A Tutorial on

Techniques for Scalable Privacy-preserving Record Linkage

Peter Christen[1], Vassilios Verykios[2], and **Dinusha Vatsalan**[1]

[1] **Research School of Computer Science,**
ANU College of Engineering and Computer Science,
The Australian National University, Canberra, Australia

[2] **School of Science and Technology,**
Hellenic Open University, Patras, Greece

Contacts: **peter.christen@anu.edu.au** / **verykios@eap.gr** /
dinusha.vatsalan@anu.edu.au

This work is partially funded by the Australian Research Council (ARC)
under Discovery Project DP130101801.

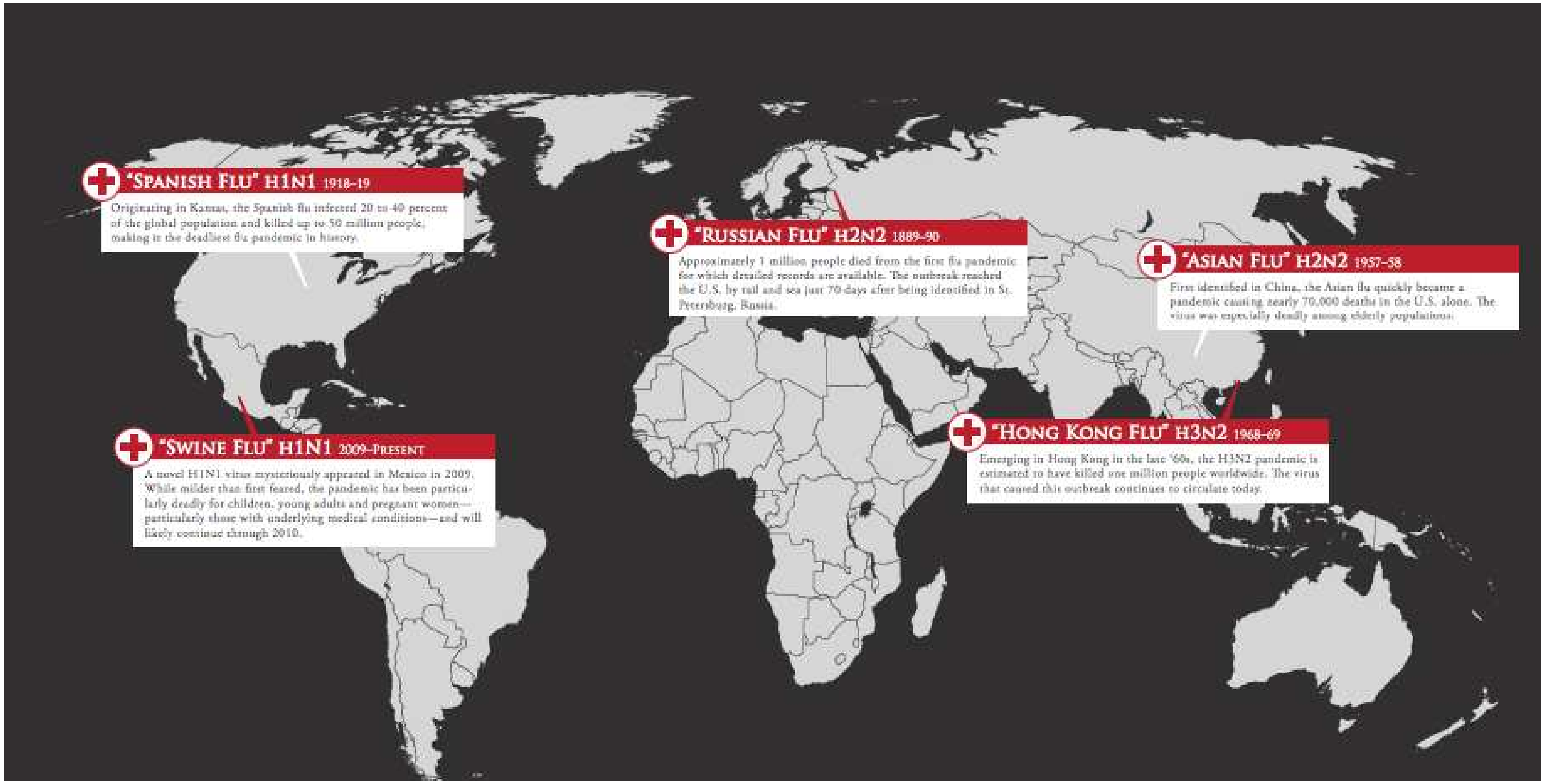
Motivation

- 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 (1)



