École Pour l'Informatique et les Techniques Avancées – EPITA

Masters program

Course: Data Privacy by Design

Created & Coordinated by: M. Salman Nadeem

Data Privacy by Design (DPbD)

Course schedule (tentative)

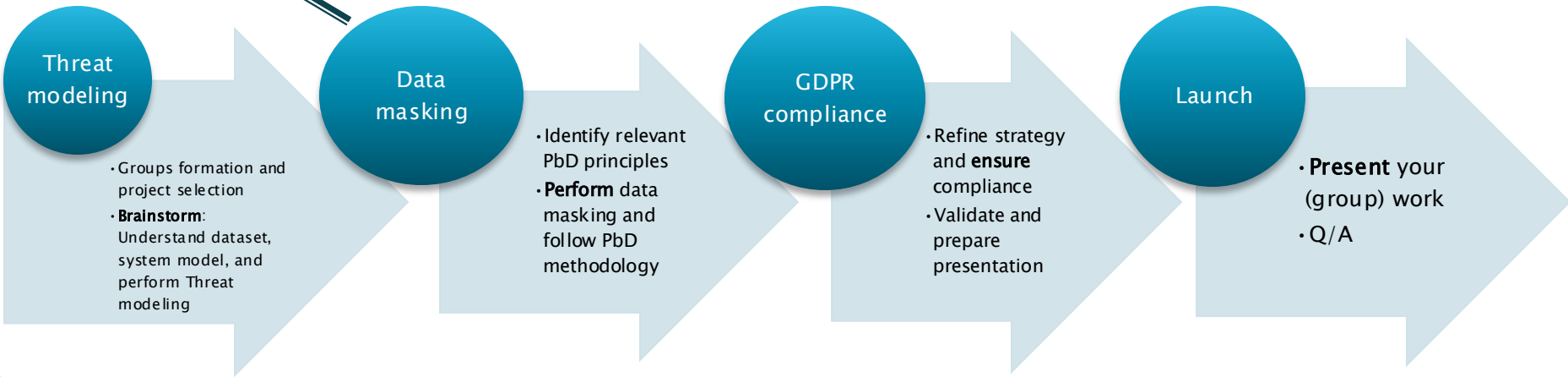
Date	No.	Topics	Duration (in hours)
(check Zeus platform for the date & time)	1	Introduction, DPbD fundamentals with case studies	3
(check Zeus platform for the date & time)	2	**Data privacy risks, Crypto. Package, Data masking (Anonymization vs Pseudonymisation)**	**3**
(check **Zeus** platform for the date & time)	3	Privacy Enhancing Technologies (PETs), DPbD and General Data Protection Regulation (GDPR)	3
(check **Zeus** platform for the date & time)	4	Recap, Conclusion	3
(check **Zeus** platform for the date & time)	5	Final evaluation	3
		Total	***15 hours***

GRADING criteria:
Class participation comprising attendance & reactivity: 10%
Class activities (quizzes): 30%
Final project (individual report & group presentation): 60%

Lecture 2 Outline

1. Common data privacy risks
2. Crypto package
3. Data masking (Anonymization vs Pseudonymisation)
4. Privacy Enhancing Technologies (PETs)

QUIZ



Hands-on → Final project

OWASP Top 10 Privacy Risks project

#	Type	Title	Frequency	Impact	Description
P1	👩‍💻	Web Application Vulnerabilities	High	Very high	Vulnerability is a key problem in any system that guards or operates on sensitive user data. Failure to suitably design and implement an application, detect a problem or promptly apply a fix (patch) is likely to result in a privacy breach. This risk also encompasses the OWASP Top 10 List of web application vulnerabilities and the risks resulting from them.
P2	👤👩‍💻	Operator-sided Data Leakage	High	Very high	Failure to prevent the leakage of any information containing or related to user data, or the data itself, to any unauthorized party resulting in loss of data confidentiality. Introduced either due to intentional malicious breach or unintentional mistake e.g. caused by insufficient access management controls, insecure storage, duplication of data or a lack of awareness.
P3	👤👩‍💻	Insufficient Data Breach Response	High	Very high	Not informing the affected persons (data subjects) about a possible breach or data leak, resulting either from intentional or unintentional events; failure to remedy the situation by fixing the cause; not attempting to limit the leaks.
P4	👤	Consent on Everything	Very high	High	Aggregation or inappropriate use of consent to legitimate processing. Consent is "on everything" and not collected separately for each purpose (e.g. use of website and profiling for advertising).
P5	👤	Non-transparent Policies, Terms and Conditions	Very high	High	Not providing sufficient information to describing how data is processed, such as its collection, storage, and processing. Failure to make this information easily-accessible and understandable for non-lawyers.
P6	👤👩‍💻	Insufficient Deletion of Personal Data	High	High	Failure to effectively and/or timely delete personal data after termination of the specified purpose or upon request.
P7	👤👩‍💻	Insufficient Data Quality	Medium	Very high	The use of outdated, incorrect or bogus user data. Failure to update or correct the data.
P8	👩‍💻	Missing or insufficient Session Expiration	Medium	Very high	Failure to effectively enforce session termination. May result in collection of additional user-data without the user's consent or awareness.
P9	👤👩‍💻	Inability of users to access and modify data	High	High	Users do not have the ability to access, change or delete data related to them.
P10	👤	Collection of data not required for the user-consented purpose	High	High	Collecting descriptive, demographic or any other user-related data that are not needed for the purposes of the system. Applies also to data for which the user did not provide consent.

