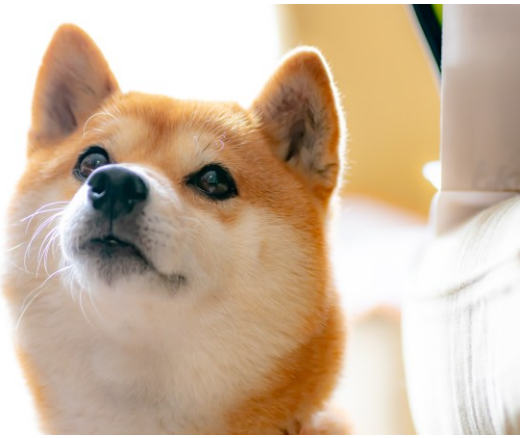
Privacy-Enhancing Technologies (PETs)

Paolo Falcarin

Ca' Foscari University of Venice

Department of Enviromental Sciences, Informatics and Statistics

paolo.falcarin@unive.it



CM0626 – Software Security

15 April 2025

Privacy-Enhancing Technologies (PETs)

Ca' Foscari University of Venice

- PETs (Privacy-Enhancing Technologies) minimize and protect personal data
- PETs are about achieving a technically enforced privacy outcome
- They allow for analysis of large data sets in a way where no one person's information is ever disclosed

Privacy Enhancing Technologies

Ca' Foscari University of Venice

- Secure Multiparty Computation (MPC)
- Fully Homomorphic Encryption (FHE)
- Functional Encryption (FE)
- Differential Privacy (DP)
- Federated Learning (FL)

They use formal mathematical reasoning to obtain provable privacy and security guarantees