"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 (2)

- 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 (including travel and immigration records; doctors, emergency and hospital admissions; drug purchases in pharmacies; animal health data; etc.)
- Privacy concerns arise if such data are stored and linked at a central location
- Private patient data and confidential data from health care organisations must be kept secure, while still allowing linking and analysis

Tutorial Outline

- Background to record linkage and PPRL
  - Applications, history, challenges, the record linkage and PPRL process
  - Scenarios, a definition, and a taxonomy for PPRL
- Exact and approximate PPRL techniques
  - Basic protocols for PPRL (two and three parties)
  - Hash-encoding for exact matching, and ↙ Tea break key techniques for approximate comparison
- Selected key techniques for scalable PPRL
  - Incl. private blocking; Bloom filters; hybrid, public reference, and differential privacy approaches, etc.
- Conclusions and challenges

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 (1)

- Computer assisted record linkage goes back as far as the 1950s (based on ad-hoc heuristic methods)
- Basic ideas of probabilistic linkage were introduced by *Newcombe & Kennedy* (1962)
- Theoretical foundation by *Fellegi & Sunter* (1969)
  - Compare common record attributes (or fields)
  - Compute matching weights based on frequency ratios (global or value specific) and error estimates
  - Sum of the matching weights is used to classify a pair of records as a *match*, *non-match*, or *potential match*
  - Problems: Estimating errors and thresholds, assumption of independence, and *clerical review*

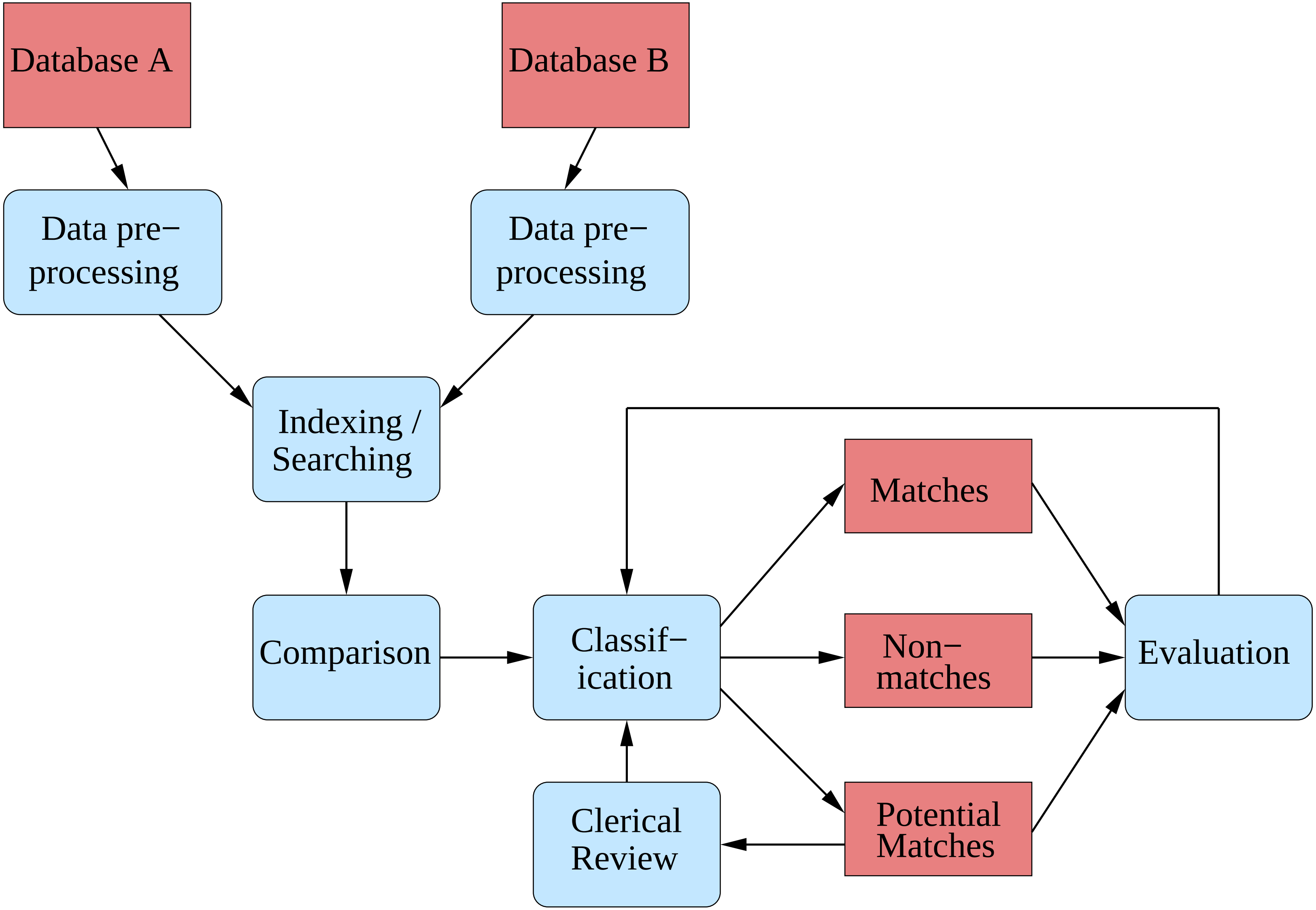
A short history of record linkage (2)