Version 2.0 – 2021

Source: https://owasp.org/www-project-top-10-privacy-risks

OWASP Top 10 Privacy Risks project (*Some Examples*)

P2: Operator-sided Data Leakage

- Lack of awareness
- Poor access management
- Unnecessary copies of personal data

Dark archives

Shadow IT

...

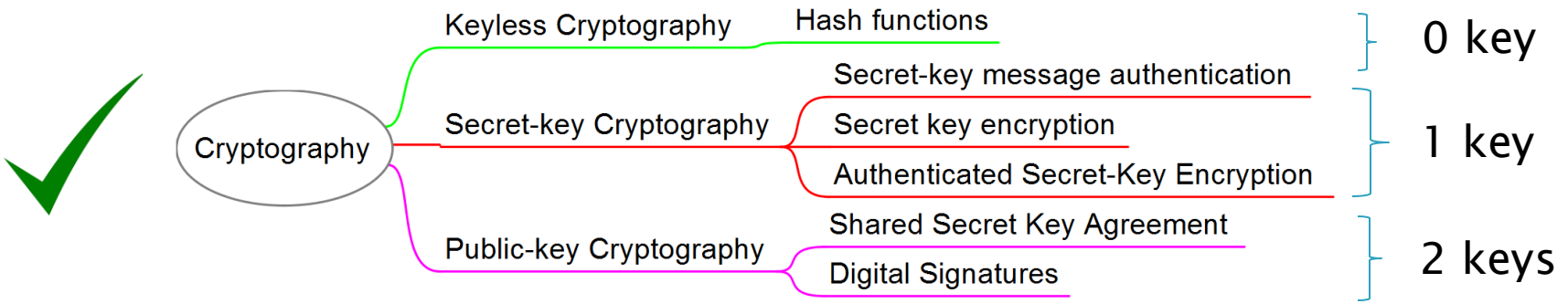
P5: Non-transparent Policies, Terms & Conditions

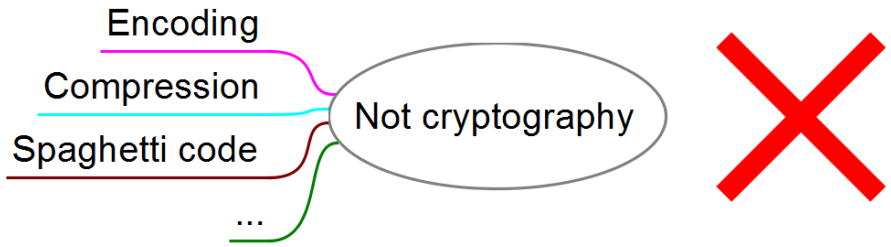
- Privacy Policies (PP), Terms & Conditions (ToU/ToS) are not up-to-date, inaccurate, incomplete or hard to find
  - Data processing is not explained sufficiently
  - Conditions are too long and users do not read them

Source: https://commonterms.org

Useful projects:
- Polisis (AI-powered summary of PP/ToS/ToU)
- https://tosdr.org (PP & ToS grades)

Overview of cryptography concepts





Cryptographic feature?

- Simply put: Using **Mathematics** to secure an application
- Cryptographic algorithms can generally be grouped by two criteria's:
  1. How much information must be supplied by the developer?
  2. What is the intended goal? (primitives to achieve):
     - Confidentiality? **(use encryption)**
     - Integrity? **(use signature)**
     - Authenticity? **(use signature)**
     - Non-repudiation? **(use signature)**
     - Deniability (or repudiation)? **(use signature)**

First Rule of Cryptography

Don't Implement it Yourself!

