UCLA Samueli
Electrical & Computer Engineering

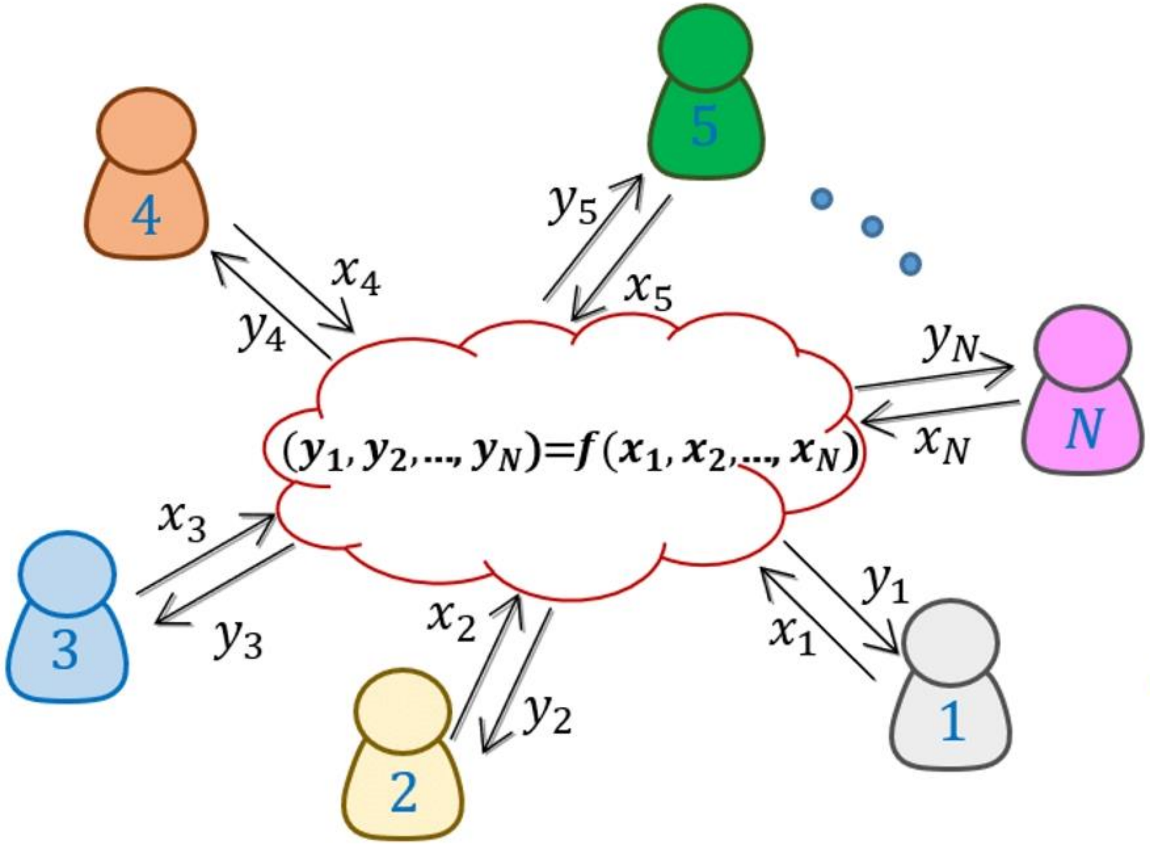
ECE 209AS (03/19/2020) Final Presentation

P2I: Privacy Preserving Inferencing for Medical Cyberphysical Systems

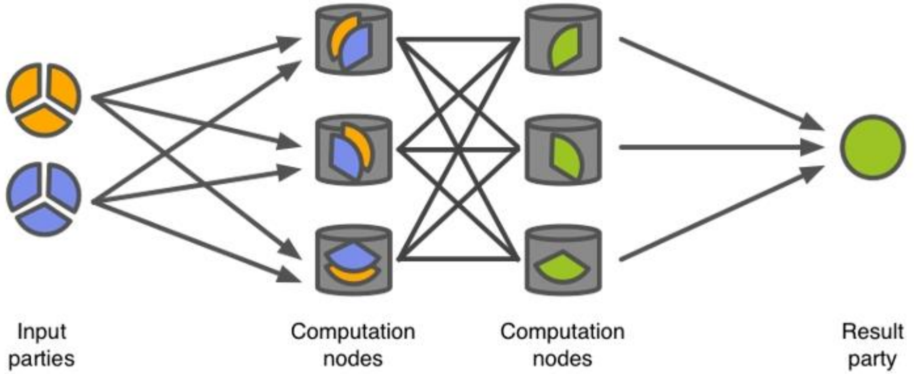
Team Members: **Swapnil Sayan Saha, Vivek Jain and Brian Wang**

