Fundamentals of Privacy Engineering

Riccardo Scandariato
Institute of Software Security, TUHH, Germany
ric***do . scanda***to @ tuhh.de

Master Course “Secure Software Engineering”
Summer Semester 2022

Agenda

- Privacy and Security
- Privacy by Design
- Privacy Engineering
- LINDDUN
- Privacy Design Strategies
- Privacy-Enhancing Technologies

Privacy or Security?

Hypothetical Scenario:

- You download an app in your smartphone.
- You agree on the privacy policy and grant the app access to certain information stored in your phone:
  - *Contacts, Location, Photos, Videos …*

✗ **SCN-1**: Imagine that the app developer turns around you and sells your information to 3rd parties **without your permission**!

✗ **SCN-2**: Now imagine that the company developing the app suffers a breach, **exposing your information** to cybercriminals!

Are these scenarios describing **privacy** or **security** violations?
What is the difference? Is there even a difference?

Security

Security and **privacy** are two closely related concepts. However, there are some key differences:

- **Security** is a broader concept. It is concerned with the protection of *assets* against malicious attackers.
  - Preventing unauthorized access to assets via breaches or leaks regardless of who the unauthorized party is.
- Assets can be critical infrastructure, money, or **private information**.
- Security measures (e.g., firewalls, user authentication, network) are implemented to deter unauthorized access.

Security


PROVIDERS
STRATEGY
DETECTION
TRAILS
PROMOTIONS
HACKERS
PUBLIC
DEVICE
ISSUES
PASSWORDS
BUSINESS
LOGS
BREACH
STRATEGY
CYBER
SECURE
ATTACKS
VULNERABLE
MODEL
RESPONSE
COMPUTING
SECURITY
APPS
INSIDER
TRAILS
PROTECTION
PRIVACY
GOVERNMENT
MALWARE
SOFTWARE
SHARING
BROWSER
AUDIT
INFRASTRUCTURE
LEGAL
INTERNET
TECHNOLOGY
SECURE
NETWORK
ACCESS

Privacy

Privacy is more concerned with the responsible use of personal data:

- Ensure that data is *processed*, *stored*, and *shared* **in compliance** with a set of data protection mandates, principles, and rights:
  - *confidentiality*
  - *integrity*
  - *transparency*
  - *right to be informed, access, rectification, "to be forgotten" …*
- Privacy, unlike security, cannot stand on its own:
  - Privacy is achieved in the practice through **security controls**.
  - In short, **we cannot have privacy without security**!

Privacy or Security?

Let's go back to the hypothetical scenarios…

✗ **SCN-1**: Imagine that the app developer turns around you and sells your information to 3rd parties **without your permission**!

- This is a **privacy violation**

✗ **SCN-2**: Now imagine that the company developing the app suffers a breach, **exposing your information** to cybercriminals!

- This is again a **privacy violation**
- But it is also a **security failure**

In both cases, the developer **failed** to protect your **privacy**

Privacy by Design

Like security, privacy is also seen as a quality attribute of software.

✗ It is often an "after thought" instead of a priority!

At its core, **Privacy-by-Design (PbD)** consists of incorporating privacy into networked data systems and technologies **by default:**

- An organizational priority.
- Embedded into every standard, protocol, and process.

PbD is translated into 7 **foundational principles**:

1. **Proactive, not reactive; preventive not remedial:** Anticipate, identify, and prevent invasive events before they happen. Do not wait for risks to materialize!

Privacy by Design

2. **Privacy as the default setting:** Ensure personal data is automatically protected in all information systems or business processes, without requiring any further action.
3. **Privacy embedded into design:** Privacy mechanisms should not be add-ons, but fully integrated components of a system.
4. **Full functionality – Positive-Sum, not Zero-Sum:** Privacy and security should not compete against other legitimate interests and design objectives. They should be embedded into the system without impairing its full functionality.
5. **End-to-end security – Full lifecycle protection:** All data should be securely retained as needed and destroyed when no longer needed.

