Privacy-preserving techniques for Machine Learning

Bogdan Cebere

OpenMined

Agenda

- What is privacy about?
- Privacy enhancing technologies
- Private set intersection
- Homomorphic encryption
- Demo: Evaluation over encrypted images
- Q&A

#whoami

- Software developer @Bitdefender.
- Crypto team member @OpenMined.

What is privacy about?

Reference: https://courses.openmined.org/

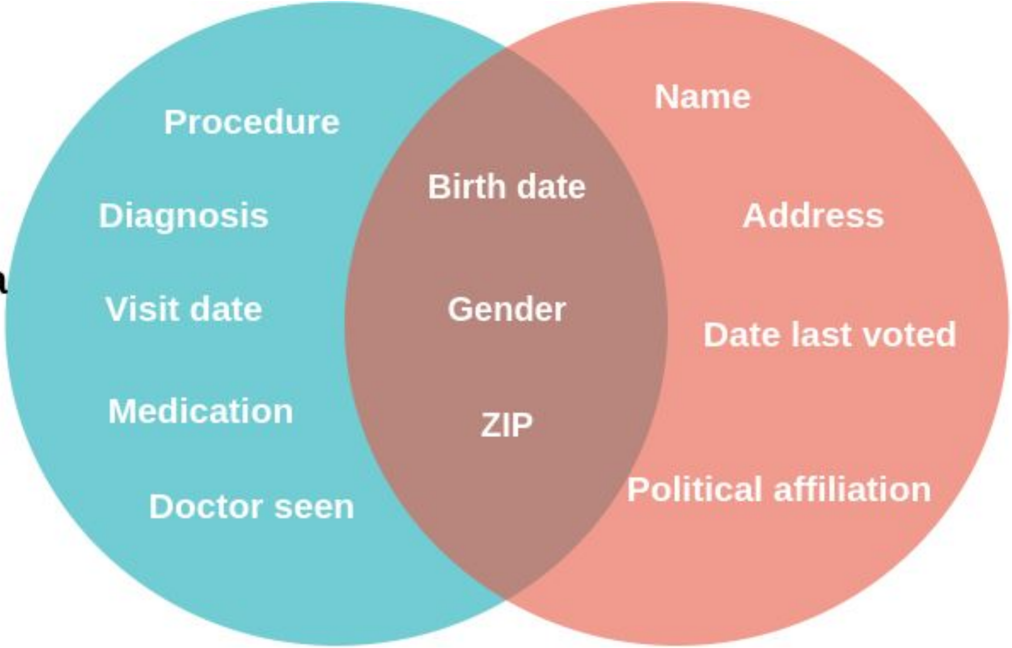
What is privacy about?

- Data Anonymization

Reference: https://courses.openmined.org/

Data anonymization doesn’t help

DATASET 1
Anonymized medical data



DATASET 2
Public voters list

Credits: Aurélien Bellet, Privacy Preserving Machine Learning Course

What is privacy about?

- Data Anonymization
- Hiding behavioral patterns

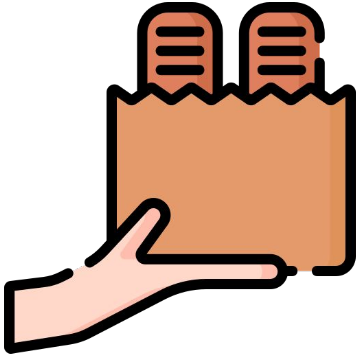
Hiding behavioral patterns doesn't help

The biggest privacy risk is actually in the change of the behavior

Reference: https://courses.openmined.org/

Hiding behavioral patterns doesn't help

The biggest privacy risk is actually in the change of the behavior



Reference: https://courses.openmined.org/

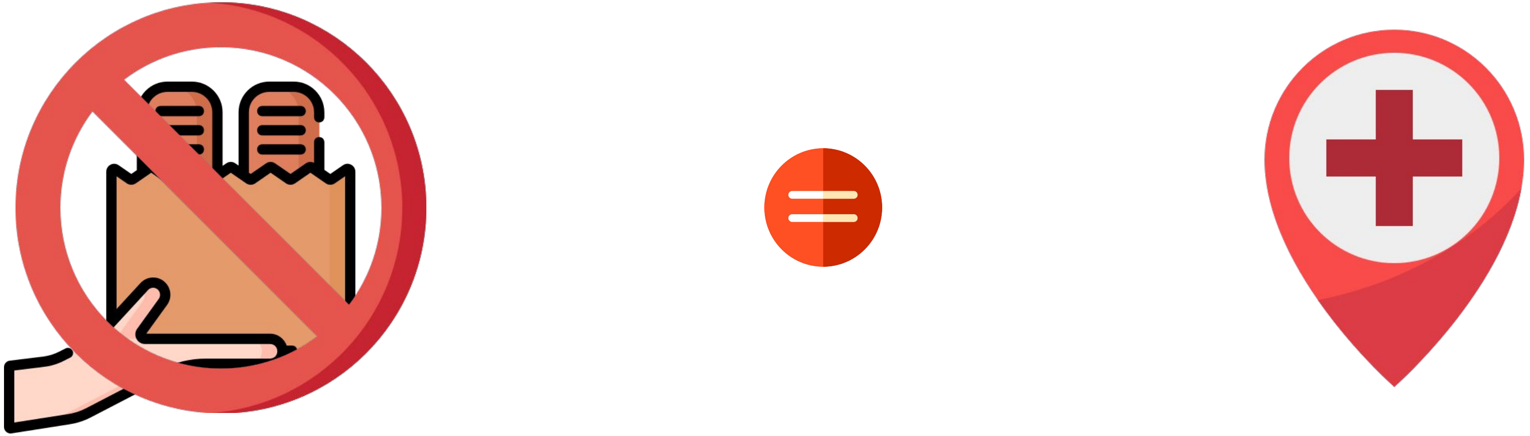
Hiding behavioral patterns doesn’t help

- The biggest privacy risk is actually in the change of the behavior

Reference: https://courses.openmined.org/

Hiding behavioral patterns doesn’t help

The biggest privacy risk is actually in the change of the behavior



Reference: https://courses.openmined.org/

What is privacy about?

- Data Anonymization
- Hiding behavioral patterns
- Individuals

Hiding only individuals doesn’t help

Strava Global Heatmap incident.

Article: https://thehustle.co/strava-heat-map-military-bases

What is privacy about?

- Data Anonymization
- Hiding behavioral patterns
- Identity and individuals
- Information

What is privacy about?

- Data Anonymization
- Hiding behavioral patterns
- Identity and individuals
- Information

What is privacy about?

- Society runs on **information flows**.

Reference: https://courses.openmined.org/

What is privacy about?

- Society runs on **information flows**.
- **Privacy** is not about the information itself but about the way the **information flows**.

Reference: https://courses.openmined.org/

What is privacy about?

- Society runs on **information flows**.
- **Privacy** is not about the information itself, but about the way the **information flows**.
- More specifically, **privacy** is about giving strong guarantees about the **context** in which the **information flows**.

Reference: https://courses.openmined.org/

What is privacy about?

- Society runs on **information flows**.
- **Privacy** is not about the information itself, but about the way the **information flows**.
- More specifically, **privacy** is about giving strong guarantees about the **context** in which the **information flows**.
- **Contextual integrity**(Nissenbaum et al.) asserts that an **ideal information flow** is one that would enable us to collaborate over information while ensuring that information is used only for the context-relative ‘approved’ purposes.

Reference: https://courses.openmined.org/

Privacy bottlenecks

Privacy bottlenecks

1. The Copy Problem

Privacy bottlenecks

1. The Copy Problem
2. The Bundling Problem

Privacy bottlenecks

1. The Copy Problem
2. The Bundling Problem
3. The Recursive Enforcement Problem

Structured transparency

- Many socially valuable activities depend on **sensitive information**: medical research, political coordination, personalized digital services, etc.

A. Trask et al, Beyond Privacy Trade-offs with Structured Transparency. https://arxiv.org/pdf/2012.08347.pdf

Structured transparency

- Many socially valuable activities depend on **sensitive information**: medical research, political coordination, personalized digital services, etc.
- Usually, there is a **privacy trade-off**: we can benefit from data analysis or retain data privacy, but not both.

A. Trask et al, Beyond Privacy Trade-offs with Structured Transparency. https://arxiv.org/pdf/2012.08347.pdf

Structured transparency

- Many socially valuable activities depend on **sensitive information**: medical research, political coordination, personalized digital services, etc.
- Usually, there is a **privacy trade-off**: we can benefit from data analysis or retain data privacy, but not both.
- **Structured Transparency**(Trask et al.) enables productive uses of information without also enabling undesired misuse.

A. Trask et al, Beyond Privacy Trade-offs with Structured Transparency. https://arxiv.org/pdf/2012.08347.pdf

Introducing

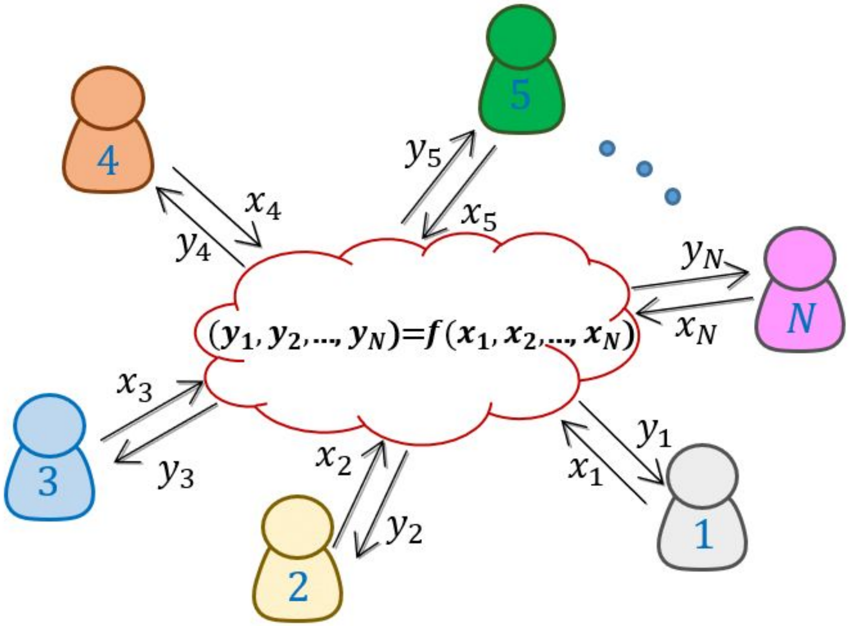
Privacy-Enhancing Technologies

OpenMined

Privacy-Enhancing Technologies

Secure multi-party computation

- **Secure Multiparty Computation** is a technique that allows parties to carry out **distributed computing** tasks safely while keeping their inputs secret.
- **Downside:** Significant communication overhead.
- Real-life use cases: **Boston wage gap**, **Google Advertising conversion.**



