Privacy Threats

Nella lezione precedente, abbiamo visto che non è affatto semplice definire la privacy, in quanto vi sono molte definizioni di quest'ultima ed in particolare, Solove (importante studioso, che ha fornito una delle prima definizioni di privacy) ha sostenuto: "É inutile perdere tempo a definire che cos'è la privacy, pensiamo invece a quali sono i modi in cui la privacy può essere compromessa." → Solove, quindi, ha prodotto una tassonomia (ovvero una classificazione) di tutti i possibili attacchi e threads, che possono andare a compromettere le informazioni personali di un individuo. In particolare, le categorie di attacco, sono raggruppate in base alle tre fasi principali, con cui un'organizzazione tipicamente processi i dati personali di un individuo, ovvero:

1. la fase di **Information Collection** → rappresenta la fase iniziale, in cui l'organizzazione raccoglie le informazioni personali dei propri clienti e/o dei propri dipendenti;
2. la fase di **Information Processing** → in cui le informazioni raccolte dalla fase precedente, vengono elaborate e memorizzate;
3. la fase di **Information Dissemination** → in cui le informazioni potrebbero essere condivise con altre organizzazioni.

Solove, in realtà, considerava un'altra categoria di attacchi (che non è associata a nessuna delle tre fasi viste appena sopra), ma che riguarda una serie di attacchi, in cui la privacy (intesa come sfera personale dell'individuo) può essere compromessa, ad esempio dal Governo o da enti pubblici.

A questo punto, analizziamo in maniera approfondita le diverse categorie di attacco associate a ciascuna delle tre fasi con cui un'organizzazione processa i dati. Abbiamo, quindi, che:

- la prima categoria di attacchi riguarda la fase di Information Collection, ovverosia la fase in cui i dati vengono raccolti da un'organizzazione. In particolare, in questa prima fase, vi sono due categorie di attacchi:
    - attacchi di **Surveillance** → in cui tipicamente le organizzazioni, o meglio dire gli attaccanti, raccolgono informazioni riguardo agli individui e ne profilano le attività (ovvero che vanno a creare un profilo delle attività di ogni individuo). In particolare, vi sono due tipologie di attacchi di Surveillance:
        1. attacchi di Surveillance che coinvolgono le **organizzazioni**;

2. attacchi di Surveillance che coinvolgono i **Governi**.

   Un esempio di attacco di Surveillance è: Entro il 2020 doveva essere obbligatorio installare nelle abitazioni gli Smart Meters (ovvero i contatori intelligenti), con l'obiettivo di risparmiare sulla bolletta. Vi è stato, però, un grande dibattito sull'installazione di questi contatori intelligenti, perchè vi erano delle problematiche relative alla privacy, in quanto tutte le organizzazioni che fornivano l'installazione di questi dispositivi, potenzialmente potevano monitorare tutte le attività che facevamo all'interno di casa e capire cosa facevamo all'interno delle nostre abitazioni (per esempio, capire quando ci svegliamo e andiamo a dormire, quando rientriamo da lavoro).

   Un altro esempio di attacchi di Surveillance, che in questo caso riguarda il Governo americano, riguarda il gioco Angry Birds, il quale raccoglieva informazioni sull'età dell'utente, la posizione geografica e l'orientamento sessuale.

- attacchi di **Interrogation** → in cui gli attaccanti tentano di portare gli utenti a rivelare informazioni personali (un esempio, infatti, sono gli attacchi di phishing).

- la seconda categoria di attacchi riguarda la fase di Information Processing, ovverosia la fase in cui i dati vengono analizzati ed elaborati. In particolare, in questa seconda fase, vi sono diverse tipologie di attacchi:
  - attacchi di **Aggregation** → attacchi che compromettono la proprietà di linkability, ovvero l'attaccante è in grado di combinare diverse informazioni, relative ad un utente, e inferire nuove informazioni personali dell'utente (per esempio: immaginiamoci di andare al supermercato e alla cassa utilizziamo la carta fedeltà del supermercato. In questo modo, il supermercato vede quali prodotti acquistiamo e può fornirci delle offerte ad hoc, rispetto al nostro stile di vita e abitudini);
  - attacchi di **Identification** → in cui l'attaccante è in grado di determinare a quale utente appartiene una determinata informazione (per esempio: l'attaccante è in grado di associare un record di un DB dell'ospedale, ad uno specifico utente);
  - attacchi di **Insecurity** → comprendono tutti quegli attacchi, che consistono in un comportamento negligente dell'organizzazione che raccoglie i dati. Comprendono, quindi, la negligenza nel proteggere le informazioni memorizzate da fughe di notizie e accessi impropri;

