Chapter 6: An Application of PETs: Privacy Preserving Record Linkage (PPRL)

Lecture PETs4DS:
Privacy Enhancing Technologies for Data Science

Parts of this slide set are based on the
“Tutorial on Techniques for Scalable Privacy-preserving Record Linkage”,
by Christen, Verykios and Vatsalan, at CIKM 2013.

Dr. Benjamin Heitmann and Prof. Dr. Stefan Decker
Informatik 5
Lehrstuhl Prof. Decker

Topics of the lecture so far

- Privacy threat analysis
- Anonymisation
- Approaches for private computation:
  - Adversary models applicable to private computation
  - Privacy Enhancing Technologies (PETs) applicable to private computation

How does this match the real world?

- Are there real world use cases requiring PETs?
- Which PETs are realistic for real world use cases?
- Challenges:
  - Scalability
  - Realistic adversaries
- Did we gain a vocabulary and an understanding which is relevant?

An Application of PETs:
Privacy Preserving Record Linkage (PPRL)

Based on
“Tutorial on Techniques for Scalable Privacy-preserving Record Linkage”,
by Christen, Verykios and Vatsalan, at CIKM 2013.

RWTH AACHEN UNIVERSITY

Overview of Privacy Preserving Record Linkage (PPRL) topics

- Motivation
- Definition
- A taxonomy for PPRL approaches
- Details of different approaches for PPRL

Motivation

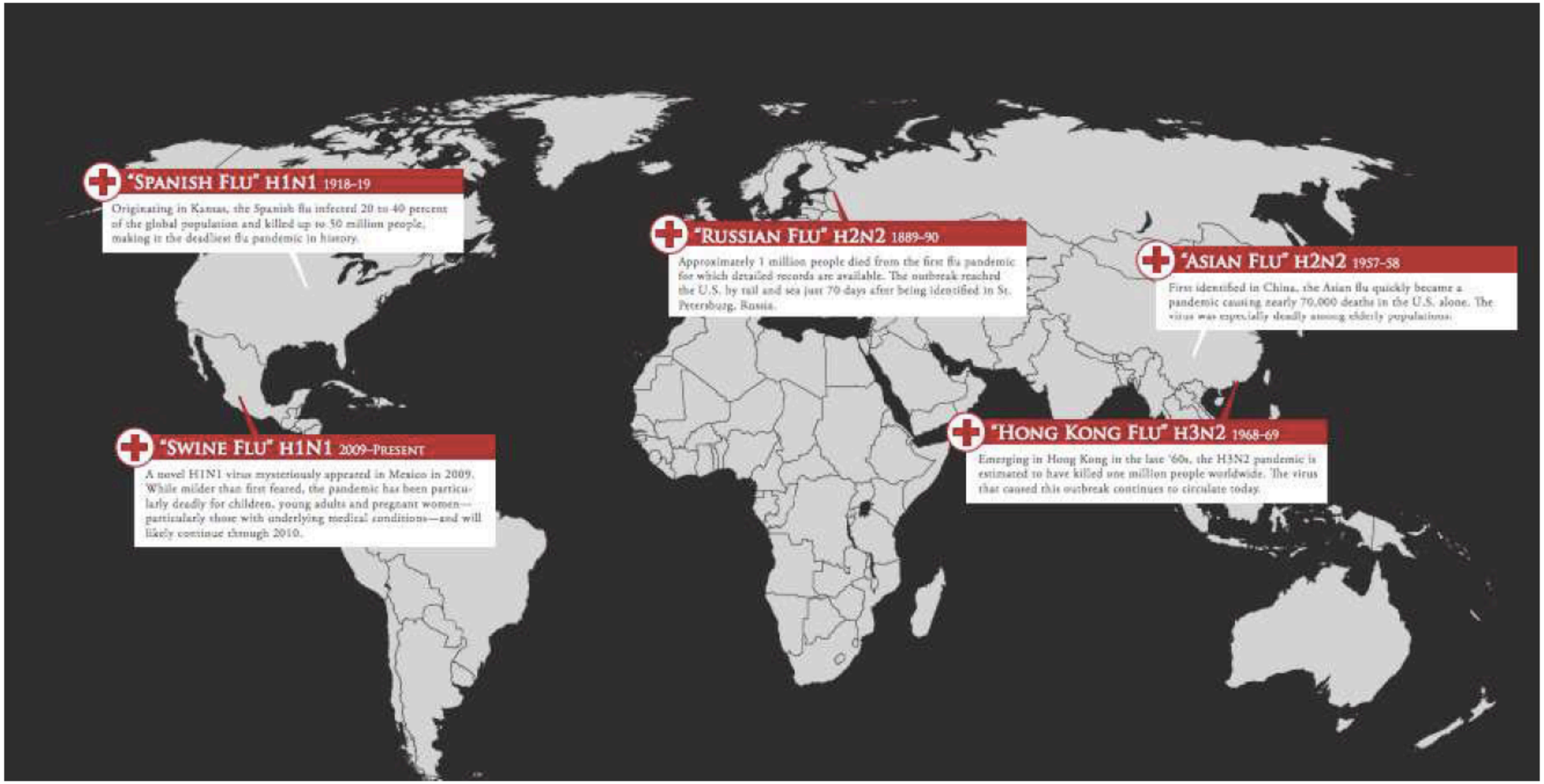
- Large amounts of data are being collected both by organisations in the private and public sectors, as well as by individuals
- Much of these data are about people, or they are generated by people
  - Financial, shopping, and travel transactions
  - Electronic health and financial records
  - Tax, social security, and census records
  - Emails, tweets, SMSs, blog posts, etc.

- Analysing (mining) such data can provide huge benefits to businesses and governments

Motivation (continued)

- Often data from different sources need to be integrated and linked
  - To improve data quality
  - To enrich data with additional information
  - To allow data analyses that are impossible on individual databases
- Lack of unique entity identifiers means that linking is often based on personal information
- When databases are linked across organisations, maintaining privacy and confidentiality is vital
- This is where privacy-preserving record linkage (PPRL) can help

Motivating example: health surveillance



"Spanish Flu" H1N1 1918–19

Originating in Kansas, the Spanish flu infected 20 to 40 percent of the global population and killed up to 50 million people, making it the deadliest flu pandemic in history.