A recurring problem: server compromise

Ca' Foscari University of Venice

Twitter 2018:
330,000,000

Capital One 2019:
100,000,000

Facebook 2021:
533,000,000

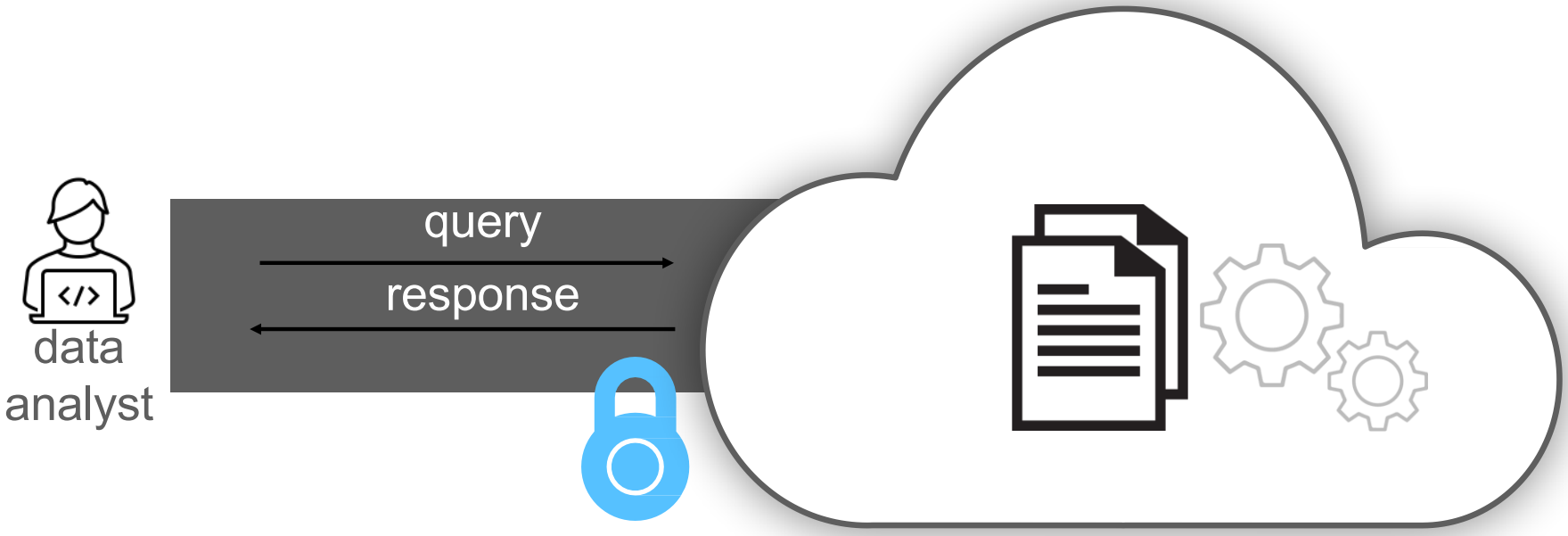
user records breached

Traditional security is insufficient

Use of encryption in practice

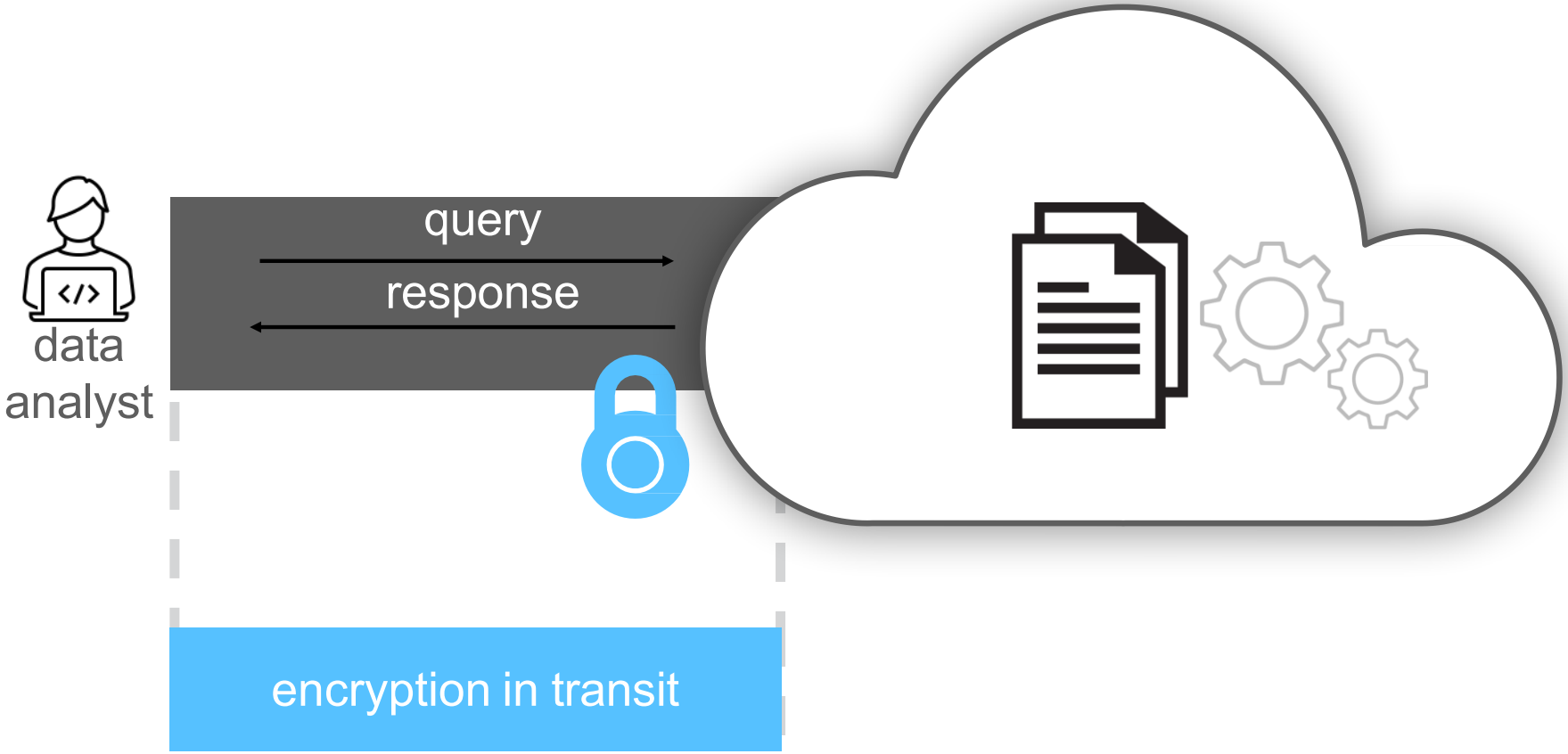
Use of encryption in practice

Ca' Foscari University of Venice



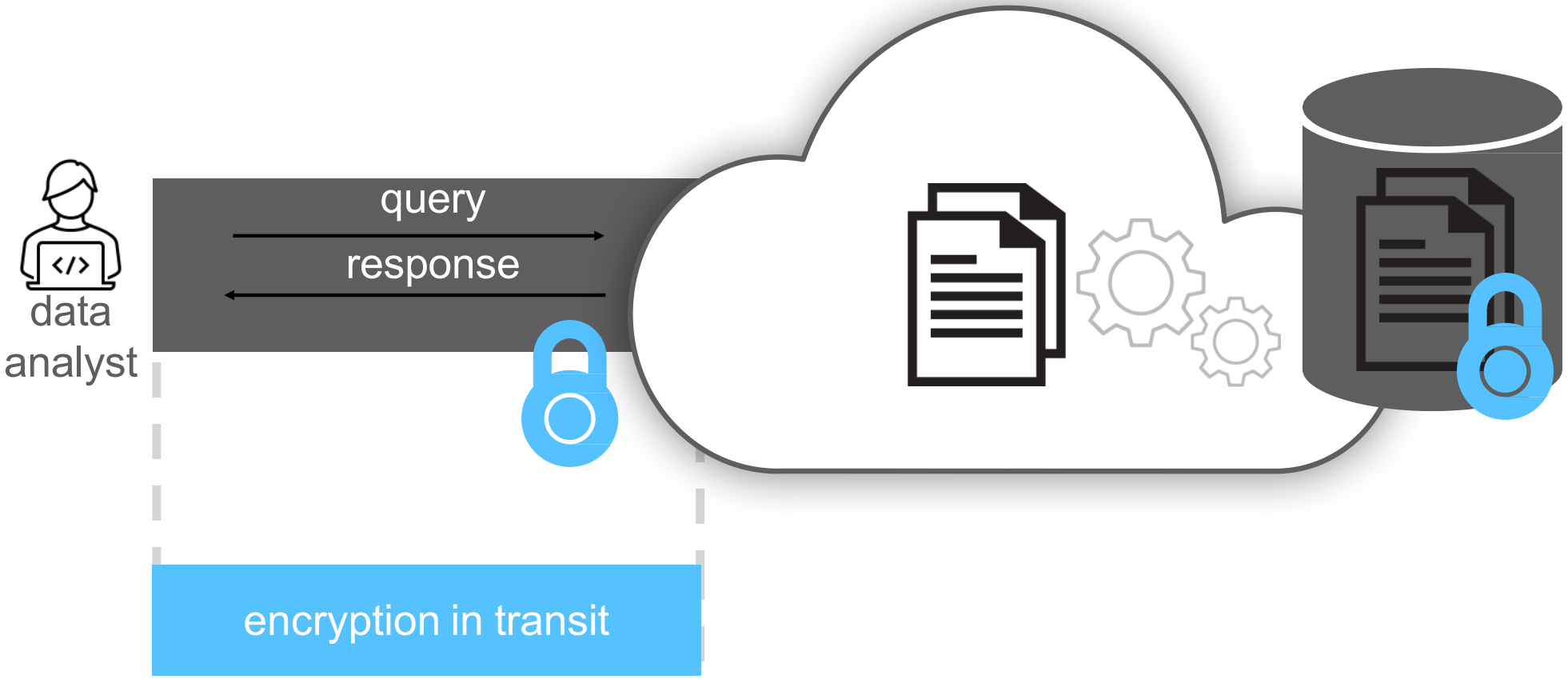
Use of encryption in practice

Ca' Foscari University of Venice



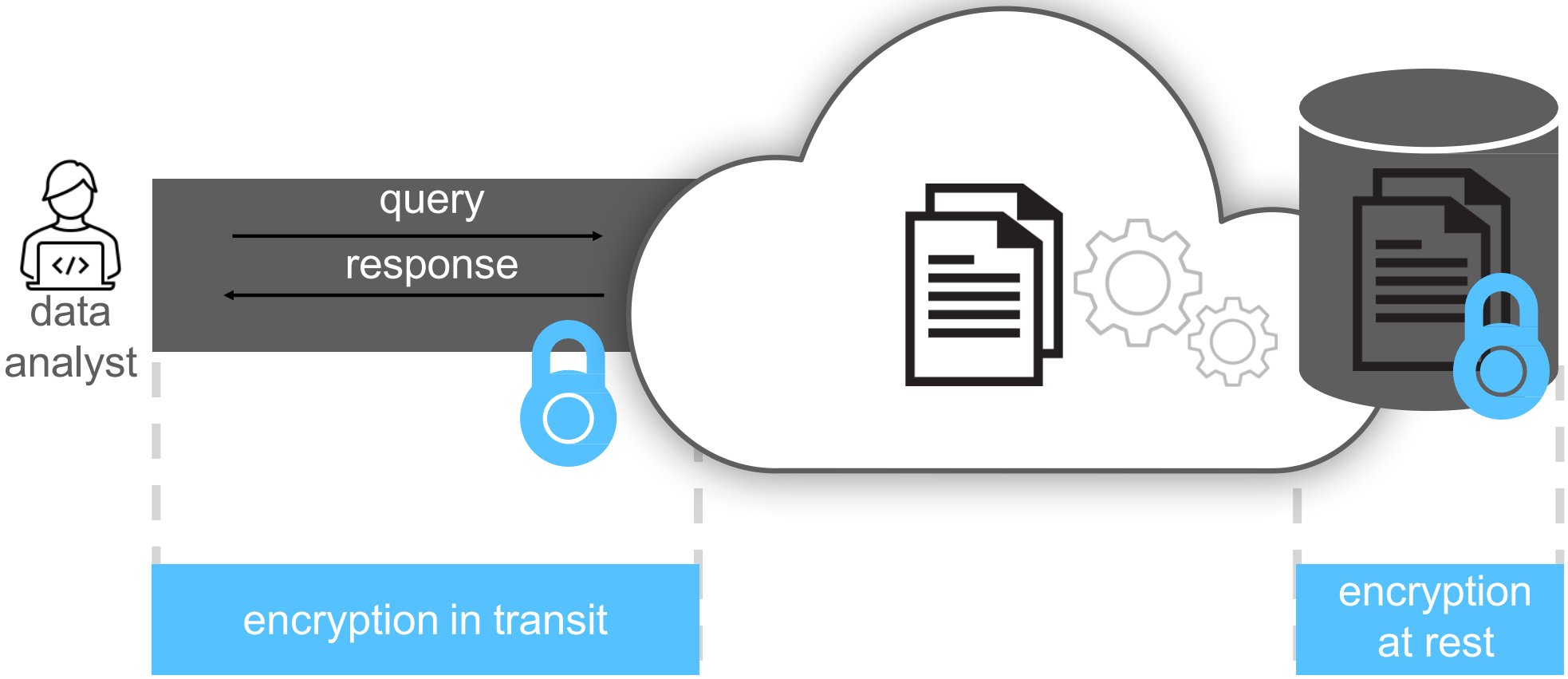
Use of encryption in practice

Ca' Foscari University of Venice



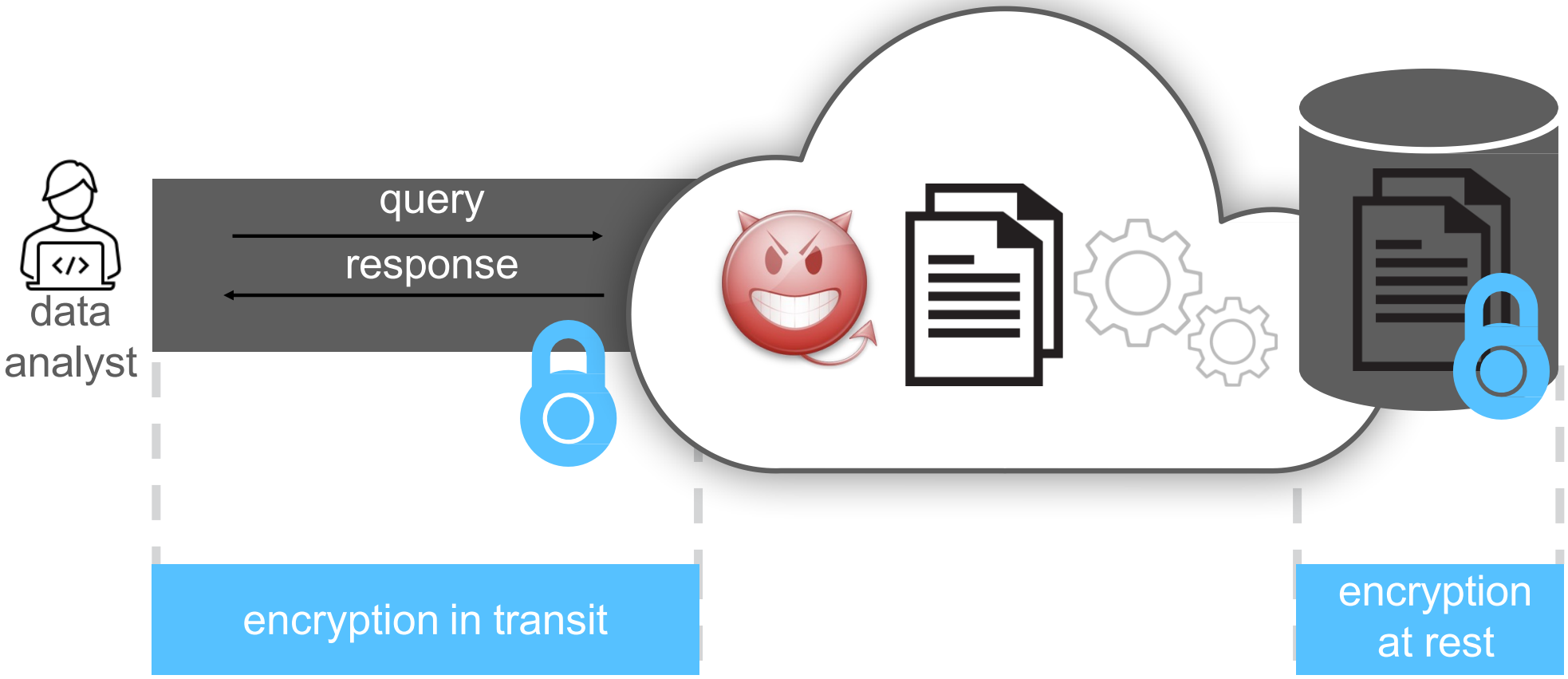
Use of encryption in practice

Ca' Foscari University of Venice



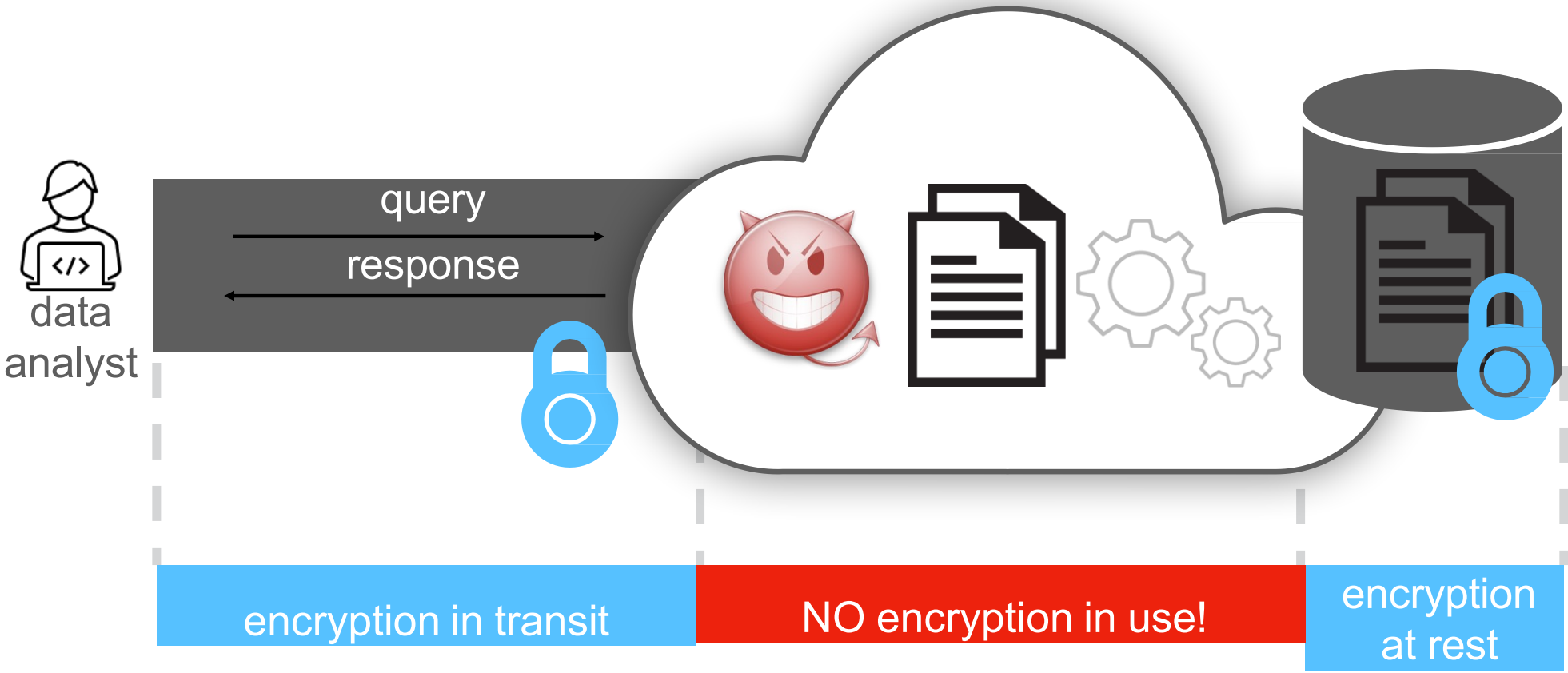
Use of encryption in practice

Ca' Foscari University of Venice



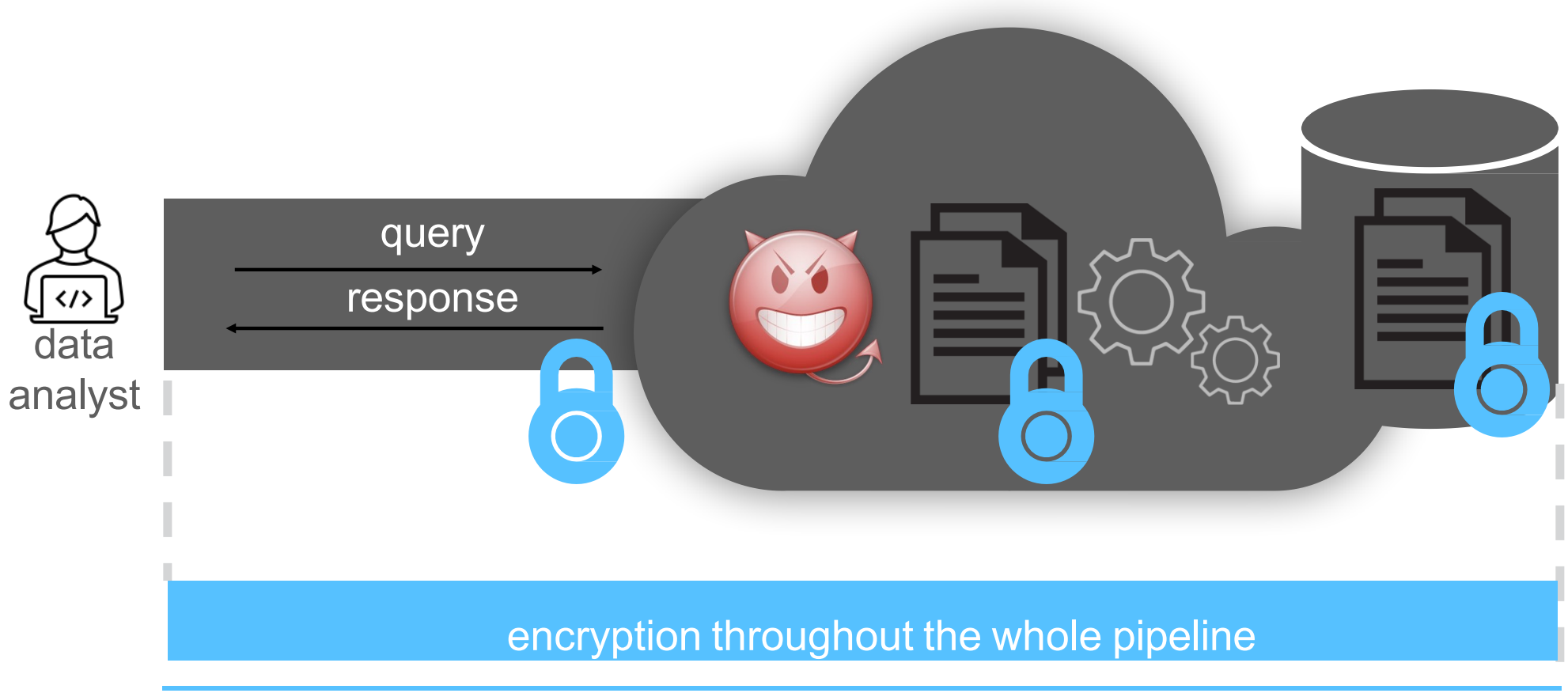
Use of encryption in practice

Ca' Foscari University of Venice



Secure computation:
keep data encrypted during processing

Ca' Foscari University of Venice



Next Generation Cryptography

Ca' Foscari University of Venice

- Protect data in transit (in the network)
  - Encryption, Digital signatures
- Protect data at rest (in cloud storage)
  - Encryption
- Protect data in use (while processing it)
  - Secure Multiparty computation
  - Fully Homomorphic Encryption

How to protect data while computing on it?

Ca' Foscari University of Venice

- How to reason whether the output reveals private information
- When it comes to sensitive information, such as:
  - medical records
  - financial transactions
  - genomic material

=> The computations cannot be performed in clear.

