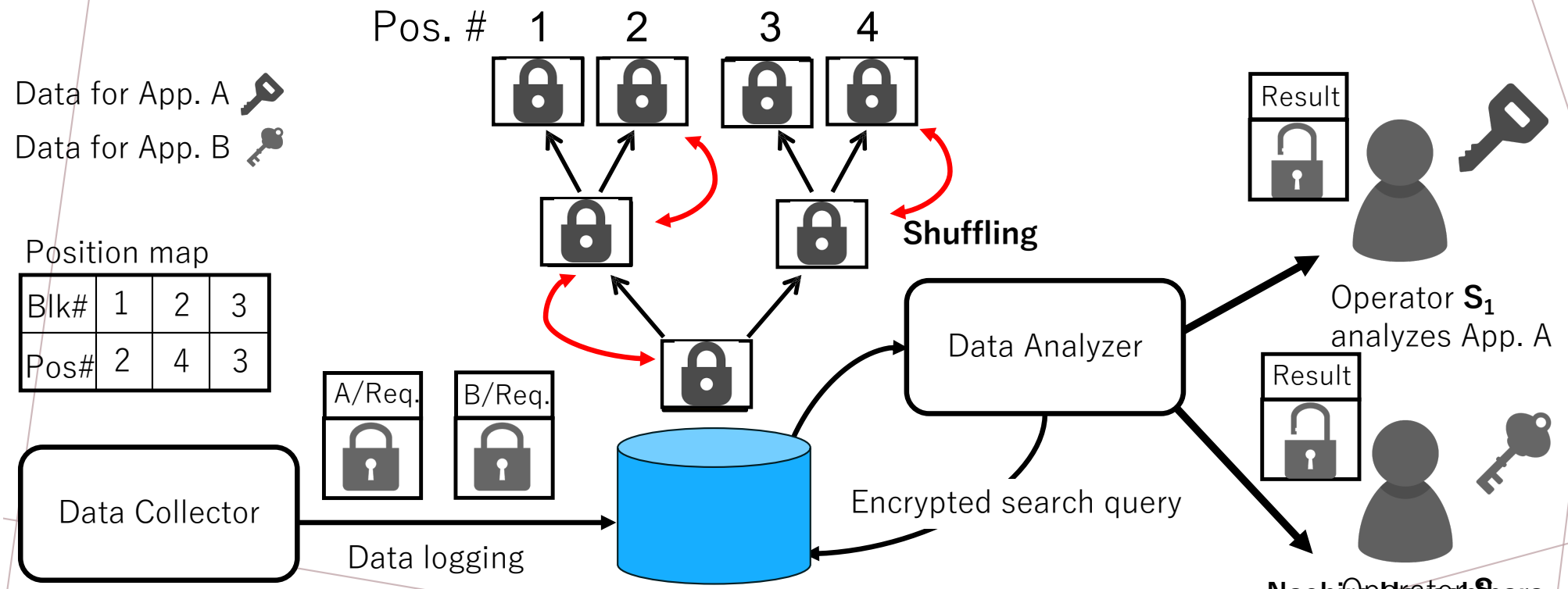
Naohiro Hayashibara
Kyoto Sangyo University, Japan

Obfuscated Logging

- Distributed Tracing
  - Logging requests
    - Monitoring and analyzing the performance of whole system
    - Anomaly detection using distributed tracing
    - Tools: OpenTelemetry, Zipkin, Jaeger, etc.
  - Issue
    - Collected data could be revealed
    - Encrypted DB does not hide access pattern to data
  - Key Technologies
    - An extension of ORAM
    - Searchable encryption



Obfuscated Logging



Summary

- Safe offloading for Microservice Architectures
  - Using obfuscation and MPC
    - Leader election
    - Distributed tracing
  - Minimizing data and information exposed to untrusted nodes.
  - Improves performance, reliability without increasing security risk.
- Future work
  - Implement prototypes and performance measurement.
  - Formal verification (e.g., model checking)