Networked & Embedded Systems Laboratory

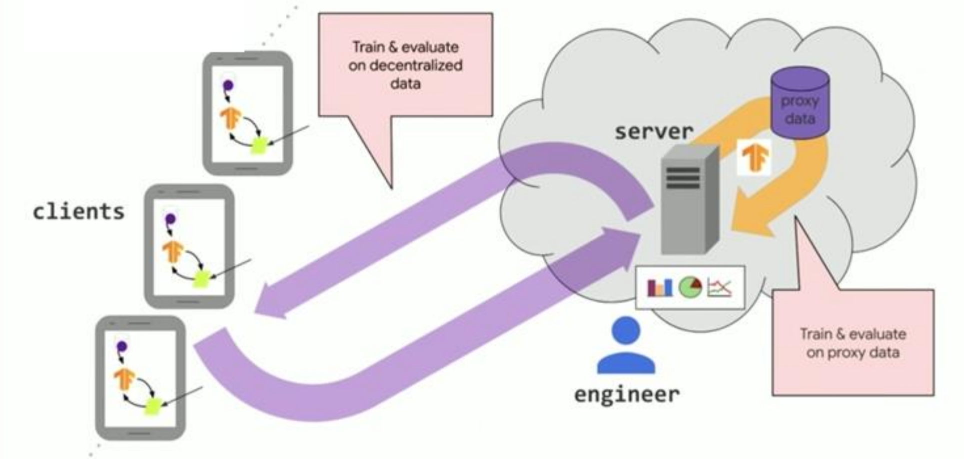
Problem Statement and Importance



Perform collaborative computation without revealing raw data

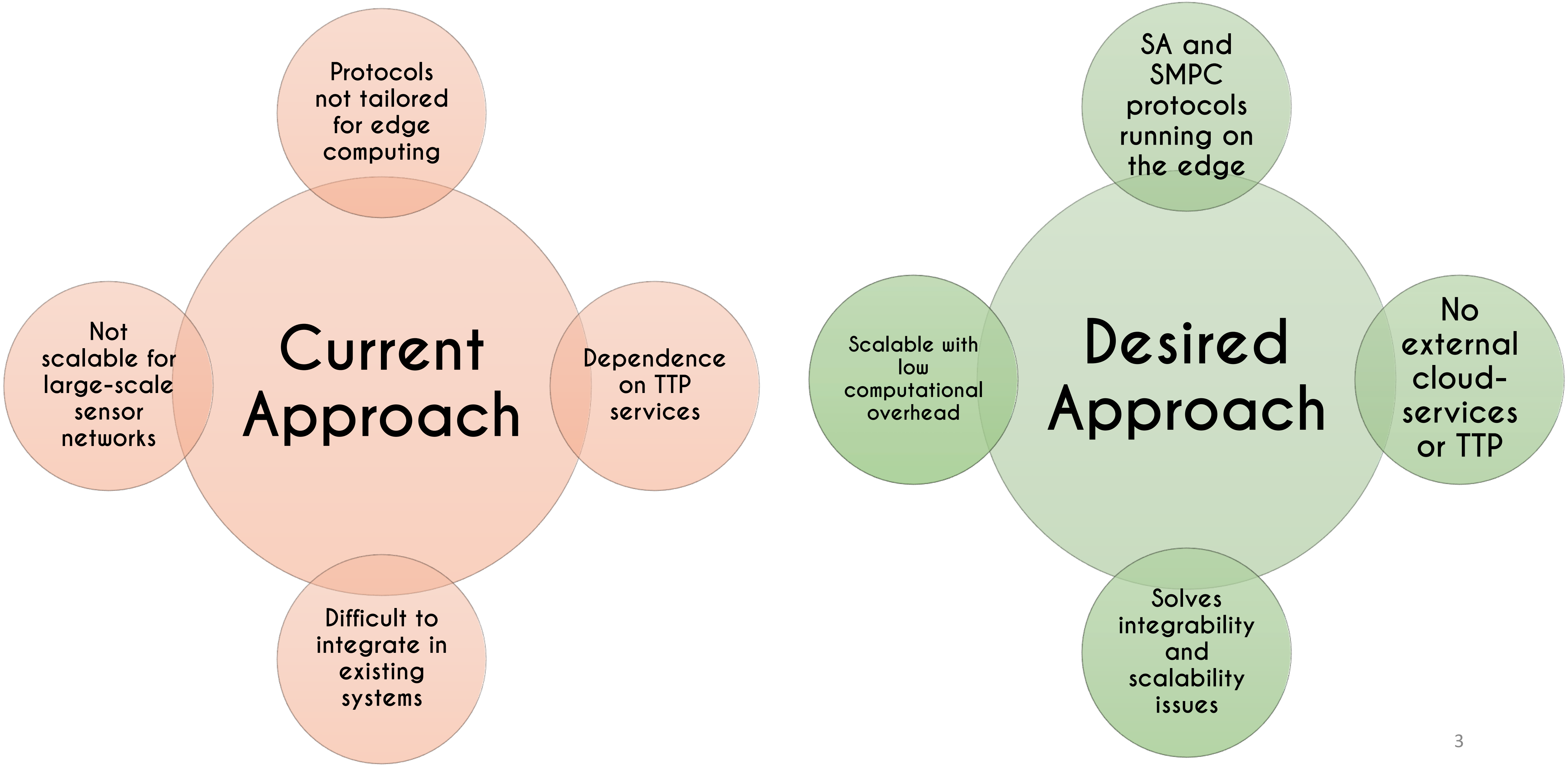


Secure Multi-Party Computation



Secure Aggregation

Problem Statement and Importance (contd.)



Overall Project Goals and Specific Aims

Specific Aims

Implement state-of-the-art SMPC and SA protocols in small-scale virtual MCPS, with computation occurring at the edge.

↓

Benchmark standard performance, privacy and security metrics of implemented SMPC and SA.

↓

Tune the parameters of state-of-the-art SMPC and SA protocols, focusing on reducing computational overhead.

↓

Implement a scalable and robust star-exchange MCPS topology embracing SMPC protocols void of centralized cloud inferencing.