Privacy by Design

Ca' Foscari University of Venice

- Privacy patterns are design solutions to common privacy problems
  - a way to translate "privacy-by-design" into practical advice for software engineering.
- Design patterns can help *document common practices* and *standardize terminology*.
- https://privacypatterns.org/

Secure Multi-Party Computation

Secure Multiparty Computation (MPC)

Ca' Foscari University of Venice

- MPC guarantees that we do not reveal anything about the input data than the output of a computation previously agreed upon.
  - Mathematical guarantees: if anyone can learn anything more, then they can solve a hard mathematical problem

- Examples
  - Number of common users
  - Histogram of spent value of users who have seen ads for a product campaign
  - ML model predicting user preferences

MPC Timeline

Ca' Foscari University of Venice

- 1982: First MPC construction: Millionaire's problem
  - MPC protocols for any functionality
  - Asymptotic improvements of protocols for general and specific functionalities
- 2004: First implementation of MPC construction
  - Multitude of new implementations and fast pace concrete efficiency improvements

MPC Timeline

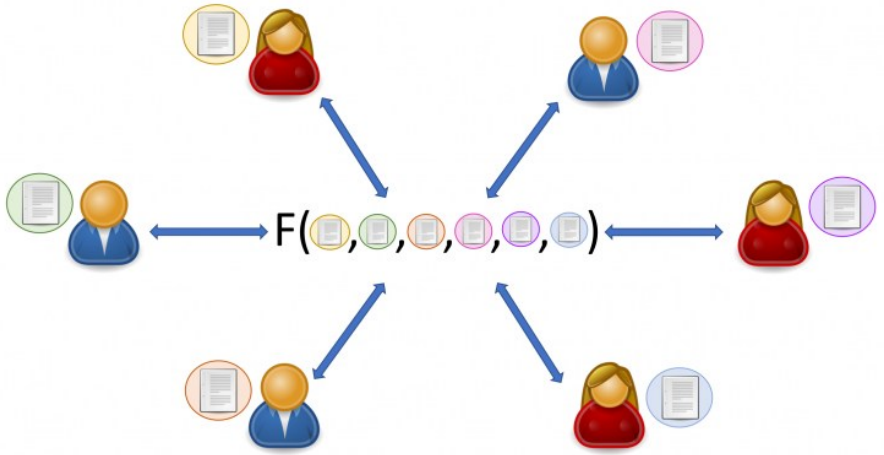
Ca' Foscari University of Venice

- 2009: First use of MPC solution implementation in practice: Sugar beet auction
  - Many orders of magnitude efficiency improvements
  - Many general MPC frameworks
  - Many highly optimized solutions for concrete tasks
  - Many MPC startups
  - Tool used in various industry settings
- 2022: MPC startup acquired for the first time
  - Continuous practical improvement
  - Broader users
  - Standardization efforts
  - MPC Alliance

Secure Multi-Party Computation

Ca' Foscari University of Venice

- In MPC, two or more parties would like to jointly compute a function F on their inputs, while keeping these inputs private.
- The first attempt to realize MPC is the one by Yao in 1982, who proposed a two-party computation (2PC) protocol based on Garbled Circuits in order to solve its famous Millionaires' problem



https://crysp.uwaterloo.ca/courses/pet/W11/cache/www.cs.wisc.edu/areas/sec/yao1982-ocr.pdf

Secure Multi-Party Computation

Ca' Foscari University of Venice

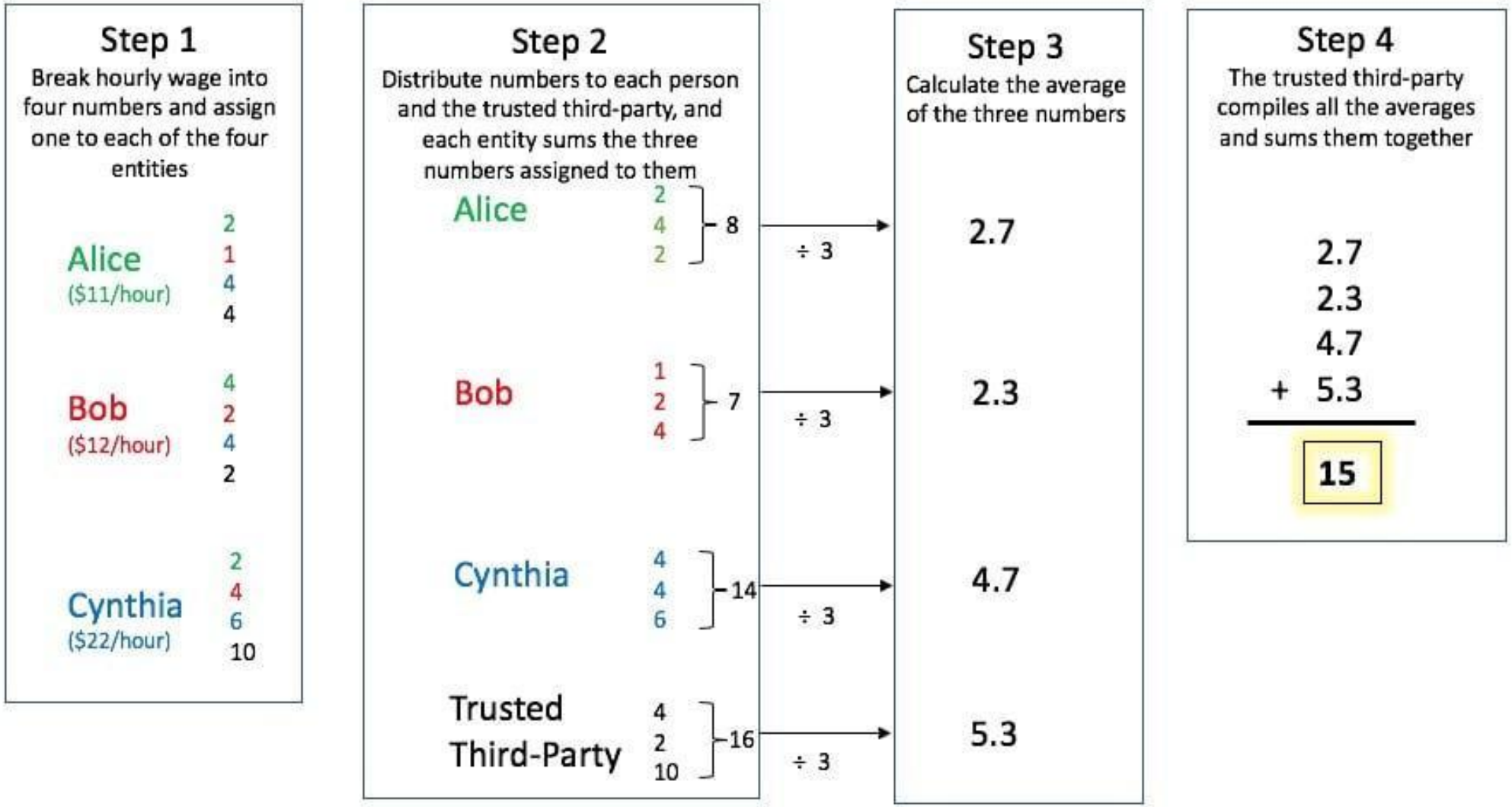
Millionaires' problem:

- two or more millionaires would like to learn who among them is the richest, without revealing to the others the money they own.
- Subsequent works solved the MPC problem by using secret sharing (binary and arithmetic), Beaver's triples, oblivious transfer, etc..
- In MPC, the input data is masked or secret shared, and the final result is generally revealed to all parties.
- The advantage of MPC is the somewhat low computational overhead, but it requires multiple rounds of interaction between the parties involved in the protocol.

Evans, David, Vladimir Kolesnikov, and Mike Rosulek. "A pragmatic introduction to secure multi-party computation." Foundations and Trends in Privacy and Security 2, no. 2-3 (2017). URL https://securecomputation.org

https://jonasspenger.github.io/blog/bgw-protocol-and-beaver-triples

MPC Example


Step 1
Break hourly wage into four numbers and assign one to each of the four entities
Alice
($11/hour)
2
1
4
4
Bob
($12/hour)
4
2
4
2
Cynthia
($22/hour)
2
4
6
10
Step 2
Distribute numbers to each person and the trusted third-party, and each entity sums the three numbers assigned to them
Alice
2
4
2
8
Bob
1
2
4
7
Cynthia
4
4
6
14
Trusted Third-Party
4
2
10
16
÷ 3
÷ 3
÷ 3
÷ 3
Step 3
Calculate the average of the three numbers
2.7
2.3
4.7
5.3
Step 4
The trusted third-party compiles all the averages and sums them together
2.7
2.3
4.7
+ 5.3
15

Applications of MPC: Data Analytics

Ca' Foscari University of Venice

- Big firms that deal with confidential user data, like healthcare companies or financial institutions, can collect data securely from an anonymous pool of users, compute, analyze, and gain insights from the data using MPC.
- This way, users will not reveal their personal information, and these organizations can analyse the data for insights without uncovering it.
- This same method can be applied to autonomous cars, shipping or truck fleets, and aircraft fleets, with recipients like car companies, city planners, and service providers benefitting from the insights without knowing any details on the information provided.

Applications of MPC: Genetic Testing

Ca' Foscari University of Venice

- MPC can also be utilized in genetic testing.
- Patients can access their genetic profiles privately and securely without revealing any confidential information on their:
  - metabolism rates
  - family traits
  - hereditary disease information
  - other data that they would rather not share

Applications of MPC: blockchain

Multi-party computation blockchain applications

- Threshold multi-signature (or multisig) technology schemes are a subfield of multi-party computation and can perform similar functions as a private key on the blockchain, including public address generation and transaction signing.
- With MPC blockchain applications, the private keys of a crypto wallet can be split (shards) among several parties in such a way that for any function to be performed, a minimum number of people holding key shares have to be involved.
- If some participants within the group become dishonest, they will not be able to have their way unless they reach the threshold for signing transactions.
  - Usually, these participants do not know each other.

Applications of MPC: crypto wallets

Ca' Foscari University of Venice

- There have historically been a few solutions for safely keeping private keys, those being either hot storage, cold storage, or hardware based storage.
- Multi-party computation protocols enhance private key security and, by extension, digital wallet security.
- Most cryptocurrency wallets use private keys, usually stored in a particular 'trusted' device.
  - This method primarily depends on the strength of the device's security measures, it also presents a single point of failure.
- Using MPC, a single private key is split up between multiple entities, making it more difficult for attackers to compromise the digital wallet since they have to attack multiple points simultaneously.

Multi-party Computation

Ca' Foscari University of Venice

- Multi-party computation has evolved over the years and remains a crucial breakthrough in the world of cryptography today. From sealed-bid auctions to crypto wallets, MPC protocols are being leveraged in various applications.
- From a blockchain perspective, leveraging MPC for security of digital wallets is a critical, immediate application that multiple parties are invested on.
  - The era of seed phrases may be over.
  - As of now, several wallet providers have already started opting MPC to offer better protection and an improved user experience.
- In the coming year, hopefully, we'll have more real-life applications similar to what happened with public key authentication.

Fully Homomorphic Encryption

A Brief History of Cryptography

Ca' Foscari University of Venice

- In the beginning, there was ***symmetric*** encryption.

Julius Ceasar (100-44 BC)

Message: **ATTACK AT DAWN**

A Brief History of Cryptography

Ca' Foscari University of Venice

- If you had the key, you could **encrypt…**

Julius Ceasar (100-44 BC)

DWWDFN DW GDZQ

Message:	**ATTACK AT DAWN**
Key: +3	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Ciphertext:	**DWWDFN DW GDZQ**

A Brief History of Cryptography

Ca' Foscari University of Venice

- If you had the key, you could **decrypt…**

Julius Ceasar (100-44 BC)

DWWDFN DW GDZQ

Ciphertext:	**DWWDFN DW GDZQ**
Key: -3	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Message:	**ATTACK AT DAWN**

A Brief History of Cryptography

Ca' Foscari University of Venice

- If you had the key, you could **decrypt…**

Julius Ceasar (100-44 BC)

DWWDFN DW GDZQ

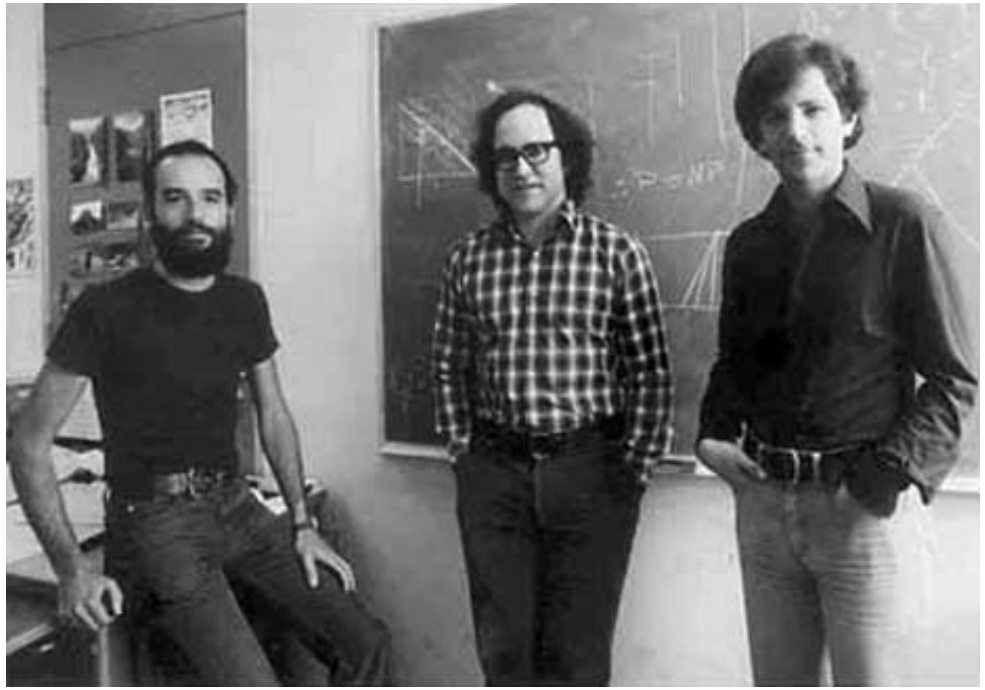
Symmetric Encryption:

Encryption and Decryption use the same key

A Brief History of Cryptography (1970s)

Ca' Foscari University of Venice

Merkle, Hellman and Diffie (1976)



Shamir, Rivest and Adleman (1978)

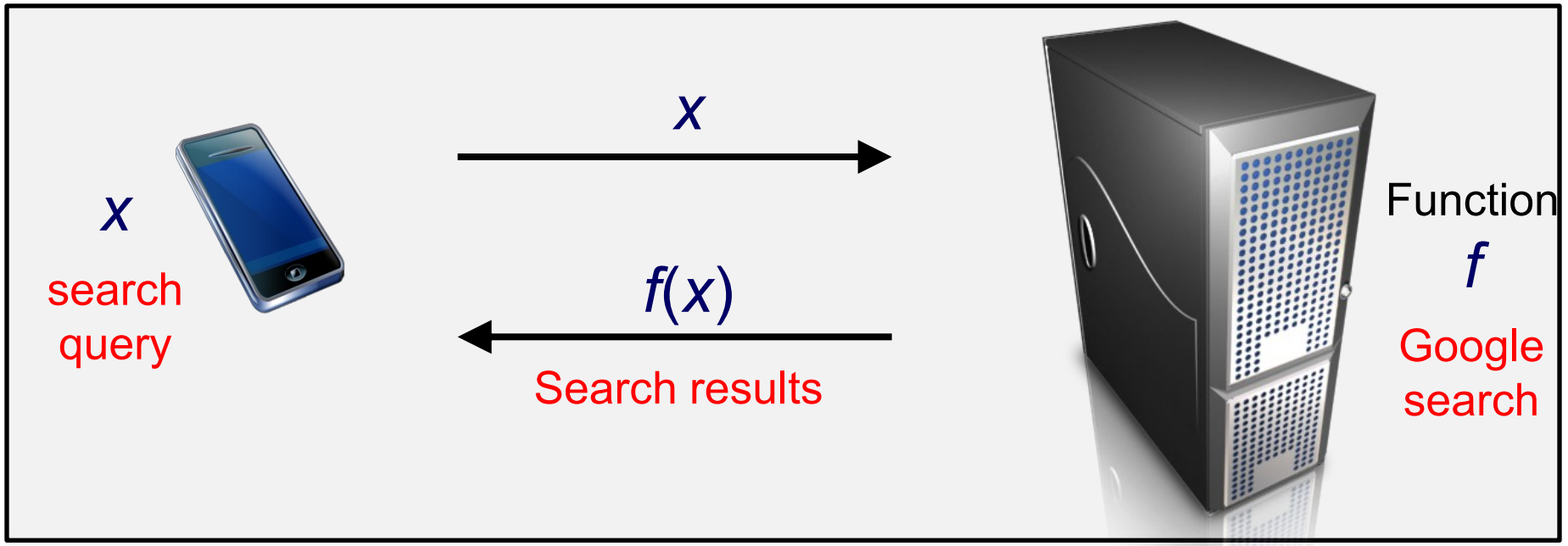
Asymmetric Encryption: foundation of e-commerce

Encryption uses a public key, Decryption uses the secret key

Privacy

Ca' Foscari University of Venice

- What else can we do with encrypted data, anyway?