- Strong interest in the last decade from computer science (from many research fields, including data mining, AI, knowledge engineering, information retrieval, information systems, databases, and digital libraries)
- 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

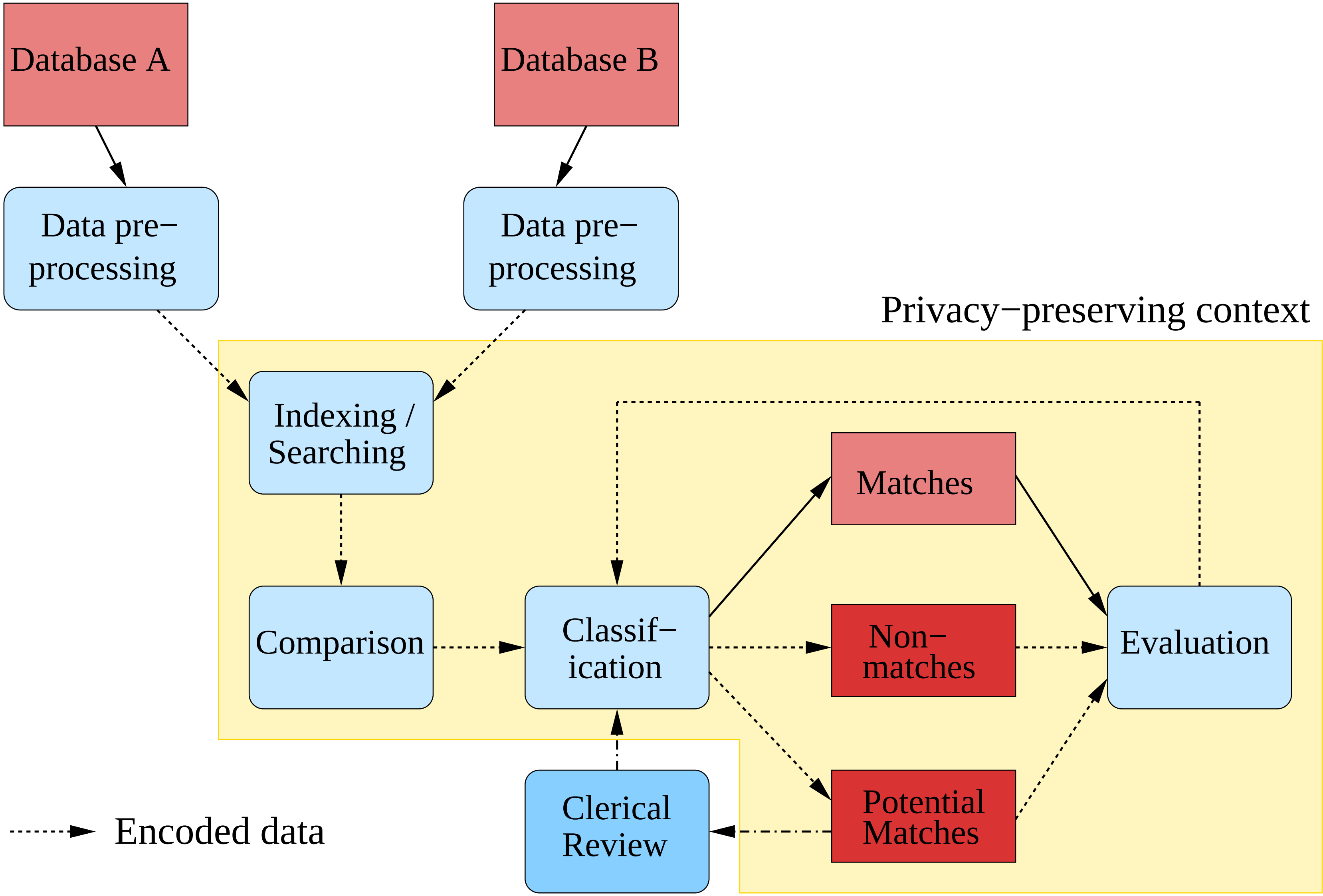
Record linkage challenges

- 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
  (because personal information, like names and addresses, are commonly required for linking)