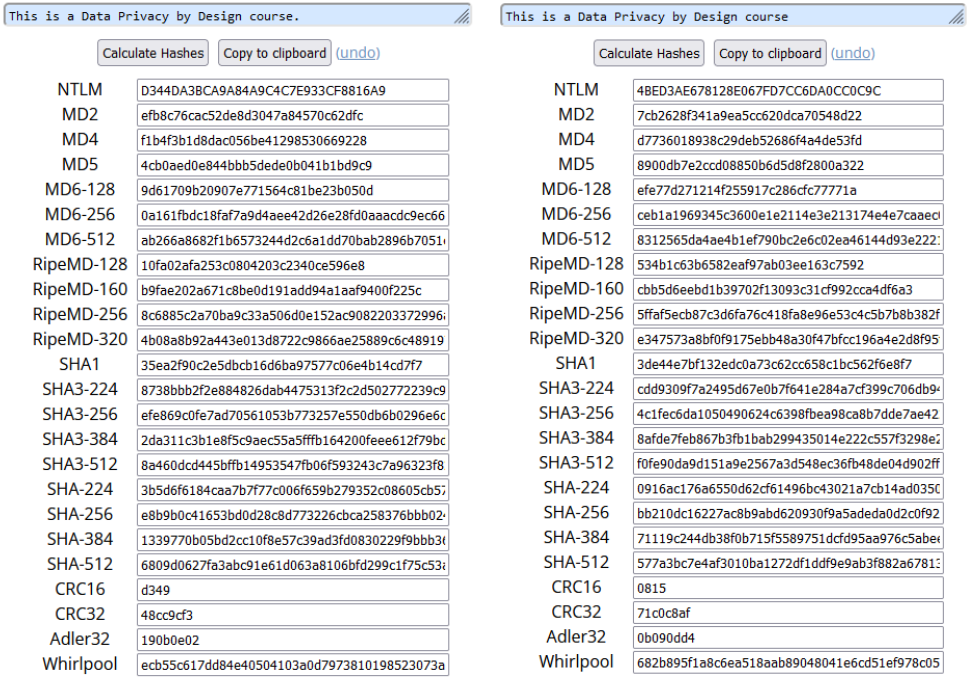
- Best left to the experts
  - Feel free to tinker
  - But don't deploy your experiments in production
- Always use a publicly scrutinized high-level crypto-library
- Crypto-library comparison example:

Name of implementation	Initiative	Main implementation language	Open-source software	Software license	Latest release
Botan	Jack Lloyd	C++	Yes	Simplified BSD	3.2.0 (October 9, 2023; 7 months ago[1]) [±]
Bouncy Castle	Legion of the Bouncy Castle Inc.	Java, C#	Yes	MIT License	**Java** 1.77 / November 13, 2023; 6 months ago[2] **Java LTS** BC-LJA 2.73.5 / March 1, 2024; 3 months ago[3] **Java FIPS** BC-FJA 1.0.2.4 / September 28, 2023; 8 months ago[4] **C#** 2.3.0 / February 5, 2024; 4 months ago[5] **C# FIPS** BC-FNA 1.0.2 / February 28, 2023; 15 months ago[6]

Source: https://en.wikipedia.org/wiki/Comparison_of_cryptography_libraries

Keyless Cryptography

- Hash function (one-way data transformations):
  - Accepts one input & returns a fixed-size output (depending on the algorithm)
  - Any change to the input will result in a drastically different hash output
  - Must not be reversed from hash output to the original message – **and that's the goal**



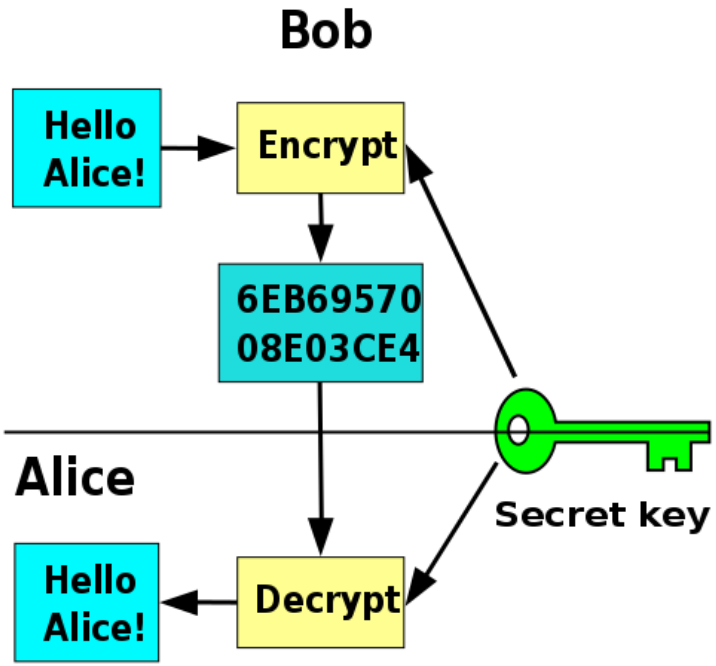
Source: https://www.browserling.com/tools/all-hashes