    - attacchi di **Secondary use** → in cui i dati, che inizialmente sono stati raccolti dall'organizzazione (per raggiungere un determinato scopo), successivamente vengono utilizzati dall'organizzazione per effettuare altre tipologie di analisi, senza che l'utente venga a conoscenza di ciò;
    - attacchi di **Exclusion** → rappresentano tutte quelle situazioni, in cui l'utente non è a conoscenza del fatto, che i propri dati vengono raccolti e vengono analizzati da un'organizzazione ed essendo che non ne è a conoscenza, l'utente non può partecipare al loro trattamento e utilizzo.
- la terza categoria di attacchi riguarda la fase di Information Dissemination, ovverosia la fase in cui i dati vengono condivisi con altre organizzazioni. In particolare, in questa terza fase, vi sono diverse tipologie di attacchi:
    - attacchi di **Breach of confidentiality** → l'organizzazione infrange la promessa di mantenere riservate le informazioni personali di un utente, dato che non vengono implementate misure di sicurezza adeguate, e di conseguenza, condividere tali informazioni con altre organizzazioni, senza informare l'utente (un esempio di questa tipologia di attacco, è l'attacco di data breach che ha subito Equifax, la quale si occupava di calcolare il credit-score dei cittadini americani e dei cittadini britannici);
    - attacchi di **Disclosure** → situazione, in cui informazioni sensibili e/o personali dell'utente vengono condivide, causando un danno alla reputazione dell'utente stesso verso gli altri utenti;
    - attacchi di **Exposure** → consistono in tutti quegli attacchi, in cui un attaccante oppure un'altra organizzazione rende pubblici fotografie, video e documenti, che rivelano informazioni che gli utenti volevano mantenere private;
    - attacchi di **Increased accessibility** → consistono in tutte quelle situazioni, in cui volontariamente o non involontariamente, un'organizzazione oppure un attaccante rendono pubbliche informazioni personali di determinati utenti;
    - attacchi di **Blackmail** → consistono in tutte quelle situazioni, in cui principalmente gli attaccanti minacciano un individuo oppure un'organizzazione di rivelare informazioni personali dell'utente stesso oppure informazioni personali dei dipendenti e/o dei clienti dell'organizzazione;
    - attacchi di **Appropriation** → consistono in tutte quelle situazioni, in cui un attaccante utilizza informazioni personali, di un altro individuo, per raggiungere i propri scopi (vi sono molti attacchi di Appropriation resi

possibili, grazie alle informazioni che pubblichiamo sui nostri social networks);

    - attacchi di **Distortion** → consistono nella diffusione di informazioni false o fuorvianti su determinati individui o su determinate organizzazioni, che possono provocare un danno alla reputazione degli individui o delle organizzazioni.

- l'ultima categoria di attacchi, è quella che non è legata ad una fase del trattamento dei dati personali. In particolare, abbiamo due tipologie di attacchi:
    - attacchi di **Intrusion** → riguardano atti invasivi, che disturbano la tranquillità o la solitudine dell'individuo, quindi vanno a compromettono la privacy quotidiana degli individui (per esempio: i vicini di casa che installano delle telecamere di sorveglianza che puntano verso il nostro giardino. Un altro esempio di questo attacco è il Cyberbullismo);
    - attacchi di **Decisional interference** → comportano l'intrusione del Governo, nelle decisioni della persona interessata in merito ai suoi affari privati.

Privacy Enhancing Technologies (PETS)

"**Come facciamo a difenderci da tutte le tipologie di attacco visti precedentemente?**" Esistono diverse tipologie di soluzioni, che possiamo adottare per ridurre il rischio, che la privacy degli utenti venga compromessa. Queste soluzioni possono essere applicate a diversi livelli all'interno dell'architettura informatica di un sistema. Tipicamente, abbiamo soluzioni che possono essere applicate:

- al livello di rete;
- oppure possono essere applicate per proteggere i dati, quando essi vengono memorizzati nel Cloud o su un Database su un Server dell'organizzazione;
- oppure possiamo avere soluzioni lato Client.

Le soluzioni possono essere applicate sia dall'utente stesso (quindi lato Client), sia dall'organizzazione. La prima categoria di tecnologie, che andiamo a considerare, può essere chiamata **Data Protection Technologies** → l'obiettivo di queste tecnologie, è quello di aiutare l'organizzazione a dimostrare, che quest'ultima rispetta tutti i principi/regole legati alla protezione dei dati personali. Per esempio, il regolamento europeo per la protezione dei dati personali, prevede che l'organizzazione che raccoglie i dati e li processa, debba garantire la sicurezza dei

dati → questo si traduce nel prevedere e adottare misure di sicurezza adeguate, come per esempio:

- la cifratura dei dati, sia quando essi sono memorizzati sia quando vengono trasmessi;
- autenticazione e autorizzazione dei dipendenti che trattano dati personali;
- principio del “right to be forgotten”, ovvero che le organizzazioni devono mantenere una copia dei dati degli utenti, solamente fino a quando ne hanno bisogno e poi dovrebbero eliminarli automaticamente;
- registrazione sicura degli accessi ai dati e delle attività di elaborazione a fini di revisione.