The record linkage process



The PPRL process



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 (insurers or police)

Example scenario (2): Crime investigation

- A national crime investigation unit is tasked with fighting against crimes that are of national significance (such as 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; and 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 unit (significantly reducing the risk of privacy breaches)

A definition of 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)

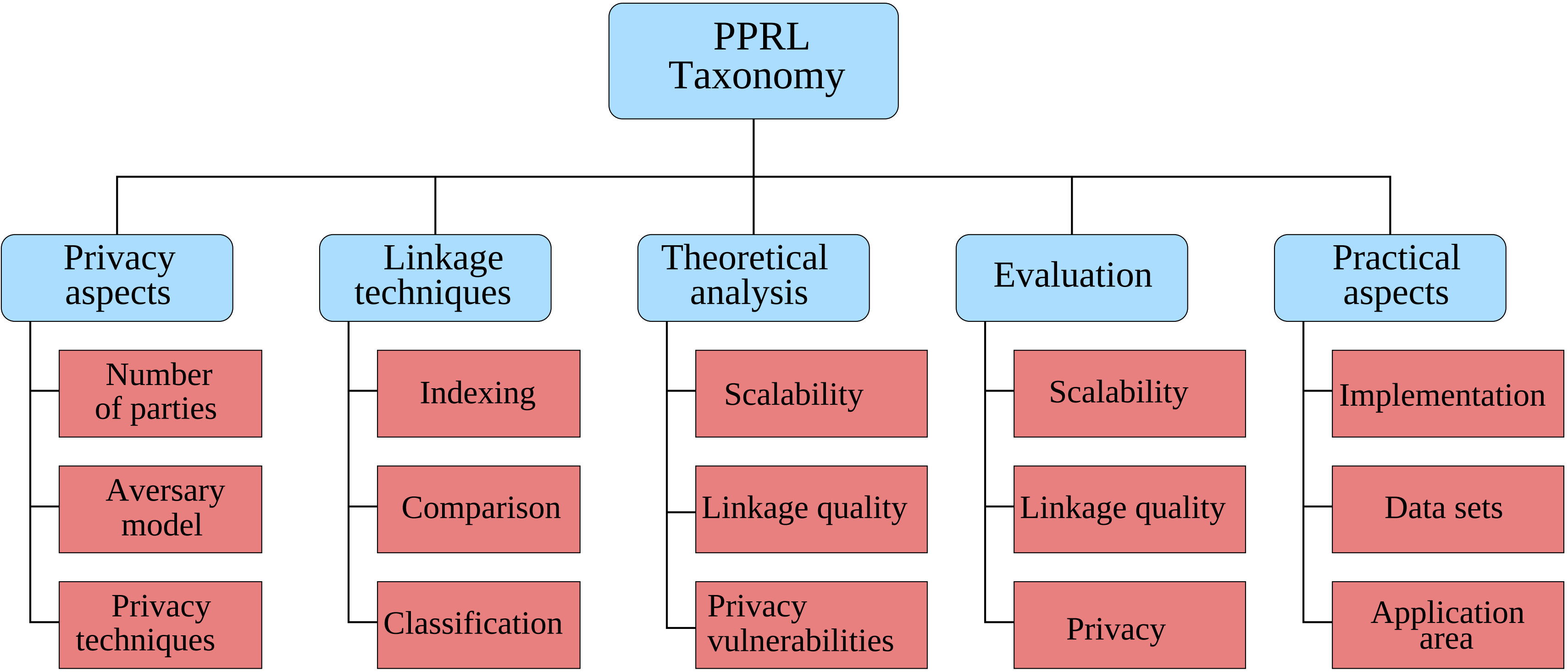
A taxonomy for PPRL (1)

- Characterise PPRL techniques along fifteen dimensions with the aim to
  - Get a clearer picture of current approaches to PPRL
  - Specify gaps between record linkage and PPRL
  - Identify directions for future research in PPRL
- Five major areas for assessing PPRL techniques

For more on this taxonomy, see:

A taxonomy of privacy-preserving record linkage techniques
Dinusha Vatsalan, Peter Christen, and Vassilios Verykios
Elsevier Information Systems, 38(6), September 2013

A taxonomy for PPRL (2)