"Russian Flu" H2N2 1889–90

Approximately 1 million people died from the first flu pandemic for which detailed records are available. The outbreak reached the U.S. by rail and sea just 70 days after being identified in St. Petersburg, Russia.

"Asian Flu" H2N2 1957–58

First identified in China, the Asian flu quickly became a pandemic causing nearly 70,000 deaths in the U.S. alone. The virus was especially deadly among elderly populations.

"Swine Flu" H1N1 2009–Present

A novel H1N1 virus mysteriously appeared in Mexico in 2009. While milder than first feared, the pandemic has been particularly deadly for children, young adults and pregnant women—particularly those with underlying medical conditions—and will likely continue through 2010.

"Hong Kong Flu" H3N2 1968–69

Emerging in Hong Kong in the late '60s, the H3N2 pandemic is estimated to have killed one million people worldwide. The virus that caused this outbreak continues to circulate today.

Motivating example: health surveillance

- Preventing the outbreak of epidemics requires monitoring of occurrences of unusual patterns in symptoms (in real time!)
- Data from many different sources will need to be collected:
  - travel and immigration records;
  - doctors, emergency and hospital admissions;
  - drug purchases in pharmacies;
  - animal health data;
- Privacy concerns arise if such data are stored and linked at a central location

- **Requirement:** Private patient data and confidential data from health care organisations must be kept secure, while still allowing linking and analysis

What is record linkage?

- The process of linking records that represent the same entity in one or more databases
  (patient, customer, business name, etc.)
  - Also known as *data matching*, *entity resolution*, *data linkage*, *object identification*, *identity uncertainty*, *merge-purge*, etc.
- Major challenge is that unique entity identifiers are often not available in the databases to be linked (or if available, they are not consistent)

  E.g., which of these records represent the same person?

Dr Smith, Peter	*42 Miller Street 2602 O'Connor*
Pete Smith	*42 Miller St 2600 Canberra A.C.T.*
P. Smithers	*24 Mill Rd 2600 Canberra ACT*

Applications of record linkage

- **Applications of record linkage**
  - Remove duplicates in a data set (de-duplication)
  - Merge new records into a larger master data set
  - Compile data for longitudinal (over time) studies
  - Clean and enrich data sets for data mining projects
  - Geocode matching (with reference address data)

- **Example application areas**
  - Immigration, taxation, social security, census
  - Fraud detection, law enforcement, national security
  - Business mailing lists, exchange of customer data
  - Social, health, and biomedical research

A short history of record linkage

- Research on computer assisted record linkage goes back as far as the 1950s.
- Strong interest in the last decade from many research fields in computer science:
  - data mining
  - AI
  - knowledge engineering
  - information retrieval

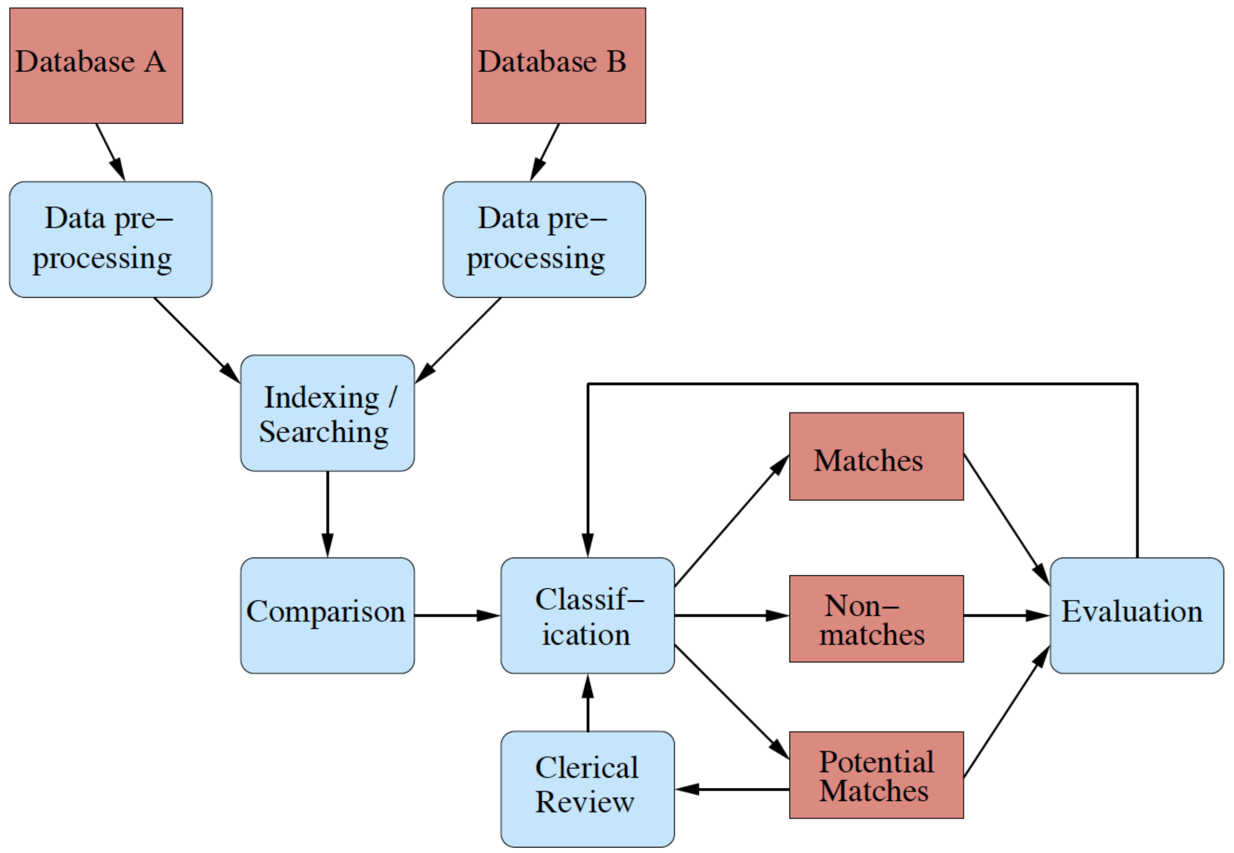
- Many different techniques have been developed
- Major focus is on scalability to large databases and linkage quality:
  - Various indexing/blocking techniques to efficiently and effectively generate candidate record pairs
  - Various machine learning-based classification techniques, both supervised and unsupervised, as well as active learning based