Privacy by Design

6. **Visibility and transparency – keep it open:** Assure stakeholders that business practices and technologies are operating according to objectives and subjects to independent verification.
7. **Respect for user privacy – keep it user-centric:** Individual privacy interests must be supported by strong privacy defaults, appropriate notice, and user-friendly options.

Many of these principles are embedded in the EU GDPR/DSGVO ⚖️

2. [1] The controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed. [2] That obligation applies to the amount of personal data collected, the extent of their processing, the period of their storage and their accessibility. [3] In particular, such measures shall ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons.

[General Data Protection Regulation (GDPR) - Article 25]

Privacy Engineering

It is a field of study concerned with the **systematic elicitation** and **implementation** of privacy requirements in socio-technical systems.

- PbD = "what to do" ⇒ privacy engineering = "how" to do it.
- Engineers are key players since they are responsible for devising the technical architecture of the system and creating the code.

Typical sources of privacy requirements are:

1) **Accepted privacy definitions:** For different authors, privacy may be a matter of *non-intrusion*, *seclusion*, *limitation*, *control*, etc.
   - Different theories provide different conceptual frameworks.
2) **User concerns**: Activities such as *data transfer*, *storage*, and *processing* can trigger privacy concerns among the stakeholders.

LUNDDUN

There are several PE methods defining activities that introduce privacy at different stages of software development life cycle:

- *ProPAn, PRIPARE, STRAP, QTMM,* ***LINDDUN****…*
- Overall, these PE methods define steps and a collection of software artifacts that support them (e.g., DFDs, threat catalogs, etc.).

LINDUNN is a systematic framework that employs threat modeling for assessing the privacy of information systems:

- ✓ Consists of **6 steps**. The first 3 steps are *problem-oriented* as they aid the identification of threats.
- ✓ The last 3 steps are *solution-oriented* as they seek to translate threats into mitigation actions and strategies.
- ✓ Can be applied **multiple times** at different stages of the life cycle.