L’altra categoria di tecnologie, ha a che fare con la proprietà di Awareness (=consapevolezza) e infatti, prendono il nome di **User Awareness Technologies** → queste tecnologie, quindi, riguardano il fatto che gli utenti dovrebbero avere il controllo **di chi** ha accesso ai propri dati personali. Ovviamente, avere controllo di chi ha accesso ai propri dati personali, implica che l’utente deve essere consapevole:

- **di chi** ha accesso ai propri dati;
- **con chi** (questi dati) vengono condivisi.

In questa categoria di tecnologie, quindi, rientrano una serie di applicazioni, che da un lato fanno visualizzare all’utente, gli altri utenti che hanno accesso ai propri dati e dall’altro lato, invece, gli forniscono gli strumenti per restringere l’accesso ai dati a solo gli utenti fidati (un esempio di queste applicazioni è **Privacy Bird**, il quale ha come obiettivo quello di facilitare gli utenti nella lettura delle politiche di privacy).

Un’altra categoria di tecnologie, riguarda le **tecniche di Anonimizzazione** → queste tecniche hanno l’obiettivo di proteggere l’utente, da attacchi che possono:

- andare a ricostruire l’identità dell’utente;
- andare a compromettere la proprietà di linkability.

In particolare, le tecniche di anonimizzazione che andremo a vedere, garantiscono la proprietà di anonimità e di linkability, quando i dati vengono memorizzati all’interno di un Database. Esse (ovvero le tecniche) garantiscono, che un attaccante (che ottenga l’accesso al Database anonimizzato) non sia in grado di risalire all’identità di uno dei record del Database.

Infine, un’ultima categorie di tecnologie, riguarda le tecnologie utilizzate per garantire la soft privacy, ovverosia l’utente è responsabile di proteggere i propri dati personali,

prima di condividerli con un'organizzazione.

Data Anonymisation

Abbiamo visto, che le organizzazioni raccolgono una grande quantità di dati sulle attività che facciamo online e spesso questi dati vengono condivisi per supportare analisi e ricerche. In questo contesto, vogliamo evitare che un attaccante, che ha accesso al Database che è stato condiviso, possa inferire informazioni sugli individui, i cui dati sono contenuti all'interno del Database. In particolare, vogliamo evitare che l'attaccante sia in grado di capire a chi appartiene un determinato record all'interno del Database e da questo possa inferire altre informazioni personali sull'individuo. Vediamo, allora, un esempio:

ID	QID			SA
Name	Zipcode	Age	Sex	Disease
Izzy	47677	29	F	Ovarian Cancer
Rose	47602	22	F	Ovarian Cancer
Bob	47678	27	M	Prostate Cancer
John	47905	43	M	Flu
Alice	47909	52	F	Heart Disease
Fred	47906	47	M	Heart Disease

Medical Data

Siamo nel contesto di un ospedale, il quale mantiene traccia, all'interno di un Database, quali sono i pazienti che vengono ospitalizzati. In particolare, il Database memorizza le seguenti informazioni:

- nome paziente;
- codice postale;
- età;
- sesso;
- patologia per cui il paziente è stato ospitalizzato.

Immaginiamoci, poi che questo Database voglia essere condiviso con un'organizzazione che fa analisi statistiche, per capire effettivamente quali sono i principali motivi per cui i pazienti vengono ricoverati. Quello che vogliamo evitare, è che un attaccante (che accede al Database) possa capire che un record appartenga ad uno specifico individuo, in modo tale che l'attaccante non riesca ad inferire informazioni sensibili (come ad esempio: la patologia) dell'individuo.

Per capire come possiamo proteggerci da queste tipologie di attacco, dobbiamo capire le tipologie di attributi, che possiamo trovare all'interno di un Database mantenuto dall'organizzazione → tipicamente, all'interno di un Database possiamo avere:

- **Explicit identifiers** → sono tutte quelle informazioni, che portano all'identificazione diretta di un utente all'interno del Database. Alcuni esempi di explicit identifiers sono:
  - nome;
  - cognome;
  - numero di passaporto;
  - codice fiscale;

  quindi si tratta di identificatori univoci associati (all'interno del Database) ad uno specifico utente e di conseguenza, permettono di tracciare tutte le informazioni e tutte le attività dell'utente.
- **Quasi-identifiers** → possono portare all'identificazione **indiretta** di un individuo, i cui dati sono contenuti all'interno di un Database. Alcuni esempi di attributi quasi-identifiers sono:
  - data di nascita;
  - età;
  - sesso;
  - codice postale;
  - numero di telefono.

  quindi, si tratta di informazioni che sono comunemente note e di conseguenza, l'attaccante andando a combinare i valori degli attributi quasi-identifiers (presenti nel Database) con i vari degli attributi quasi-identifiers presenti online, è in grado di inferire l'identità dell'utente;
- **Sensitive attributes** → questi attributi sono ciò di cui i ricercatori hanno bisogno per condurre le analisi e di conseguenza, essi vengono sempre rilasciati direttamente ai ricercatori. Alcuni esempi sono:
  - stipendio;
  - malattie.