Record linkage challenges

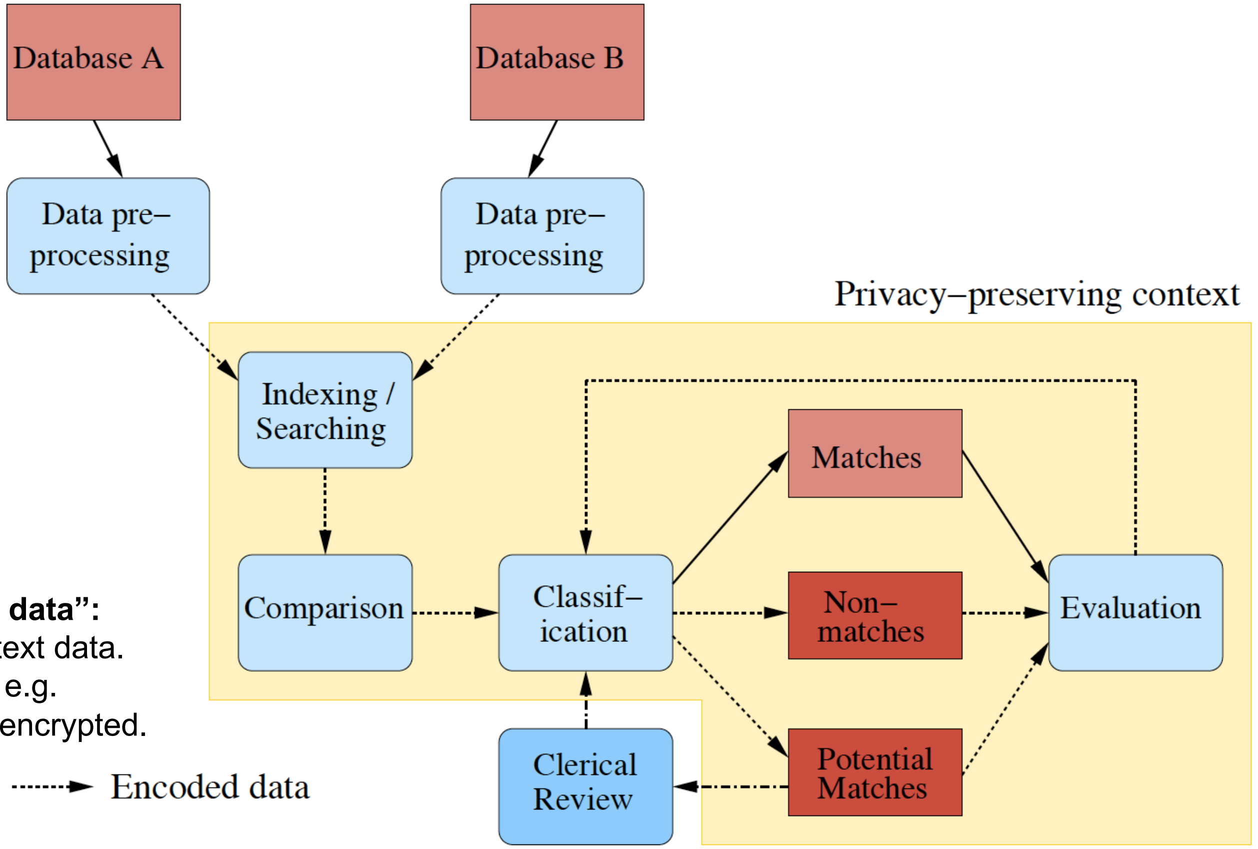
- No unique entity identifiers available

- **Real world data is dirty:**
  - typographical errors and variations, missing and out-of-date values, different coding schemes, etc.
- **Scalability:**
  - Naïve comparison of all record pairs is quadratic
  - Remove likely no-matches as efficiently as possible
- **No training data in many linkage applications**
  - No record pairs with known true match status
- **Privacy and confidentiality:**
  - personally identifiable information (PIIs), like names and addresses, are commonly required for linking

The (general) record linkage process



The privacy preserving record linkage process (PPRL)



“Encoded data”:
Non-plain text data.
It could be e.g.
hashed or encrypted.

Example scenario (1): Public health research

- A research group is interested in analysing the effects of car accidents upon the health system
  - Most common types of injuries?
  - Financial burden upon the public health system?
  - General health of people after they were involved in a serious car accident?
- They need access to data from hospitals, doctors, car and health insurers, and from the police
  - All identifying data have to be given to the researchers, or alternatively a trusted record linkage unit.
- This might prevent an organisation from being able or willing to participate
  - e.g. insurers or police

Example scenario (2): Crime investigation

- A national crime investigation unit is tasked with fighting against crimes that are of national significance, e.g. organised crime syndicates.
- This unit will likely manage various national databases which draw from different sources, including law enforcement and tax agencies, Internet service providers, and financial institutions.