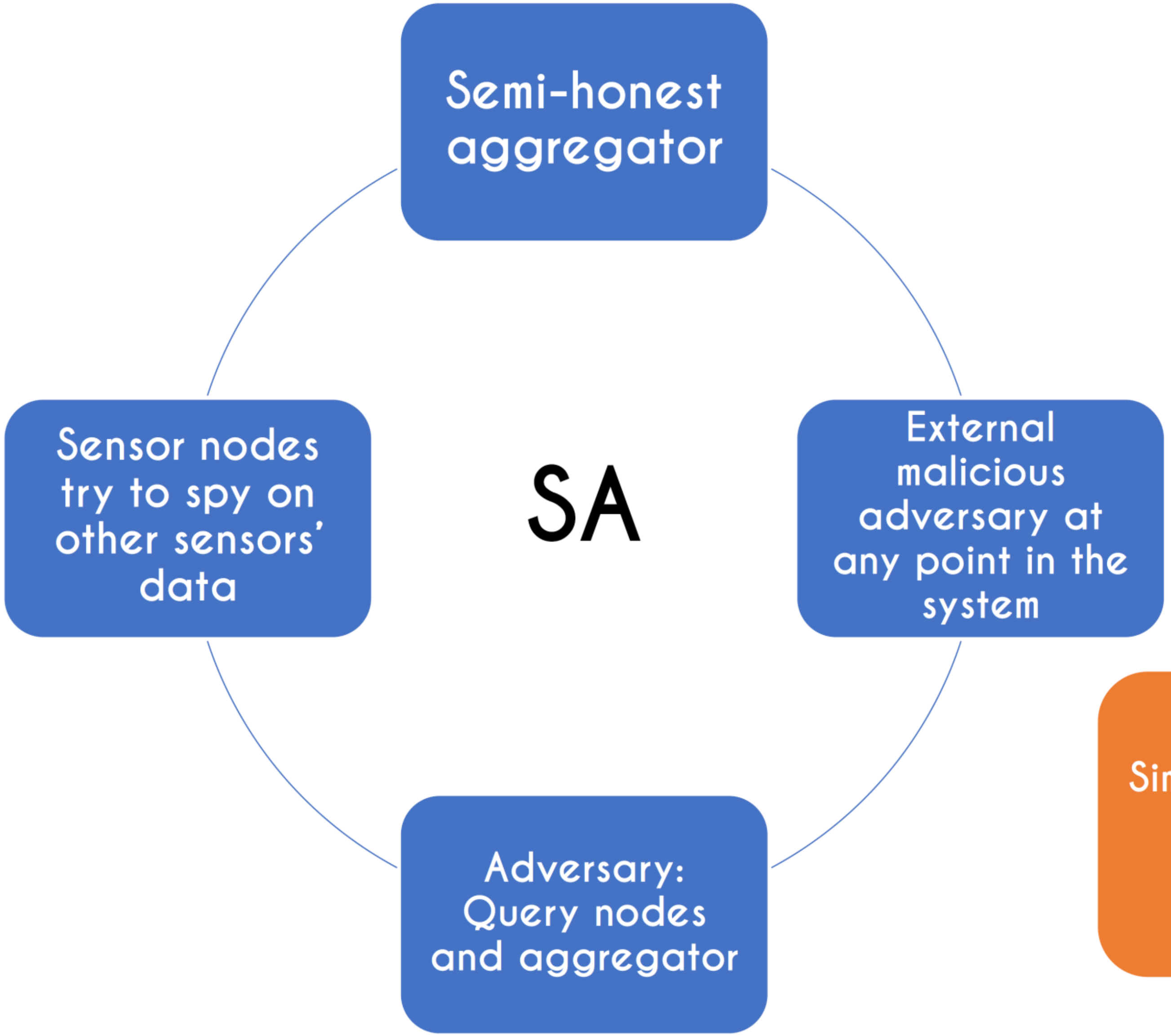
Deliverables / Goals

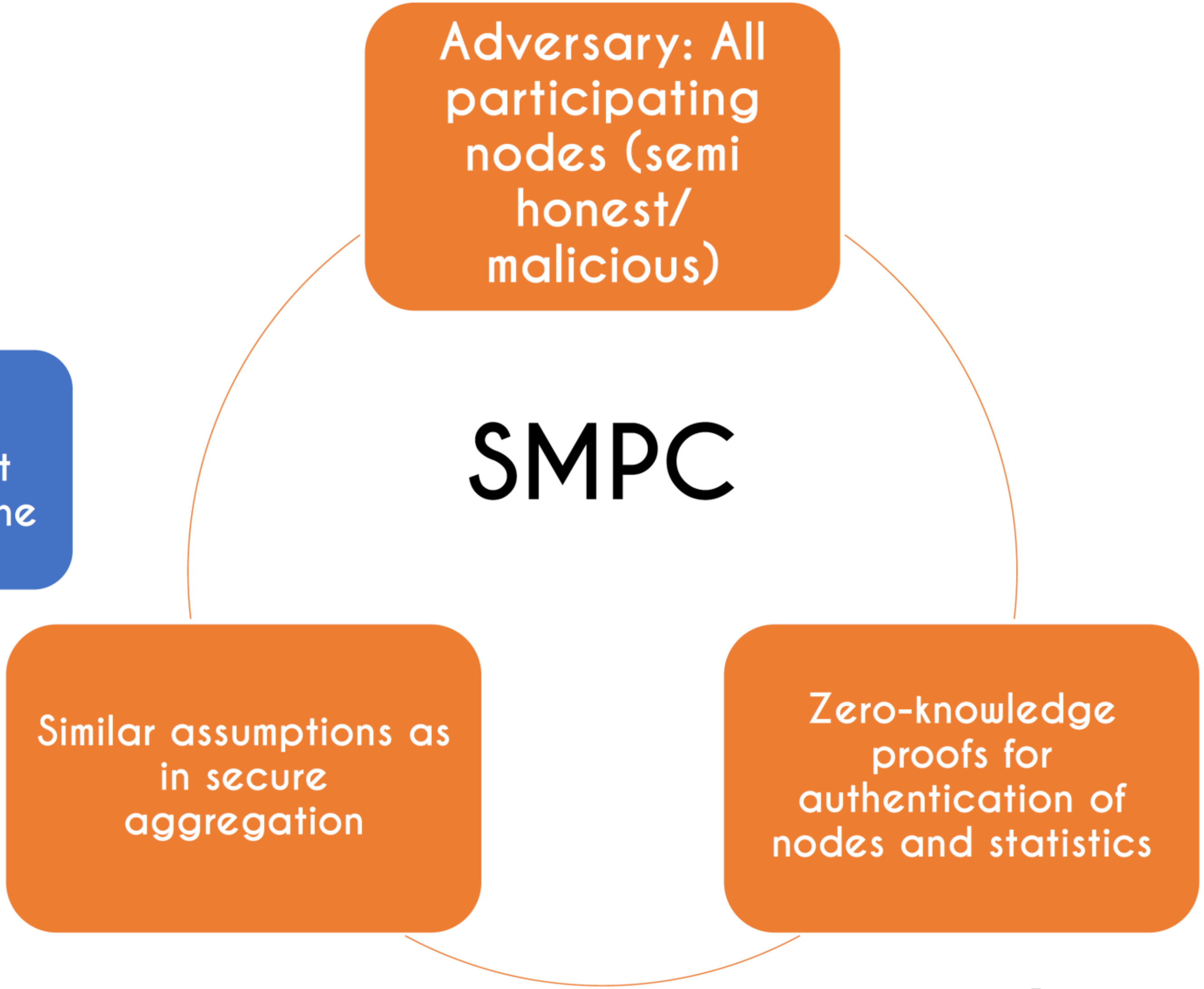
Benchmark and tune SMPC and SA protocols for resource-constrained settings

↓

A real-time scalable and robust privacy preserving inferencing system at the edge for MCPS