Un primo approccio, quindi, che si può adottare per evitare questi attacchi di re-identificazione, consiste nell'andare a mascherare gli **Explicit identifiers**. In particolare, vi sono due tecniche che possiamo utilizzare:

1. una tecnica per gli Explicit identifiers che assumono un **valore numerico** → questa tecnica prende il nome di **Tokenization**;
2. una tecnica per gli Explicit identifiers che sono una **stringa** → questa tecnica prende il nome di **Substitution**.

La tecnica di **Tokenization** consiste nel trasformare un valore numerico, univoco per l'utente e contenuto nel Database, in un altro valore numerico da cui non si può risalire al valore originario → quindi, anche se l'attaccante ottenesse accesso al Database con valore tokenizzato, non sarebbe in grado al valore originare.

La tecnica, invece, di **Substitution** consiste nell'andare a sostituire una stringa univoca per l'utente, con un'altra stringa presa da un Database di stringhe comuni della nazione (per esempio: il nome e il cognome di un paziente del Database dell'ospedale, può essere sostituito con un nome preso dal Database di nomi comuni italiani, quale per esempio Mario Rossi) → in questo modo, andiamo a spezzare il legame che c'è tra l'identità dell'utente e l'Explicit identifier.

> 💡 Mascherare, però, solamente gli Explicit identifiers non è sufficiente per proteggersi dagli attacchi di re-identificazione.

Introduciamo, allora, il concetto di **K-Anonymity** → ovvero che possiamo ripartire un Database in classi di equivalenza, le quali non sono altro che insiemi di record, dove il valore degli attributi Quasi-identifiers è esattamente uguale per tutti i record → questo viene fatto, perchè se si hanno almeno **k records** indistinguibili l'uno dall'altro, per quanto riguarda il valore degli attributi Quasi-identifiers, l'attaccante ha **probabilità 1/k** di re-identificare l'utente a cui è associato uno specifico record nel Database. Vediamo un esempio:

Original Database

Name	Zipcode	Age	Disease
Hilary	47677	29	Heart Disease
Jenny	47602	22	Heart Disease
Bob	47678	27	Heart Disease
Izzy	47905	43	Flu
John	47909	52	Heart Disease
Fred	47906	47	Cancer
Sam	47605	30	Heart Disease
Carl	47673	36	Cancer
Sarah	47607	32	Cancer

Released Database

Zipcode	Age	Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
4790*	≥40	Flu
4790*	≥40	Heart Disease
4790*	≥40	Cancer
476**	3*	Heart Disease
476**	3*	Cancer
476**	3*	Cancer

vediamo che siamo andati ad eliminare il nome, ovverosia l'identificatore esplicito, e successivamente abbiamo suddivido il Database in tre classi di equivalenza, in cui il valore degli attributi Quasi-identifiers (età e codice postale) è esattamente uguale per tutti i records presenti nella stessa classe di equivalenza → **siccome nel Released Database, abbiamo almeno tre records che hanno lo stesso valore degli attributi Quasi-identifiers è un Database 3-anonymous**.

Nella pratica, possiamo implementare il concetto di **K-Anonymity** attraverso due tecniche, che possiamo utilizzare in combinazione per ripartire il Database in classi di equivalenza, dove gli attributi Quasi-identifiers hanno lo stesso valore. In particolare, queste due tecniche sono:

- la tecnica di **Generalizzazione** → consiste nell'andare a prendere un valore di un attributo Quasi-identifiers e lo sostituiamo con un valore meno specifico, che però mantiene la stessa informazione (o meglio dire lo stesso livello di informazione) dell'attributo originario. Per esempio, abbiamo il codice postale che è formato da 5 numeri, di cui:
  - il primo numero rappresenta la regione;
  - il secondo e il terzo numero rappresentano la città;
  - il quarto e il quinto numero rappresentano la zona specifica della città.

  Possiamo generalizzare il valore del codice postale, andando ad eliminare le ultime due cifre e mantenendo, quindi, solamente le informazioni relative alla città in cui si trova l'utente. Il livello successivo di generalizzazione, potrebbe essere di eliminare anche il secondo e il terzo numero, in modo tale da mantenere solamente le informazioni relative alla regione in cui vive l'utente → **chiaramente, maggiore è il livello di generalizzazione, minori saranno le informazioni che mantengo dell'attributo Quasi-identifier originale e quindi l'utilità del valore potrebbe venire meno ai fini dell'analisi**.
- la tecnica di **Soppressione** → consiste nell'andare ad eliminare interamente il valore dell'attributo Quasi-identifier dalla classe di equivalenza, in quanto non vi è altro modo di generare una classe di equivalenza, in cui i valori degli attributi sono esattamente tutti identici. Chiaramente, per eseguire questa tecnica dobbiamo bilanciare due aspetti:
  - l'utilità del valore dell'attributo Quasi-identifier, ai fini dell'analisi;
  - la relazione che vi è tra l'attributo sensitivo su cui si vuole fare l'analisi e l'attributo Quasi-identifier.