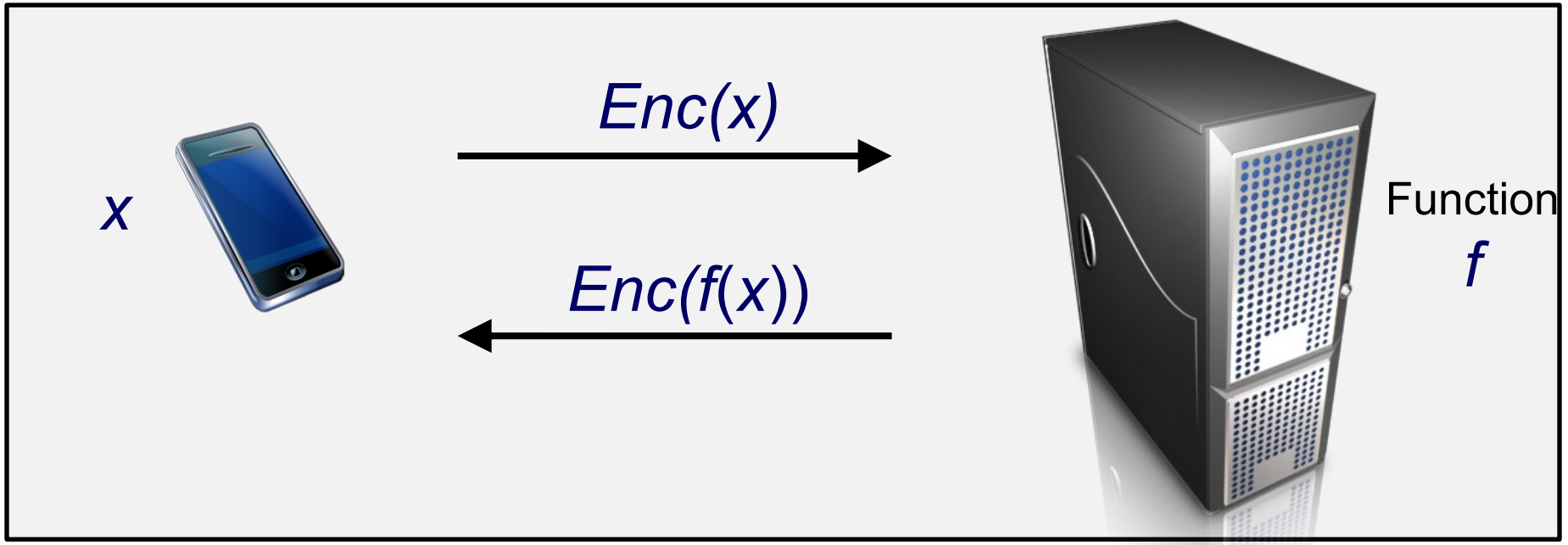


WANT PRIVACY!

Computing on Encrypted Data

Ca' Foscari University of Venice

- What else can we do with encrypted data, anyway?



WANT PRIVACY!

Computing on Encrypted Data

Ca' Foscari University of Venice

- Some people noted the algebraic structure in RSA…

$$E(m_1) = {m_1}^e \qquad E(m_2) = {m_2}^e$$

Ergo … $E(m_1) \times E(m_2)$

$$= {m_1}^e \times {m_2}^e$$

$$= (m_1 \times m_2)^e$$

$$= E(m_1 \times m_2)$$

RSA is multiplicatively homomorphic

(but not additively homomorphic)

Other Encryptions were additively homomorphic

(but not multiplicatively homomorphic)

Multiplicative Homomorphism

$$E(m_1) \times E(m_2) = E(m_1 \times m_2)$$

Additive Homomorphism

$$E(m_1) + E(m_2) = E(m_1 + m_2)$$

Computing on Encrypted Data

Ca' Foscari University of Venice

- What people really wanted was the ability to do arbitrary computing on encrypted data…
- … and this required the ability to compute *both* sums *and* products …
- … on the same encrypted data set!

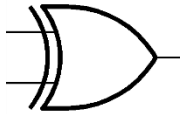
Computing on Encrypted Data

Ca' Foscari University of Venice

Why SUMs and PRODUCTs?

SUM

=

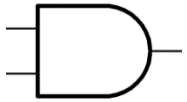


XOR

0 XOR 0	0
1 XOR 0	1
0 XOR 1	1
1 XOR 1	0

PRODUCT

=



AND

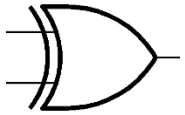
0 AND 0	0
1 AND 0	0
0 AND 1	0
1 AND 1	1

Computing on Encrypted Data

Ca' Foscari University of Venice

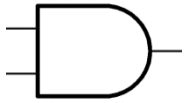
Because {XOR,AND} is Turing-complete …

… any function is a combination of XOR and AND gates



XOR

0 XOR 0	0
1 XOR 0	1
0 XOR 1	1
1 XOR 1	0



AND

0 AND 0	0
1 AND 0	0
0 AND 1	0
1 AND 1	1

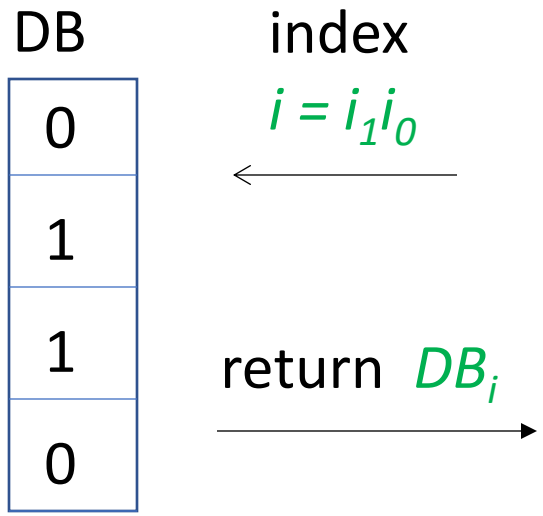
Computing on Encrypted Data

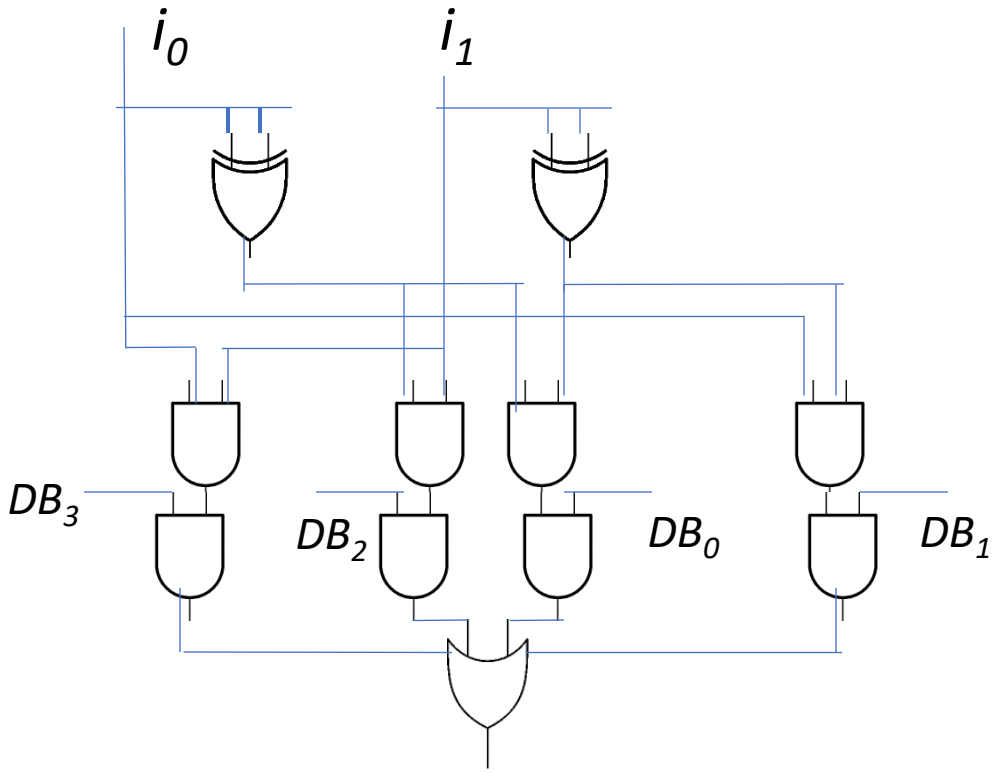
Ca' Foscari University of Venice

Because {XOR,AND} is Turing-complete …

… any function is a combination of XOR and AND gates

Example: Indexing a database





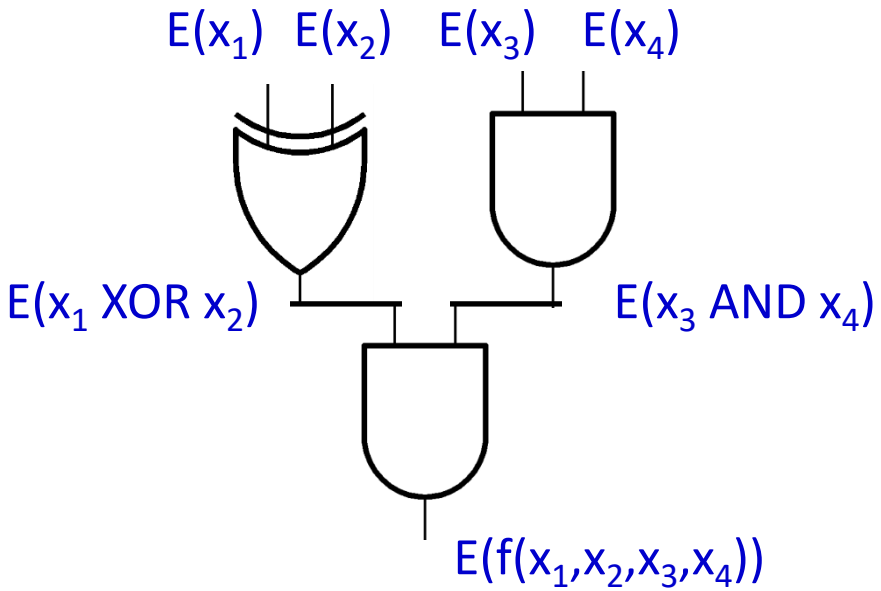
Computing on Encrypted Data

Ca' Foscari University of Venice

Because {XOR,AND} is Turing-complete …

… if you can compute sums and products on encrypted bits

… you can compute ***ANY*** function on encrypted inputs



Computing on Encrypted Data

Ca' Foscari University of Venice

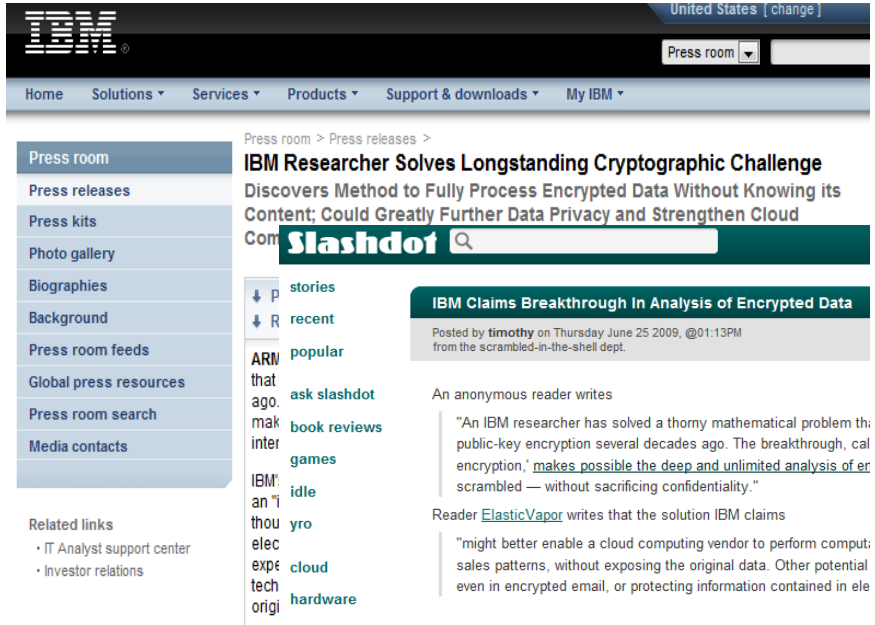
Fully-Homomorphic Encryption!