Crypto hashes & Password hashes (don't confuse them)

Simple Hashes	Password Hash (schemes)
• Fast • Only one input: The message • E.g., SHA, Whirlpool, …	• Intentionally slow • At least three inputs: 1. The password 2. A per-user salt 3. A cost factor (how expensive to make the computation) i.e., random/fixed iterations • E.g., Scrypt, Bcrypt, Argon2

Secret Key Cryptography

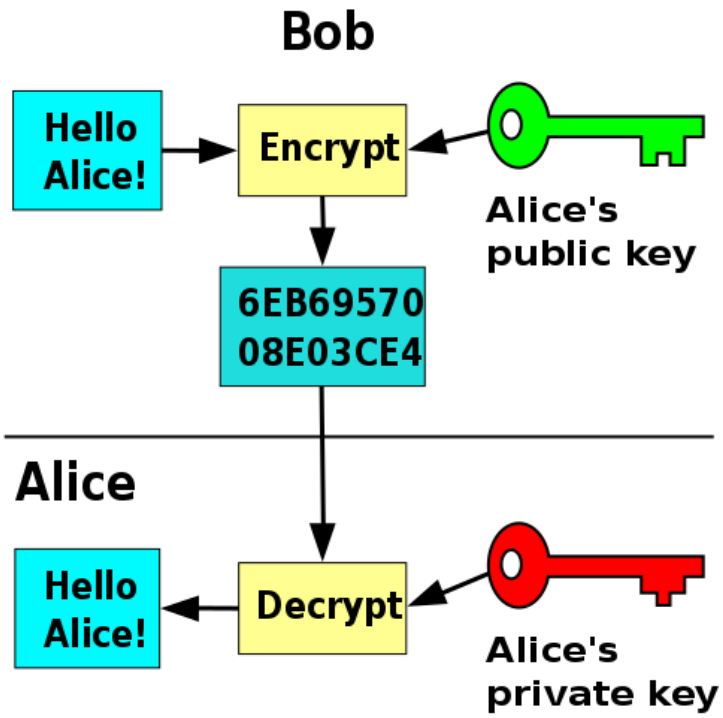
- Typically require two pieces of input: The message and a secret key
  - A secret key should be a unique string of random bytes
  - The secret key must be only known to sender and intended recipient, and nobody else!



Source: https://commons.wikimedia.org/wiki/File:Symmetric_key_encryption.svg

Public Key Cryptography

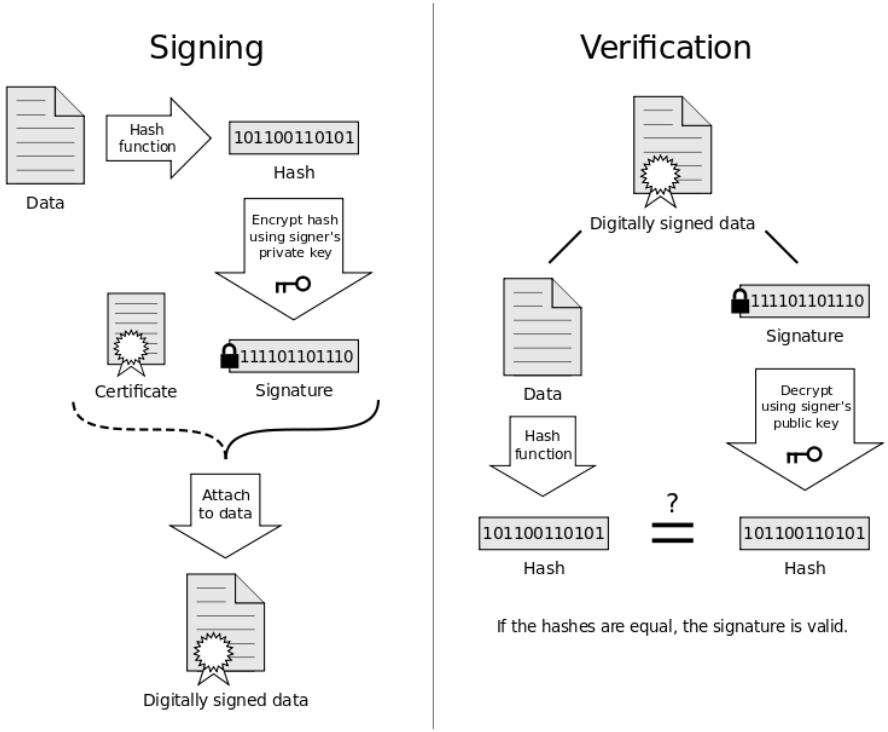
- Unlike secret key encryption, in public key cryptography, each participant has two keys (or a keypair):
  - **Private key**: never shared, used for:
    - Signing a message
    - Decrypting a message
  - **Public key**: mathematically related to the private key, shared with everyone
    - Used for encrypting a message
    - Used for verifying digital signatures