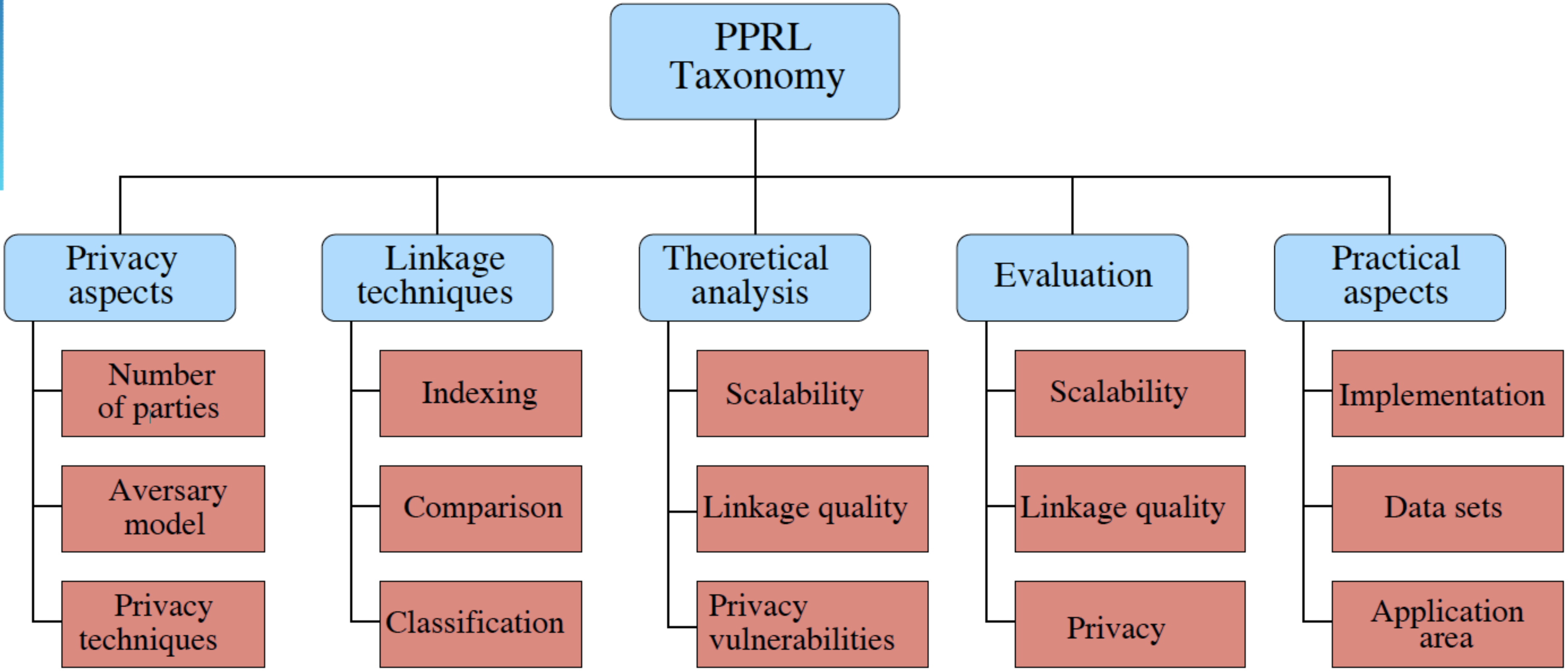
- **These data are highly sensitive:**
  - Storage, retrieval, analysis and sharing must be tightly regulated.
  - Collecting such data in one place makes them vulnerable to outsider attacks and internal adversaries.
- Ideally, only linked records, such as those of suspicious individuals, are available to the crime unit
  - This significantly reduces the risk of privacy breaches.

A definition of Privacy Preserving Record Linkage (PPRL)

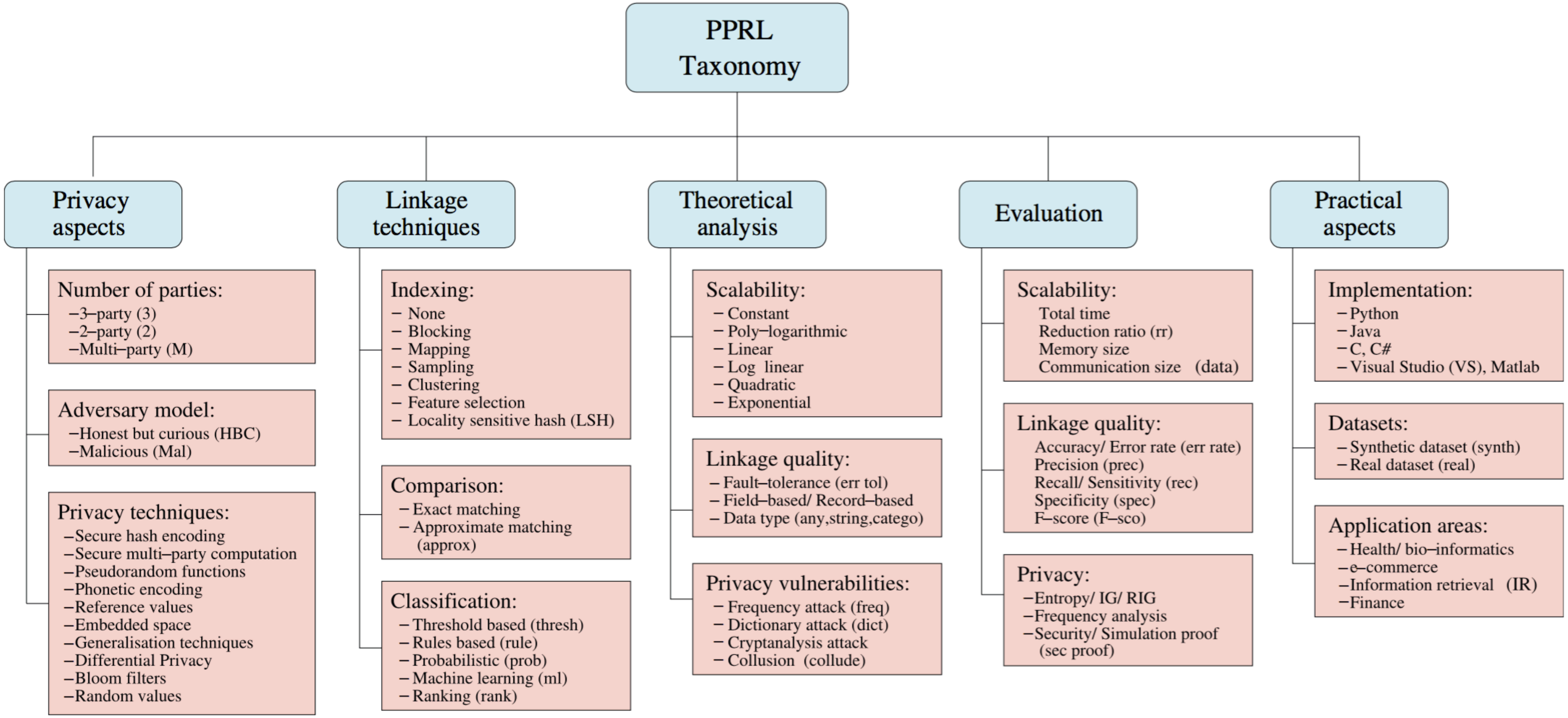
- Assume $O_1 \cdots O_d$ are the *d* owners of their respective databases $D_1 \cdots D_d$
- They wish to determine which of their records $r_1^i \in D_1$, $r_2^j \in D_2, \cdots$, and $r_d^k \in D_d$, match according to a decision model $C(\mathrm{r}_1^i, r_2^j, \cdots, r_d^k)$ that classifies pairs (or groups) of records into one of the two classes *M* of matches, and *U* of non-matches
- $O_1 \cdots O_d$ do not wish to reveal their actual records $r_1^i \cdots r_d^k$ with any other party