Un altro aspetto da tenere in considerazione, quando anonimizziamo i dati e che effettivamente rende il processo di anonimizzazione complicato, è la **conoscenza dell'attaccante**.

Vediamo un esempio di utilizzo delle tecniche di generalizzazione e soppressione. Partiamo, allora, dal nostro Database originale e poi applichiamo le due tecniche:

Name	Zipcode	Age	Sex	Disease
Hilary	47677	29	F	Heart Disease
Jenny	47673	22	F	Heart Disease
Bob	47678	27	M	Heart Disease
Izzy	47905	43	F	Flu
John	47909	52	M	Heart Disease
Fred	47906	47	M	Cancer
Sam	47605	30	M	Heart Disease
Carl	47602	36	M	Cancer
Sarah	47607	32	F	Cancer

Database originale

Zipcode	Age	Sex	Disease
47677	10-29	F	Heart Disease
47673	10-29	F	Heart Disease
47678	10-29	M	Heart Disease
47909	50-69	M	Heart Disease
47906	50-69	M	Cancer
47605	30-49	M	Heart Disease
47602	30-49	M	Cancer
47607	30-49	F	Cancer

generalizziamo l'età, sostituendo il valore dell'età con degli intervalli di età di 20 anni.

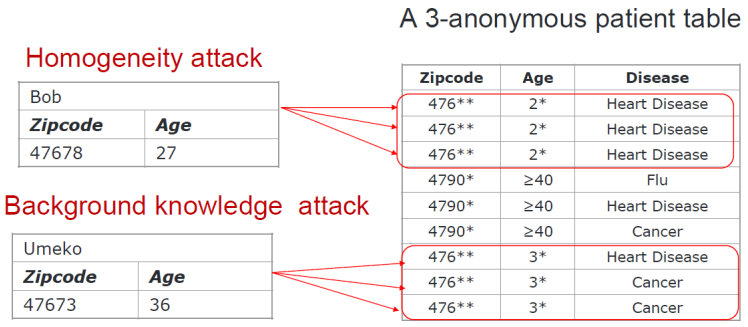
Zipcode	Age	Sex	Disease
*	10-29	*	Heart Disease
*	10-29	*	Heart Disease
*	10-29	*	Heart Disease
*	50-69	M	Heart Disease
*	50-69	M	Cancer
*	30-49	*	Heart Disease
*	30-49	*	Cancer
*	30-49	*	Cancer

andiamo a sopprimere il valore dello zipcode, in quanto sono tutti valori univoci e del sesso. In questo modo, otteniamo un Database 2-anonimo.

L'anonimità, però, non è una soluzione perfetta, in quanto la K-Anonymity non garantisce la privacy (ovvero non va a proteggere l'attributo sensitivo su cui viene fatta l'analisi) se:

- i valori sensibili in una classe di equivalenza non presentano diversità;
- l'attaccante ha una conoscenza di base.

A sostegno di ciò, immaginiamoci di essere un attaccante e sappiamo che le informazioni dell'utente Bob sono contenute all'interno del Database e in più so (perchè ad esempio, sono il suo vicino di casa) che Bob ha 27 anni e il suo codice postale è 47678:



avendo queste informazioni, quindi, l'attaccante può inferire che Bob ha una malattia al cuore, nonostante le informazioni esplicite siano state eliminate e nonostante gli attributi Quasi-identifiers siano stati generalizzati → questo attacco prende il nome di **Homogeneity Attack**, che consiste nel rivelare il valore dell'attributo sensitivo su cui tipicamente vogliamo condurre un'analisi.

Un altro esempio è: l'attaccante conosce a priori, che l'utente Umeko ha 36 anni e il suo codice postale è 47673. Inoltre, è risaputo che i giapponesi (come Umeko) hanno una bassa probabilità di avere problemi al cuore. L'attaccante, quindi, può inferire che Umeko ha il cancro → questo attacco prende il nome di **Background knowledge attack**.

Questo ha portato a definire un'altra nozione di anonimizzazione, ovverosia si va a definire la **L-diversity** → essa impone, che all'interno di una classe di equivalenza, dobbiamo avere almeno ***L*** valori diversi che vengono assunti dall'attributo sensitivo. In questo modo, si evitano situazioni in cui si ha una classe di equivalenza, in cui il valore dell'attributo sensibile è uguale per tutti i record della classe di equivalenza. Essenzialmente, quindi, soddisfiamo la nozione di L-diversity, se abbiamo una classe di equivalenza in cui il valore dell'attributo sensibile assume almeno *L* valori distinti. Vediamo un esempio:

....	Disease
	HIV
	HIV
	HIV
	HIV
	HIV
	HIV
	HIV
	Bronchitis
	Pneumonia

Abbiamo che il valore sensitivo, ovvero la malattia, assume 3 valori diversi.