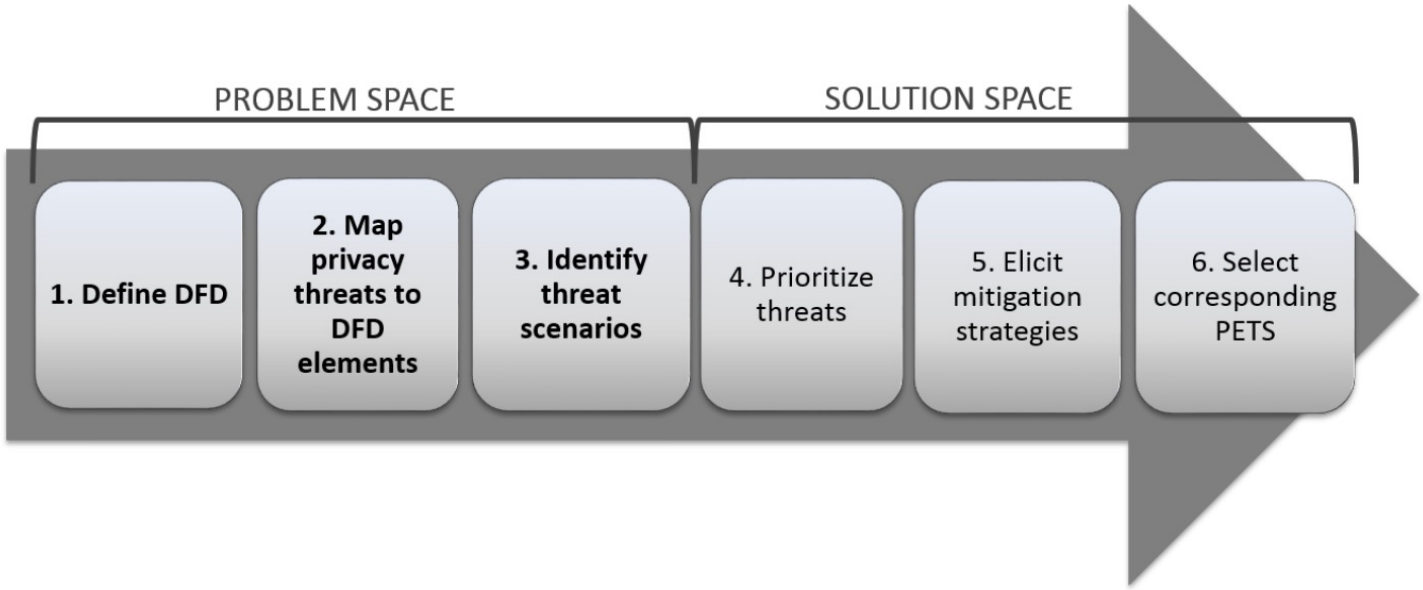
LUNDDUN

LINDDUN is **model-based** ⇒ Uses DFDs for representation and analysis

- Each element of the DFD is thoroughly examined for privacy threats.

LINDDUN is **knowledge-based** ⇒ Provides a threat catalog

- Contains common attack paths for a set of privacy threat categories: ***L****inkability,* ***I****dentifiability,* ***N****on-repudiation,* ***D****etectability,* ***D****isclosure of information,* ***U****nawareness,* ***N****on-compliance* ⇒ ***LINDDUN***



Step 1: Define DFD

A DFD is describes how information moves across the system using 4 types of building blocks:

- **Process (P)**: a unit of work that operates the data.
- **Data flow (DF)**: a named flow of data through a system of processes.
- **Data store (DS)**: a logical repository of data (passive container).
- **External entity (E)**: a source or destination of data, such as a system, users, or third-party services.

Optionally, trust boundaries can be added to the DFD to indicate places in which parties with different privileges interact.

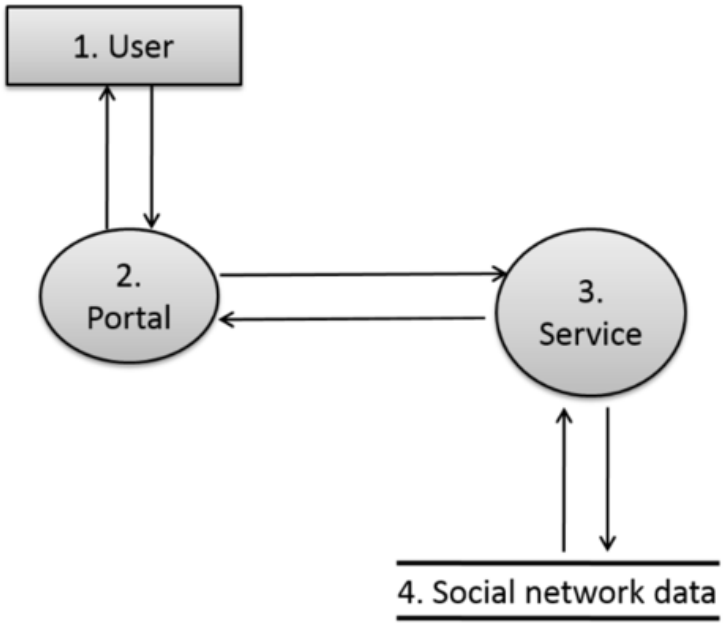
Step 1: Define DFD

Guidelines for DFD elaboration from requirements:

1. *Start with a "level 0" DFD:*
   - *One **main process** representing the system.*
   - *Connect this process to the **actors/users** of the system and the **external entities** (e.g., 3rd-party services and components).*
2. *Decompose the process into one or more **data stores** and **internal processes:***
   - *Processes communicating to external entities (facades, portals, websites).*
   - *Processes responsible for the databases like repository processes (Remember: databases are passive containers).*
   - *The internal operation of the system will require multiple processes in order to represent its complexity.*
3. *Add dataflows to **connect** all the above DFD elements.*

Step 1: Define DFD

Example: A social network platform in which users interact through a web portal. The portal forwards the users' requests (e.g., add friends, share content, etc.) to a service that manages them. Eventually, such a service stores the necessary information inside a database.



Step 2: Mapping the DFD to LINDUNN threat categories

In LINDDUN, each **DFD element type** is potentially subject to **specific privacy threats** grouped around 7 high-level categories.