Source: https://en.wikipedia.org/wiki/Public-key_cryptography#/media/File:Public_key_encryption.svg

Digital Signatures

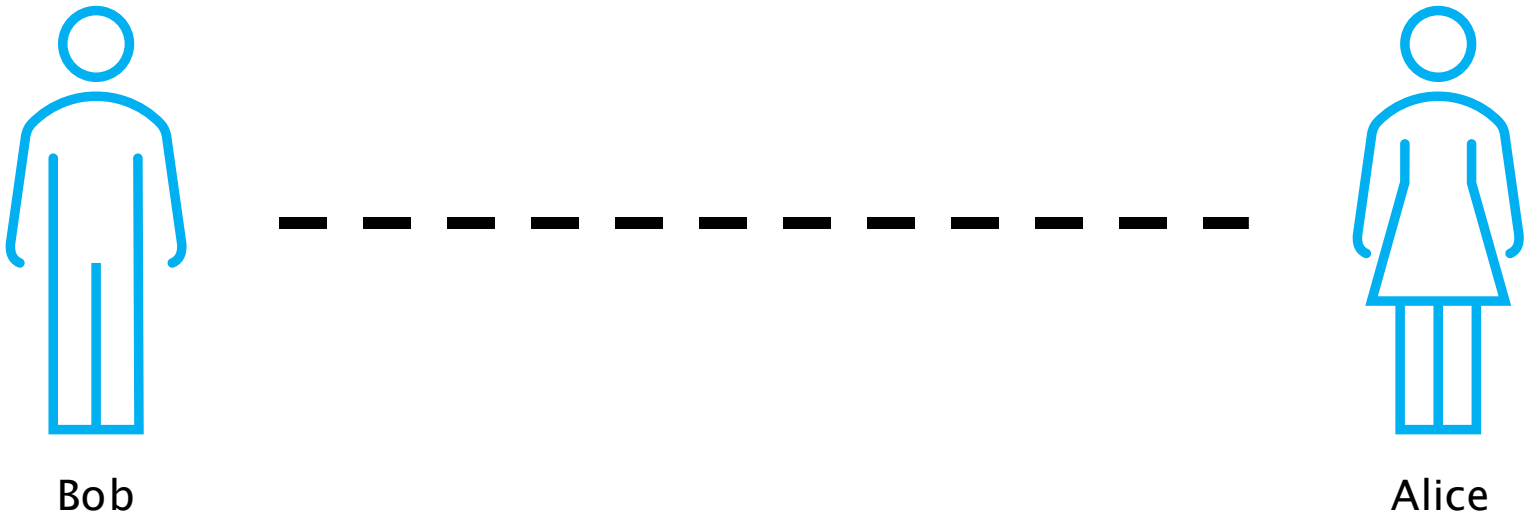
- A **digital signature** is calculated from a **message** and a **private key**:
  - Algorithm such as EdDSA (Edwards-curve Digital Signature Algorithm) or RSA (Rivest–Shamir–Adleman) are commonly used
  - Anyone else with a copy of respective **public key** can verify that a particular message was signed by someone's private key



Source:
https://commons.wikimedia.org/wiki/File:Digital_Signature_diagram.svg

Open exercise

- Propose a secure and private protocol enabling two parties to communicate:



Data masking

- Process of obfuscating original/sensitive data
- The two main categories include:

Anonymization

Information rendered anonymous, such that the data subject is no longer identifiable