Però, questa nozione di L-diversity non protegge da attacchi, che possono inferire il valore dell'attributo sensitivo relativo ad un determinato utente, perchè **non tiene conto della distribuzione dei valori all'interno di una classe di equivalenza** (per esempio, nell'esempio sopra, se noi sappiamo che all'interno del Database vi è l'utente Bob, possiamo inferire che con molta probabilità abbia l'HIV, dato che 8 records su 10 riguardano l'HIV) → per questo motivo, è stata introdotta una nozione più forte di diversity, che non impone solamente che i valori siano diversi all'interno di una classe di equivalenza, ma impone anche che **l'entropia** della classe di equivalenza sia maggiore o uguale al logaritmo dell'entropia della classe di equivalenza → **con questa definizione, quindi, ci assicuriamo che anche la distribuzione dei valori, all'interno di una classe di equivalenza, sia diversa**. In particolare, questa nozione la definisco nel seguente modo:

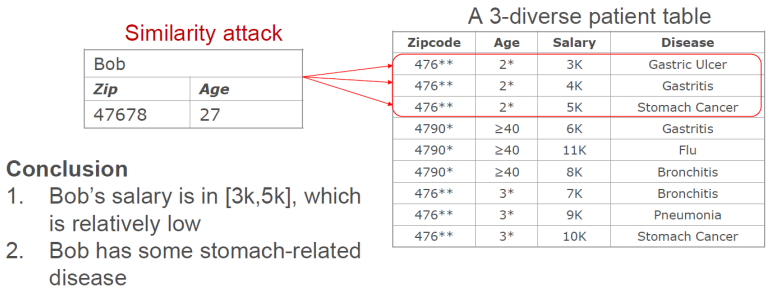
- assumo che l'insieme **s**, sia l'insieme dei valori dell'attributo sensitivo;
- considero la frazione dei records, ovverosia il numero degli elementi all'interno del Database, che possono assumere uno specifico valore del dominio **s**;
- successivamente calcolo l'entropia, che è data da: la sommatoria (per tutti i valori che può assumere l'attributo sensitivo) della frazione dei records moltiplicata per il logaritmo della frazione dei records.

Entropy ℓ-diversity. The entropy of an equivalence class E is defined to be

$$Entropy(E) = -\sum_{s \in S} p(E, s) \log p(E, s)$$

in which S is the domain of the sensitive attribute, and $p(E, s)$ is the fraction of records in E that have sensitive value s.

Una volta che abbiamo calcolato l'entropia di una classe di equivalenza, vado ad imporre che tale valore dell'entropia sia maggiore o uguale al logaritmo dell'entropia della classe di equivalenza. Anche con questa definizione, però, abbiamo dei problemi e per capirli vediamo un esempio:



abbiamo il nostro solito Database, che è sia 3-anonymous sia 3-diverse. Supponiamo, sempre, che l'attaccante conosca che l'utente Bob abbia 27 anni e il suo codice postale è 47678. Con queste informazioni, l'attaccante riesce ad individuare la classe di equivalenza a cui appartiene Bob e quindi può inferire, che Bob ha problemi di stomaco e abbia un salario tra i 3000-5000€ → capiamo, allora, che **la nozione di L-diversity non tiene conto delle relazioni semantiche che ci sono tra i dati** (dato che tutte le malattie della classe di equivalenza di Bob, riguardano l'apparato digerente). Inoltre, **la nozione di L-diversity non considera la distribuzione complessiva dei valori sensibili**.

Per questi motivi, è stata proposta un'ulteriore nozione di anonimizzazione, ovvero la **nozione di t-closeness** → l'idea alla base di questa nozione, è che dobbiamo assicurarci che **la distribuzione dei valori, all'interno di una classe di equivalenza, sia simile (= si avvicini) alla distribuzione dello stesso attributo sensitivo a livello dell'intero Database**. Vediamo un esempio:

Zipcode	Age	Salary	Disease
476**	2*	3K	Gastric Ulcer
476**	2*	4K	Gastritis
476**	2*	5K	Stomach Cancer
4790*	≥40	6K	Gastritis
4790*	≥40	11K	Flu
4790*	≥40	8K	Bronchitis
476**	3*	7K	Bronchitis
476**	3*	9K	Pneumonia
476**	3*	10K	Stomach Cancer

abbiamo un Database sia 3-anonymous sia 3-diverse e per applicare la nozione di t-closeness, dobbiamo:

- calcolare la distribuzione dei valori sensitivi, a livello dell'intero Database;

- confrontare la distribuzione dei valori sensitivi, a livello dell'intero Database, con la distribuzione dei valori sensitivi all'interno delle classi di equivalenza;
- determinare il discostamento tra la distribuzione dei valori sensitivi, a livello dell'intero Database, e la distribuzione dei valori sensitivi all'interno delle classi di equivalenza.

Per riuscire a fare questo, la nozione di t-closeness utilizza il concetto di **Earth Mover Distance** → questo concetto va a quantificare qual è l'effort che bisogna impiegare, per trasformare la distribuzione dei valori sensitivi all'interno delle classi di equivalenza, nella distribuzione dei valori sensitivi, a livello dell'intero Database.