Mapping Template

Threat categories	E	DF	DS	P
Linkability	X	X	X	X
Identifiability	X	X	X	X
Non-repudiation		X	X	X
Detectability		X	X	X
Disclosure of information		X	X	X
Unawareness	X			
Non-compliance		X	X	X

Each DFD element is subject to certain privacy threats, and the nature of the potential privacy threat is determined by the DFD element type.

Can "unawareness" be a threat to a datastore?
Why? Why not?

Step 2: Mapping the DFD to LINDUNN threat categories

Linkability

Being able to sufficiently distinguish whether 2 IOI (items of interest) are linked or not, even *without* knowing the actual identity of the subject of the linkable IOI.

Not being able to hide the link between two or more actions/identities/pieces of information.

Linkability can result in severe privacy issues only when linkable data leads to identification or inference:

- **Identification**: A data subject can be recognized by linking several (pseudo-)anonymous data (e.g., street name + gender + age).
- **Inference**: We can deduce relationships from certain related properties leading to severe cases of *discrimination* (e.g., people living in a certain neighborhood are prone to particular diseases).

Step 2: Mapping the DFD to LINDUNN threat categories

Identifiability

Being able to sufficiently identify the subject within a set of subjects (i.e. the anonymity set).

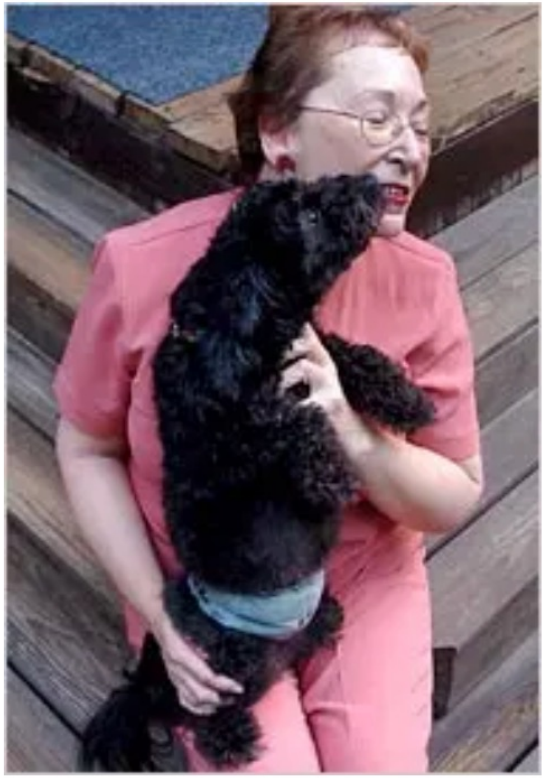
Not being able to hide the link between the identity and the IOI (an action or piece of information).

Identifiability is often a consequence of linking data to the same subject:

- Data are considered *de-identified* when no identifiers (e.g., social security number, full name and address, birth date) are stored.
- Pseudo-identifiers (e.g., birth year (instead of birthdate), city (instead of full address), etc.) can also lead to an identification.

⇒ The more information is linked together, the smaller the anonymity set will be. **Can ratings of a song/movie lead to identifiability?**

Step 2: Mapping the DFD to LINDUNN threat categories



"No. **4417749** conducted hundreds of searches over a three-month period on topics ranging from "numb fingers" to "60 single men" to "dog that urinates on everything."

"…search by search, click by click, the identity of AOL user No. 4417749 became easier to discern."

"…AOL removed the search data from its site over the weekend and apologized for its release." ☺

Thelma Arnold's identity was betrayed by AOL records of her Web searches, like ones for her dog, Dudley, who clearly has a problem.
Erik S. Lesser for The New York Times

https://www.nytimes.com/2006/08/09/technology/09aol.html

Internet users can be easily **identified** by **compiling** their **search queries**.

Step 2: Mapping the DFD to LINDUNN threat categories

Non-repudiation*

Having irrefutable evidence concerning the occurrence or non-occurrence of an event or action. [3]

*An attacker may be interested on demonstrating that a user has said, done, or knows something. This threat is a **security goal** as well.