  (they are, however, prepared to disclose to each other, or to an external party, the outcomes of the matching process — certain attribute values of record pairs in class *M* — to allow further analysis)

A taxonomy for PPRL: Overview



A taxonomy for PPRL: Details



PPRL Taxonomy: Privacy aspects

- **Number of parties involved in a protocol**
  - Two-party protocol: Two database owners only
  - Three-party protocol: Require a (trusted) third party
- **Adversary model:**
  - Based on models used in cryptography:
  - Honest-but-curious or malicious behaviour
- **Privacy technologies** — many different approaches, including:
  - One-way hash encoding,
  - Anonymisation and generalisation
  - Secure multi-party computation (SMPC)
  - Differential privacy
  - Bloom filters
  - Public reference values
  - Phonetic encoding

PPRL Taxonomy: Linkage techniques

- **Indexing / blocking:**
  - Indexing aims to identify candidate record pairs that likely correspond to matches
  - Different techniques used: blocking, sampling, generalisation, clustering, hashing, binning, etc.
- **Comparison:**
  - Exact or approximate (consider partial similarities, like “vest” and “west”, or “peter” and “pedro”)
- **Classification:**
  - Based on the similarities calculated between records
  - Various techniques, including similarity threshold, rules, ranking, probabilistic, or machine learning based

PPRL Taxonomy: Theoretical analysis

- **Scalability:**
  - of computation and communication, usually done using ‘big O’ notation — O(n), O(n2), etc.
- **Linkage quality:**
  - Fault (error) tolerance
  - Field- or record-based (matching)
  - Data types (strings, numerical, age, dates, etc.)
- **Privacy vulnerabilities**
  - Different types of attack (frequency, dictionary, linkage, and crypt-analysis)
  - Collusion between parties

PPRL Taxonomy: Evaluation

- **Scalability:**
  - We can measure run-time and memory usage
  - Implementation independent measures are based on the number of candidate record pairs generated

- **Linkage quality:**
  - Classifying record pairs as matches or non-matchesis a binary classification problem
  - Use traditional accuracy measures (precision, recall, etc.)

- **Privacy:**
  - Least 'standardised' area of evaluation, with various measures used.
  - Information gain, simulation proofs, disclosure risk, or probability of re-identification.

PPRL Taxonomy: Practical Aspects

- **Data sets:**
  - Real-world data sets or synthetic data sets
  - Public data (from repositories) or confidential data

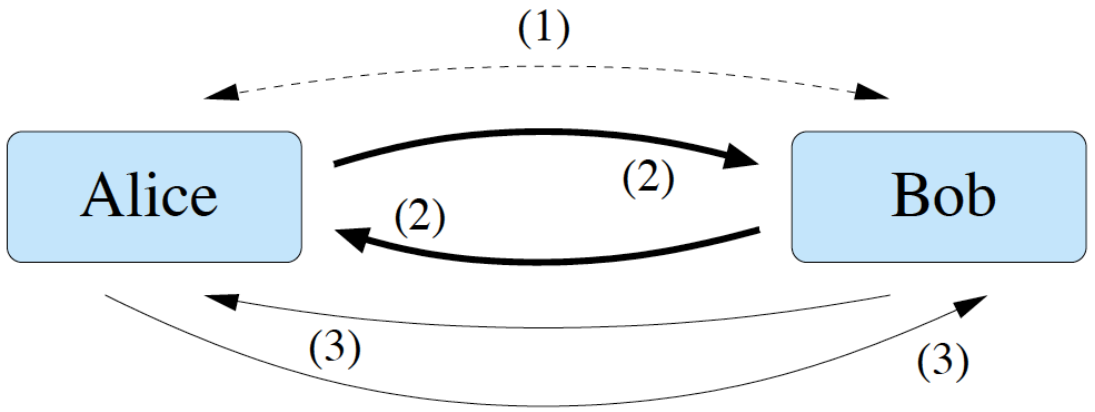
- **Targeted application areas:**
  - Include health care, census, business, finance, etc.

Basic protocols for PPRL

- **Two basic types of protocols:**
  - **Two-party protocol:** Only the two database owners who wish to link their data
  - **Three-party protocols:** Use a (trusted) third party (linkage unit) to conduct the linkage

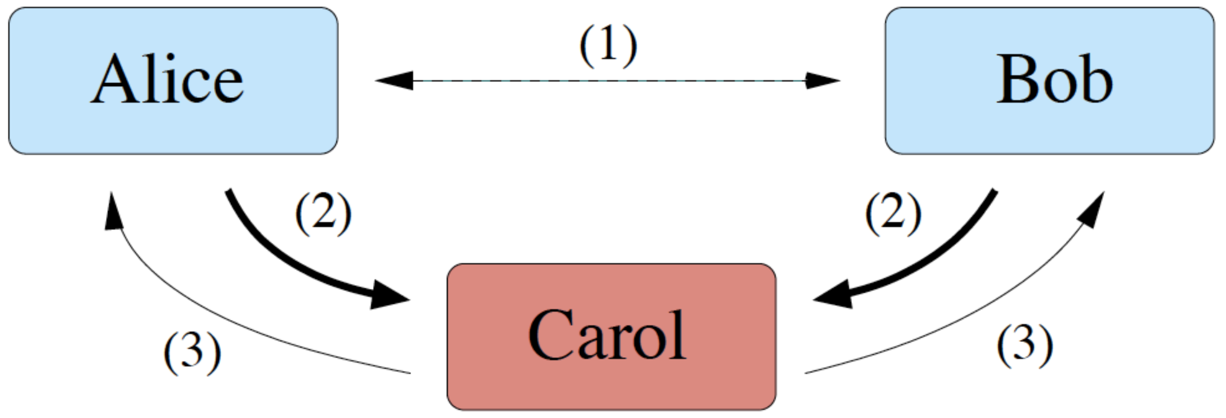
- **Generally, three main communication steps**
  1. Exchange of which attributes to use in a linkage, pre-processing methods, encoding functions, parameters, secret keys, etc.
  2. Exchange of the somehow encoded database records
  3. Exchange of records (or selected attribute values, or identifiers only, of records) classified as matches.