Prendiamo, allora, l'esempio di prima e applichiamo nella pratica, la nozione di t-closeness:

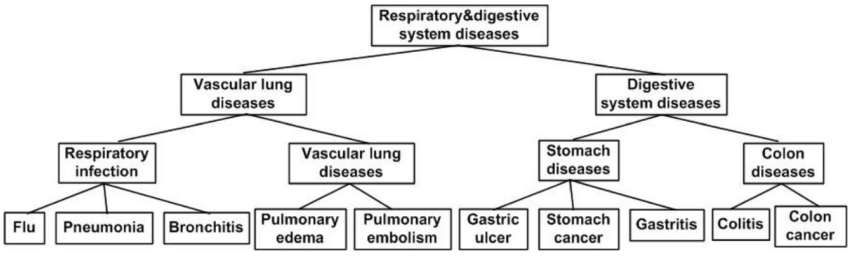
Disease
Gastric Ulcer
Gastritis
Stomach Cancer
Gastritis
Flu
Bronchitis
Bronchitis
Pneumonia
Stomach Cancer

{Gastric Ulcer, Gastritis, Stomach Cancer, Flu, Bronchitis , Pneumonia}

P1={Gastric Ulcer, Gastritis, Stomach Cancer} D[P1,Q]=0.5

P2={Gastric Ulcer, Stomach Cancer, Pneumonia} D[P2,Q]= 0.278

vediamo che abbiamo le due classi di equivalenza *P1* e *P2* e come primo passo, andiamo a creare una gerarchia (un albero) di valori, in cui abbiamo alle foglie i valori dei nostri attributi sensibili del Database e ai livelli più alti, abbiamo le categorie a cui appartengono i valori degli attributi sensibili:



Per calcolare la distanza tra la distribuzione dei valori che assume l'attributo sensitivo, all'interno di una classe di equivalenza, e la distribuzione dei valori che assume l'attributo sensitivo a livello dell'intero Database, dobbiamo andare a calcolare, sulla base della gerarchia, la distanza tra i valori che abbiamo nella classe di equivalenza, con gli stessi valori che abbiamo nella distribuzione a livello dell'intero Database. Per esempio: prendiamo i valori della classe di equivalenza *P1*. Devo calcolare la distanza che ho tra i valori di P1 con i valori rimanenti del Database (quindi: Flu, Bronchitis e Pneumonia). "**Come facciamo a calcolare la**

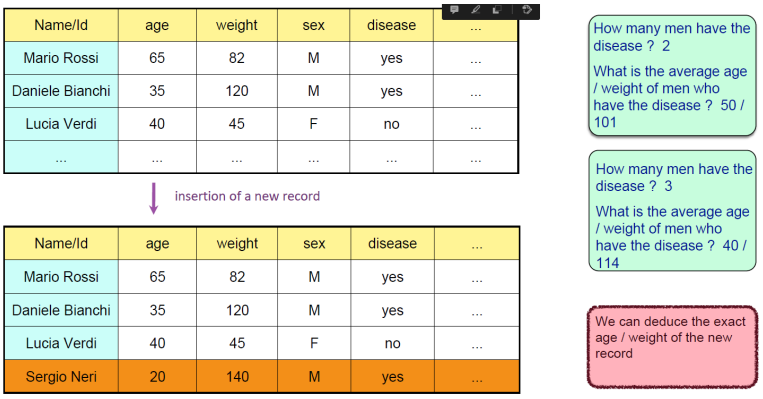
distanza tra due valori?” Per calcolare la distanza, devo guardare qual è il nodo genitore (in comune) per entrambi i valori. Individuato il nodo genitore, in comune tra i due valori, vado a calcolare l’altezza del sotto-albero e lo divido per l’intera altezza dell’albero. Per esempio: l’altezza totale dell’albero è 3 e supponiamo di voler calcolare la distanza tra Gastric ulcer e Flu. Il nodo in comune tra questi due valori è il nodo radice e di conseguenza la distanza tra questi due valori è: altezza sotto-albero/altezza albero = 3/3 = 1. Supponiamo, ancora, di voler calcolare la distanza tra Flu e Pneumonia, che hanno lo stesso nodo padre. In questo caso la distanza è: 1/3. Per calcolare, quindi, la distanza della classe di equivalenza *P1*, devo sommare tutte le distanze dei valori di cui è composta e poi divido il valore per il numero totale di valori, a livello del Database (nel nostro caso 6).

Capiamo, allora, che anonimizzare i dati non basta per proteggerli, perchè abbiamo visto che ogni nozione di anonimizzazione protegge **parzialmente** i dati, da attacchi che possono portare ad un’identificazione dell’utente oppure a rivelare informazioni personali e sensibili dell’utente.