PPRL Taxonomy

- Privacy aspects
  - Number of parties
  - Aversary model
  - Privacy techniques
- Linkage techniques
  - Indexing
  - Comparison
  - Classification
- Theoretical analysis
  - Scalability
  - Linkage quality
  - Privacy vulnerabilities
- Evaluation
  - Scalability
  - Linkage quality
  - Privacy
- Practical aspects
  - Implementation
  - Data sets
  - Application area

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
  - One-way hash encoding, generalisation, secure multi-party computation, differential privacy, Bloom filters, public reference values, phonetic encoding, random extra values, and various others

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

Taxonomy: Theoretical analysis

- Scalability (of computation and communication, usually done using ‘big *O*’ notation — $O(n)$, $O(n^2)$, 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

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-matches* is a binary classification problem, so we can use traditional accuracy measures (precision, recall, etc.)
- Privacy
  - Least 'standardised' area of evaluation, with various measures used (such as information gain, simulation proofs, disclosure risk, or probability of re-identification)

Taxonomy: Practical aspects

- Implementation
  - Programming language used (if implemented), or only theoretical proof-of-concept
  - Sometimes no details are published
- 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.
  - Sometimes not specified

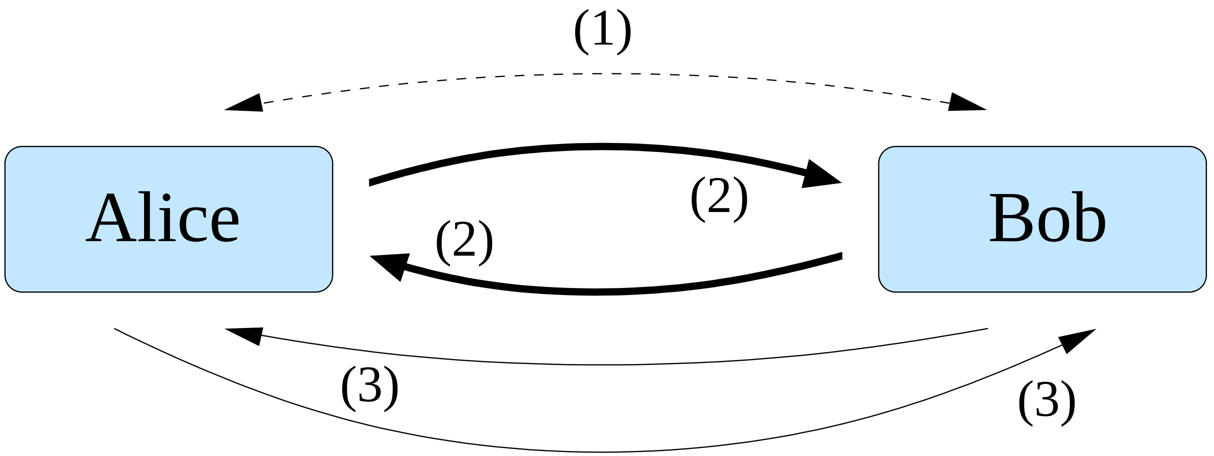
Tutorial Outline

- Background to record linkage and PPRL
  - Applications, history, challenges, the record linkage and PPRL process
  - Scenarios, a definition, and a taxonomy for PPRL
- Exact and approximate PPRL techniques
  - Basic protocols for PPRL (two and three parties)
  - Hash-encoding for exact matching, and ↙ Tea break key techniques for approximate comparison
- Selected key techniques for scalable PPRL
  - Incl. private blocking; Bloom filters; hybrid, public reference, and differential privacy approaches, etc.
- Conclusions and challenges

Basic protocols for PPRL

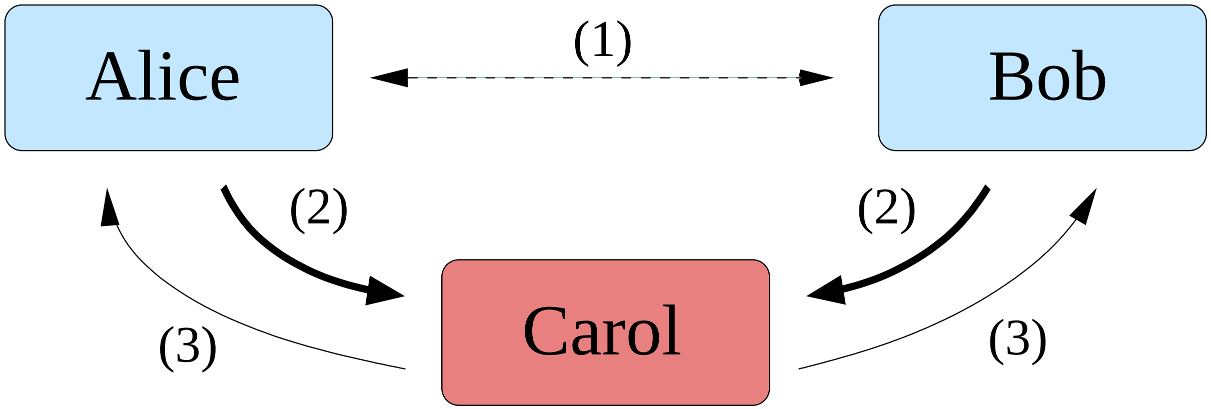
- Two basic types of protocols
  - Two-party protocol: Only the two database owners who wish to link their data
  - Three-party protocols: Use a (trusted) third party (linkage unit) to conduct the linkage
- Generally, three main communication steps
  1. Exchange of which attributes to use in a linkage, 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