Detectability

An attacker can sufficiently distinguish whether an item of interest (IOI) exists or not.

Disclosure of Information

Exposing information to someone not authorized to see it.

Unawareness

Not understanding the consequences of sharing personal information in the past, present, or future.

Step 2: Mapping the DFD to LINDUNN threat categories

Non-compliance

Not following the (data protection) legislation, the advertised policies or the existing user consents

The system as a **data controller must** determine the purposes for which and the means by which personal data is processed ⇒ **privacy policy**

- A **policy** specifies a set of rules with respect to data protection. These are general rules determined by the systems' stakeholders.
- The system should allow users to **grant** or **revoke** permissions over the collection and processing of their personal data

It is very important to ensure that policies are properly implemented, and users' consent is acknowledged and respected.

- This threat is closely related to **legislation**. LINDDUN alone **cannot guarantee** full compliance. Regulatory threats should be analyzed by legal experts!

Step 2: Mapping the DFD to LINDUNN threat categories

Example: We compute a list of generic threats to the modelled system using (i) ***LINDUNN mapping template****, and (ii) the system's DFD elements. We mark with gray those threats deemed as irrelevant.*

Computed Threats

	Threat target	L	I	N	D	D	U	N
Data Store	Social network DB	1	4			7		10
Data Flow	User data stream (user – portal)	2	5			8		10*
	Service data stream (portal – service)							10*
	DB data stream (service – DB)							10*
Process	Portal							10*
	Social network service							10*
Entity	User	3	6				9	

The threats we will take into consideration are marked with a number. We use **10*** to indicate noncompliance threats affecting the whole system.

Step 3: Elicit + document threats

This is the core execution step in LINDDUN but also the most tedious and it is thus divided into three sub-activities.

3.1 Refining threats via threat tree patterns

- For each 'X' in the mapping table (see step 2), LINDDUN provides a list of concrete threats (organized into **trees**) that need to be considered.
- The tree shows the specific **preconditions (vulnerabilities)** for a given threat category that can be exploited in a privacy-attack scenario.
- We will examine the branches of the trees corresponding to the **threats computed in Step 2** to identify potential privacy violations.

LINDDUN provides a **threat tree catalog** on its website for supporting this step

Step 3: Elicit + document threats

3.2 Documenting assumptions

Certain leaf nodes (or even entire branches) may not be deemed relevant to the system under analysis and will thus not be considered.

- Assumptions are explicit or implicit choices to trust an element of the system (e.g., human, piece of software) to behave as expected.
- Assumptions should be documented for instance as a free text linked to the corresponding **misuse case** (see next) for traceability purposes.

3.3 Documenting threats using a threat template

Applicable threats (i.e., those deemed relevant for the system under analysis) should be documented as **misuse cases**.

- Misuse case: A use case from the misactor's point of view.
- LINDUNN provides a **template** for this purpose.

Threat description template

The proposed misuse case structure is described below (*optional fields are indicated with **):

Summary: provides a brief description of the threat.

Assets, stakeholders and threats*: describes the assets being threatened, their importance to the different stakeholders, and what the potential damage is if the misuse case succeeds.

Primary misactor: describes the type of misactor performing the misuse-case. Possible types are insiders, people with a certain technical skill, and so on. Also, some misuse cases could occur accidentally whereas other are most likely to be performed intentionally.

Basic Flow: discusses the normal flow of actions, resulting in a successful attack for the misactor.

Alternative Flows*: describes the other ways the misuse can occur.

Trigger*: describes how and when the misuse case is initiated.

Preconditions*: precondition that the system must meet for the attack to be feasible.

Leaf node(s)*: refers to the leaf node(s) of the threat tree(s) the threat corresponds to.

Root node(s)*: refers to the root node(s) of the threat tree(s) that were examined for the threat.

DFD element(s)*: lists all DFD elements to which this threat is applicable.

Remarks(*): Although optional, related assumptions should be mentioned here.