Naohiro Hayashibara
Kyoto Sangyo University, Japan

Naohiro Hayashibara
Kyoto Sangyo University, Japan

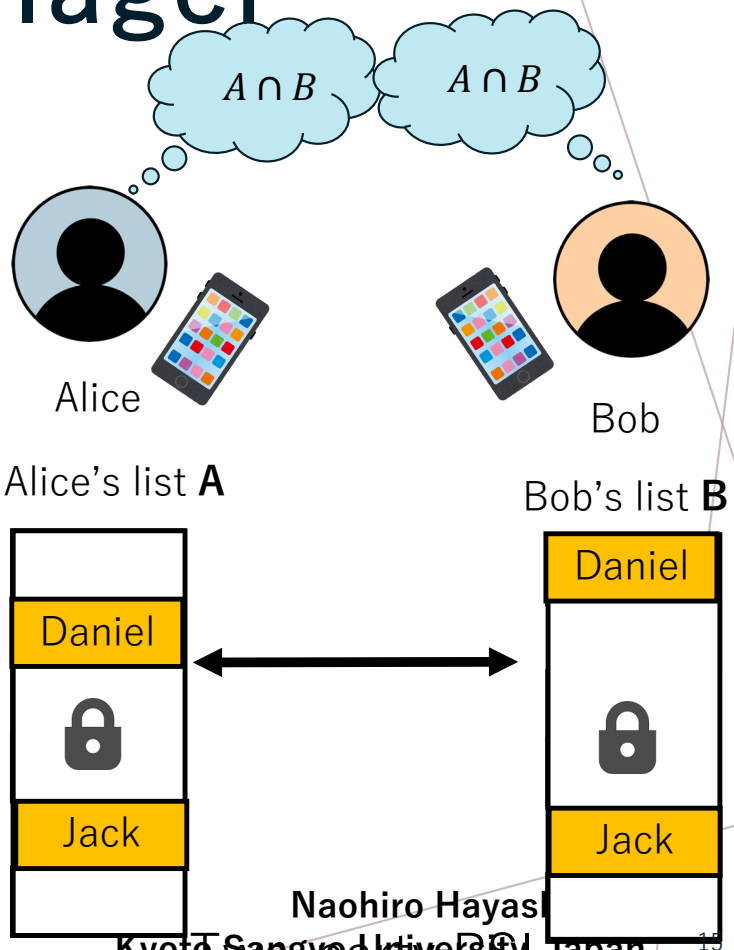
Solutions with obfuscation and MPC

- Leader Election
  - Single Secret Leader Election (SSLE)
    - Trusted Execution Environment (TEE)
- Distributed Tracing
  - Obfuscated Logging
    - An extension of Oblivious Random Access Memory (ORAM)
    - Searchable encryption
- Resource Allocation and Contention Awareness
  - Oblivious Scheduling for Microservices
    - Private Set Interaction (PSI)