Se non possiamo anonimizzare i dati, quindi, possiamo cambiare l’**approccio di condivisione** dei dati, ovvero: invece di anonimizzare i dati e condividerli con organizzazioni (che poi possono fare analisi per scopi di ricerca), possiamo mantenere il **Database in house** e consentire ai ricercatori di effettuare le ricerche sul Database stesso. Per capire se effettivamente questo approccio funziona, guardiamo un esempio:

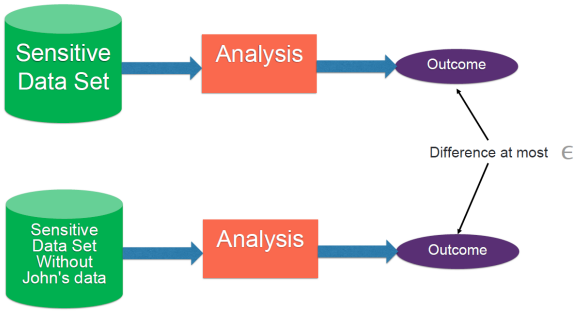
Name/Id	age	weight	sex	disease	...
Mario Rossi	65	82	M	yes	...
Daniele Bianchi	35	120	M	yes	...
Lucia Verdi	40	45	F	no	...
...	...	...	...	...	...

abbiamo un Database, in cui non viene tenuta traccia del nome della malattia, ma sappiamo solamente se il paziente ha la malattia oppure no. Inoltre, supponiamo di concedere ai ricercatori di effettuare solamente delle query aggregate sul Database (quindi, ad esempio, i ricercatori possono chiedersi: Quante persone hanno la malattia? Qual è la media delle persone che hanno la malattia?). **Teoricamente**, allora, se concediamo di effettuare solamente delle query aggregate, è molto meno probabile che l’attaccante venga a conoscenza di informazioni personali o sensibili degli utenti del Database. Diciamo teoricamente, perchè:



Capiamo, allora, che la restrizione solamente alle query aggregate non è sufficiente, in quanto anche queste query possono far trapelare informazioni sugli individui.

Per questo motivo, è stata introdotta la nozione di **Differential Privacy** → l'idea alla base della Differential privacy è il risultato di un'analisi, che viene fatta su un Database che contiene informazioni sensibili, non rivela alcuna informazione relativa agli utenti del Database. Per riuscire a fare ciò, la Differential privacy rende il risultato dell'analisi sul Database che **contiene** le informazioni sensibili degli utenti e l'analisi sul Database che **non contiene** le informazioni sensibili degli utenti, praticamente identiche e indistinguibili (differenziano per un parametro **epsilon**, il quale quantifica qual è la similarità del risultato dell'analisi eseguita sul Database con le informazioni personali con il risulto dell'analisi eseguita sul Database senza le informazioni personali)→ questo fa si, che l'attaccante non riesca ad inferire nulla sugli utenti del Database.



> 💡 Notiamo un particolare: la Differential privacy non garantisce il fatto, che il rischio della privacy degli utenti sia pari a zero (ovverosia eliminato), bensì il rischio della privacy degli utenti è pari al rischio che avrebbero analizzando i dati di altri utenti. In altre parole, **il rischio della privacy degli utenti è limitato alle informazioni, che un attaccante può inferire sull'utente, sulla base delle informazioni di altri utenti**.

A questo punto, capiamo solamente l'intuizione che sta dietro alla nozione di Differential Privacy → in particolare, vengono presi i valori degli attributi sensitivi (come per esempio: i valori della malattia) e viene aggiunto del **rumore**, dove il rumore segue una distribuzione probabilistica. Per implementarla nella pratica, viene utilizzato il concetto di **sensitività globale** → supponiamo di avere una funzione *F* e vogliamo ottenere la versione di tale funzione che soddisfa la nozione di Differential Privacy. Per riuscire a fare ciò, dobbiamo:

- calcolare la **sensitività** della funzione, che viene definita come la variazione massima, che possiamo avere nel calcolo della funzione *F*, quando la funzione *F* è applicata su un Database *D* e un Database *D'*, dove la differenza tra i due Database è che abbiamo sostituito un record con un altro, oppure abbiamo eliminato un record (la differenza tra i due Database, quindi, è un record) → la sensitività globale, quindi, determina la massima differenza che possiamo avere nell'output della nostra funzione *F*, se tolgo un qualsiasi record dal Database *D*.

$$\text{Global Sensitivity of f is } S(f) = \max_{\text{dist}(D, D') = 1} |f(D) - f(D')|$$

- una volta calcolata la sensitività della funzione, per riuscire ad ottenere effettivamente la funzione che soddisfa la funzione di Differential Privacy, dobbiamo sommare all'output della funzione (calcolata sul Database *D*), i valori che può assumere una variabile *Z*, che segue una distribuzione di Laplace, moltiplicato per la sensitività della funzione e diviso epsilon.

L'applicazione pratica della Differential Privacy, consiste nel calcolare funzioni statistiche sui dati (quali per esempio: media e mediana), per allenare i meccanismi di machine learning e per implementare un'invariante delle tecniche di anonimizzazione viste precedentemente.