Step 3: Elicit + document threats

MUC 1 – Linkability of social network database (data store)

Summary: Data entries can be linked to the same person (without necessarily revealing the persons identity).

Assets, stakeholders, threats: The user's Personal Identifiable Information (PII)

- Data entries can be linked to each other revealing the persons identity.
- The misactor can build a profile of a user's online activities (interests, actives time, comments, updates, etc.).

Primary misactor: skilled insider/skilled outsider.

Basic Flow:

1. The misactor gains access to the database.
2. The misactor can link the data entries together and possibly re-identify the data subject from the data content.

Step 3: Elicit + document threats

MUC 1 – Linkability of social network database (data store)

Alternative Flow:

1. The misactor gains access to the database.
2. Each data entry is linked to a pseudonym.
3. The misactor can link the different pseudonyms together (linkability of entity).
4. Based on the pseudonyms, the misactor can link the different data entries.

Trigger: by misactor, can always happen.

Preconditions:

- No or insufficient protection of the data store.
- No or insufficient data anonymization techniques or strong data mining applied.

Step 4: Prioritization of threats (risk assessment)

Before moving forward and looking for suitable **mitigation actions**, the identified threats must be prioritized.

- Time and budget limitations make treating all threats unfeasible.
- Only the most important ones will be addressed in the requirements specification and in the design of the solution.
- Risk assessment techniques provide support for this stage.

In general, **risk** is calculated as a function of the **likelihood** of the attack scenario and its **impact (or consequence level)**.

⇒ The LINDDUN framework is independent from the risk assessment technique that is used.

⇒ The analyst is **free to pick** the technique of choice (e.g., OWASP).

Step 5: Elicit mitigation strategies | Step 6: PETs

Misuse cases can help extracting a set of (positive) system requirements:

- Some requirements are “straight-forward” and correspond to a set of **elementary privacy objectives**.
  - LINDDUN provides a mapping table supporting this task.
- More detailed **mitigation strategies** may be necessary in the practice
  - LINDDUN provides a taxonomy of mitigation strategies along with their corresponding requirements and solutions.

⇒ Mitigation strategies (or tactics) capture a **high-level view** of common techniques used in the practice to prevent privacy threats.

At the final step (Step 6), privacy strategies are translated into a set of **Privacy Enhancing Technologies** (PETs)

Step 5: Elicit mitigation strategies | Step 6: PETs

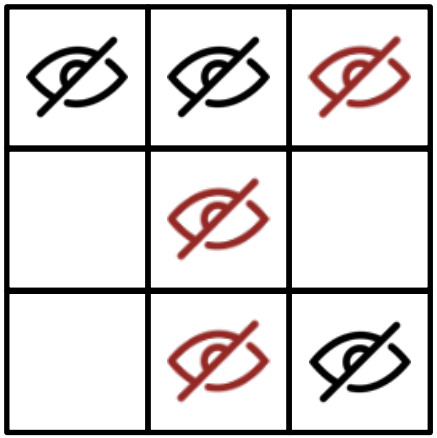
LINDDUN threats	Elementary privacy objectives
Linkability of (E,E)	Unlinkability of (E,E)
Linkability of (DF,DF)	Unlinkability of (DF,DF)
Linkability of (DS,DS)	Unlinkability of (DS,DS)
Linkability of (P,P)	Unlinkability of (P,P)
Identifiability of (E,E)	Anonymity / pseudonymity of (E,E)
Identifiability of (E,DF)	Anonymity / pseudonymity of (E,DF)
Identifiability of (E,DS)	Anonymity / pseudonymity of (E,DS)
Identifiability of (E,P)	Anonymity / pseudonymity of (E,P)
Non-repudiation of (E,DF)	Plausible deniability of (E,DF)
Non-repudiation of (E,DS)	Plausible deniability of (E,DS)
Non-repudiation of (E,P)	Plausible deniability of (E,P)
Detectability of DF	Undetectability of DF
Detectability of DS	Undetectability of DS
Detectability of P	Undetectability of P
Information Disclosure of DF	Confidentiality of DF
Information Disclosure of DS	Confidentiality of DS
Information Disclosure of P	Confidentiality of P
Content Unawareness of E	Content awareness of E
Policy and consent Non-compliance of the system	Policy and consent compliance of the system

Straight-forward Requirements

Privacy Design Strategies

Privacy and data protection by design can be achieved through a set of design strategies:

- Minimize: System designers should ensure that only the **minimal necessary** personal info is collected.
- Hide: Confidentiality of the data is ensured either by **encrypting**, **pseudonymizing** or **anonymizing** data in transit or storage.