Two-party protocol



- 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

Three-party protocol



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

Hash-encoding for PPRL (1)

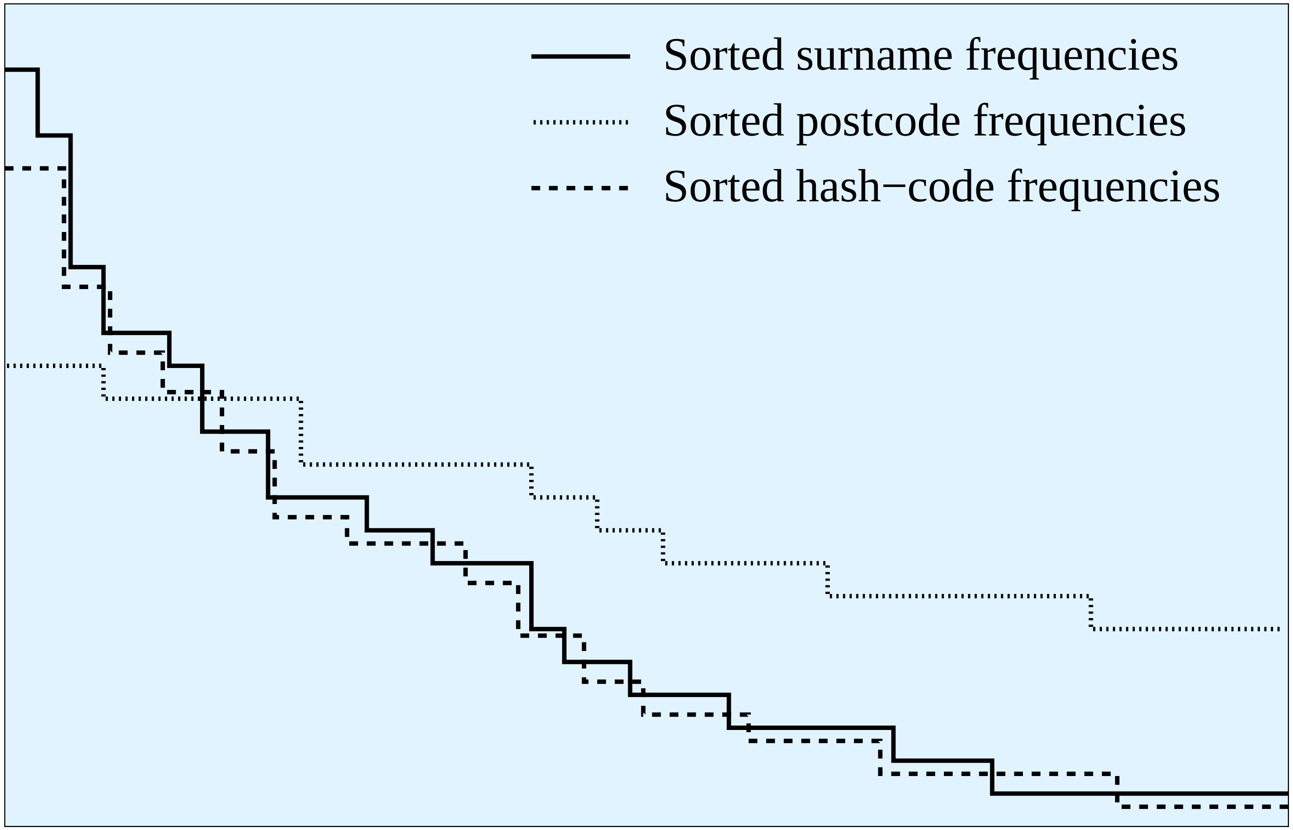
- A basic building block of many PPRL protocols
- Idea: Use a one-way hash-encoding function to encode values, then compare these hash-codes
  - One-way hash functions like MD5 (message digest) or SHA (secure hash algorithm)
  - Convert a string into a hash-code (MD5 128 bits, SHA-1 160 bits, SHA-2 224–512 bits)
- For example:

  ‘peter’ $\rightarrow$ ‘101010. . .100101’ or ‘4R#x+Y4i9!e@t4o]’
  ‘pete’ $\rightarrow$ ‘011101. . .011010’ or ‘Z5%o-(7Tq1@?7iE/’
- Single character difference in input values results in completely different hash codes

Hash-encoding for PPRL (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)

Frequency attack example



- 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

- 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
- First PPRL approaches based on hash-encoding were developed by French health researchers (Dusserre, Quantin, Bouzelat, et al., 1995)

Approximate string matching (1)

- 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’ → [‘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

ANU
THE AUSTRALIAN NATIONAL UNIVERSITY

Approximate string matching (2)

- Edit-distance based approximate comparisons
  - The number of basic character edits (insert, delete, substitute) needed to convert one string into another
  - Can be calculated using a dynamic programming algorithm (of quadratic complexity in length of strings)
  - Convert distance into a similarity as
    sim_{ED} = 1 - $dist_{ED}$ / max(l_1, l_2)
    ($l_1 =$ length of string s_1, $l_2 =$ length of s_2)
  - With $s_1 =$ 'peter' and $s_2 =$ 'pete': $l_1 = 5$, $l_2 = 4$, $dist_{ED} = 1$ (delete 'r'), $sim_{ED} = 1$ - 1/5 = 4/5 = 0.8
  - Variations consider transposition of two adjacent characters, allow different edit costs, or allow for gaps

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

Secure edit-distance for PPRL (1)

- Proposed by Atallah et al. (WPES, 2003)
- Calculate edit distance between two strings such that parties only learn the final edit-distance (two party protocol)
- Basic idea: The dynamic programming matrix is split across the two parties: $M = M_A + M_B$

M		**g**	**a**	**y**	**l**	**e**
	0	1	2	3	4	5
g	1	**0**	1	2	3	4
a	2	1	**0**	1	2	3
i	3	2	1	**1**	2	2
l	4	3	2	2	**1**	**2**

'gail' → substitute 'i' with 'y', and insert 'e' → 'gayle'

Secure edit-distance for PPRL (2)

- Matrix *M* is built row-wise
  - Element *M[i,j]* is the number of edits needed to convert $s_1[0{:}i]$ into $s_2[0{:}j]$
  - Calculated as:

    if $s_1[i] = s_2[j]$ **then** $M[i,j]=M[i-1, j-1]$

    else $M[i,j]=\min(M[i-1, j-1] + S(s_1[i], s_2[j]),$ (substitute)

    $M[i-1, j] + D(s_1[i]),$ (delete)

    $M[i, j-1] + I(s_2[j]))$ (insert)

    (often the different ‘costs’ are set to 1)
- At each step of the protocol, *Alice* and *Bob* need to determine the minimum of three values, without learning at which position the minimum occurred

Secure edit-distance for PPRL (3)

Alice – ‘gail’

M_A		?	?	?	?	?
	0	0	0	0	0	0
g	1					
a	2					
i	3					
l	4					

Bob – ‘gayle’

M_B		**g**	**a**	**y**	**l**	**e**
	0	1	2	3	4	5
?	0					
?	0					
?	0					
?	0					

⇓

Alice

M_A		?	?	?	?	?
	0	0	0	0	0	0
g	1	**-0.3**	0.7	1.1	0.7	1.4
a	2	0.9	**0.4**	0.5	0.5	1.3
i	3	0.1	0.3	**0.1**	1.5	0.6
l	4	1.5	1.3	0.8	**0.4**	***1.4***

⇓

Bob

M_B		**g**	**a**	**y**	**l**	**e**
	0	1	2	3	4	5
?	0	**0.3**	0.3	0.9	2.3	2.6
?	0	0.1	**-0.4**	0.5	1.5	1.7
?	0	1.9	0.7	**0.9**	0.5	1.4
?	0	1.5	0.7	1.2	**0.6**	***0.6***

Secure edit-distance for PPRL (4)