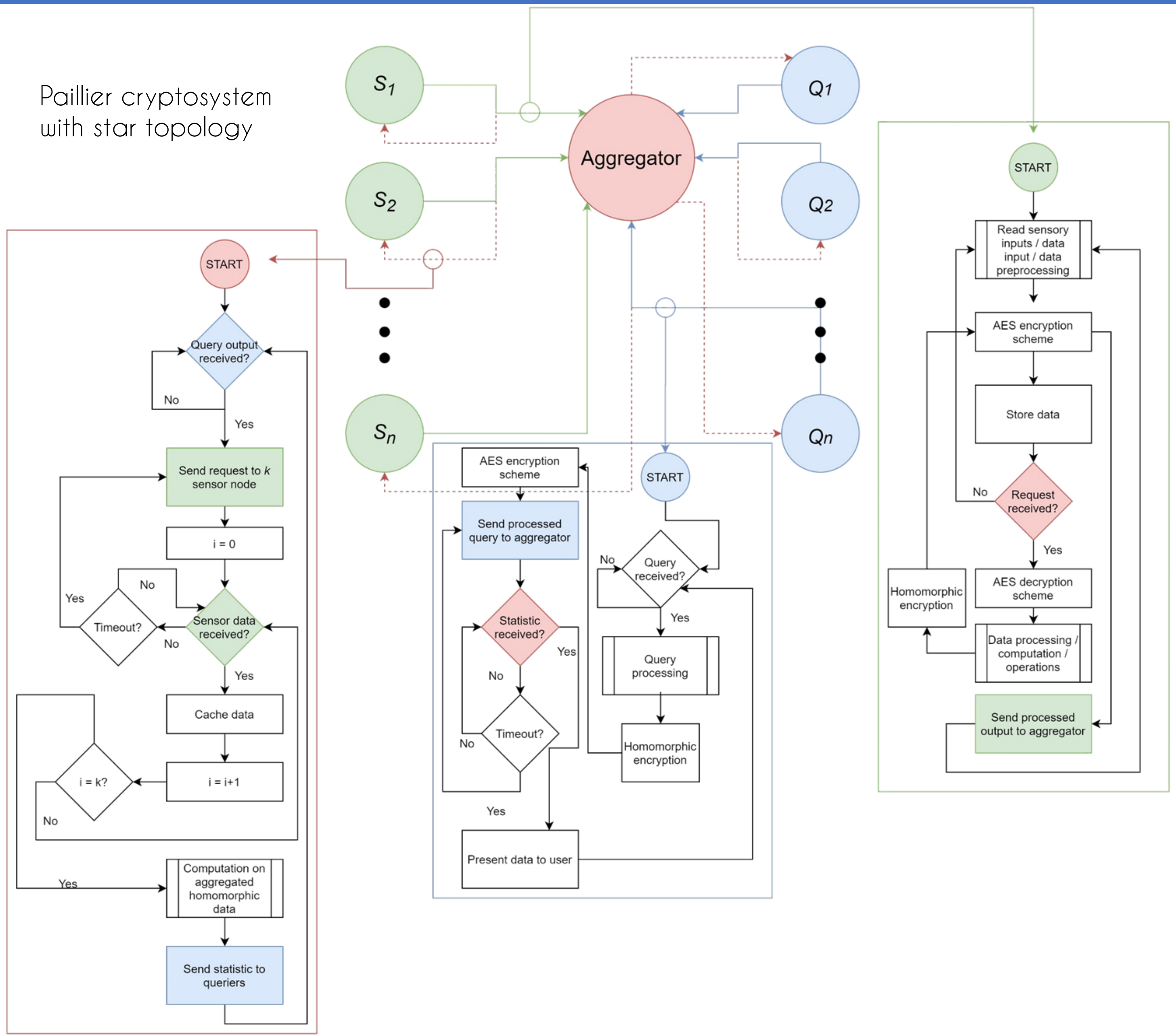
Attack Model





Technical Approach (Secure Agg.)

Paillier cryptosystem
with star topology



Key Generation:

- $pk = (n, g)$
  - $n = pq, GCD(pq, (p-1)(q-1)) = 1$
  - $g \in \mathbb{Z}^*_{n^2}$
- $sk = (\lambda, \mu)$
  - $\lambda = LMC(p-1, q-1)$
  - $\mu = (\frac{g^\lambda \mod n^2 - 1}{n})^{-1} \mod n$

Encrypt message into ciphertext:

- $c = g^m \cdot r^n \mod n^2, r \in \mathbb{Z}_n$

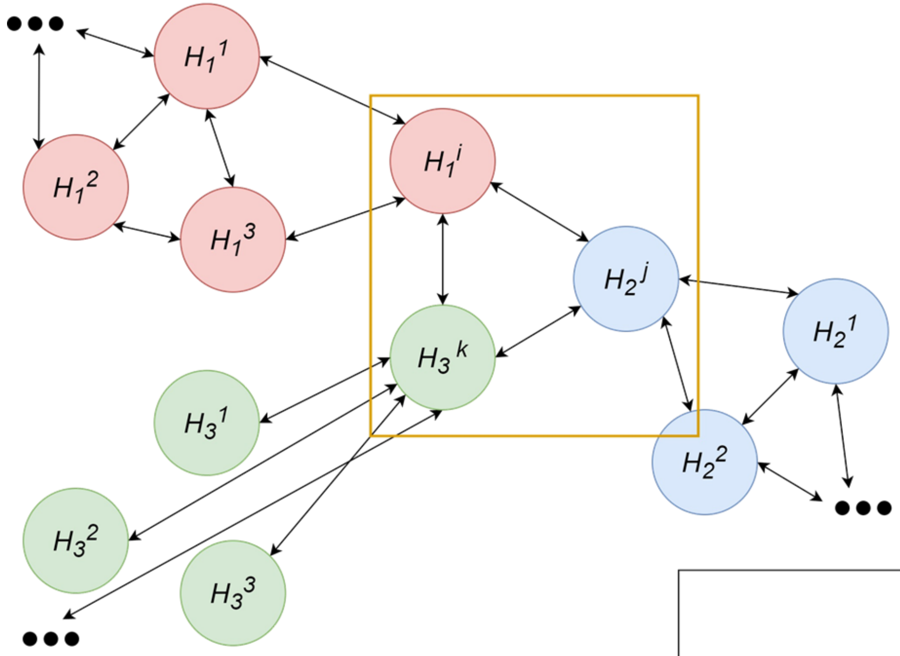
Decrypt ciphertext into message:

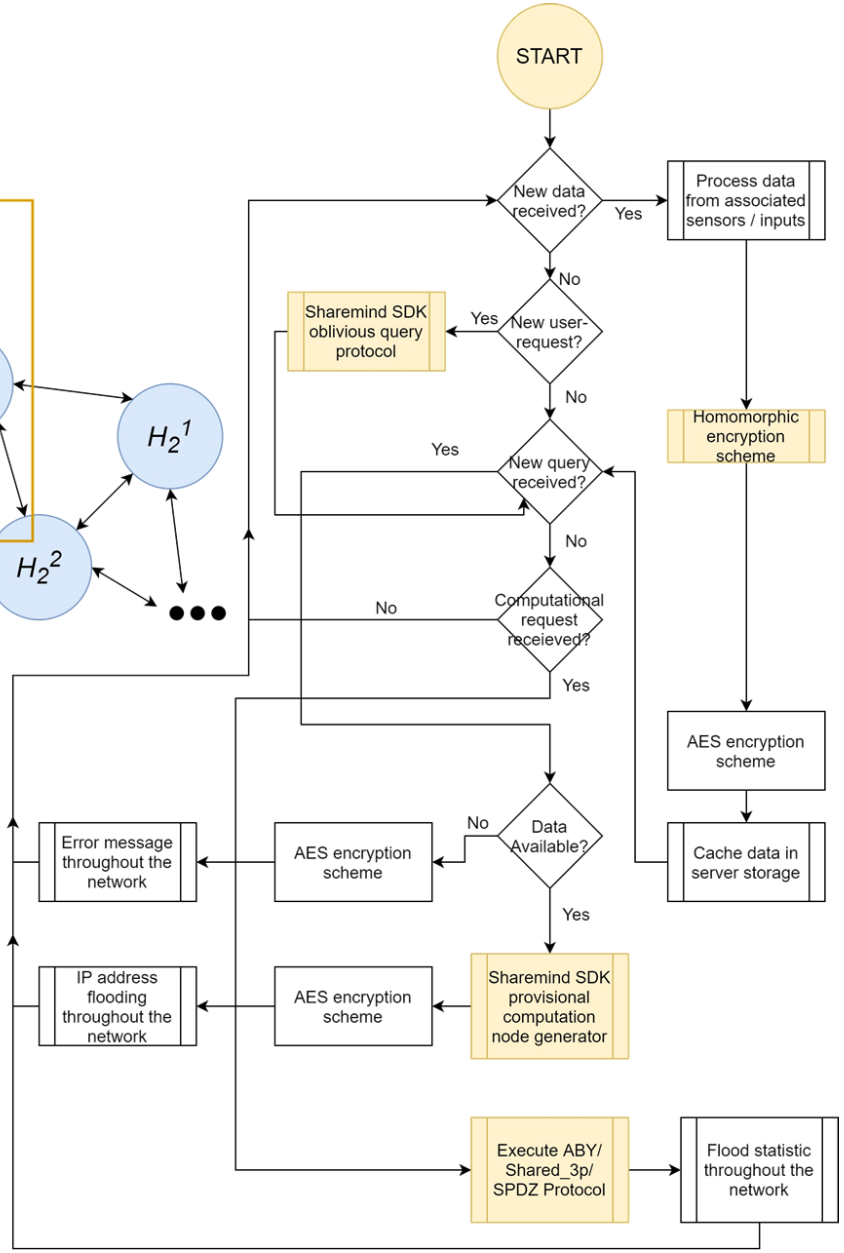
- $m = \frac{c^\lambda \mod n^2 - 1}{n} \cdot \mu \mod n$

Homomorphic property:

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n.$$
$$D(E(m_1, r_1) \cdot g^{m_2} \bmod n^2) = m_1 + m_2 \bmod n.$$
$$D(E(m_1, r_1)^k \bmod n^2) = km_1 \bmod n.$$

Technical Approach (SMPC)





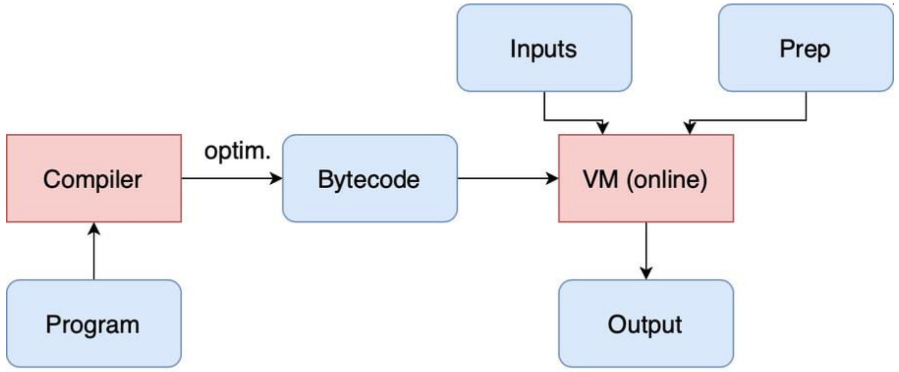
sharemind

CYBERNETICA

Technical Approach (SMPC)

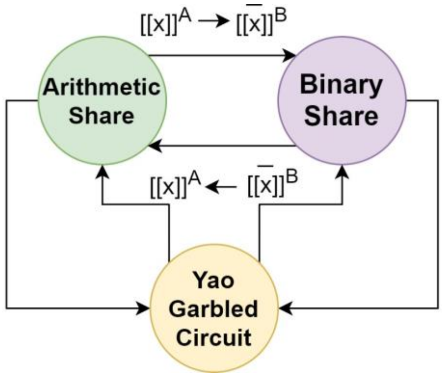
SPDZ:

- An input $a \in \mathbb{F}_{p^k}$ is represented as $<a> = (\delta, (a_1, \ldots, a_n), (\gamma(a)_1, \ldots, \gamma(a)_n))$, a_i is a share of a and $\gamma(a)_i$ is the MAC share authenticating a under a SPDZ global key α (not revealed until end). Player i holds a_i, $\gamma(a)_i$ and δ is public.
- Correct SPDZ execution: $a = \sum_i a_i$, $\alpha(a + \delta) = \sum_i \gamma(a)_i$
- Two phases – offline: generates precomputed values (independent of the function); online: executes designated function using the values.



ABY

- Arithmetic, binary and Yao 3PC
- 3PC with secret sharing for privacy preserving machine learning and database joins (PSI, Union, etc.); secure against semi-honest adversaries;
- Randomly goes back and forth between A, B and Y.



$$[[x]]^A = \sum_i x_i$$

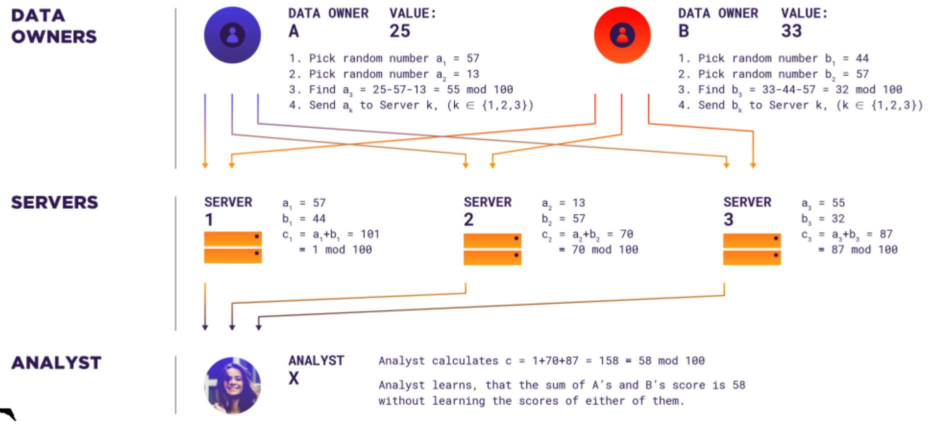
$$[[x]]^B = x_1 \oplus x_2 \oplus x_3$$

$$[[x]]^Y = LSB(x_1 \oplus x_2)$$

$$x = x_1 + x_2 + x_3$$

Shared3p

- Sharemind's proprietary MPC.
- 3PC with secret sharing; secure against semi-honest adversaries
- Uses the additive secret sharing scheme in the ring $Z_2(32)$.