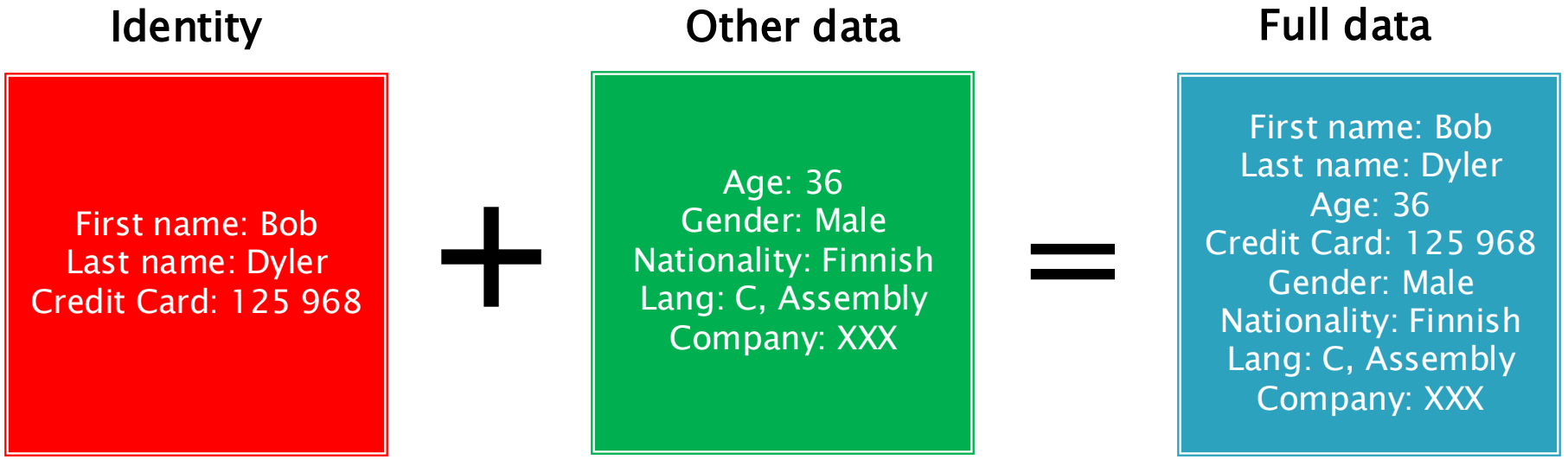
Pseudonymization

Information rendered neither anonymous nor directly identifying

Anonymization vs Pseudonymisation

Key difference	Anonymization	Pseudonymization
Concept	Data should not be re-identifiable	Data should allow for some form of re-identification
Data	Mainly used for sensitive personal information such as: Names, IDs (CC, ID, …), Addresses, Phone numbers, etc	Any data
Usage scenario	Mostly one-time (e.g., passing data to a 3rd party). Preferable in cases where Personal data (and the legal obligation to protect it) should no longer exist	Mostly multi-time (e.g., operational/ Transactional data). Preferable in cases when Personal data is needed at different stages of processing

Anonymization in a Glance



Unique identifiers: advisable to anonymize

Quasi-identifiers: advisable to pseudonymize

Pseudonymized data can be attributed when the identity is added to the data

Anonymization techniques, including noise addition and data aggregation, may lead to a reduction in the dataset's utility.
→ Maintaining the desired level of utility in the anonymized dataset is of paramount importance!

Pseudonymisation & Data masking (in general)

- Pseudonymization is used when re-identification is necessary for the purpose of processing:
  - When personal data is utilized for transactional operations (e.g., handling in a relational database)
- Data masking techniques (miscellaneous):
  - Scrambling/Obfuscation (e.g., Name: Bob -> KC0)
  - Encryption/Hashing (e.g., Name: Tyler -> 8cbx2)
  - Substitution and/or shuffling (e.g., Credit Card no. : 125 978 -> X25 798; X=1, 97->79)
  - Tokenization (e.g., Credit Card: 125 968 -> akjcn809)
  - Blurring (approximation): (e.g., Age: 36 -> Above 30)
  - …

Data masking techniques (categorization)

Category	Sub-category	Techniques	Application scenario
Anonymization	Randomization	Noise addition	Numeric data
		Permutation	Numeric data (high utility requirement)
		Differential privacy	Big data statistics
	Generalization	Aggregation	Big data statistics
		K-anonymity	
		L-diversity	
		T-closeness	
Pseudonymization		Encryption (AES256)	Data needs to be reversible
		Hash (HMAC-SHA256)	Fixed length value
		Tokenization	Keep data format such as ID

Source: Hands-On Security in DevOps – Tony Hsu (2018), Chapter 6 (Security Architecture and Design Principles): Data masking (Page 99)
Referenced from Art. 29 WP (now EPDP) opinion (0829/14/EN – WP216)