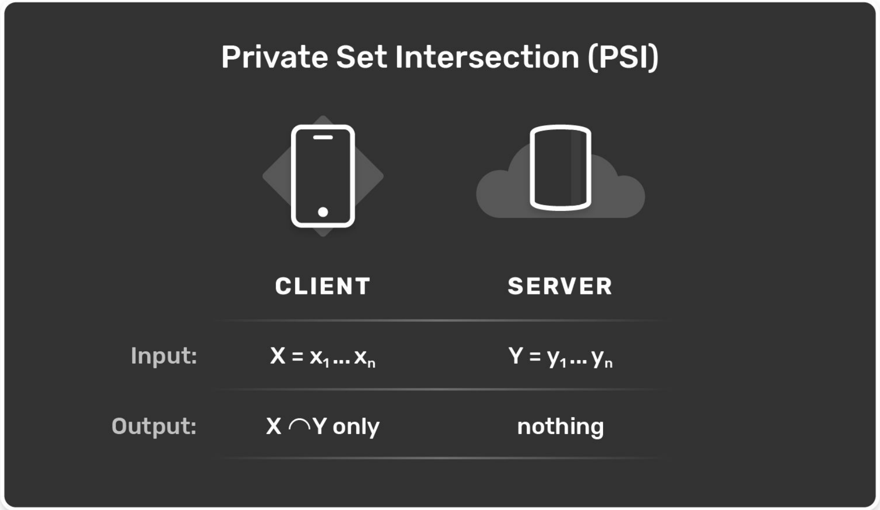
Read more: https://blog.openmined.org/what-is-secure-multi-party-computation/

Privacy-Enhancing Technologies

Private set intersection

- **Private set intersection** is a cryptographic technique that allows two parties to compare data without exposing their raw data to the other party.

- **Real life use cases**: Private Contact Discovery, DNA testing, Contact tracing.

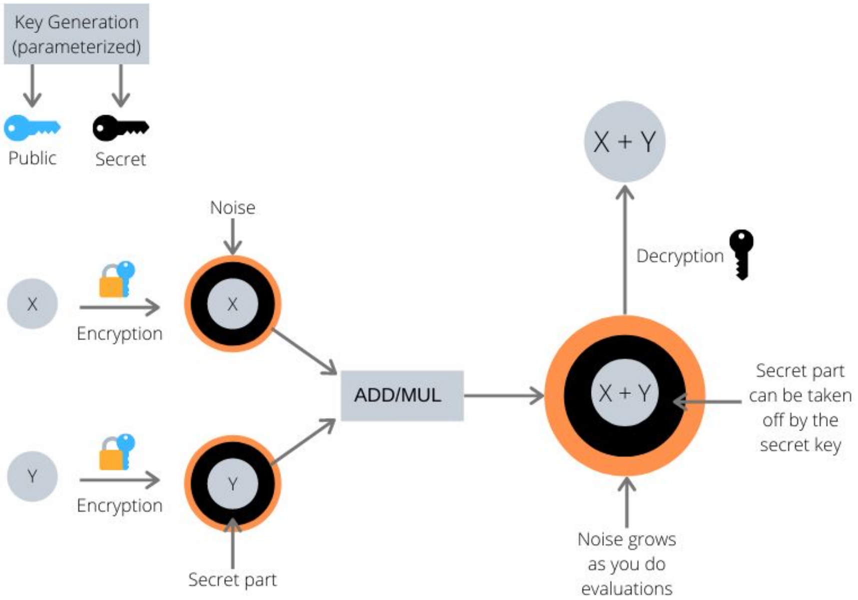


Read more: https://blog.openmined.org/private-set-intersection/

Privacy-Enhancing Technologies

Homomorphic encryption

- **Homomorphic encryption** computes arbitrary mathematical functions on encrypted data sets.
- **Downside:** Computationally expensive.
- **Real-life use cases**: Microsoft Edge password manager, South Korea Personal Credit Rating System

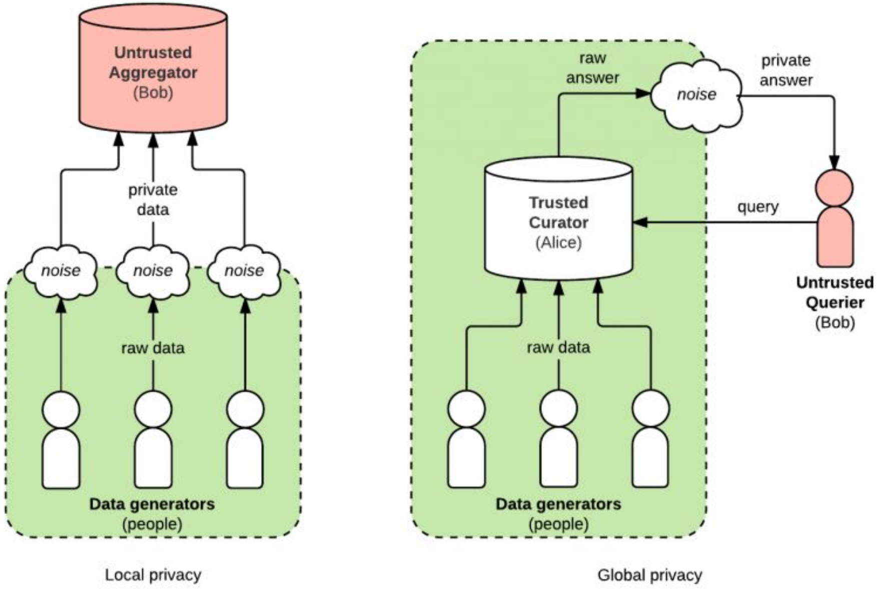


Read more: https://blog.openmined.org/what-is-homomorphic-encryption/

Privacy-Enhancing Technologies

Differential privacy

- **Differential privacy** is a system for publicly sharing information about a dataset by describing the patterns of groups within the dataset while withholding information about individuals in the dataset.
- **Downside:** Lossy
- **Real life use case:** 2020 Census

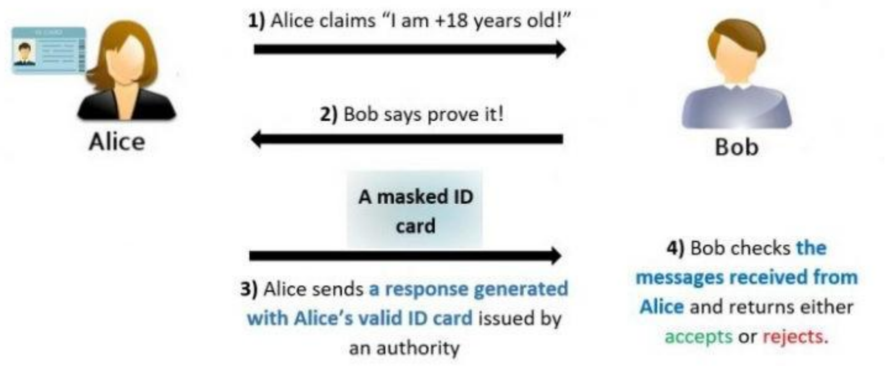


Read more: https://en.wikipedia.org/wiki/Differential_privacy

Privacy-Enhancing Technologies

Zero knowledge proofs

- A **zero-knowledge proof** is a method by which one party (t**he prover**) can prove to another party (**the verifier**) that they know a value x, without conveying any information apart from the fact that they know the value x.
- **Real life use cases:** blockchain validations, authentication, banking loans**.**



Read more: https://en.wikipedia.org/wiki/Zero-knowledge_proof

Wasn't this about machine learning?

Wasn’t this about machine learning?

- Several ML models, GPUs, but only a few datasets.

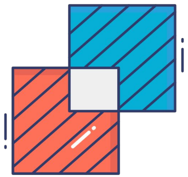
Wasn’t this about machine learning?

- Several ML models, GPUs, but only a few datasets.
- Interesting datasets contain sensitive data or are hard to get.

Wasn’t this about machine learning?

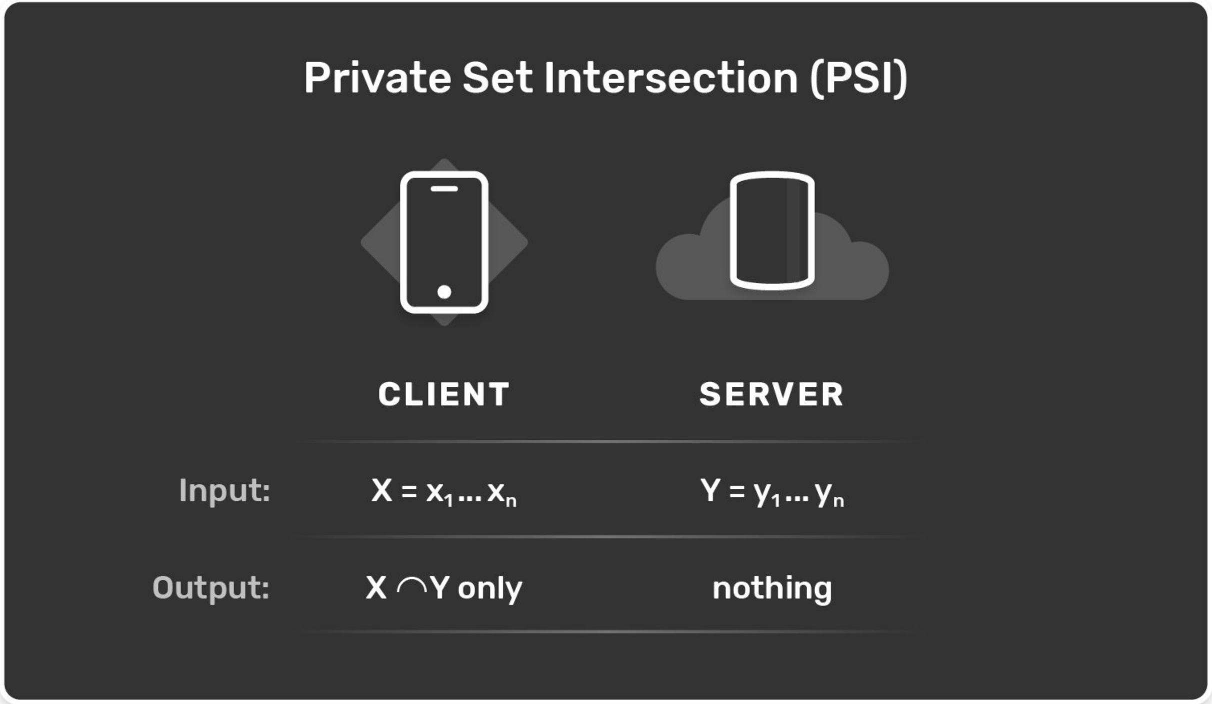
- Several ML models, GPUs, but only a few datasets.
- Interesting datasets contain sensitive data or are hard to get.
- Solving privacy can unlock machine learning applications in critical domains like healthcare.