Privacy Design strategies

- Separate: Personal data should be stored and processed in a **distributed** way.
- Aggregate: Storage of *individualized data* should be **restricted** as much as possible and replaced by *aggregates* when possible.
- Inform: Respondents should be made **informed** what information about them is being collected and processed for which reasons.
- Control: Respondents should be able to **consult**, **modify** and **delete** the information about them.
- Enforce: Privacy policies should be put in place and **enforced**.
- Demonstrate: Data controllers ought to **document** all collection and analysis processes conducted on personal information.

Example: Data Minimization

A *recruiter*, **Rob**, calls a *potential employee*, **Ed**, for a job interview:

- **Rob** wants to know whether Ed is willing to work for the salary Rob's company is offering.
- However, **Ed** does not want to reveal his true salary requirements!!!
- **Sally**, a consultant, is the intermediary between Ed and Bob. She will be in charge of answering Rob's and Ed's salary questions.

Solution: Ed and Bob give Sally the salary offer and requirements. She makes the comparison and reports the result to them.

Question: Is this solution correct? Can you spot any issues?

Example: Data Minimization

The proposed solution has the following issues:

✗ Sally could judge Ed's salary (*"oh is he earning THAT much?!").*

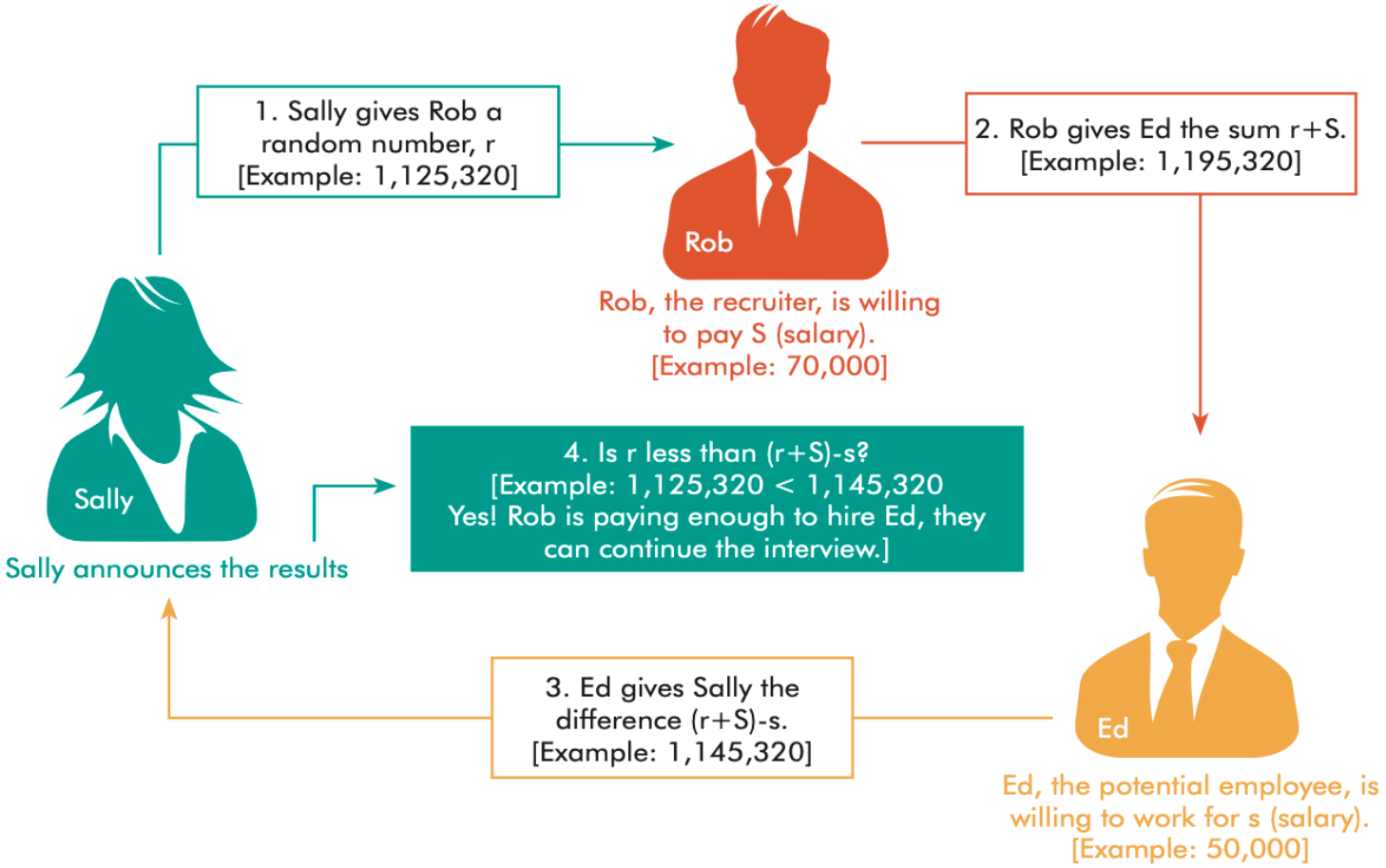
✗ She could also sell the information of Rob's offer to his competitor.