PPRL: Two-party protocols



- More challenging than three-party protocols, but more secure (no third party involved, so no collusion possible)
- Main challenge: How to hide sensitive data from the other database owner
- Step 2 “exchange of the encoded database records” is generally done over several iterations of communication

PPRL: Three party protocols



- Easier than two-party protocols, as third party (Carol) prevents database owners from directly seeing each other’s sensitive data
- **Linkage unit never sees un-encoded / un-encrypted data**
- Collusion is possible: One database owner gets access to data from the other database owner via the linkage unit

Exact Matching for PPRL with Hash-encoding (1)

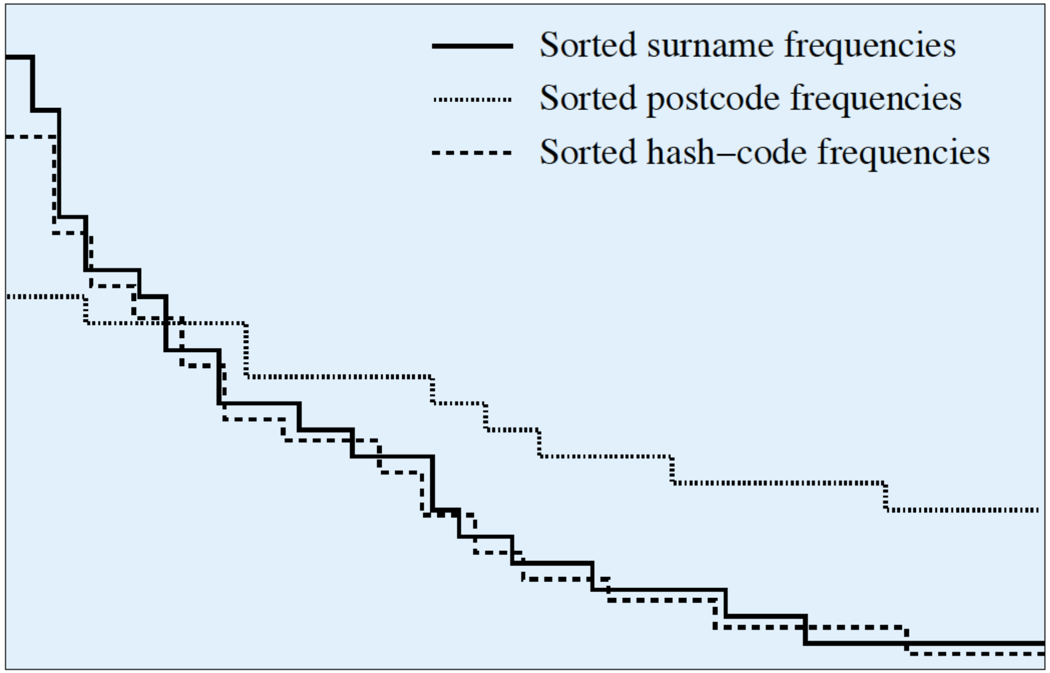
- A basic building block of many PPRL protocols
- **Idea:** Use a one-way hash-encoding function to encode values, then compare these hash-codes
  - One-way hash functions like MD5 (message digest) or SHA (secure hash algorithm)
  - Convert a string into a hash-code (MD5 128 bits, SHA-1 160 bits, SHA-2 224–512 bits)
- **For example:**
  - ‘peter’ → ‘101010. . .100101’ or ‘4R#x+Y4i9!e@t4o]’
  - ‘pete’ → ‘011101. . .011010’ or ‘Z5%o-(7Tq1@?7iE/’
- Single character difference in input values results in completely different hash codes

Exact Matching for PPRL with Hash-encoding (2)

- Having only access to hash-codes will make it nearly impossible with current computing technology to learn their original input values
  - Brute force dictionary attack (try all known possible input values) and all known hash-encoding functions.
  - Can be overcome by adding a secret key (known only to database owners) to input values before hash-encoding.
  - For example, with secret key: ‘42-rocks!’
    ‘peter’ → ‘peter42-rocks!’ → ‘i9=!e@Qt8?4#4$7B’

- Frequency attack still possible:
  - compare frequency of hash-values to frequency of known attribute values

Example of a frequency attack



- If frequency distribution of hash-encoded values closely matches the distribution of values in a (public) database, then ‘re-identification’ of values might be possible

Problems with hash-encoding for PPRL

- **Simple hash-encoding only allows for exact matching of attribute values**
  - Can to some degree be overcome by pre-processing, such as phonetic encoding (Soundex, NYSIIS, etc.)
  - Database owners clean their values, convert name variations into standard values, etc.
- **Frequency attacks are possible**
  - Can be overcome by adding random records to distort frequencies