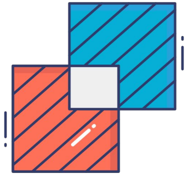
https://courses.openmined.org/



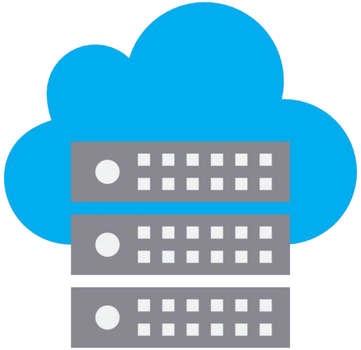
Private set intersection

Private Set Intersection

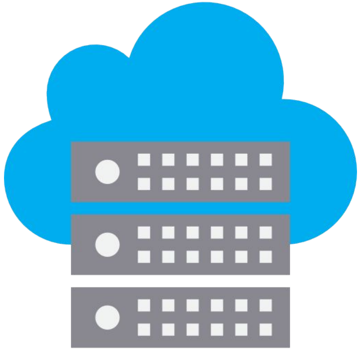


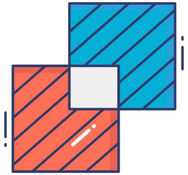


Private Set Intersection Example

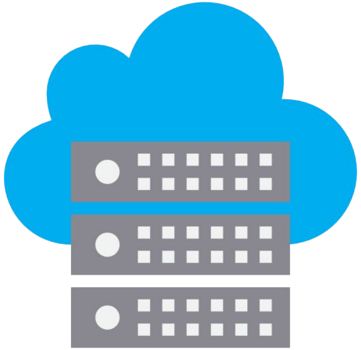


Private Set Intersection Example

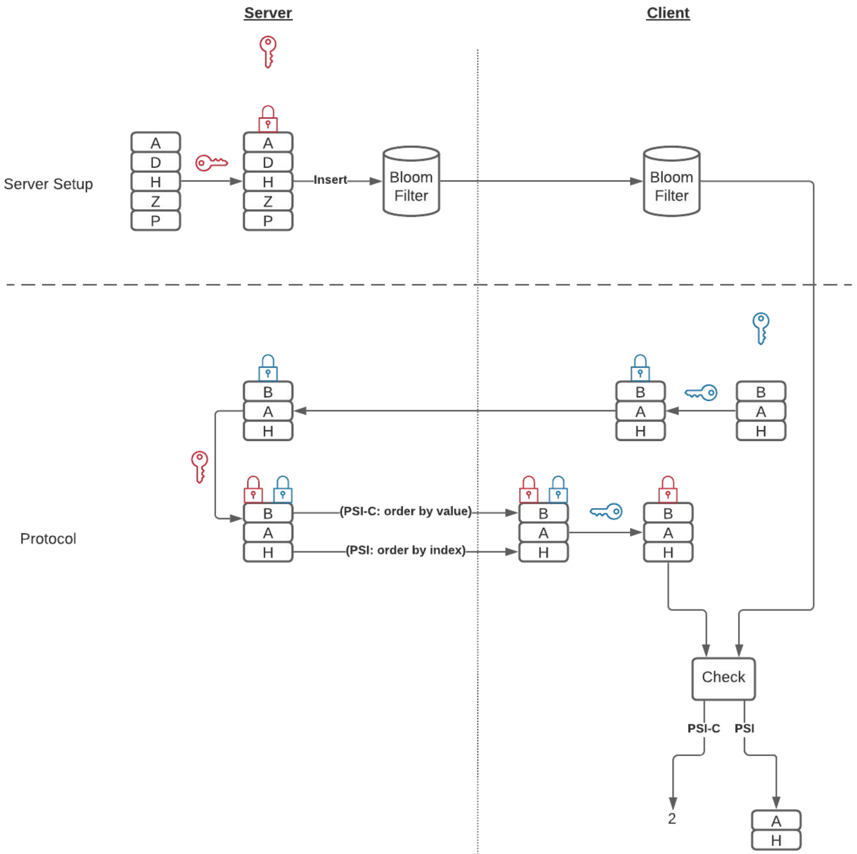




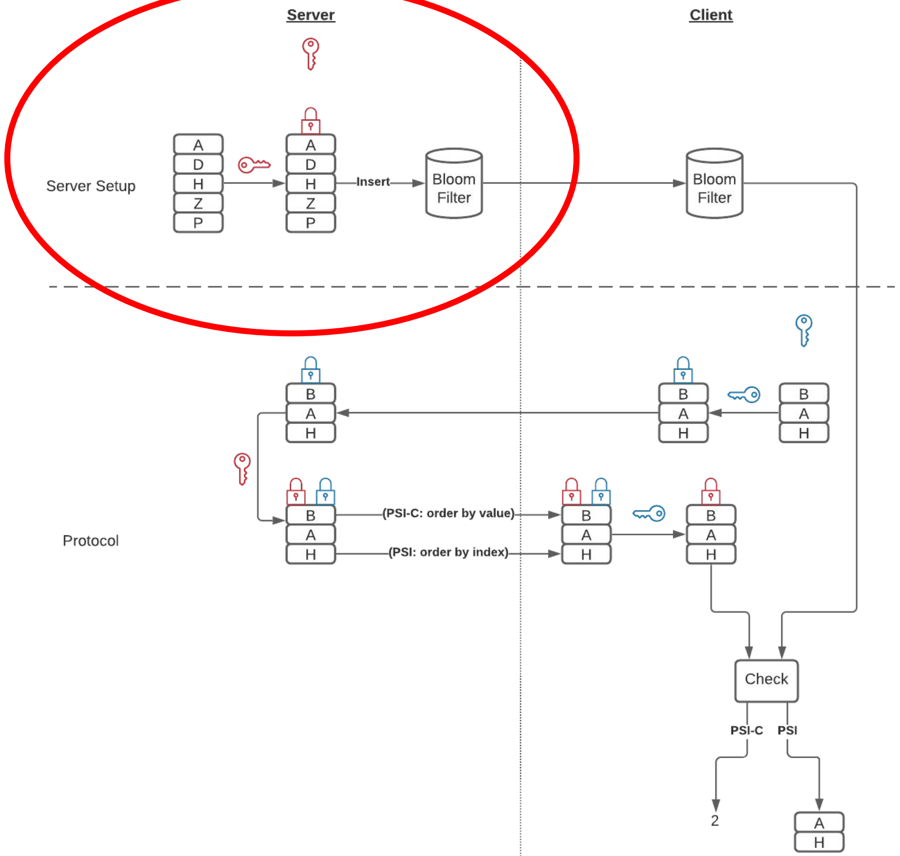
Private Set Intersection Example



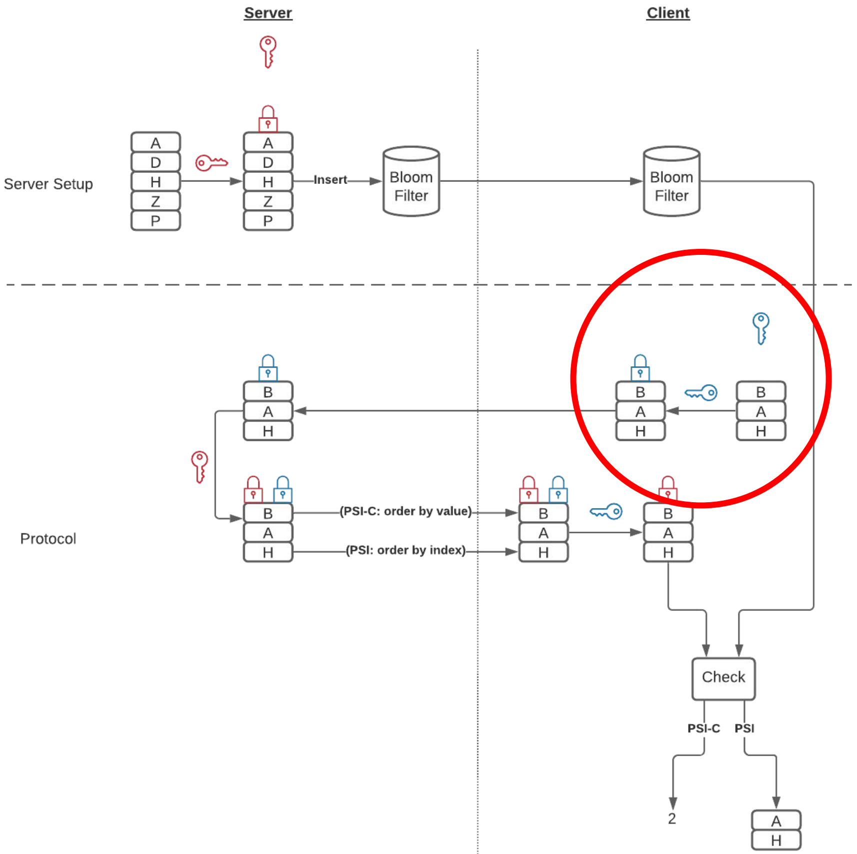
Private Set Intersection Protocol