Technical Approach (oblivious functions)

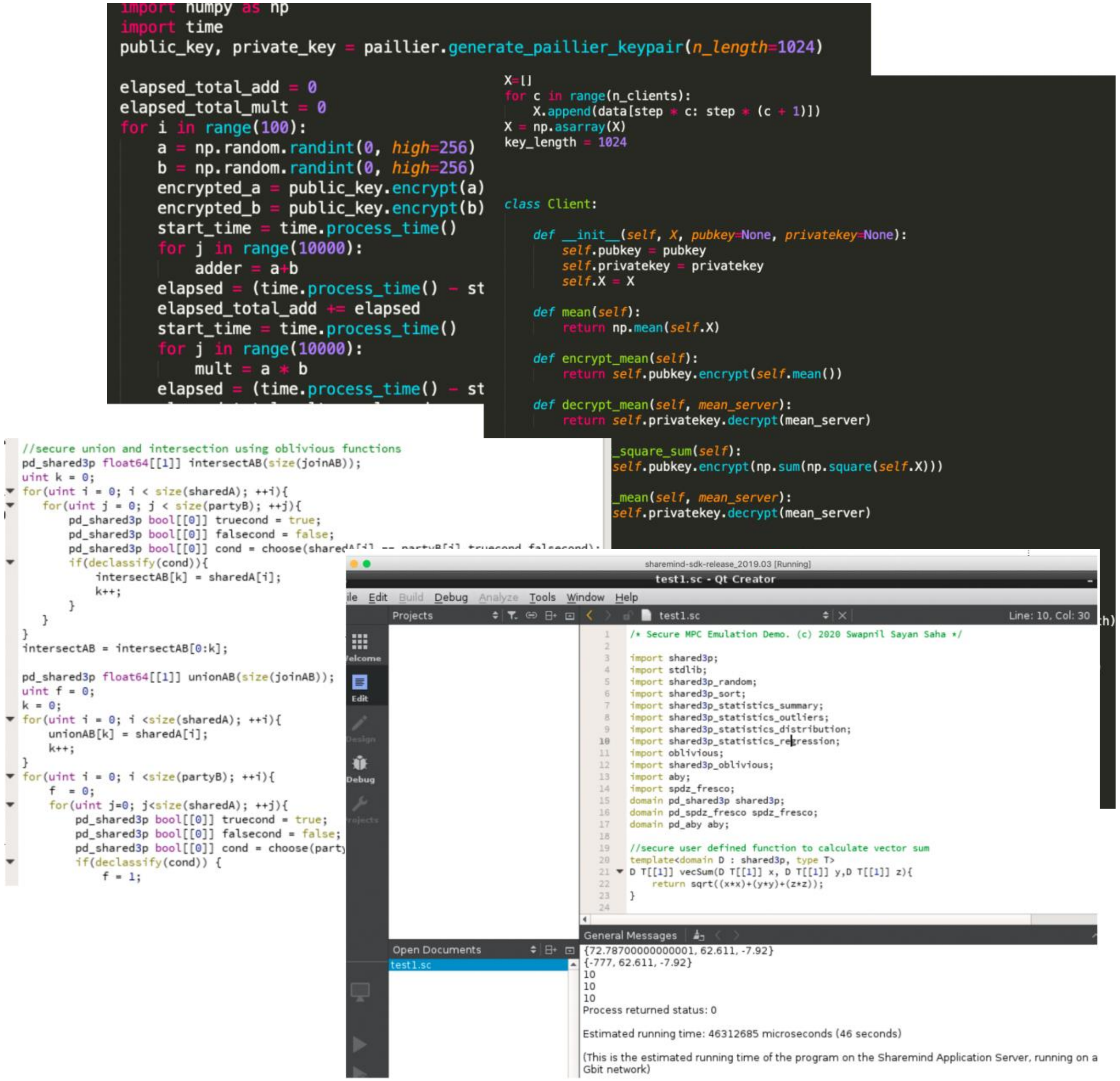
Implemented SA operations (Language: Python)

Mean	Convolution	Linear Regression
Vector Sum		

Implemented SMPC operations (Language: SecreC)

Shuffle	Quicksort	Outer join
Union	Intersection	MAD
Mean	Median	Upper Quantile
Lower Quantile	Minimum	Maximum
StdDev	Variance	Vector Sum (VS)
Outlier_MAD	Outlier_Quantile	Linear Regression
Obv_Insert		

* mean implemented for all 3 protocols

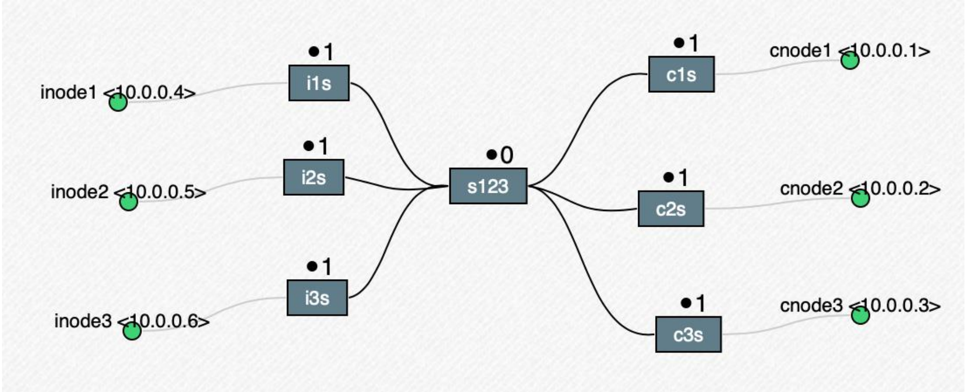


Experimental Setup

Preliminary Benchmarking and Prototyping:

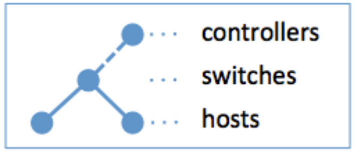
SDN Narmox Spear – Mininet

http://demo.spear.narmox.com/app/?apiurl=demo#!/mininet



Real-time Benchmarking/Implementation

> sudo mn



Success Metrics

- Low latency and execution time of PPI system in resource-constrained setting
- Integrability of code with existing system
- No compromise in security for performance
- 0 need for trusted third party