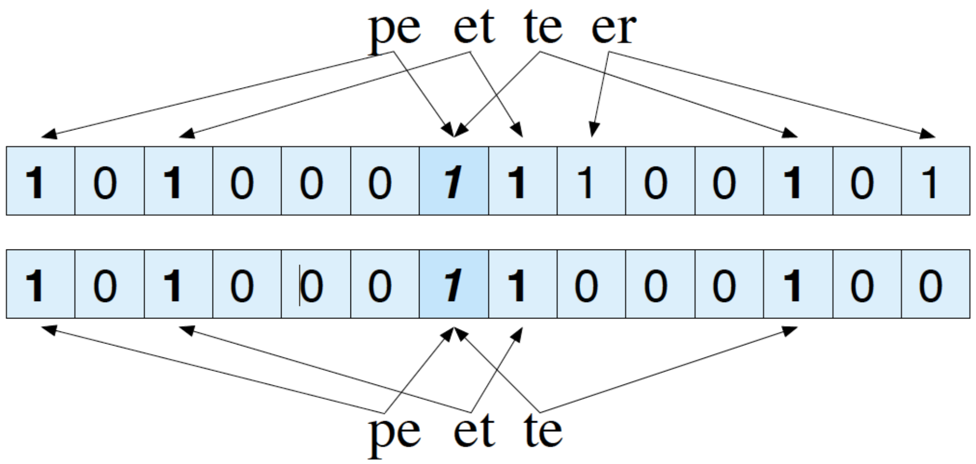
Approximate string matching for PPRL

- Aim: Calculate a normalised similarity between two strings $(0 \leq sim_{approx\_match} \leq 1)$
- Q-gram based approximate comparisons
  - Convert a string into q-grams (sub-strings of length q)
    For example, for q = 2: ‘peter’ $\rightarrow$ [‘pe’,‘et’,‘te’,‘er’]
  - Find q-grams that occur in two strings, for example using the Dice coefficient: $sim_{Dice} = 2 \times c_c / (c_1 + c_2)$ (c_c = number of common q-grams, c_1 = number of q-grams in string s_1, c_2 = number of q-grams in s_2)
  - With s_1 = ‘peter’ and s_2 = ‘pete’: $c_1 = 4$, $c_2 = 3$, $c_c = 3$ (‘pe’,‘et’,‘te’), $sim_{Dice} = 2 \times 3/(4+3) = 6/7 = 0.86$
  - Variations based on Overlap or Jaccard coefficients

Bloom filter based PPRL (1)

- Proposed by Schnell et al. (Biomed Central, 2009)
- A Bloom filter is a bit-array, where a bit is set to 1 if a hash-function $H_k(x)$ maps an element *x* of a set into this bit (elements in our case are q-grams)
  - $0 \leq H_k(x) < l$, with *l* the number of bits in Bloom filter
  - Many hash functions can be used (Schnell: $k = 30$)
  - Number of bits can be large (Schnell: $l = 1000$ bits)
- Basic idea: Map q-grams into Bloom filters using hash functions only known to database owners, send Bloom filters to a third party which calculates Dice coefficient (number of *1*-bits in Bloom filters)

Bloom filter based PPRL (2)



- *1*-bits for string ‘peter’: 7, *1*-bits for ‘pete’: 5, common *1*-bits: 5, therefore $sim_{Dice} = 2 \times 5/(7+5) = 10/12 = 0.83$
- Collisions will effect the calculated similarity values
- Number of hash functions and length of Bloom filter need to be carefully chosen

Bloom filter based PPRL (3)

- **Frequency attacks are possible**
  - Frequency of 1-bits reveals frequency of q-grams (especially problematic for short strings)
  - Using more hash functions can improve security
  - Add random (dummy) string values to hide real values

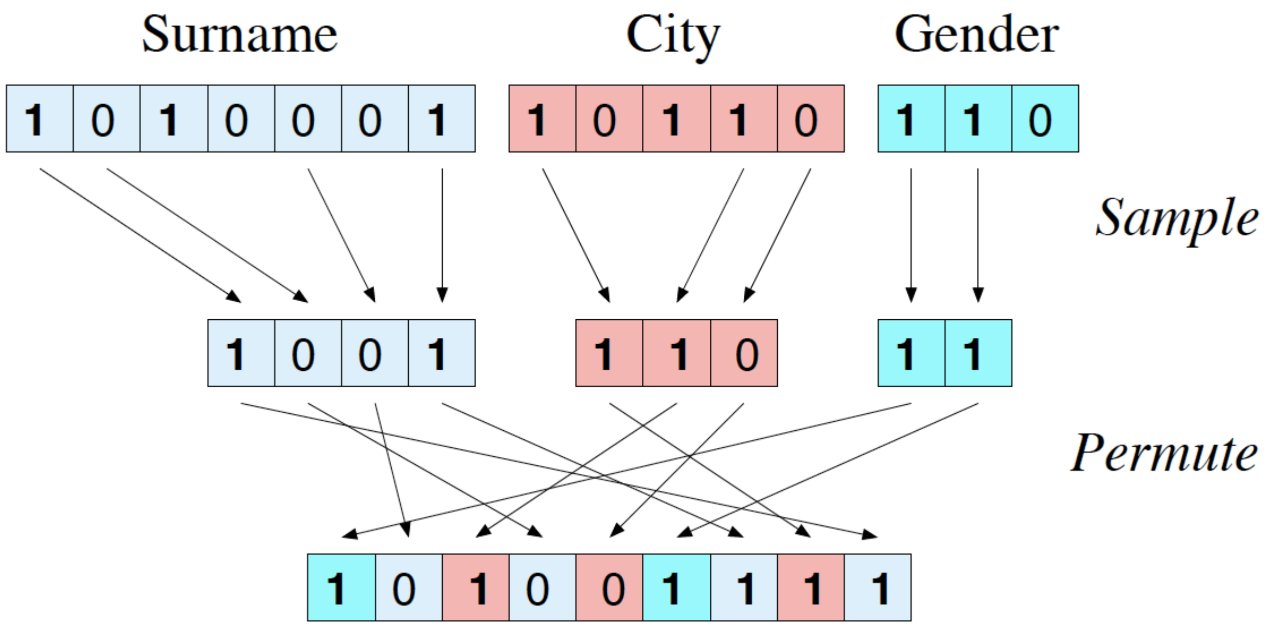
Composite Bloom filters for PPRL (1)

- The idea is to first generate Bloom filters for attributes individually, then combine them into one composite Bloom filter per record

- **Different approaches:**
  - Same number of bits from each attribute
  - Better: Sample different number of bits from attributes depending upon discriminative power of attributes
  - Even better: Attribute Bloom filters have different sizes such that they have similar percentage of 1-bits (depending upon attribute value lengths)

- Final random permutation of bits in composite Bloom filter

Composite Bloom filters for PPRL (2)



- Experimental results showed much improved security with regard to crypt-analysis attacks