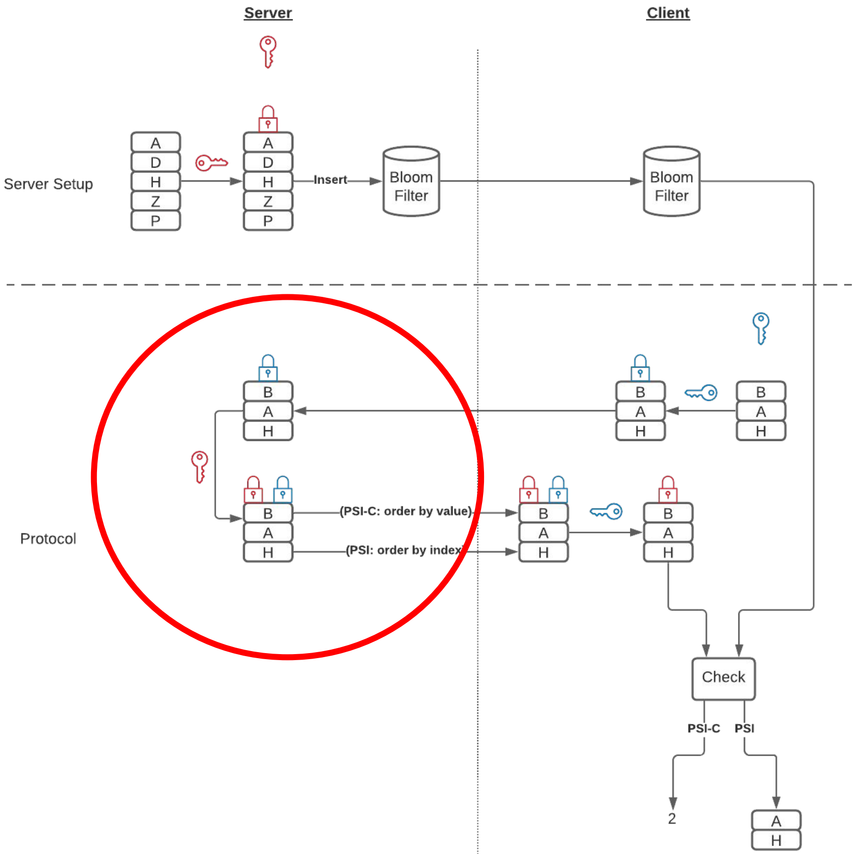
Private Set Intersection Protocol



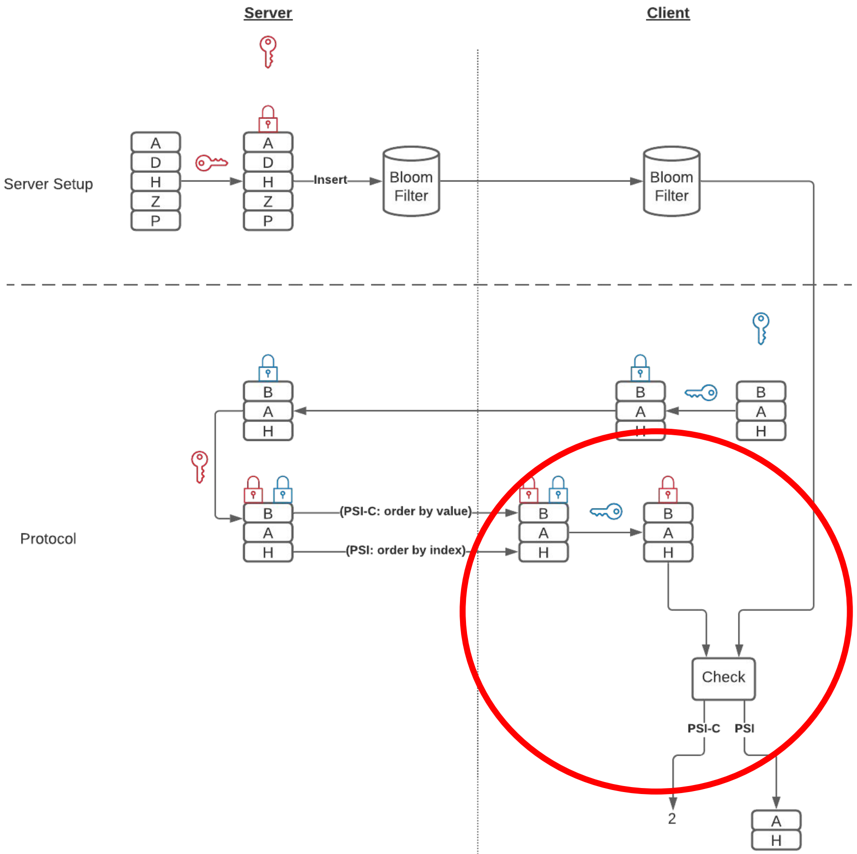
Private Set Intersection Protocol



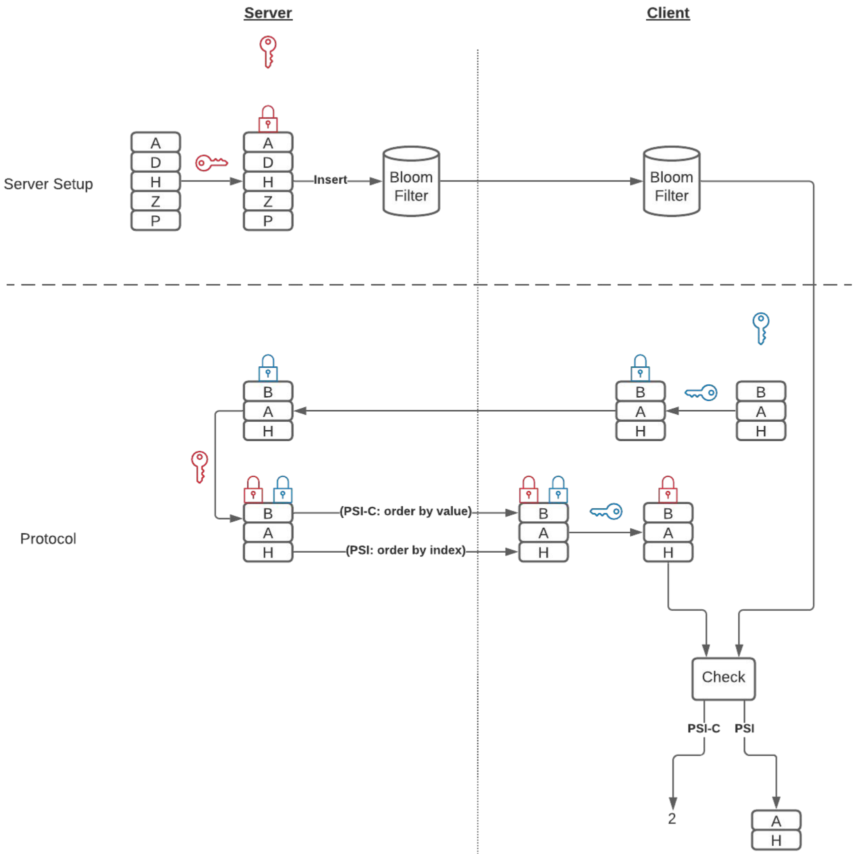
Private Set Intersection Protocol



Private Set Intersection Protocol



Private Set Intersection Protocol



Use case: Contact tracing

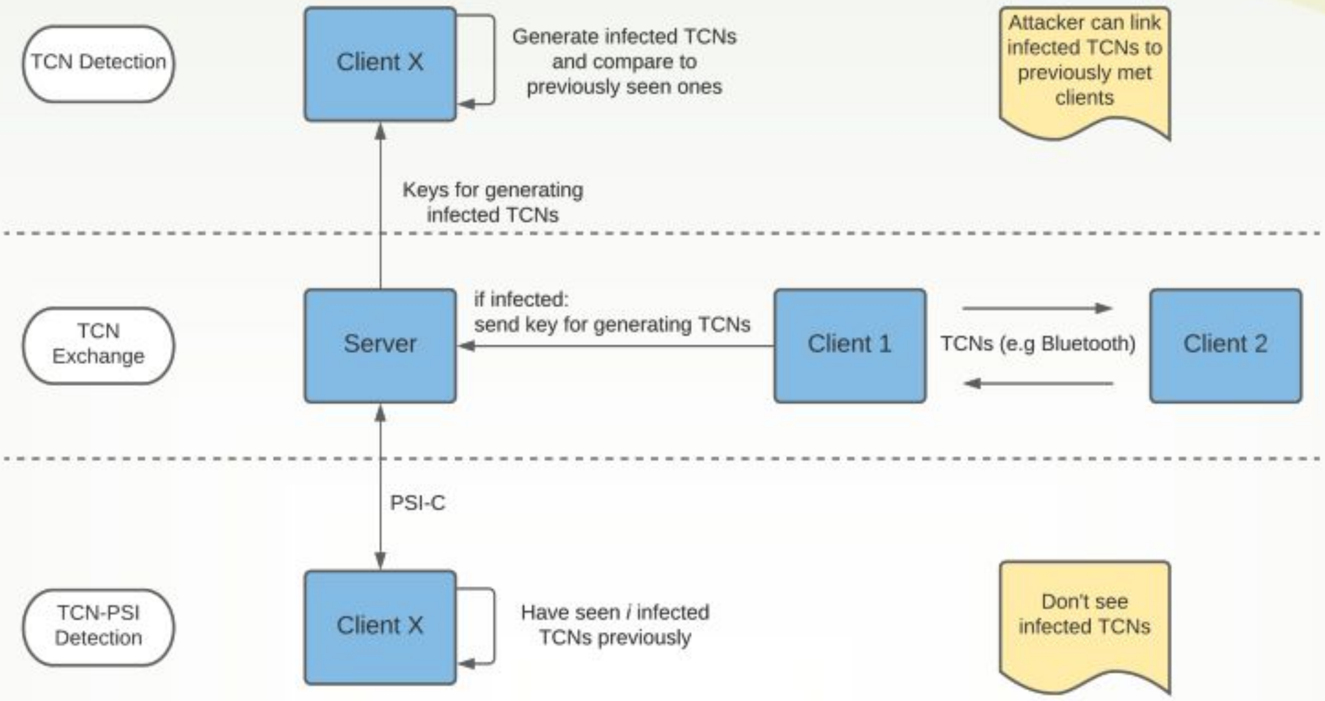
- In the course of the COVID-19 pandemic, several protocols for privacy-preserving **contact tracing** have been proposed, including DP3T, TCN, and the protocol of Apple and Google.