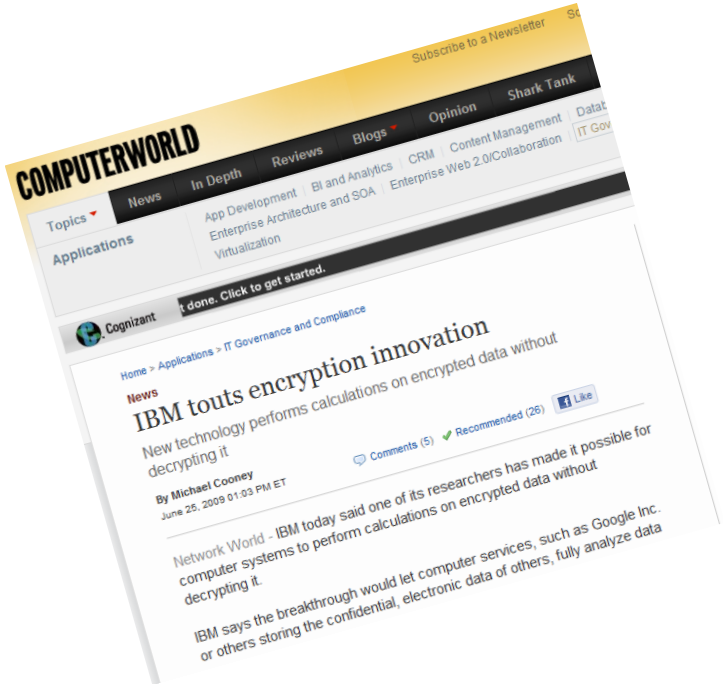
Cryptography's Holy Grail

Fully-Homomorphic Encryption

Ca' Foscari University of Venice

- Delegate arbitrary processing of data… without giving away access to it!
  - Private Cloud Computing
- People tried do this for years and years with no success... until, in 2008 ***Craig Gentry*** created the first fully homomorphic encryption scheme …





Homomorphic Encryption example

Ca' Foscari University of Venice

- We wish to add 1 and 2
- The data is encrypted so that E(1) = 33 and E(2) =54
- The encrypted data is sent to the cloud and processed: the result is 87
- The result is downloaded and decrypted to provide the final answer 3

How does it work?

Ca' Foscari University of Venice

- What sort of math we use?
- What sort of objects can we add and multiply?

Polynomials?

$$(x^2 + 6x + 1) + (x^2 - 6x) = (2x^2 + 1)$$

$$(x^2 + 6x + 1) \times (x^2 - 6x) = (x^4 - 35x^2 - 6x)$$

Matrices?

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} X \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ -1 & 3 \end{pmatrix}$$

How about integers?!?

Symmetric Encryption

Ca' Foscari University of Venice

- *Secret key:* large *odd* number **p**

*To Encrypt a bit **b**:*

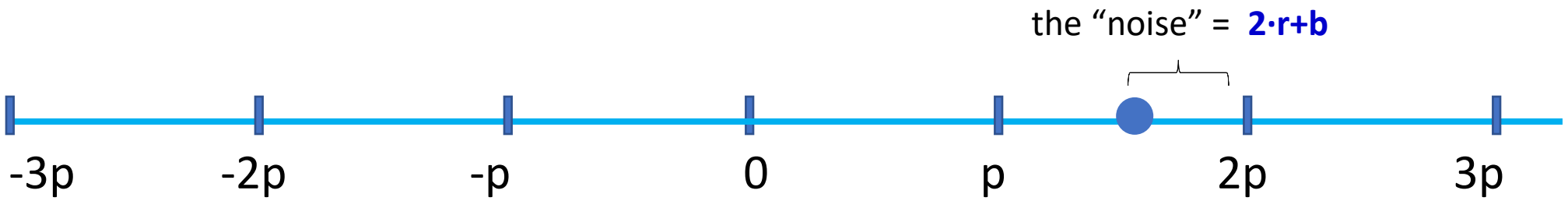
– pick a (random) "large" multiple of p, say **q·p**

– pick a (random) "small" number **2·r+b** (this is even if b=0, and odd if b=1)

– Ciphertext **c = q·p + 2·r +b**

*To Decrypt a ciphertext **c**:*

Taking ***c** mod p* recovers the noise



How secure is this?

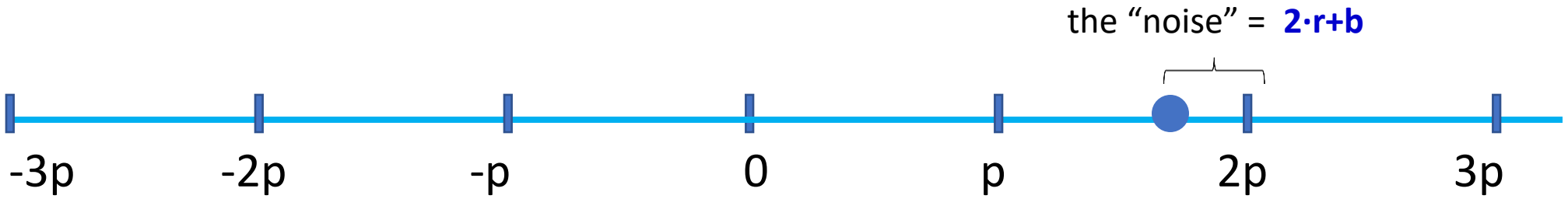
Ca' Foscari University of Venice

… if there were no noise (think r=0)

… and I give you two encryptions of 0 (q_1p & q_2p)

… then you can recover the secret key p

$$= \text{GCD}(q_1p, q_2p)$$



How secure is this?

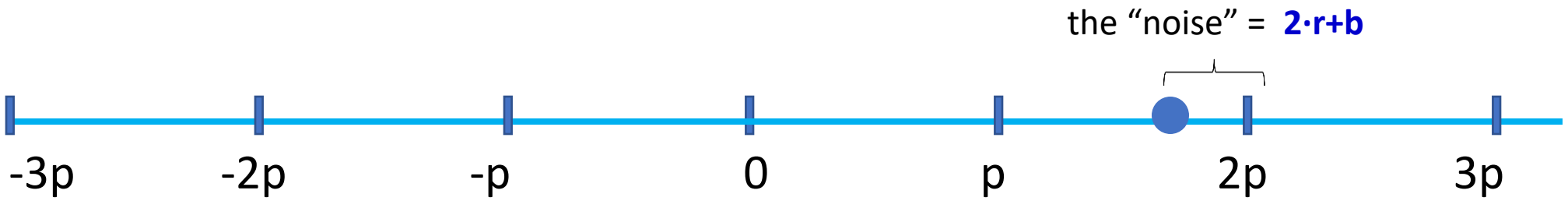
Ca' Foscari University of Venice

… but if there is noise

… the GCD attack doesn't work

… and neither does any attack

… this is called the *approximate GCD assumption*



XORing two encrypted bits

Ca' Foscari University of Venice

$\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$

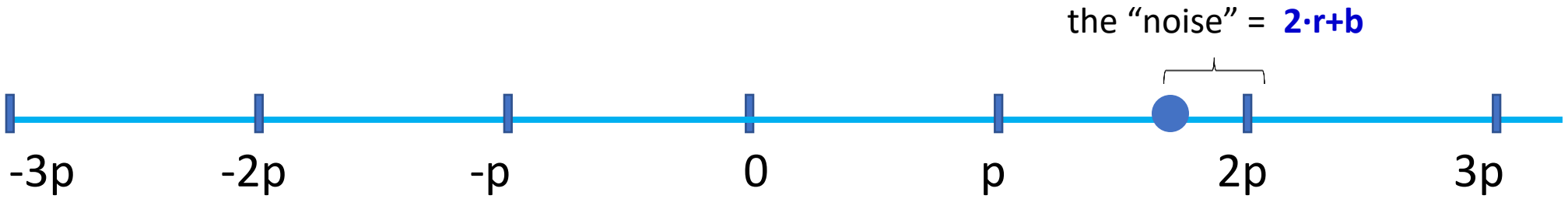
$\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$

$\mathbf{c_1 + c_2} = \mathbf{p} \cdot (q_1 + q_2) + \mathbf{2} \cdot (r_1 + r_2) + (b_1 + b_2)$

lsb= b_1 XOR b_2

Odd if $b_1=0$, $b_2=1$ (or) $b_1=1$, $b_2=0$

Even if $b_1=0$, $b_2=0$ (or) $b_1=1$, $b_2=1$



ANDing two encrypted bits

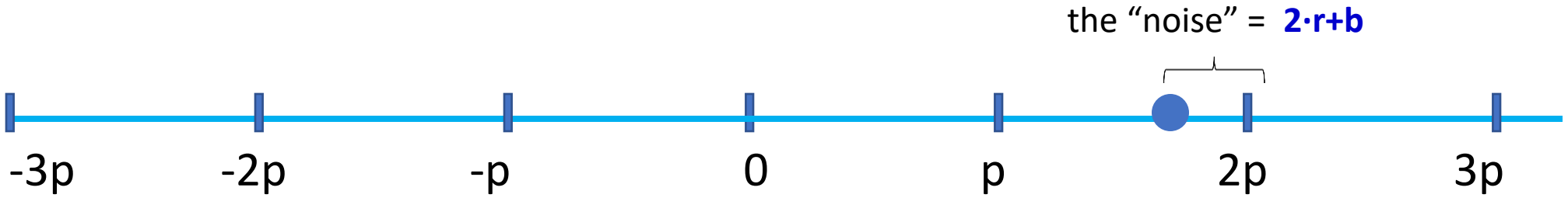
Ca' Foscari University of Venice

$\mathbf{c_1} = q_1 \cdot p + (2 \cdot r_1 + b_1)$

$\mathbf{c_2} = q_2 \cdot p + (2 \cdot r_2 + b_2)$

$\mathbf{c_1 c_2} = \mathbf{p} \cdot (c_2 \cdot q_1 + c_1 \cdot q_2 - q_1 \cdot q_2) + \underbrace{\mathbf{2} \cdot (r_1 r_2 + r_1 b_2 + r_2 b_1) + b_1 b_2}$

lsb= b_1 AND b_2



The noise grows!

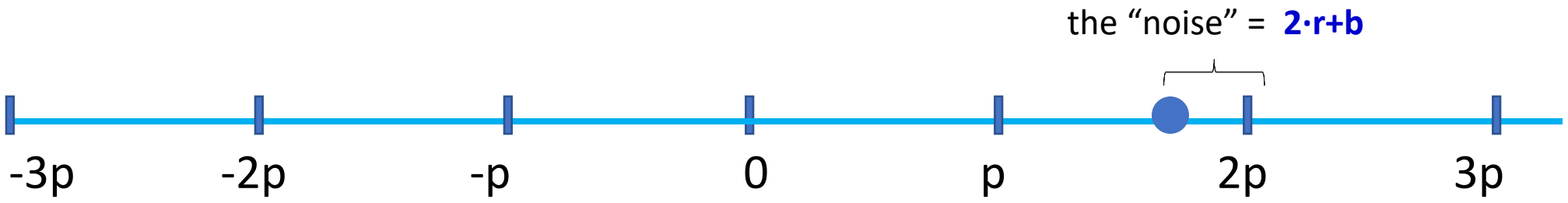
Ca' Foscari University of Venice

$$-\mathbf{c_1}+\mathbf{c_2} = \mathbf{p}\cdot(q_1 + q_2) + \underbrace{\mathbf{2}\cdot(r_1+r_2) + (b_1+b_2)}$$

noise= 2 * (initial noise)

$$-\mathbf{c_1}\mathbf{c_2} = \mathbf{p}\cdot(c_2\cdot q_1+c_1\cdot q_2-q_1\cdot q_2) + \underbrace{\mathbf{2}\cdot(r_1r_2+r_1b_2+r_2b_1) + b_1b_2}$$

noise = (initial noise)^2

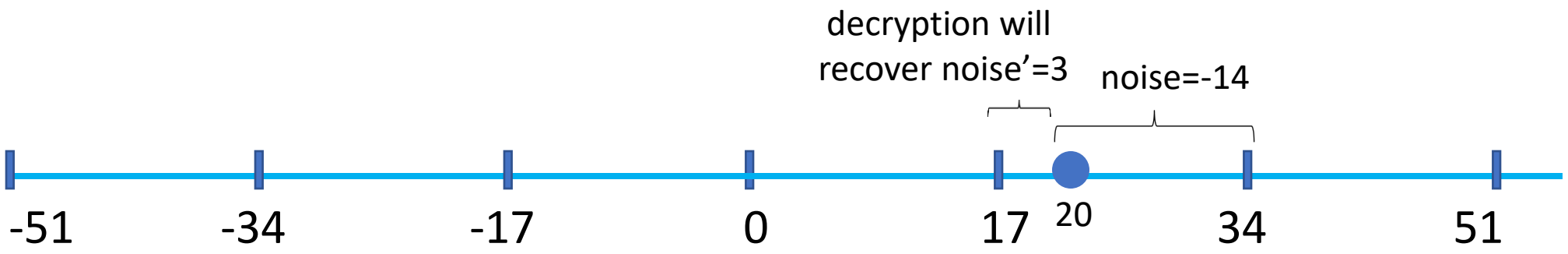


The noise grows

Ca' Foscari University of Venice

- What is the problem?
- *If the |noise| > p/2, then …*

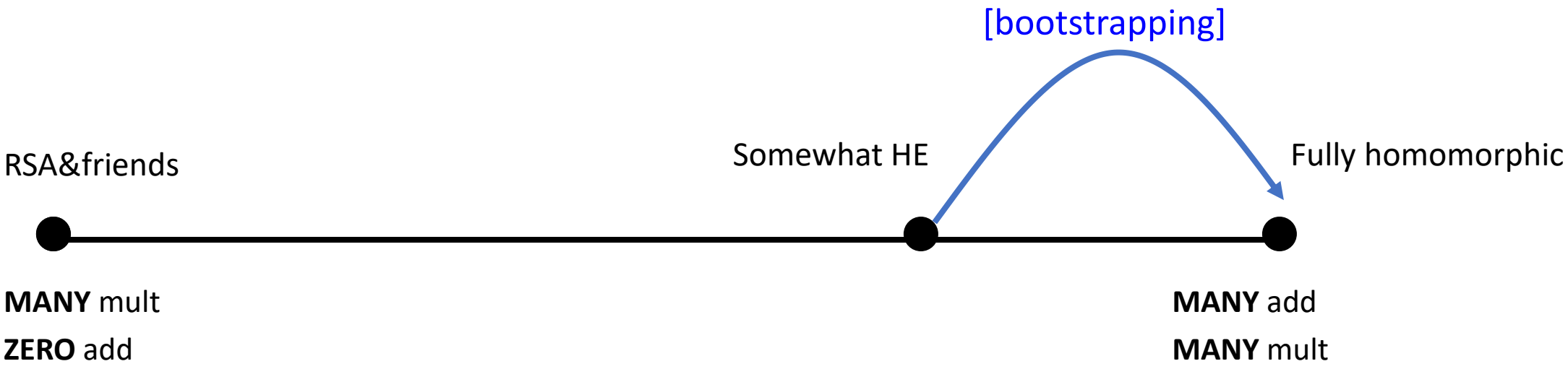
…decryption will output an incorrect bit