⇒ Therefore, Ed and Rob are **reluctant to give their data** to Sally ☹

Solution 2: Sally does not need to know the actual figures, but just **which of them is larger**.

1. Sally generates a random number "rnd" and gives it to Rob.
2. Rob add the offer amount to "rnd" and passes it to Ed.
3. Ed subtracts his salary requirement and passes the result to Sally.
4. Sally compares the number from Ed against the original "rnd".
   - *If larger, the offer is bigger* ☺
   - *If not, then is smaller* ☹

Example: Data Minimization



Privacy Enhancing Techniques (PETs)

Privacy strategies are sometimes not enough on their own:

- 👎 Often too broad and vague.
- 👎 Must be refined to be effectively used in practice.

Although PETs allow a many interpretations, they can be described as:

- ✓ Technologies that make use of **privacy design strategies.**
- ✓ Implement **privacy design patterns** or consider **protection goals.**

Example: K-Anonymity

Problem: *How to publicly release a database without compromising individual privacy?*

- Remove unique identifiers (e.g., name, social security number)? ☺
- The triple <*birthdate, gender, zip code*> suffices to uniquely identify at least 87% of US citizens in publicly available databases ☹

K-Anonymity: Attributes are suppressed or generalized until each row is identical to K-1 other rows ⇒ *k-anonymous Dataset*

⇒ In the worse case, the released dataset narrows-down an individual entry to a group of k individuals.

- Method 1: **Suppression** (replace individual attributes with a *).
- Method 2: **Generalization** (replace attributes with a broader category).

PETs: K-Anonymity

First name	Last Name	Age	Race
Harry	Stone	34	African-American
John	Kane	36	Caucasian
Beatrice	Stone	34	African-American
John	Delgado	22	Hispanic

This database an be 2-Anonymized with **suppression**:

First name	Last Name	Age	Race
*	Stone	34	African-American
John	*	*	*
*	Stone	34	African-American
John	*	*	*

PETs: K-Anonymity

Overall, we can guarantee k-anonymity by replacing every cell with an *:

- This renders the database useless!!!
- The cost of a K-Anonymous solution is the number of *'s introduced.
- A **minimum cost** k-anonymity solution suppresses the fewest number of cells necessary to guarantee k-anonymity.

Share your opinion

Go to menti.com – xxxx yyyy

Questions ?

More questions at a later time?

ric***do . scanda***to @ tuhh.de