Use case: Contact tracing

- In the course of the COVID-19 pandemic, several protocols for privacy-preserving **contact tracing** have been proposed, including DP3T, TCN, and the protocol of Apple and Google.

- Previous work has shown that these protocols can be susceptible to **linkage attacks**.

Use case: Contact tracing



Source code: https://github.com/bcebere/TCN-PSI

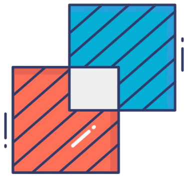
Use case: Private Vertical Federated Machine Learning

PyVertical: https://github.com/OpenMined/PyVertical

Use case: Private Vertical Federated Machine Learning

- In **federated learning**, a machine learning model is to be trained on data held by multiple parties.

PyVertical: https://github.com/OpenMined/PyVertical

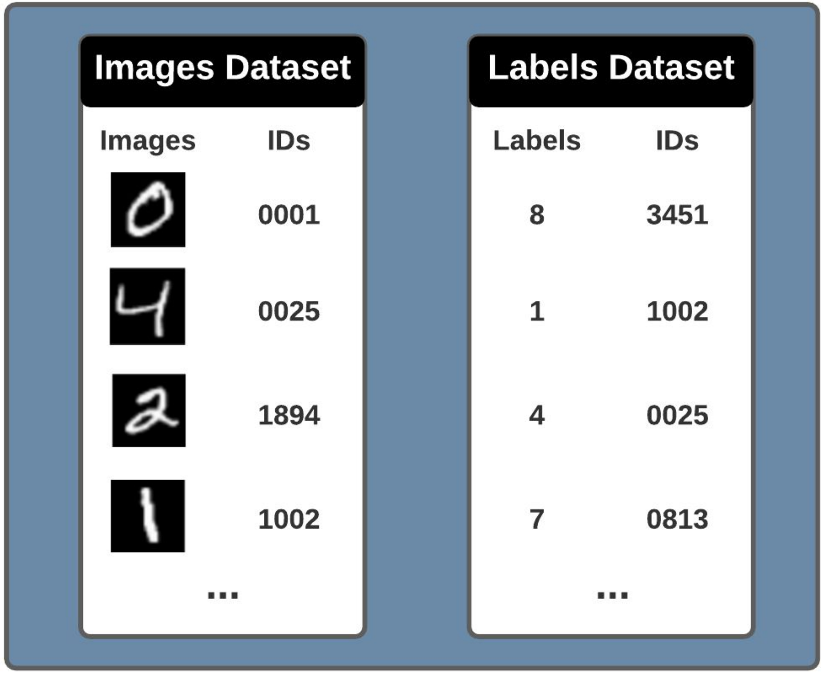


Use case: Private Vertical Federated Machine Learning

- In **federated learning**, a machine learning model is to be trained on data held by multiple parties.
- **Vertically distributed data** are datasets that share partial information about the same entity, differing in the features of each dataset.

PyVertical: https://github.com/OpenMined/PyVertical

Use case: Private Vertical Federated Machine Learning



PyVertical: https://github.com/OpenMined/PyVertical

Use case: Private Vertical Federated Machine Learning

- In **federated learning**, a machine learning model is to be trained on data held by multiple parties.
- **Vertically distributed data** are datasets that share partial information about the same entity, differing in the features of each dataset.
- **Vertical Federated Learning** applies federated learning to vertically distributed data.

Use case: Private Vertical Federated Machine Learning

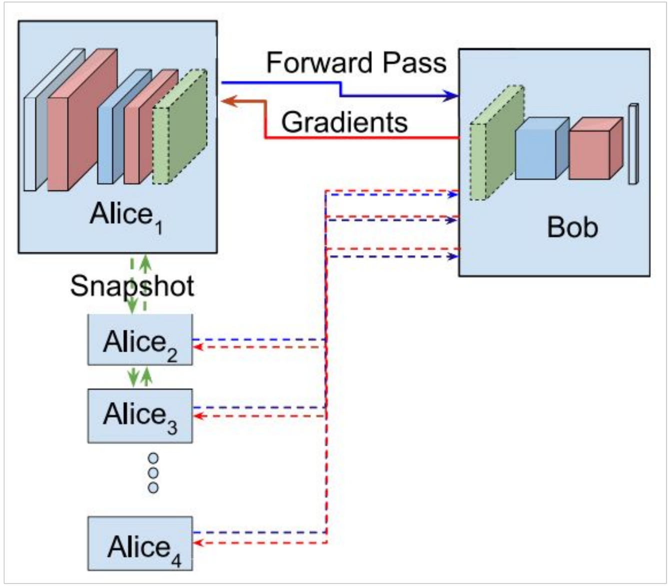
- In **federated learning**, a machine learning model is to be trained on data held by multiple parties.
- **Vertically distributed data** are datasets that share partial information about the same entity, differing in the features of each dataset.
- **Vertical Federated Learning** applies federated learning to vertically distributed data.
- **Example**: Different hospitals may have differing data about the same patient, but cannot simply merge this data across institutions due to privacy reasons.

PyVertical: https://github.com/OpenMined/PyVertical

Use case: Private Vertical Federated Machine Learning

- **Split Neural Network** (SplitNN): the Neural Network (NN) is split among participants, and each model segment acts as a self-contained NN.

- Each model segment trains and forwards its result to the next segment until completion.



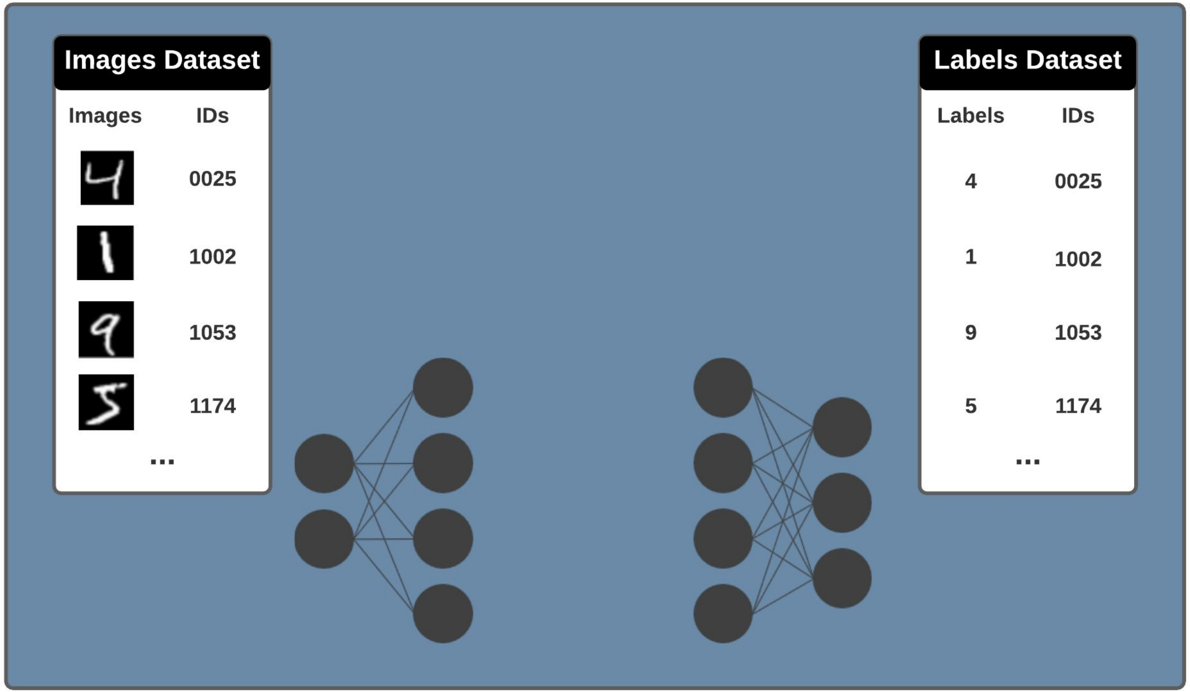
Use case: Private Vertical Federated Machine Learning

PyVertical: https://github.com/OpenMined/PyVertical

Use case: Private Vertical Federated Machine Learning

PyVertical: https://github.com/OpenMined/PyVertical

Use case: Private Vertical Federated Machine Learning



PyVertical: https://github.com/OpenMined/PyVertical

Private Set Intersection

- PSI source code: https://github.com/OpenMined/PSI
- "Asymmetric Private Set Intersection with Applications to Contact Tracing and Private Vertical Federated Machine Learning", *NeurIPS 2020 PPML Workshop*, https://arxiv.org/pdf/2011.09350.pdf

Homomorphic encryption

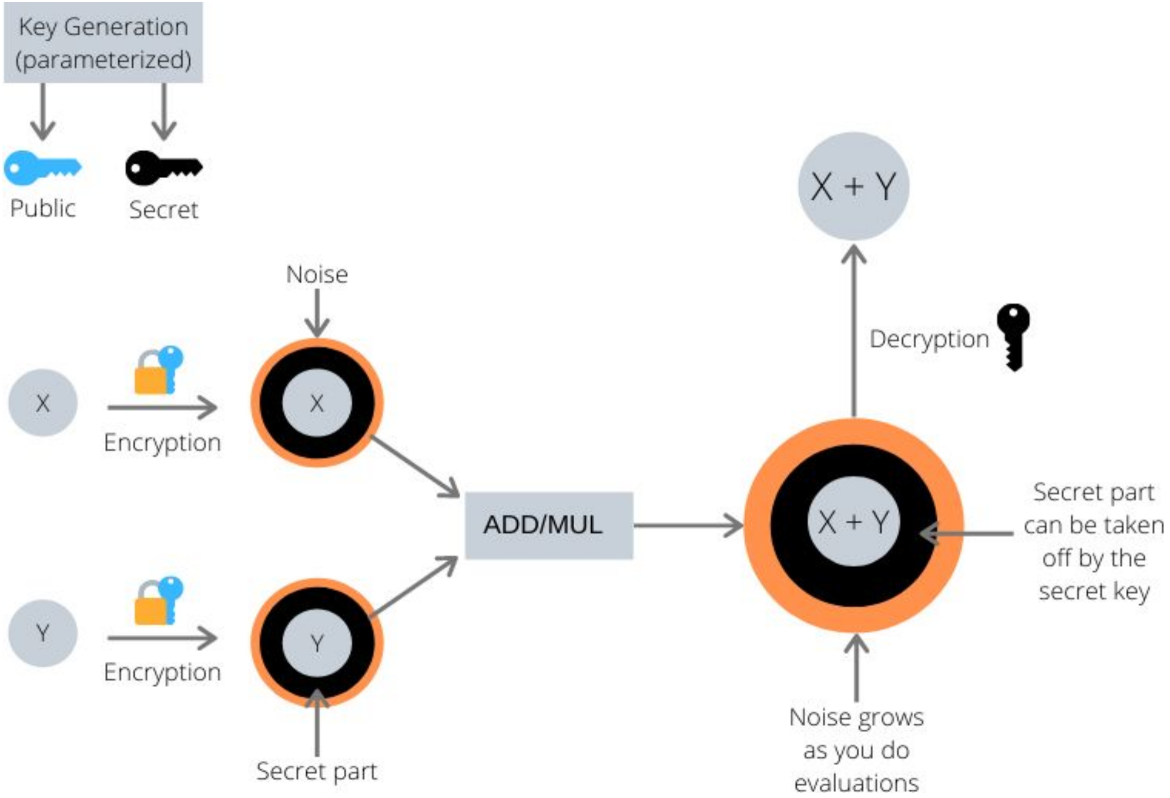
Homomorphic Encryption

Why do we love HE?