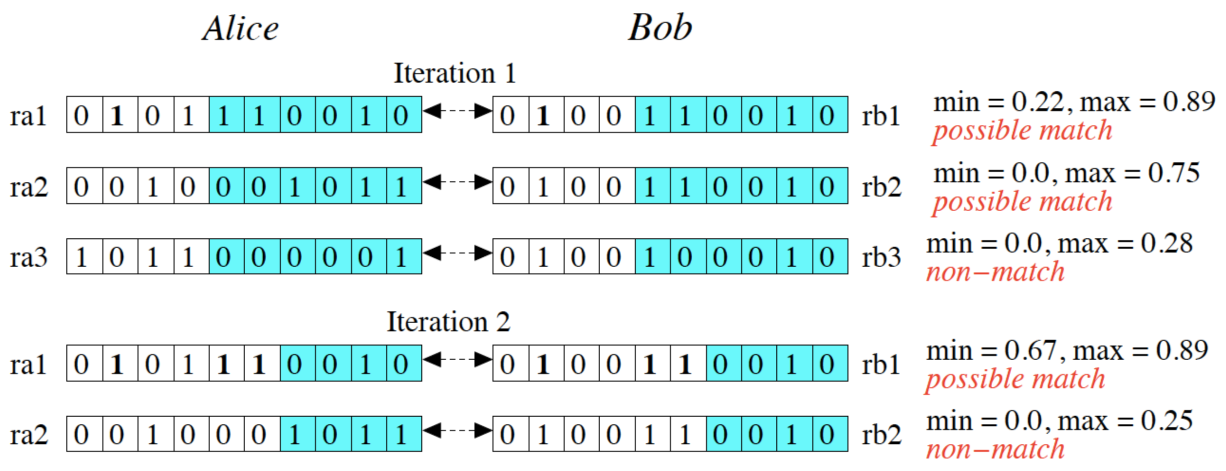
Two-party Bloom filter protocol for PPRL (1)

- Proposed by Vatsalan et al. (AusDM, 2012)
- Iteratively exchange certain bits from the Bloom filters between database owners

- Calculate the minimum Dice-coefficient similarity from the bits exchanged, and classify record pairs as matches, non-matches, and possible matches

- Pairs classified as possible matches are taken to the next iteration
  - The number of bits revealed in each iteration is calculated such that the risk of revealing more bits for non-matches is minimised
  - Minimum similarity of possible matches increases as more bits are revealed

Two-party Bloom filter protocol for PPRL (2)



- Each party knows how many 1-bits are set in total in a Bloom filter received from the other party
- In iteration 1, for example, there is one unrevealed 1-bit in *ra3*, and so the maximum possible Dice similarity with *rb3* is: *max*($sim_{Dice}(ra3, rb3)$) $= 2\times1/(4+3)= 2/7 = 0.28$

A hybrid approach to PPRL (1)

- Proposed by Inan et al. (ICDE, 2008)
- Use k-anonymity to generalise (sanitise) databases and find ‘blocks’ of possible matching record pairs

- **Basic idea:** In a first step, generate value generalisation hierarchies (VGH); in a second step calculate distances between records with same generalised values using a secure multiparty computation (SMC) approach (based on homomorphic encryption)

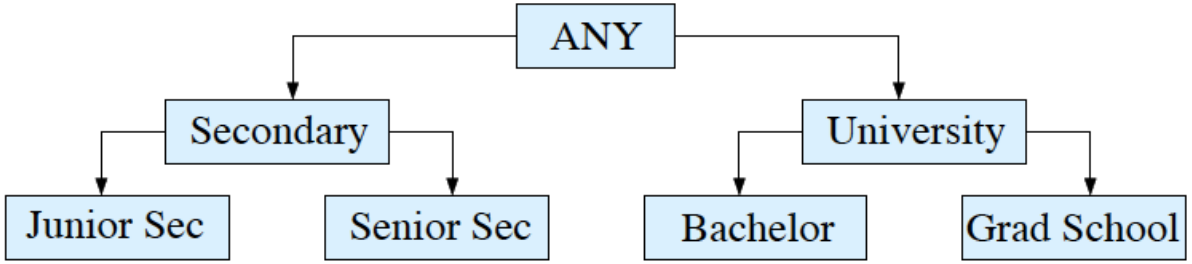
- VGHs are hierarchical tree-like structures where a node at each level is a generalisation of its descendants

A hybrid approach to PPRL (2)

ID	Education	Age
r1	Junior Sec	22
r2	Senior Sec	16
r3	Junior Sec	27
r4	Bachelor	33
r5	Bachelor	39
r6	Grad School	34

ID	Education	Age
r1’	Secondary	[1–32]
r2’	Secondary	[1–32]
r3’	Secondary	[1–32]
r4’	University	[33–39]
r5’	University	[33–39]
r6’	University	[33–39]

3-anonymous generalisation



A hybrid approach to PPRL (3)

- Generalised and hash-encoded attribute values are sent to the third party, which can classify record pairs as matches, non-matches or possible matches (depending upon how many generalised attribute values two records have in common)

- SMC approach is used to calculate similarities of possible matches (computationally more expensive)

- User can set threshold to tune between precision and recall of the resulting matched record pairs
- Main drawback: Cannot be applied on alphanumeric values (such as names) that do not have a VGH

Summary

- Significant advances to achieving the goal of PPRL have been developed in recent years
  - Various approaches based on different techniques
  - Can link records securely, approximately, and in a (somewhat) scalable fashion
- So far, most PPRL techniques concentrated on approximate matching techniques, and on making PPRL more scalable to large databases
- However, no large-scale comparative evaluations of PPRL techniques have been published
- Only limited investigation of classification and linking assessment in PPRL

Challenges and future work

- **Improved classification for PPRL:**
  - Mostly simple threshold based classification is used
  - No investigation into advanced methods, such as collective entity resolution techniques
  - Supervised classification is difficult — no training data in most situations
- **Assessing linkage quality and completeness:**
  - How to assess linkage quality (precision and recall)?
  - How many classified matches are true matches?
  - How many true matches have we found?
  - Evaluating actual record values is not possible (as this would reveal sensitive information)
- **PPRL on multiple databases:**
  - Most work so far is limited to linking two databases (in reality often databases from several organisations)
  - Pair-wise linking does not scale up
  - Preventing collusion between (sub-groups of) parties becomes more difficult