Implementation / Demo

Real-time secure aggregation demo on Mininet:

https://www.youtube.com/watch?v=DHPKwDjj1qg

Real-time SMPC demo on Mininet:

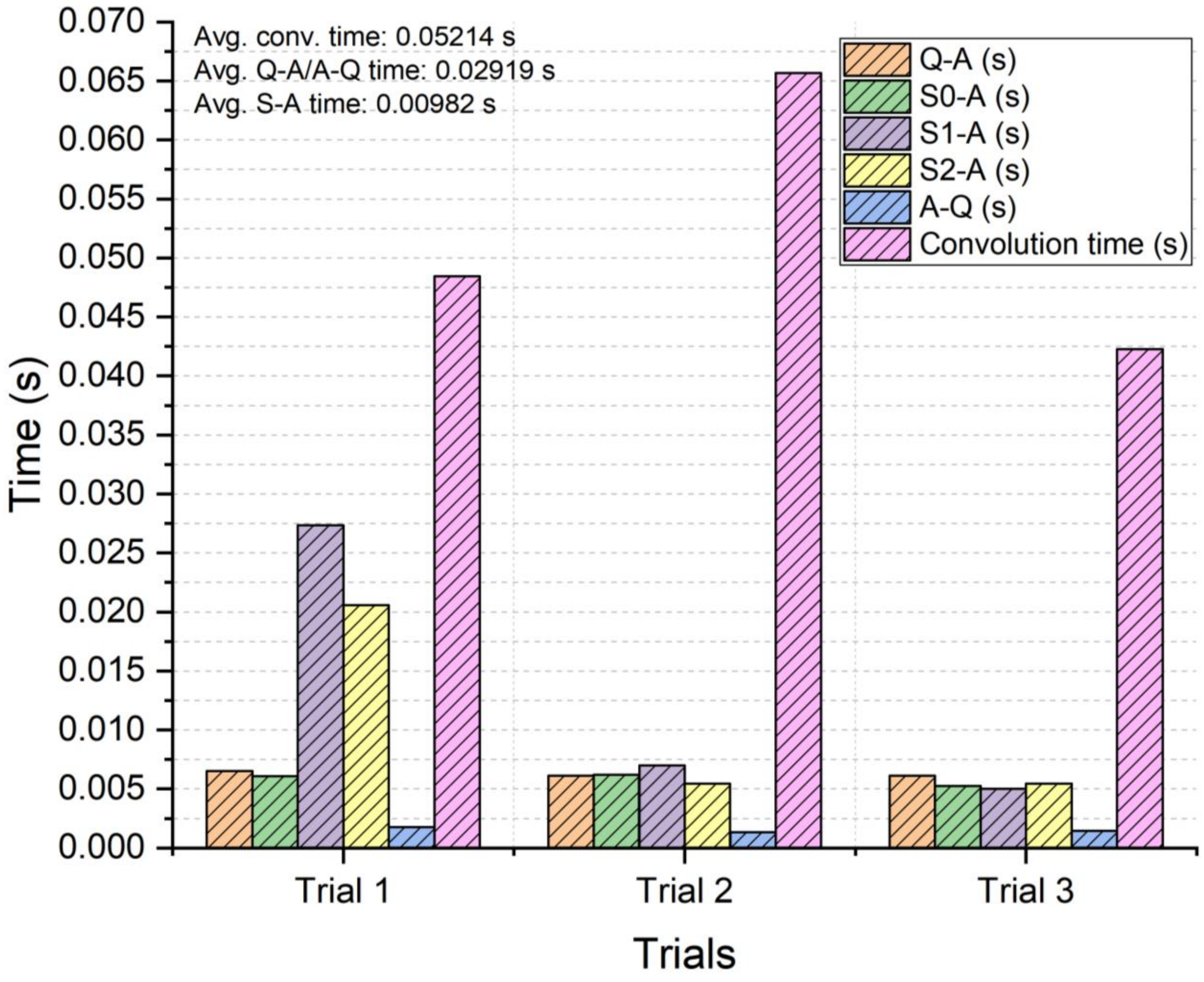
https://www.youtube.com/watch?v=t_0Hudujrkc

Key Findings (SA)

SA standalone benchmark metrics:

Parameter	I7-6700 HQ, 16 GB RAM	Raspberry Pi 4
Memory Usage	~ 8 Mb	~ 36 Mb
CPU Usage	17.6%	99.7%
Key Generation (mS)	259.37	105.66
Encryption (mS)	13.75	40.01
Decryption (mS)	15.63	12.32
Scalar Addition (nS)	46.88	336.33
Scalar Multiplication (nS)	109.38	374.01

SA Real-time benchmark metrics:



Total: 0.09 seconds

Key Findings (SMPC)