- Arbitrary mathematical functions can be computed on encrypted data sets.

Where HE needs improvements?

Homomorphic Encryption



Credits: https://blog.openmined.org/ckks-homomorphic-encryption-pytorch-pysyft-seal/

Homomorphic Encryption

1. Partially Homomorphic Encryption: RSA, ElGamal, Paillier.

Homomorphic Encryption

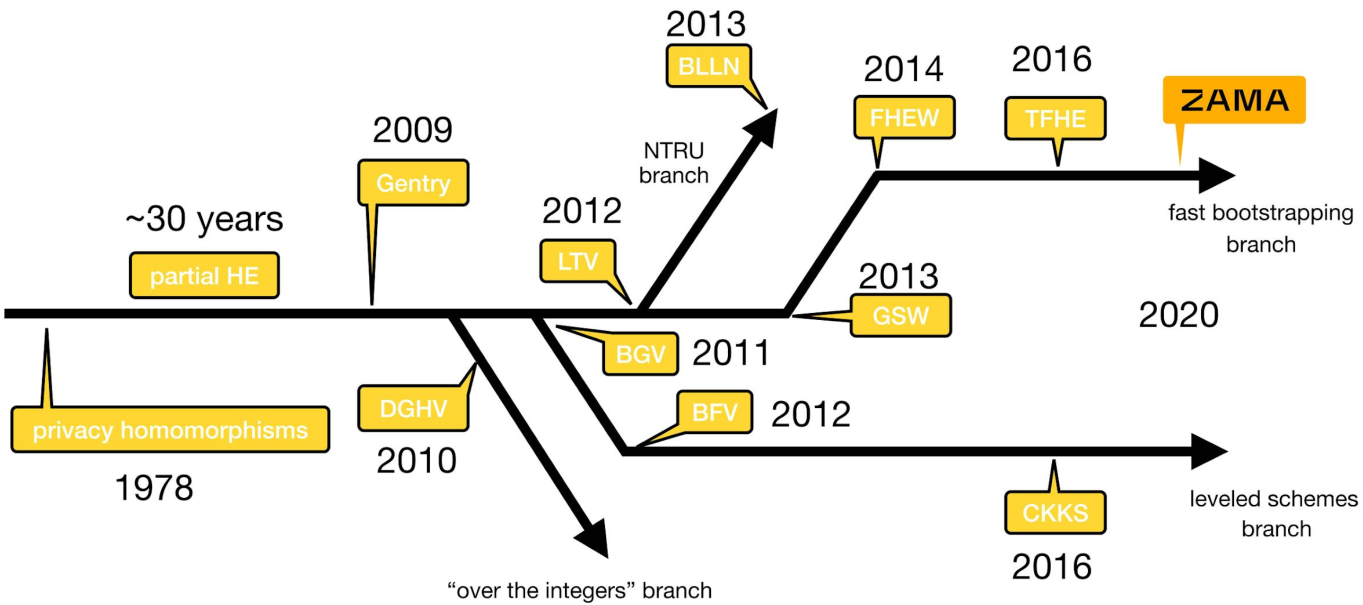
1. Partially Homomorphic Encryption: RSA, ElGamal, Paillier.
2. Leveled Homomorphic Encryption: CKKS scheme.

Homomorphic Encryption

1. Partially Homomorphic Encryption: RSA, ElGamal, Paillier.
2. Leveled Homomorphic Encryption: BFV or CKKS scheme.
3. Fully Homomorphic Encryption: TFHE, CKKS with bootstrapping.

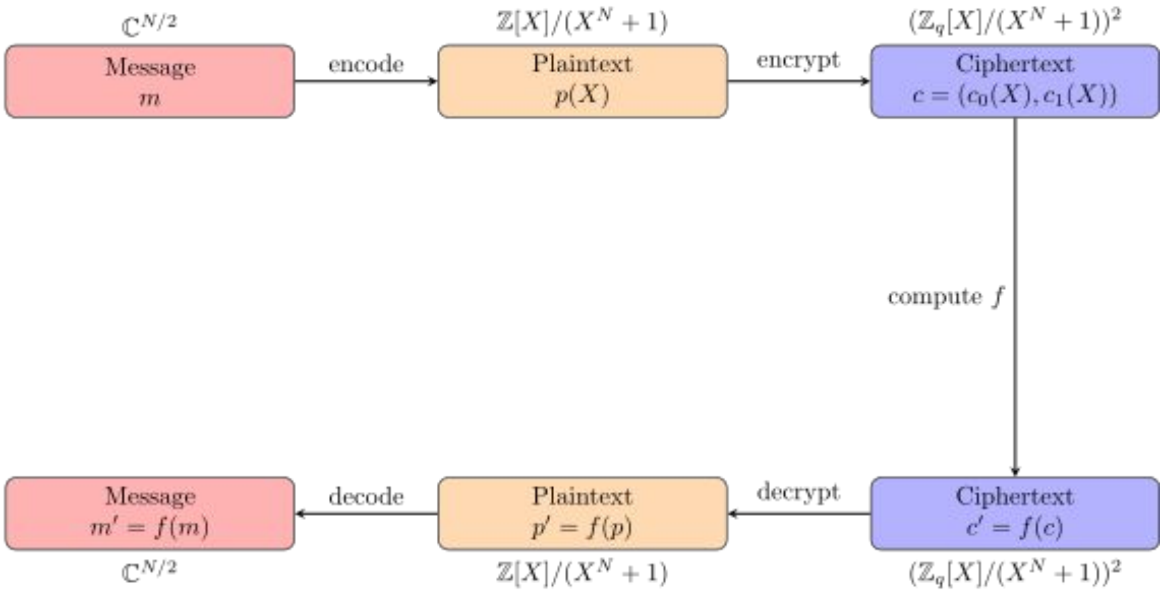
Homomorphic Encryption

A timeline of ~40 years



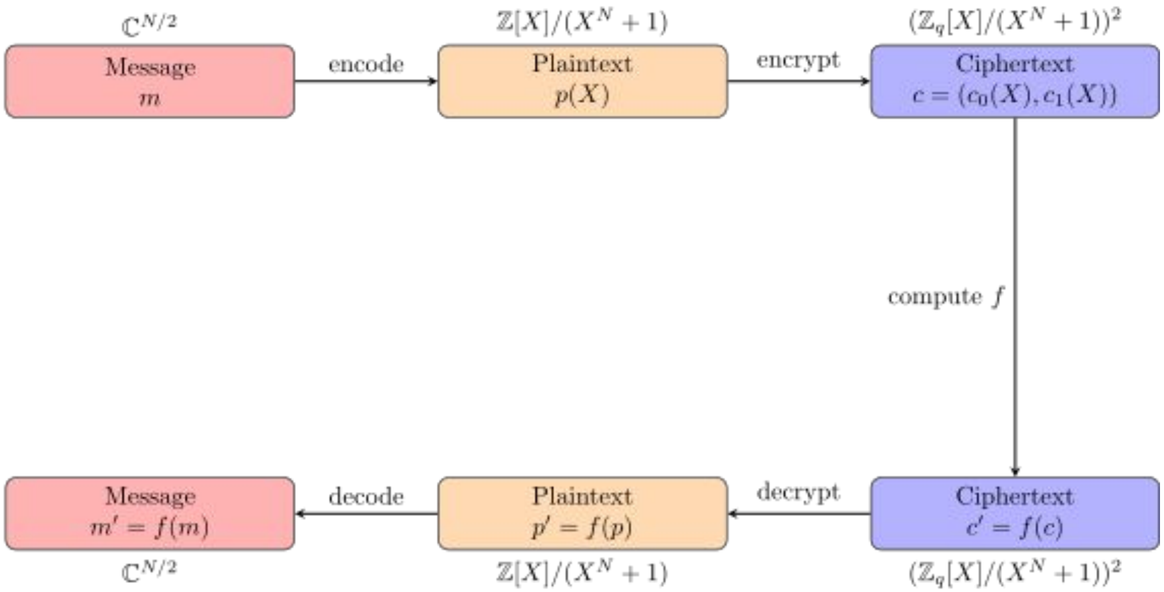
Credits: Pascal Paillier, Introduction to Fully Homomorphic Encryption

Homomorphic Encryption: High-level Overview



Credits: https://blog.openmined.org/ckks-explained-part-1-simple-encoding-and-decoding/

Homomorphic Encryption: High-level Overview



Credits: https://blog.openmined.org/ckks-explained-part-1-simple-encoding-and-decoding/