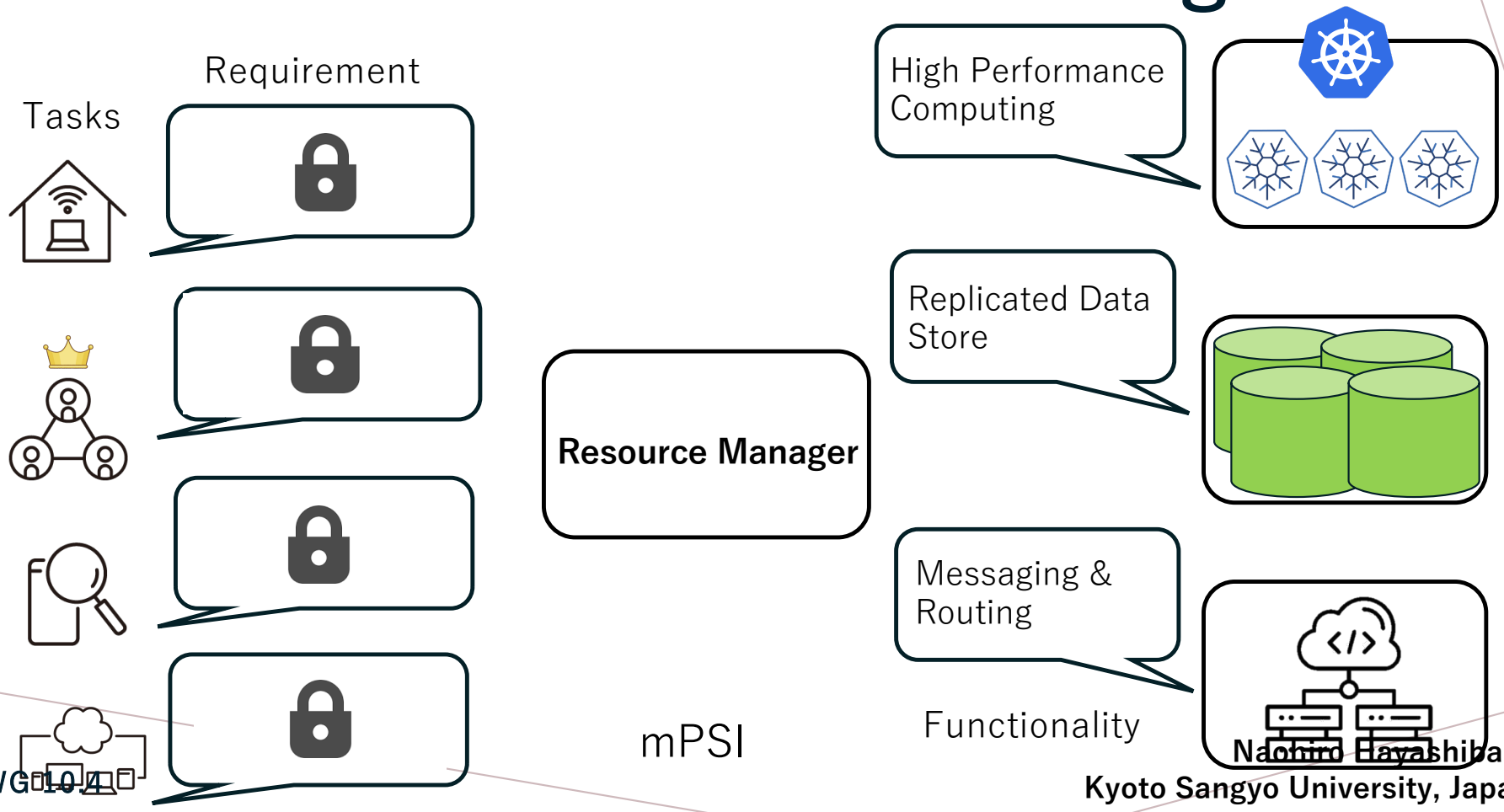
Naohiro Hayashibara
Kyoto Sangyo University, Japan

Oblivious Resource Manager

- Resource Management in Microservices
  - Resource Optimization
    - Allocating tasks taking account of requirements
  - Issue
    - All execution details should not be exposed
      - Resource and service that are used in a component
  - Purpose
    - Minimizing the information leak regarding the execution.
  - Key Technologies
    - Multiparty Private Set Intersection (mPSI) [Kolesnikov et al. 2017]



Oblivious Resource Manager



Summary

- Safe offloading for Microservice Architectures
  - Using obfuscation and MPC
    - Leader election
    - Distributed tracing
    - Resource management
  - Minimizing data and information exposed to untrusted nodes.
  - Improves performance, reliability without increasing security risk.
- Ongoing tasks
  - Implement prototypes and performance measurement.
  - Formal verification (e.g., model checking) for ensuring obfuscation.