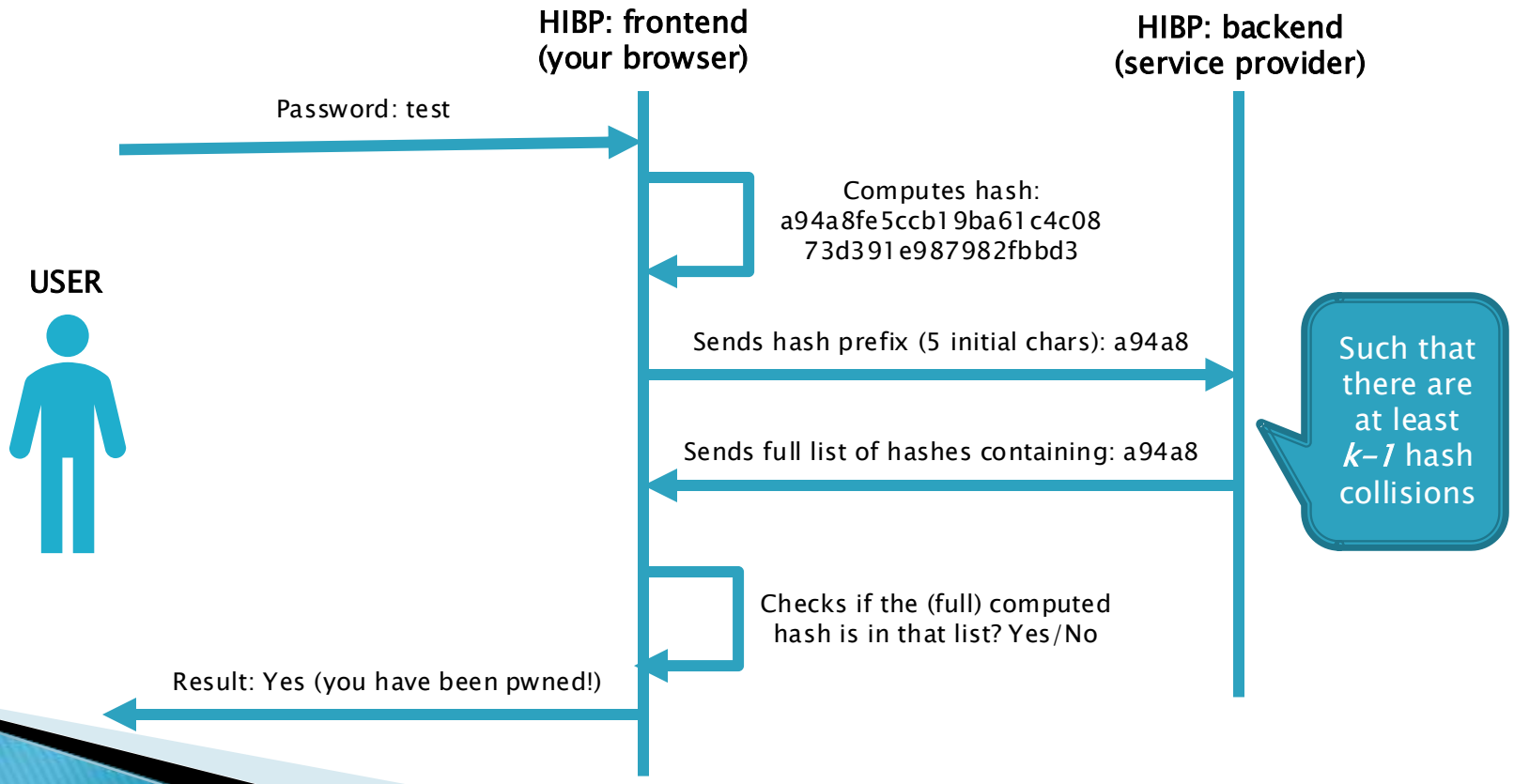
Data masking techniques (→ PETs)

- Doesn't offer a silver bullet
  - One data anonymization company, Aircloak, even acknowledges that true anonymization is extremely difficult: "as is the case with IT security, no 100% guarantee can be given, and often there is the **need for a risk assessment**" (source)
- Gazillion Anonymization techniques:
  - Often embodied as "**Privacy Enhancing Technologies**" (PETs):
    - Soft: 3rd parties can be trusted for data processing (through compliance control and audit), example technologies: differential privacy, SSL, etc
    - Hard: 3rd parties cannot be trusted, example technologies: onion routing, secret ballot, etc
- How can one assess the robustness of an anonymization technique? According to the Art. 29 WP (now EPDP) opinion (0829/14/EN WP216) on Anonymization Techniques, the criteria include:
  1. Single out an individual from a larger group
  2. Link different records related to the same individual
  3. Infer unknown information about an individual