Homomorphic Encryption: Noise everywhere

But there is a notion of **noise** in ciphertexts

$Enc(x)$ — decryptable

$Enc(x)$ — incorrect decryption

Credits: Pascal Paillier, Introduction to Fully Homomorphic Encryption

Homomorphic Encryption: Noise everywhere

$$Enc(x), Enc(y) \rightarrow Enc(x \oplus y)$$

noises are **added**

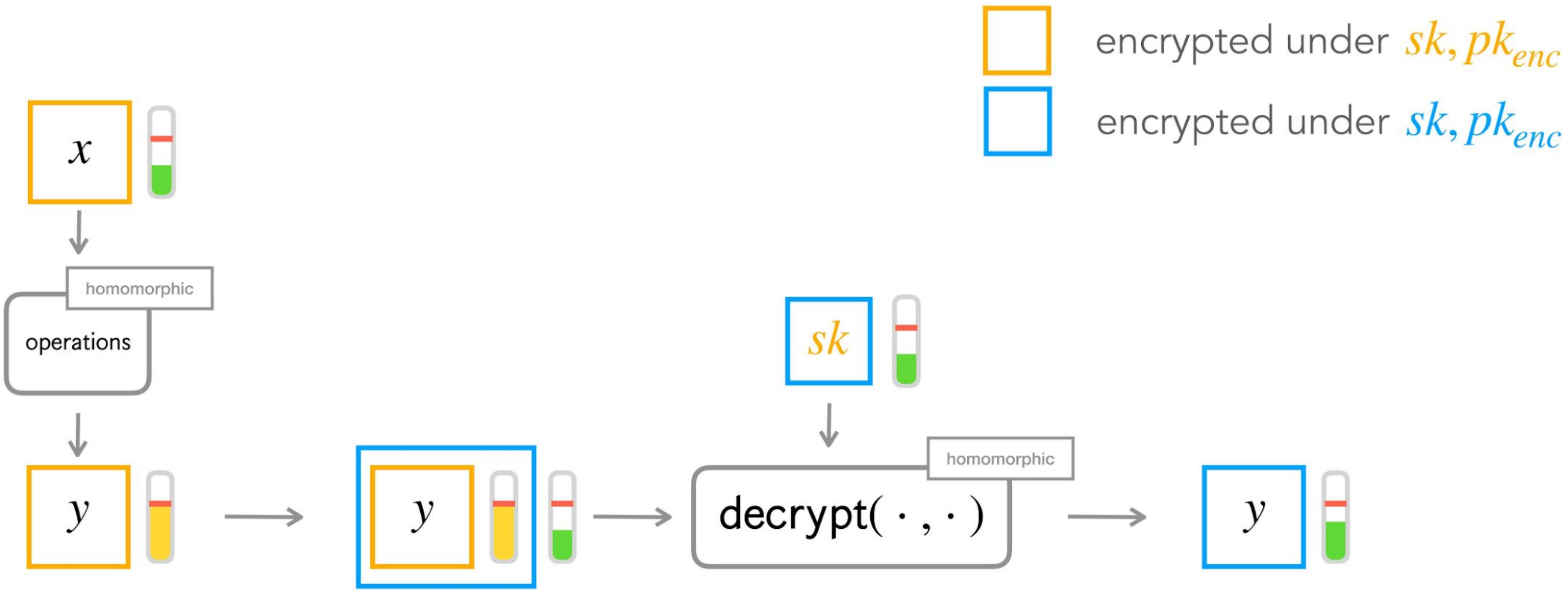
$$Enc(x), Enc(y) \rightarrow Enc(x \otimes y)$$

noises are **multiplied**

(size doubles)

Homomorphic Encryption: Bootstrapping

Homomorphic Encryption: Bootstrapping



Credits: Pascal Paillier, Introduction to Fully Homomorphic Encryption

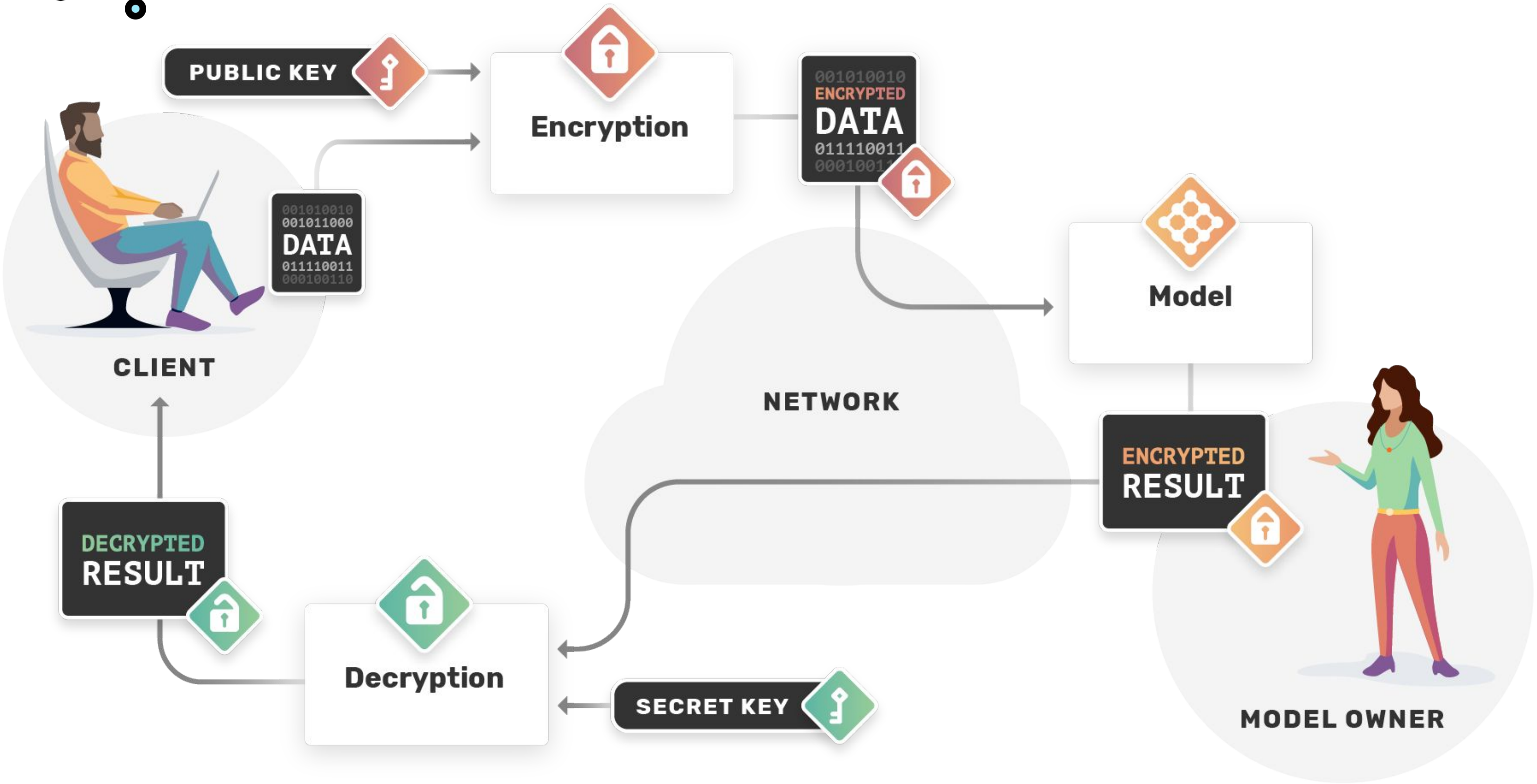
Homomorphic Encryption

Why do we love HE?

- Arbitrary mathematical functions can be computed on encrypted data sets.
- Data is decrypted less often.

Where HE needs improvements?

Homomorphic Encryption



Credits: https://blog.openmined.org/ckks-homomorphic-encryption-pytorch-pysyft-seal/

Homomorphic Encryption

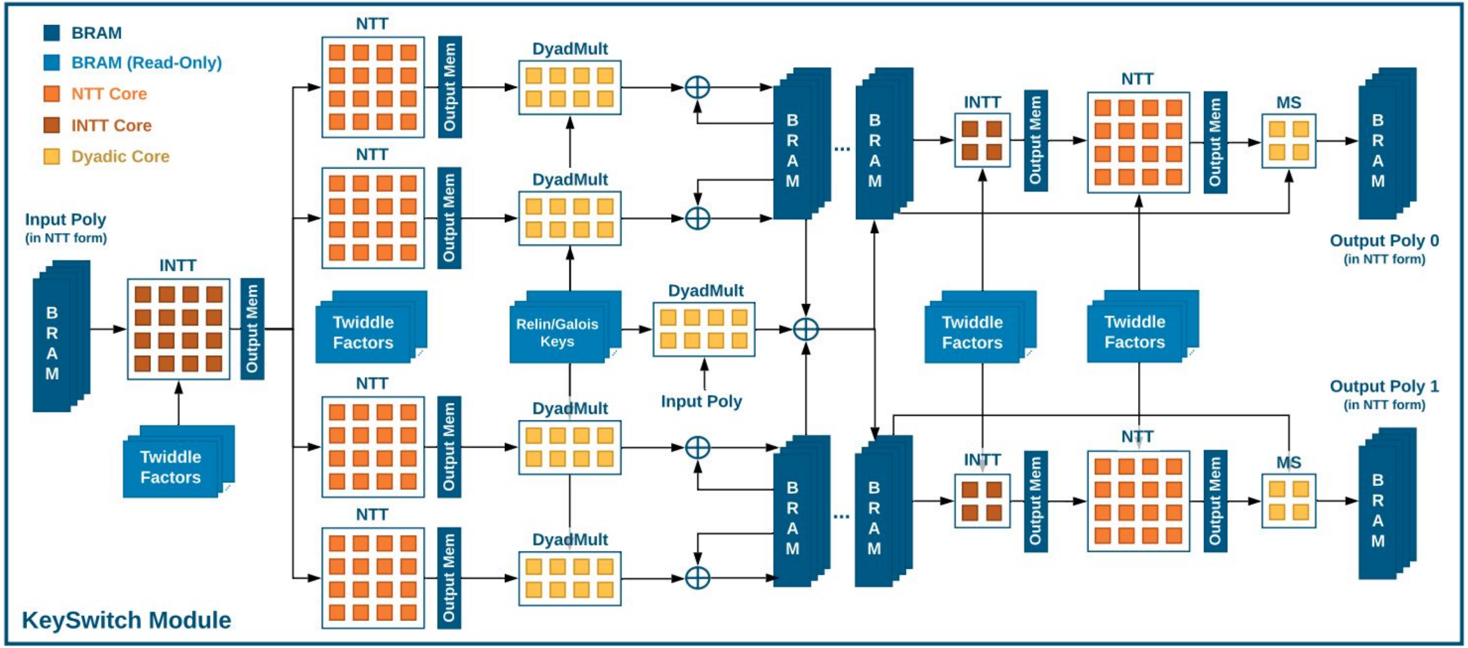
Why do we love HE?