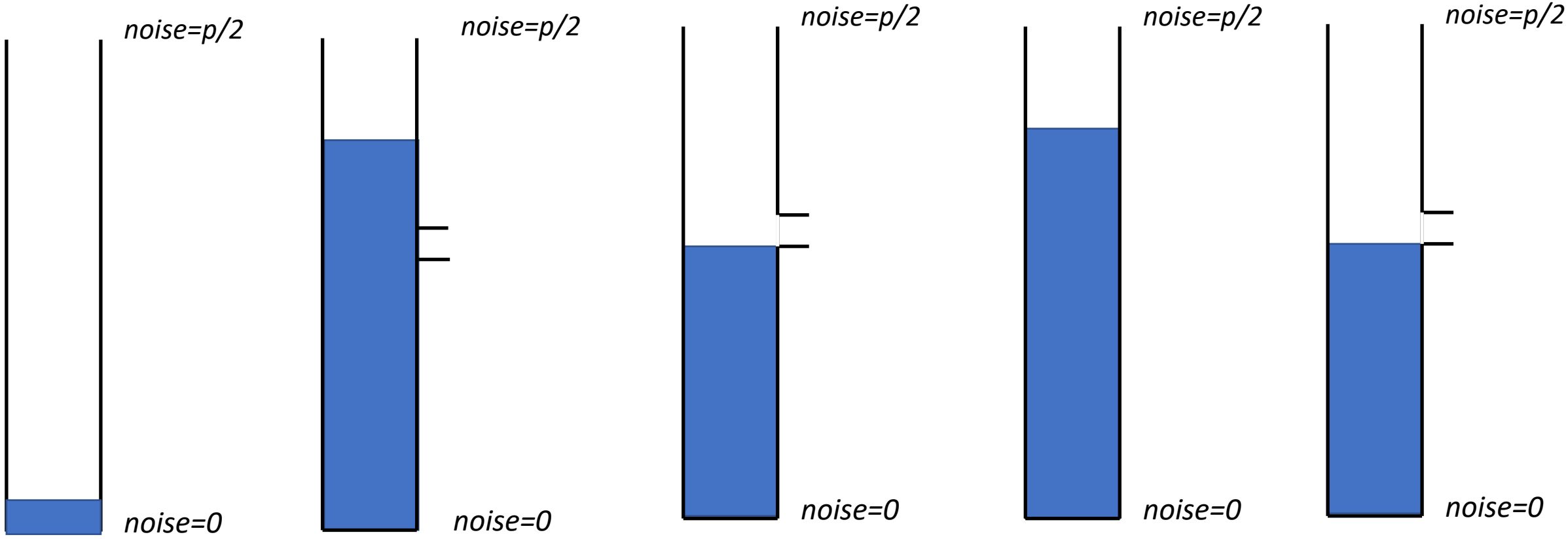
In the end

Ca' Foscari University of Venice

- *… we can do lots of additions and*
- *… some multiplications (= a "somewhat homomorphic" encryption) …*
- *enough to do many useful tasks,*
  - *e.g., database search, spam filtering etc.*



The bootstrapping method



Initial noise

Bootstrapping =
"Valve" at a fixed height

… repeat until done

Fully Homomorphic Encryption

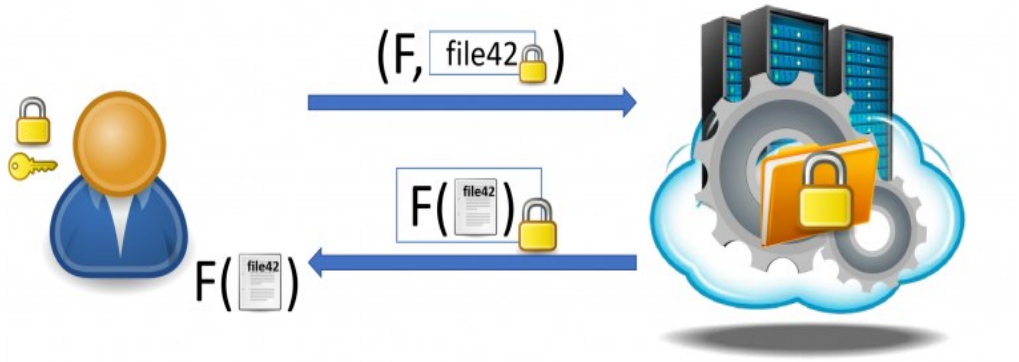
Ca' Foscari University of Venice

- In FHE the idea is to perform computations on encrypted data.
- Different types of homomorphic encryption exist:
  - additive, partial, leveled, somewhat (SHE)
- When we talk about fully, we mean that potentially every function can be evaluated, while in the other cases there are limitations.

Fully Homomorphic Encryption

Ca' Foscari University of Venice

- One of the most common scenarios where FHE can be used is in outsourced computations:
  - a client sends encrypted data to a server and asks this latter to evaluate a function F on this encrypted data.
- The inputs and outputs of the computation are encrypted with the client's secret/public key and the server manipulates only encrypted data.



Fully Homomorphic Encryption

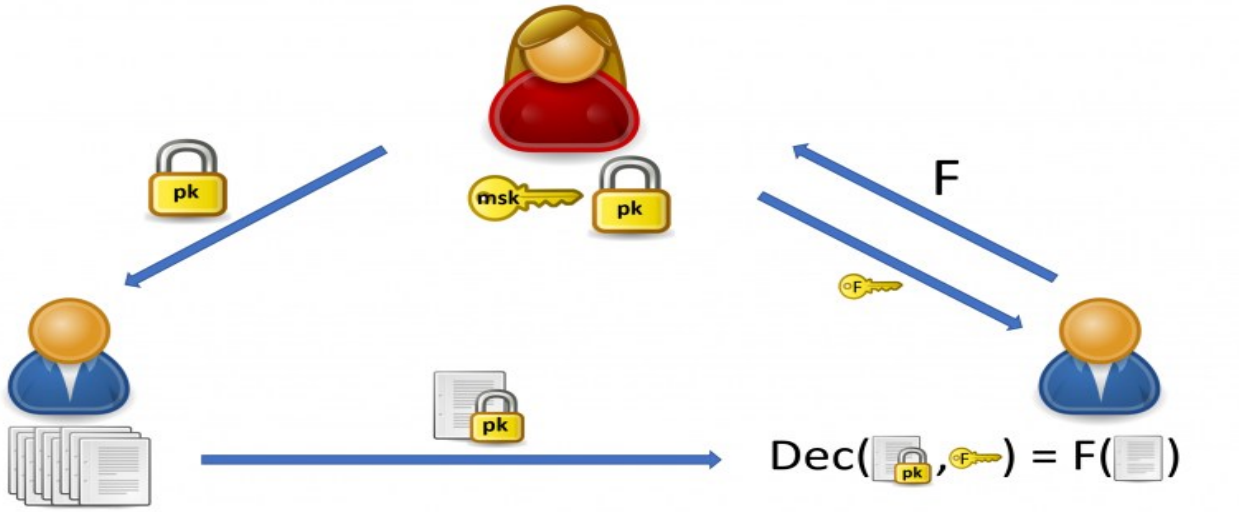
Ca' Foscari University of Venice

- The first solution was proposed by Gentry in 2009 (about 30 years after the seminal idea by Rivest, Adleman and Dertouzos), which introduced a technique called bootstrapping.
- Bootstrapping allows to refresh noisy ciphertexts and can be used to ensure a correct answer after decryption.
- In fact in the majority of the FHE solutions, the ciphertexts contain a little amount of noise that increases after each operation.
- If this noise is not controlled, the decryption can be wrong.
- FHE demands no interaction between the client and the server during the computations, but the computations are still very expensive.

Functional Encryption

Ca' Foscari University of Venice

- FE is a public key construction, on which it is possible to produce functional secret keys allowing a party to evaluate a specific function F (generally public) on an encrypted input during its decryption.
- So the input is encrypted and the output is in cleartext: the party performing the (functional) decryption learns the result of the function on the specific data, but nothing else.



Functional Encryption

Ca' Foscari University of Venice

- A practical application of this technique could be its use in spam filters, on which a user wants his email provider to learn if a specific email is spam, without allowing him to learn anything else about the content of the emails.
- FE has been properly formalized in 2011 by Boneh, Sahai and Waters. Nowadays, there are no known FE schemes that can be used to efficiently evaluate general functions.
- However, the literature proposes multiple efficient constructions to evaluate linear and quadratic functions.

https://link.springer.com/content/pdf/10.1007/978-3-642-19571-6_16.pdf

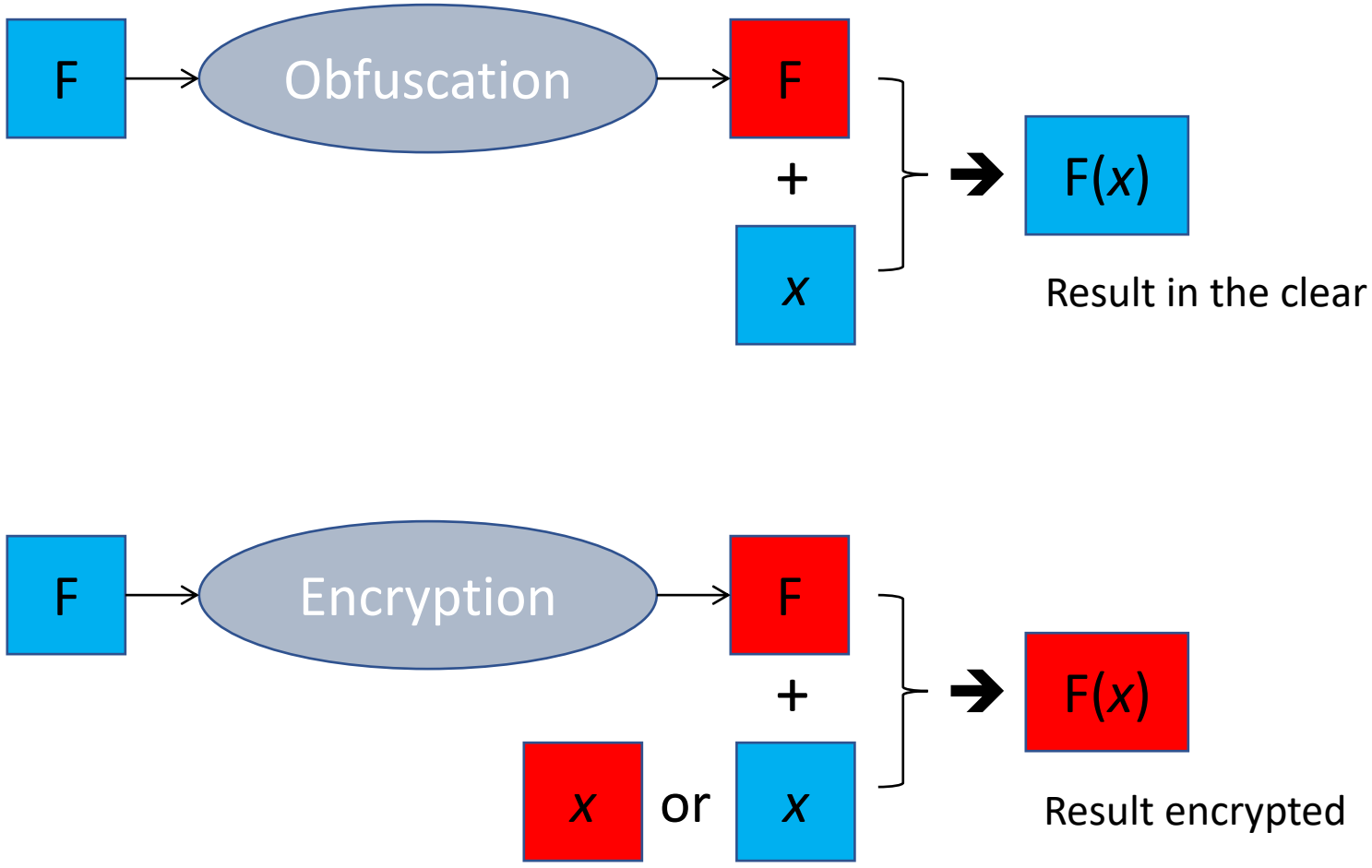
Summary

Ca' Foscari University of Venice

	MPC	**FHE**	**FE**
Input	Masked/shared	Encrypted	Encrypted
Output	Cleartext	Encrypted	Cleartext
Interaction	Yes	No	No
Computation	Efficient	Reasonable in SHE Expensive in FHE	Efficient (for linear and quadratic functions)