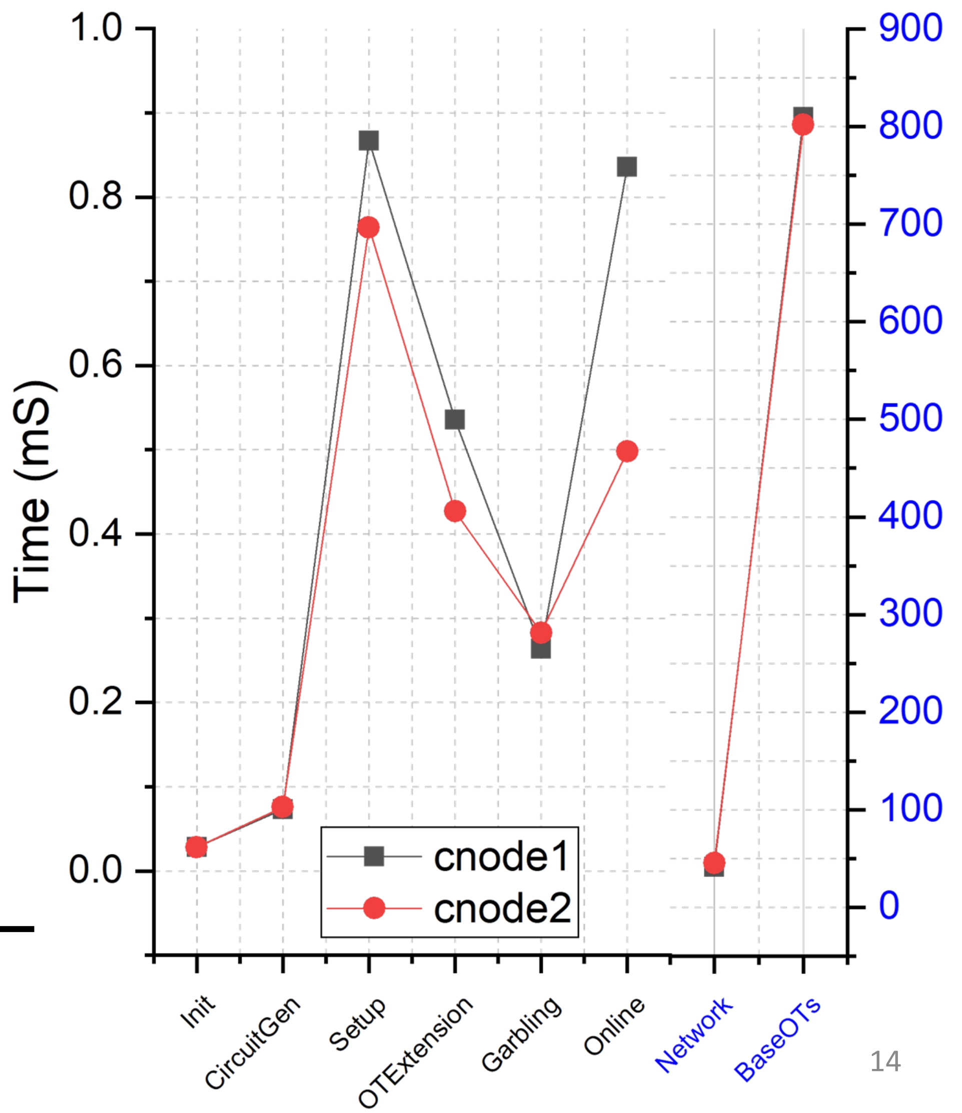
SMPC Standalone benchmark metrics:

Preliminary benchmark results* (-microseconds (CPU usage)):

	ABY	Shared 3p	SPDZ Fresco
Scalar Addition (+ encryption)	11373 (9%)	104 (11%)	18149 (10%)
Scalar Multiplication (+ encryption)	8055 (11%)	397 (14%)	25685 (10%)

* on single core AMD64 architecture, 1 GB RAM (RAM usage: 173 MB)

ABY SMPC realtime benchmark metrics (Millionaire's Problem):



Total: 1.73 seconds
cnode1: 0.854 seconds
cnode2: 0.849 seconds

Key Findings (Overall)

Prior Work and Relative Novelty

Work	Functionality	n-party?	Malicious security?	Practical?
Nikolaenko et al. [60]	ridge regression	no	no	–
Hall et al. [45]	linear regression	yes	no	–
Gascon et al. [38]	linear regression	no	no	–
Cock et al. [21]	linear regression	no	no	–
Giacomelli et al. [39]	ridge regression	no	no	–
Alexandru et al. [5]	quadratic opt.	no	no	–
SecureML [58]	linear, logistic, deep learning	no	no	–
Shokri&Shmatikov [70]	deep learning	not MPC (heuristic)	no	–
Semi-honest MPC [7]	any function	yes	no	–
Malicious MPC [28, 41, 11, 2]	any function	yes	yes	no
Our proposal, Helen: regularized linear models		yes	yes	yes

Our work

- Zheng et al. [1] hypothesized that it is not possible to achieve robust and practical MPC using existing state-of-the-art protocols on the edge.
- Helen requires powerful server-class machines to operate.

Our benchmark shows that it is possible to solve classical MPC (and SA) problems and queries on resource-constrained edge devices using existing state-of-the-art MPC (and SA) protocols.

[1]. Zheng, Wenting, et al. "Helen: Maliciously secure coopetitive learning for linear models." *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

Prior Work and Relative Novelty

- Several MPC and SA architectures have been proposed in literature for MCPS (populated in website)

- All of the proposals, while secure from an information-theoretic view, require some notion of cloud services or external trusted third party, breaking Lindell's [1] recommendations.

"...the key challenge in secure MPC is computational resource, and common errors in secure MPC include assuming semi honest behavior precluding collusions, input dependent flow, deterministic encryption and having a false notion of the absolute actions an adversary may take (rather than mathematical proof)..."

Our proposed architectures do not require any external party in the pipeline, yet achieving collaborative computing goals.

[1]. Lindell, Yehuda, and Benny Pinkas. "Secure Multiparty Computation for Privacy-Preserving Data Mining." *Journal of Privacy and Confidentiality* 1.1 (2009).

Strengths, Weaknesses and Future Directions

Strengths

- PPI system for resource-constrained setting
- Uses state-of-the-art robust security protocols
- Wide variety of possible oblivious functions
- No TTP required
- Easily integrable into existing systems

Weaknesses

- Sharemind framework requires license to operate
- Real-time SMPC and SA not tested on hardware yet
- More tests required for scalability

Member Contributions

Swapnil Sayan Saha

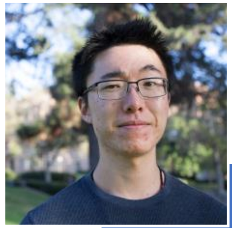
Secure MPC

- Overseeing website/GitHub repo.
- Survey of literature pertinent to the fundamentals and latest advances in SMPC, with applications in MCPS.
- Implementation and preliminary benchmarking of basic SMPC protocols and custom oblivious functions in computer simulation.
- Implementation, dependency installation and benchmarking of SMPC protocols in Raspberry Pi hardware simulation environment.
- Tuning SMPC protocols for resource-constrained environments.

Vivek Jain

Secure Aggregation

- Survey of literature pertinent to application of secure aggregation and SMPC for IoT sensor networks and MCPS
- Implementation and preliminary benchmarking of basic secure aggregation protocols and custom oblivious functions in computer simulation.
- Benchmarking secure aggregation protocols (in real-time) in Raspberry Pi environment and Mininet.
- Handling networking mechanisms in Mininet.
- Tuning secure aggregation protocols for resource-constrained environments.



Brian Wang

Implementation

- Survey of literature pertinent to application of secure multiparty computation in medical cyberphysical systems and clinical decision support systems.
- Formulating SMPC and aggregation architecture for MCPS.
- Implementation of Mininet software simulation.
- Implementation and benchmarking (real-time) of SMPC and secure aggregation protocols in Mininet and Raspberry Pi.
- Handling networking mechanisms in Mininet and Raspberry Pi 4.

THANK YOU

https://github.com/swapnilsayansaha/BVSece209as
https://swapnilsayansaha.github.io/BVSece209as/