Source: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf (alternative)

PET (soft): K-anonymity (range queries)

- If at least 'k' individuals share same quasi-identifier(s) in the same data set, then no individual can be uniquely traced
- E.g., HIBP (https://haveibeenpwned.com/Passwords) should not know your password in order to be able to tell if it was breached



Putting it all together

Don't implement the crypto. (algorithms) yourselves

+

Use appropriate crypto. key management procedures or architecture

+

Use valid (non-obsolete) ciphers & key lengths

+

Use reputed, publicly accepted and open-source cipher implementation

+

Keep crypto. libraries up-to-date

+

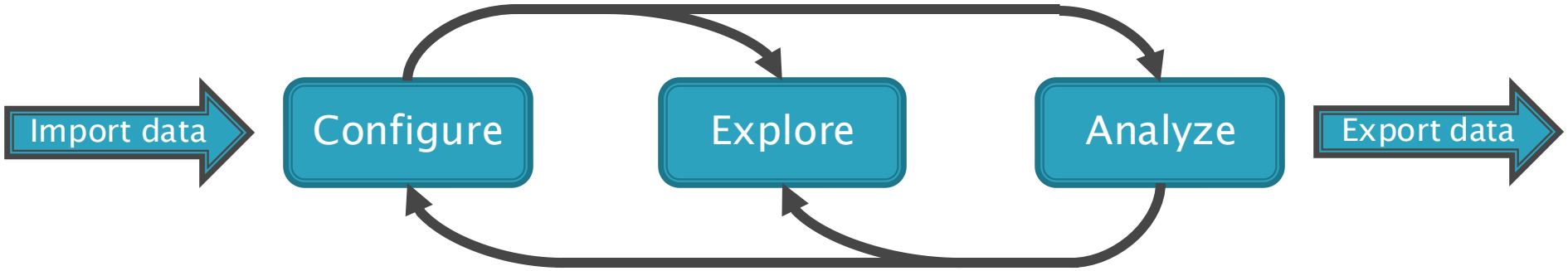
Perform testing & code review

+

Use appropriate Anonymization/pseudonymisation techniques

QUIZ…

Project: Data masking (ARX introduction)



- Iterative process to successively refine transformation until desired result is obtained

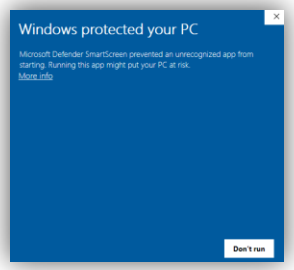
1. Define transformation model, privacy and coding model [wizard assistance]
2. Filter and analyze the solution space, and organize transformations [privacy and utility measures]
3. Compare and analyze input and output, regarding risks and utility

Project: Data masking (ARX exercise)

- Use ‘Arx.deidentifier.org’

[Opensource; Apache “License” Version 2.0]

1. Download (and install) Arx: https://arx.deidentifier.org/downloads/
   - If you get a warning that developer is not (or cannot be) verified, please ignore it.
   - For MAC users, if your device is based on ARM ISA, ARX tool will not run, as it is based on x86 ISA. Make sure Rosetta 2 is installed.



More info. -> **Run anyway**

2. Create a new project
3. Import **your project dataset**
4. Perform data masking (using **any transformation/technique of your choice**)
   GOAL: Mask direct identifiers while masking quasi-identifiers to a level that ensures an acceptable level of utility percentage
5. Generate ARX certificate and (optionally) export your project OUTPUT (anonymized) file (firstname_lastname) & upload it to the ‘Teams’ assignment section (using your EPITA account)
   - **NOTE**:
     - Do Not submit compressed files (.zip, .rar, etc).
     - Do Not submit sharepoint or external cloud links.

Deadline: See ‘Teams’ Assignment section

Project: Data masking (report)

Individual report:

- **Section 4**: Include your project ARX certificate
- **Section 5**: Explain the decisions taken in reference to the ARX certificate
  - **Input specifications**
    - Attributes and transformations
    - Configurations
  - **Output properties**:
    - Output data
    - Solutions
    - Transformations
    - Data quality models
    - Privacy models

NOTE: Collaborate on brainstorming as a group, but avoid copying and pasting your group members' work.

Deadline: See 'Teams' Assignment section

Lecture 2 ends here

- Course Slides: Go to MS Teams:
  'Data Privacy by Design'
  → "Files section"

- Send your questions by email OR via direct message using MS Teams

- Thank You!