- Arbitrary mathematical functions can be computed on encrypted data sets.
- Data is decrypted less often.
- An area of very active research.

Where HE needs improvements?

Homomorphic Encryption

- **Microsoft HEAX**: a new computing architecture, specifically designed for FHE, using FPGAs.



The "KeySwitch" module within the HEAX architecture

Homomorphic Encryption

Why do we love HE?

- Arbitrary mathematical functions can be computed on encrypted data sets.
- Data is decrypted less often.
- An area of very active research.

Where HE needs improvements?

- Hard to choose the security parameters correctly.

Homomorphic Encryption

- Plaintext data of 8.8 KB, encrypted with the CKKS scheme.

Polynomial modulus	Coefficient modulus sizes	Precision	Ciphertext serialized size	Encryption increase ratio
8192	[40, 21, 21, 21, 21, 21, 21, 40]	2**40	427.16 KB	48.52
8192	[40, 20, 40]	2**40	153.13 KB	17.39
8192	[17, 17]	2**15	38.85 KB	4.41
4096	[40, 20, 40]	2**40	78.96 KB	8.97
4096	[25, 25]	2**20	30.77 KB	3.49
4096	[18, 18]	2**16	23.86 KB	2.71
2048	[16, 16]	2**14	9.25 KB	1.05

Reference: https://github.com/OpenMined/TenSEAL/blob/master/tutorials/Tutorial%203%20-%20Benchmarks.ipynb

Homomorphic Encryption

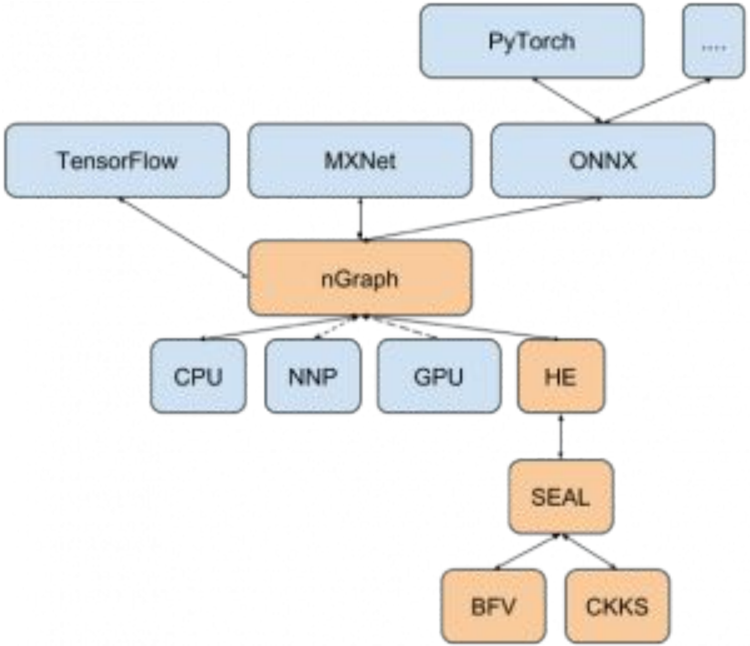
Why do we love HE?

- Arbitrary mathematical functions can be computed on encrypted data sets.
- Data is decrypted less often.
- An area of very active research.

Where HE needs improvements?

- Hard to choose the security parameters correctly.
- Computationally expensive.

Homomorphic Encryption: nGraph



Reference: https://www.intel.com/content/www/us/en/artificial-intelligence/posts/he-transformer-for-ngraph-enabling-deep-learning-on-encrypted-data.html

Homomorphic Encryption: nGraph

Table 8: MobileNetV2 results on localhost and LAN settings using complex packing, batch size 4096, 56 threads, and encryption parameters $N = 2^{12}, L = 3$ at $\lambda = 128$-bit security. Runtimes are averaged across 10 trials. Encrypting the data reduces the top-1 accuracy by an average of 0.0136%, ≈7 images in 50,000.

MobileNetV2 Model	Unencrypted Accuracy (%)		Encrypted Accuracy (%)		Runtime				Communication (MB/image)	Memory (GB)	
					Localhost		LAN				
	Top-1	Top-5	Top-1	Top-5	Amt. (ms)	Total (s)	Amt. (ms)	Total (s)		Client	Server
0.35-96	42.370	67.106	42.356 (−0.014)	67.114 (+0.008)	27	112 ± 5	71	292 ± 5	38.4	8.6	60.3
0.35-128	50.032	74.382	49.982 (−0.050)	74.358 (−0.024)	46	187 ± 4	116	475 ± 10	63.7	12.6	100.4
0.35-160	56.202	79.730	56.184 (−0.018)	79.716 (−0.014)	71	290 ± 7	197	807 ± 19	107.5	17.9	161.0
0.35-192	58.582	81.252	58.586 (+0.004)	81.252 (−0.000)	103	422 ± 23	278	1,141 ± 22	152.2	24.2	239.2
0.35-224	60.384	82.750	60.394 (+0.010)	82.768 (+0.018)	129	529 ± 18	381	1,559 ± 27	206.9	56.9	324.3

Homomorphic Encryption

Why do we love HE?

- Arbitrary mathematical functions can be computed on encrypted data sets.
- Data is decrypted less often.
- An area of very active research.

Where HE needs improvements?

- Hard to choose the security parameters correctly.
- Slow and computationally expensive.
- Difficult to prototype new ideas.

Introducing TenSEAL

https://github.com/OpenMined/TenSEAL

Library features

- Built on top of Microsoft SEAL.

Library features

- Built on top of Microsoft SEAL.
- Several types of encrypted tensors built over CKKS and BFV schemes.

Library features

- Built on top of Microsoft SEAL.
- Several types of encrypted tensors built over CKKS and BFV schemes.
- Libraries for C++ and Python, deployed for Windows, Linux and MacOS.

Library features

- Built on top of Microsoft SEAL.
- Several types of encrypted tensors built over CKKS and BFV schemes.
- Libraries for C++ and Python, deployed for Windows, Linux and MacOS.
- Serialization done with Protobuffers.

Library features

- Built on top of Microsoft SEAL.
- Several types of encrypted tensors built over CKKS and BFV schemes.
- Libraries for C++ and Python, deployed for Windows, Linux and MacOS.
- Serialization done with Protobuffers.
- Bonus: Python bindings for the SEAL API.

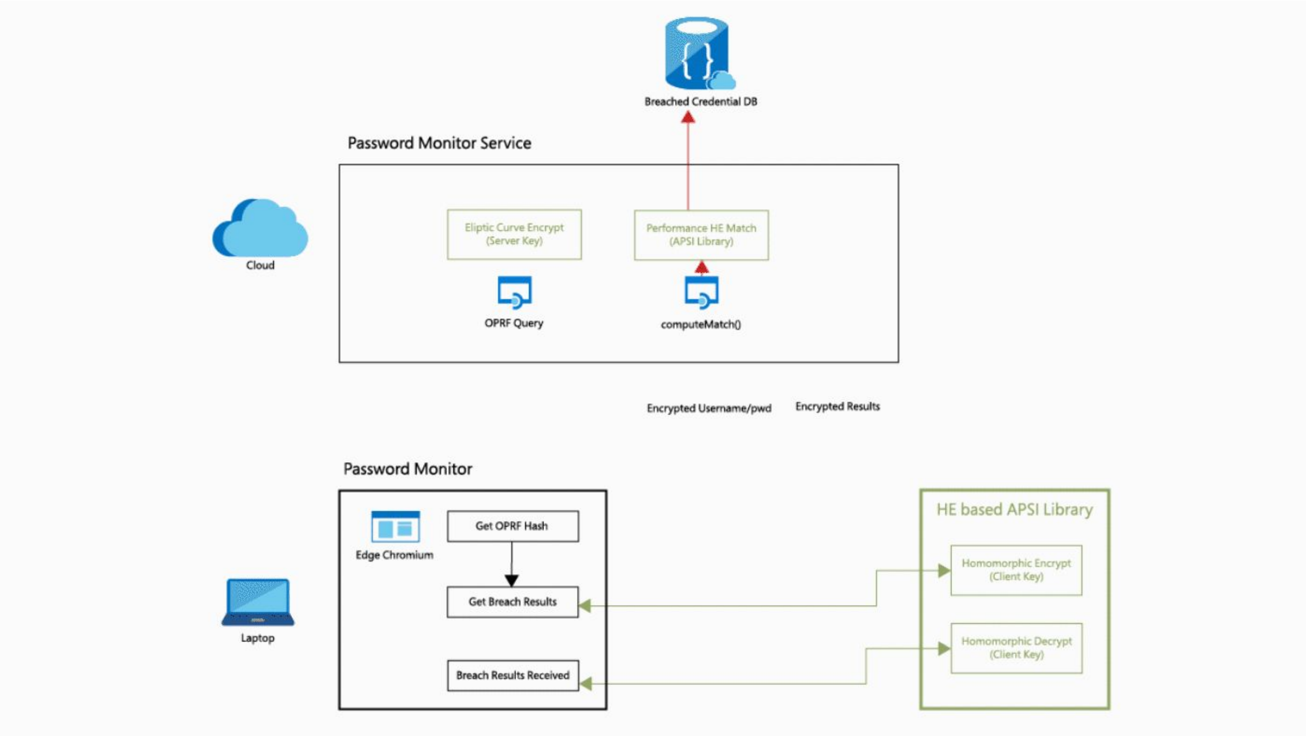
https://github.com/OpenMined/TenSEAL

Demo

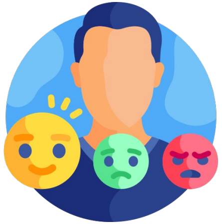
OpenMined

Homomorphic encryption in real life

Password Monitor: Safeguarding passwords in Microsoft Edge



Reference: https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/



South Korea Personal Credit Rating System

Reference: https://www.fintechdirect.net/2020/07/14/south-korea-beefs-up-personal-credit-rating-system/

Open-source is mandatory for privacy technologies

- Open-source offers transparency to your methods.

Open-source is mandatory for privacy technologies

- Open-source offers transparency to your methods.
- You cannot build trust for privacy with black boxes.

Open-source is mandatory for privacy technologies

- Open-source offers transparency to your methods.
- You cannot build trust for privacy with black boxes.
- With trust and structured transparency, you can unlock fantastic machine learning applications.

Time for Q&A