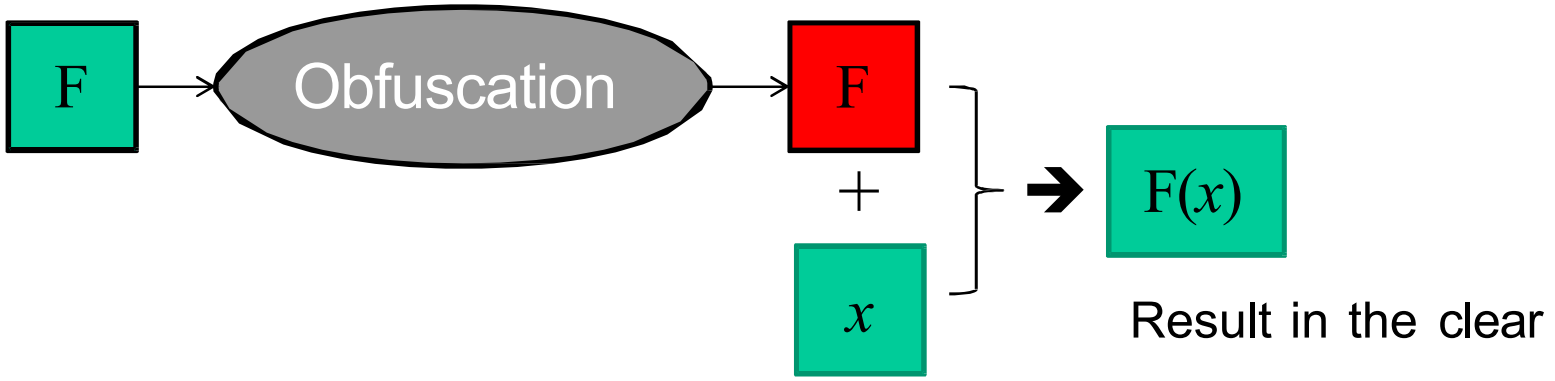
Obfuscation vs. HE

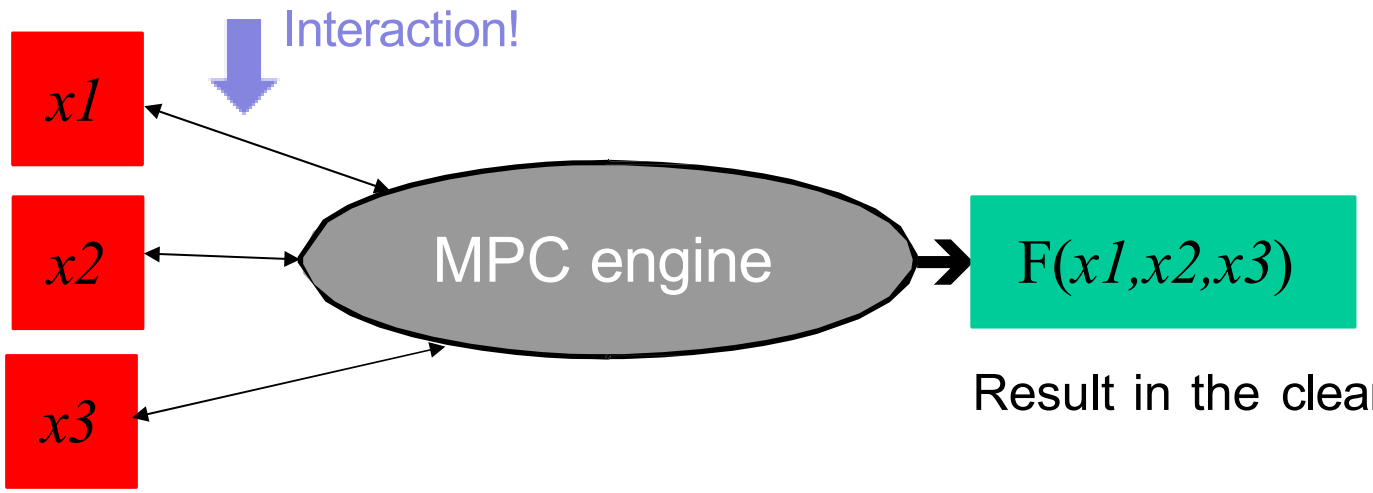
Ca' Foscari University of Venice



Obfuscation ⇔ Multiparty Computation

Ca' Foscari University of Venice





MPC feasible for more and more functions

Other PETs

Differential Privacy (DP)

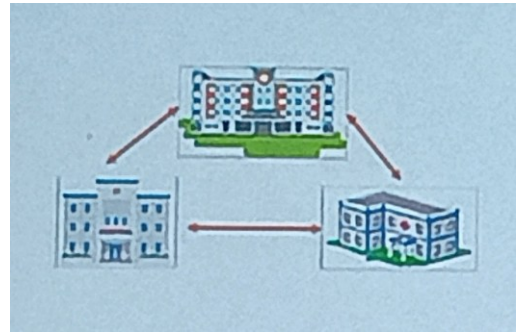
Ca' Foscari University of Venice

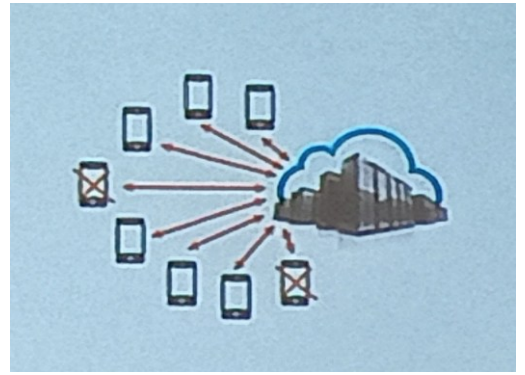
- What does the output of a computation on a database reveal about individual records /users?
  - Differential Privacy – measure and bound the difference of the computation output on two databases that differ on a single record
  - Looking at the output one cannot distinguish whether a particular record was included in the computation
  - It was introduced at Theory of Cryptography Conference (TCC) in 2006

Privacy-Preserving Computation Settings

Ca' Foscari University of Venice

- Cross-organizational Collaboration
  - Small number of parties
  - Full peer-to-peer communication
  - Computation resources at each participant
- User population Modelling
  - Weak devices
  - Star communication (non P2P)
  - Devices may drop out





Google Protected Computing

Ca' Foscari University of Venice

- A toolkit of technologies that transforms how, when, and where data is processed to technically ensure the privacy and safety of user data

1. Minimize the data footprint
   - Collection, Retention, usage and sharing of identifiable data is minimized
2. De-identify data
   - Identity is stripped from your data so it is no longer linked to you
3. Restrict access to data
   - Making it technically impossible for anyone (including Google) to access your sensitive data

Protected Computing vs PETs

Ca' Foscari University of Venice

- PETs (Privacy-Enhancing Technologies) minimize and protect personal data
  - PETs are about achieving a technically enforced privacy outcome
  - They allow for analysis of large data sets in a way where no one person's information is ever disclosed

- Protected Computing builds on top of PETs to technically ensure the privacy and safety of data with technologies like:
  - Cloud enclaves
  - Edge processing
  - End-to-End encryption

Private Join and Count

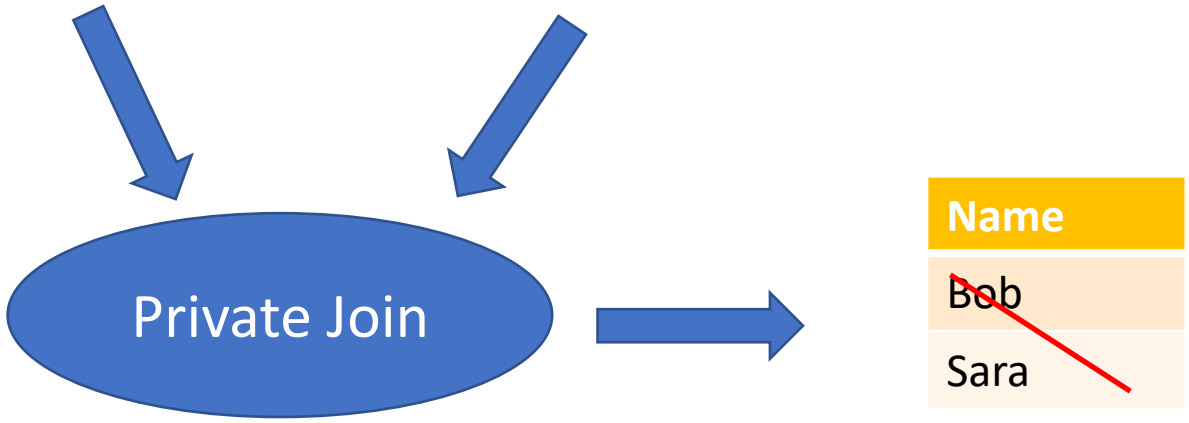
Ca' Foscari University of Venice

Bus Riders

Name
Bob
Peter
Sara

Customers

Name
Bob
Sara
Alice



Name
~~Bob~~
Sara

COUNT()

2

Private Join and Compute (PJ&C)

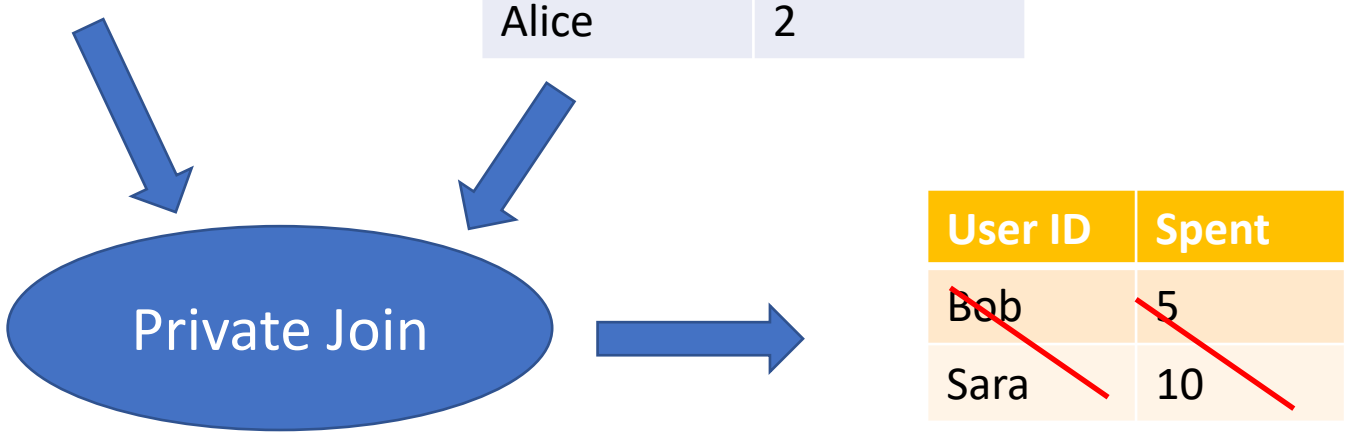
Ca' Foscari University of Venice

Bus Riders

User ID
Sara
Chris
Peter

Customer Purchases

User ID	Spent
Bob	3
Peter	10
Sara	5
Alice	2



User ID	Spent
~~Bob~~	~~5~~
Sara	10

SUM() → (2, 15)

PJ&C Libraries in C++
https://github.com/Google/private-join-and-compute

The zk-SNARK (Succinct Non-Interactive Arguments of Knowledge) is probably the most popular form of zero-knowledge proof, first coined in the 2011 **Bit+11** paper.

By 2013, zero-knowledge proof could be used in real life applications thanks to the **Pinocchio PHGR13** paper which made zk-SNARKS applicable for general computing, though at a slower pace. The **Groth16** algorithm proposed in 2016 greatly reduced computational complexity, and made zk-SNARKS so efficient, it is still the standard used today.

However, a trusted setup is essential for the security of these zero-knowledge protocols. An initial process used to generate the cryptographic parameters is necessary in order to be able to run the zero-knowledge protocol. This is done by a third-party to ensure that the cryptographic parameters are random, unpredictable, and secure.

Bulletproofs (BBBPWM17) were subsequently introduced in 2017, and **zk-STARKs (BBHR18)** in 2018. Differing from their predecessors, they are types of range proofs that do not require an initial trusted set up. The 2019 **PlonK** paper implemented the Universal Zero-Knowledge proof algorithm, which meant that the trusted setup only needs to be initiated once, while in comparison, Groth16 required every circuit to have a separate trusted set up.

- Thanks to developments in the field, zero-knowledge proofs have transitioned from being purely theoretical to having useful real-life applications in blockchain, secure communications, electronic voting, access control and gaming. As they continue to be put into commercial use, there will be even more exciting developments to advance the technology.

ZKP Example: Proof of Membership

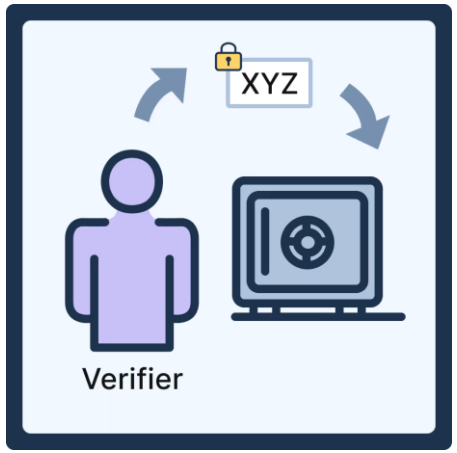
Ca' Foscari University of Venice

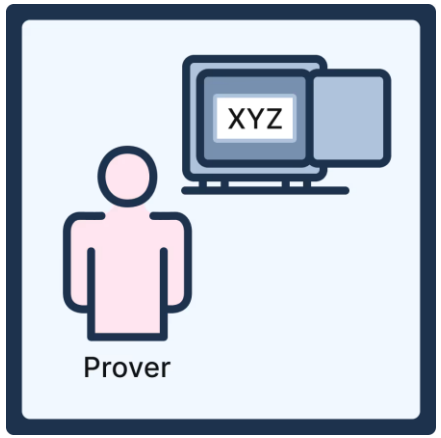
- You meet someone you do not know, yet she claims to also be a member of the group you are part of. How can you know if you can trust her
- Luckily, your group has a locked safe, and only the members of your group know the secret combination code to gain acces to the safe.
- So write a secret message and place it in the locked safe.

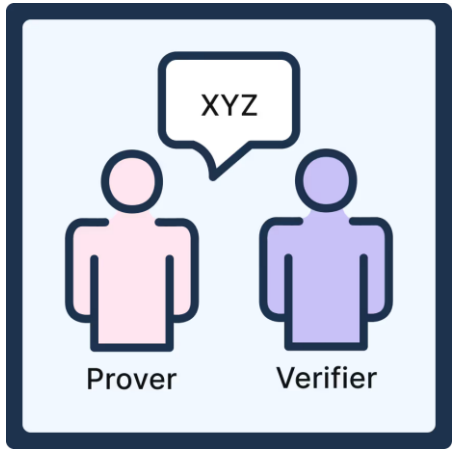
ZKP Example: Proof of Membership

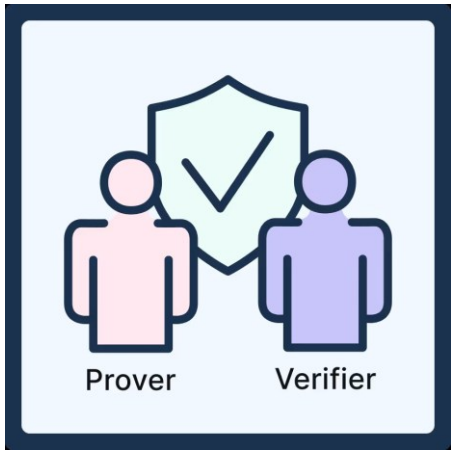
Ca' Foscari University of Venice

1. Verifier writes a secret message and put it in a locked safe
2. Prover, who meets the requirements, has knowledge of the combination code and opens the locked safe
3. Prover returns the secret message to Verifier
4. Verifier is convinced that the prover really knows the combination code and can therefore be trusted
5





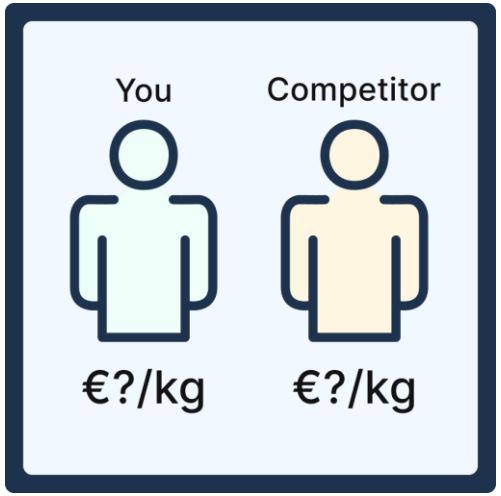




ZKP example: Opaque Pricing

Ca' Foscari University of Venice

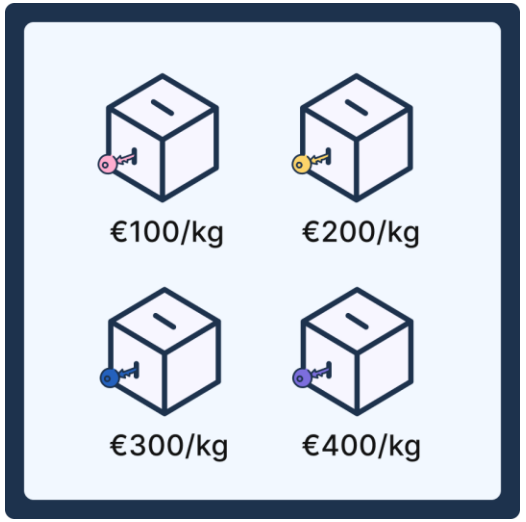
- You and a competitor discover that you are buying the same materials from the same supplier.
- You want to find out if you are paying the same price per kilogram
- However, there isn't enough trust between the both of you to share the prices you are each paying, and you are also contractually bound to not share this information.
- Assuming the market rate for the materials can only be 100, 200, 300 or 400 per kilogram, one can set up a zero-knowledge proof for this situation.

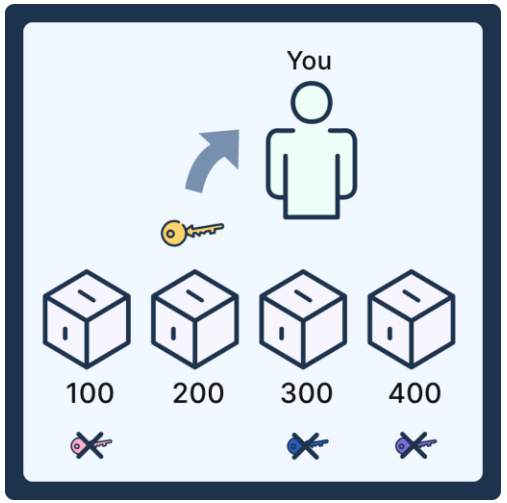


ZKP example: Opaque Pricing

Ca' Foscari University of Venice

1. We get 4 lockable lockboxes, each with a small slot that can take only a piece of paper:
   - They are labelled with price per kg, and placed in a secure, private room.
2. You go into the room alone first. Since you are paying 200 per kg, you take the key from the lockbox that is labelled 200 and destroy the keys for the other boxes. You leave the room.

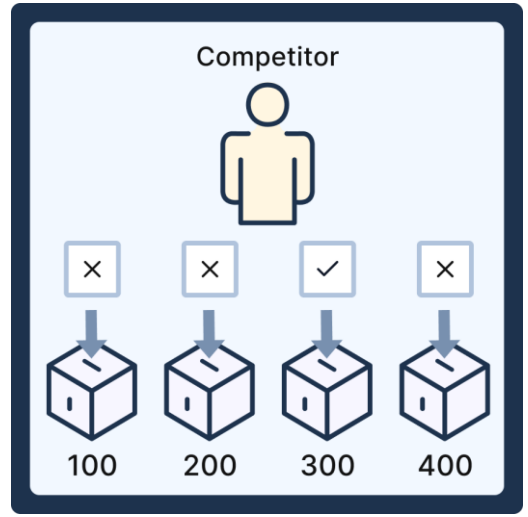


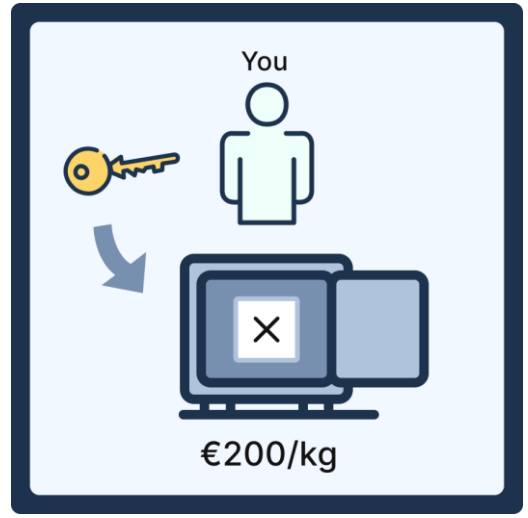


ZKP example: Opaque Pricing

Ca' Foscari University of Venice

3. Your competitor goes into the room alone with 4 pieces of paper, 1 with a check, and 3 with crosses. Because your competitor is paying 300 per kilogram, they slide the paper with a check inside the lockbox that is labelled 300, and slide the papers with crosses into the other lockboxes. They leave the room.
4. After they leave, you can return with your key that can only open the lockbox labelled 200. You find a piece of paper with a cross on it, so now you know that your competitor is not paying the same amount as you.





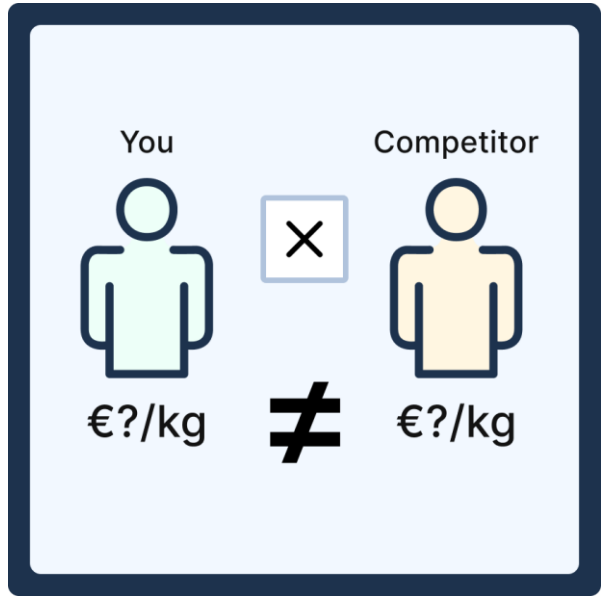
ZKP example: Opaque Pricing

Ca' Foscari University of Venice

5. Your competitor returns and sees that you have a piece of paper with a cross on it, so now they also know that you are not paying the same amount as them.

If you get a piece of paper with a check on it, both of you would know that you are paying the same amount. Since you got the paper with a cross on it, both of you know that you are not paying the same amount, but also without knowing how much the other is paying.

- Both of you leave knowing **only that you are not paying the same amount, but neither of you has gained knowledge of what the other is paying.**



Machine Learning privacy challenges

Ca' Foscari University of Venice

- Main shortcomings like lack of continual learning on edge devices and aggregating private data on central servers.
- A central ML model is built using all available training data in a centralized environment
- In mobile computing, users demand fast responses and the communication time between the user device and a central server may be too slow for a good user experience.
  - To overcome this, the model may be placed in the end-user device but then continual learning becomes a challenge since models are trained on a complete data set and the end user device does not have access to the complete dataset.
- User's data gets aggregated in a central location for machine learning training which may be against the privacy policies of certain countries and may make the data more vulnerable to data breaches

Federated Learning

Ca' Foscari University of Venice

- Can we do cross-device machine learning and analytics without centralized data collection?

- Federated Learning (FL) is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider.
- Each client's raw data are stored locally and not exchanged or transferred
- Instead focused updates intended for immediate aggregation are used to achieve the learning objective

Advances and Open problems in Federated Learning (Foundations and Trends in Machine learning 21)

Federated Learning in Mobile Apps

Ca' Foscari University of Venice

Mobile apps use machine learning for personalization, like next-word prediction, face detection, fingerprint and voice recognition.

However, traditional AI training centralizes user data, which would increase concerns about privacy, security, and data governance.

Federated learning addresses these challenges by allowing models to be trained across a network of devices without transmitting raw user data:

- **Privacy-preserving AI**: Sensitive user data remains on the device, reducing risks of data exposure while still improving model accuracy.
- **Personalized and adaptive models**: Apps can fine-tune AI models based on individual usage patterns without needing constant cloud updates.
- **Lower bandwidth usage**: Instead of uploading large datasets, only model updates are shared, making federated learning efficient for mobile networks.
- **Improved security**: By keeping data decentralized, federated learning mitigates risks associated with centralized data storage and breaches.

Federated Learning in Healthcare

Ca' Foscari University of Venice

Traditional data centralization, where hospitals and institutions pool medical records into a single repository, raises significant concerns about data governance, security, and compliance with regulations like HIPAA and GDPR.

Federated learning enables collaborative model training across multiple institutions without requiring direct data sharing.

- **Enhanced privacy and security**: Sensitive patient data remains within its original source, reducing the risks of exposure and data breaches.
- **Improved data diversity**: By training on datasets from different hospitals, research centers, and electronic health records, federated learning enables models to recognize rare diseases and improve diagnostic accuracy across diverse populations.
- **Scalable medical AI**: Machine learning models can be continuously refined on real-world data from multiple institutions, leading to more reliable predictive analytics and better patient outcomes.

Federated Learning in Autonomous Vehicles

Ca' Foscari University of Venice

Traditional cloud-based approaches can introduce latency and pose safety risks, particularly in high-density traffic scenarios where split-second decisions are critical.

Autonomous vehicles can collaboratively train models while keeping data localized.

This approach ensures that vehicles continuously refine their decision-making based on the latest road conditions, without excessive data transfer.

- **Real-time traffic and road awareness**: Vehicles can quickly process and share insights on road hazards, construction zones, or sudden weather changes, ensuring safer navigation.
- **Immediate decision-making**: Onboard AI can react faster to dynamic driving conditions, reducing dependency on remote servers and minimizing latency in critical moments.
- **Continual model improvement**: As more vehicles contribute their localized learnings, autonomous systems evolve and enhance their predictive accuracy over time.

Federated Learning in Predictive Maintenance

Ca' Foscari University of Venice

- Predictive maintenance helps manufacturers reduce downtime, extend equipment lifespan, and boost efficiency, but its implementation faces challenges, including data privacy, security, and cross-border sharing restrictions.
- Instead of aggregating information from multiple plants or customers into a central repository, federated learning allows each site to train models locally.
- **Privacy-preserving AI**: Industrial data remains on-site, eliminating concerns about sharing proprietary or sensitive operational data with external entities.
- **Cross-border compliance**: Many manufacturers operate in multiple countries, each with different data protection regulations.
- **Adaptability to diverse equipment and conditions**: Manufacturing environments vary widely based on machinery, workload, and operational settings.

Federated learning enables compliance by keeping data localized while still benefiting from collective intelligence.

Federated learning allows predictive models to be tailored to local conditions while contributing to a broader understanding of equipment failure patterns.

Conclusions

Ca' Foscari University of Venice

- Even if there is still lot of space for improvement in all the three domains, these technologies are already very promising.
- A few practical examples already exist in literature, and many others theoretical ones are left as open problems.
- Concerning the practical ones, we can already observe constructions mixing MPC with (additive) homomorphic encryption:
  - homomorphic encryption is used as a subroutine to produce correlated randomness, useful in the so-called online phase
  - https://link.springer.com/content/pdf/10.1007/978-3-642-32009-5_38.pdf
- FHE can also become "multi-party", in the schemes allowing a multi-key functionality: e-voting schemes are a practical application of this technology.
- Open problem: adding a FE decryption step after a huge FHE computation, in order to output a plaintext instead of a ciphertext.
  - No secure solution has been found yet, but such a solution (if efficient) could be a very powerful tool for secure computation.

References

Ca' Foscari University of Venice

- Computing arbitrary functions of Encrypted Data", Craig Gentry, Communications of the ACM 53(3), 2010.
- Fully Homomorphic Encryption from the Integers, Marten van Dijk, Craig Gentry, Shai Halevi, Vinod Vaikuntanathan http://eprint.iacr.org/2009/616, Eurocrypt 2010.
- Computing Blindfolded: New Developments in Fully Homomorphic Encryption, Vinod Vaikuntanathan, IEEE Foundations of Computer Science Invited Talk, 2012
- https://datatilsynet.no/en/
- https://github.com/diagonalworks/diagonal-pets
- https://github.com/mc2-project/mc2
- https://www.openfhe.org/
- https://github.com/google/fully-homomorphic-encryption
- https://fhe.org/
- https://privacypatterns.org/