- Protocol requires a secure function to calculate the minimum value in a shared vector, $\vec{c} = \vec{a} + \vec{b}$, without knowing the position of the minimum (and a variation to calculate the maximum of values)
- To check if $c_i \geq c_j$, use: $c_i \geq c_j = (a_i + b_i) \geq (a_j + b_j) \Leftrightarrow (a_i - a_j) \geq -(b_i - b_j)$
- To ‘hide’ position of minimum value, use a ‘blind and permute’ protocol based on homomorphic encryption (first *Alice* blinds *Bob*, then *Bob* blinds *Alice*)
  - Homomorphic encryption: $E(a) * E(b) = E(a * b)$
- For substitution cost, check if min($s_1[i]$, $s_2[j]$) is different from max($s_1[i]$, $s_2[j]$)

Secure edit-distance for PPRL (5)

- Atallah et al. describe several variations of their protocol for different cases of costs S(·,·), D(·), and I(·)
  - Certain applications might only allow inserts and deletions, others have substitution costs depending upon the ‘distance’ from s_1*[i]* to s_2*[j]*
- Major drawback of this protocol: For each element in *M* one communication step is required (number of communication steps is quadratic in the length of the two strings)
- Not scalable to linking large databases, or long sequences

Secure TF-IDF and Euclidean distance for PPRL (1)

- Proposed by Ravikumar et al. (PSDM, 2004)
- Use a secure dot product protocol to calculate distance metrics (two party protocol)
- TF-IDF (term-frequency, inverse document frequency)
  - Weighting scheme used to calculate Cosine similarity between text documents based on their term vectors
  - Soft TF-IDF (Cohen et al., KDD 2003) combines an approximate string comparison function with TF-IDF, leading to improved matching results
- Basic idea: Calculate stochastic dot product by sampling vector elements and use secure set intersection protocol to calculate similarity

Secure TF-IDF and Euclidean distance for PPRL (2)

- Calculate the secure dot product of two vectors $\vec{a}$ (held by *Alice*), and $\vec{b}$ (held by *Bob*) (vector elements are TF-IDF weights for tokens in records)

1. *Alice* calculates normalisation $z_A = \sum_i^n a_i$, with *n* being the dimension of vector $\vec{a}$ (*Bob* calculates z_B on his vector, also assumed to be of length *n*)
2. They each sample $k < n$ elements, $i \in \{1, \ldots, n\}$ with probability a_i / z_A into set T_A, or b_i / z_B into set T_B
3. Use secure set intersection cardinality protocol (Vaidya and Clifton, 2005) to find $v = |T_A \cap T_B|$, then average $v' = v / k$
4. Calculate dot product as: $v'' = v' * z_A * z_B$

Secure TF-IDF and Euclidean distance for PPRL (3)

- Experiments on bibliographic database *Cora* (records containing author names, article titles, dates, and venues of conferences and workshops)
- After around k = 1,000 samples (with n = 10,000, i.e. 10%), the secure stochastic scalar product achieved results comparable to the scalar product using the full vectors.
- Major drawback of this protocol: Requires k messages between *Alice* and *Bob* to calculate secure set intersection
- Not scalable to linking large databases

Q-gram based PPRL: Blindfolded record linkage (1)

- Proposed by Churches and Christen (Biomed Central, 2004 and PAKDD, 2004)
- Basic idea: Securely calculate Dice coefficient using a third party (*Carol*)
- Four step protocol
  1. *Alice* and *Bob* agree on data pre-processing steps, a one-way hash encoding algorithm, and secret key
  2. Convert their attribute values into q-gram lists, and get q-gram sub-lists (down to a certain minimum length) For example: ‘peter’ → *[‘pe’,‘et’,‘te’,‘er’], [‘et’,‘te’,‘er’], [‘pe’,‘te’,‘er’], [‘pe’,‘et’,‘er’], [‘pe’,‘et’,‘te’], [‘pe’,‘et’], [‘pe’,‘te’], [‘pe’,‘er’], [‘et’,‘te’], [‘et’,‘er’], [‘te’,‘er’]*

Q-gram based PPRL: Blindfolded record linkage (2)

- Four step protocol (continue)

3. For each record and attribute, and all q-gram sub-lists, *Alice* and *Bob* send 4-tuples to *Carol* with:
   – encrypted record identifier: **A.id** and **B.id**
   – hash encoded sub-list: **A.hsubl** and **B.hsubl**
   – num q-grams in sub-list: **A.subl_len** and **B.hsubl_len**
   – num q-grams in attribute: **A.val_len** and **B.val_len**

4. For each matching hash encoded q-gram sub-list (i.e. **A.hsubl** = **B.hsubl**), and for each unique pair of encrypted record identifiers, *Carol* can calculate the Dice co-efficient as

$$\mathit{sim}_{Dice} = \frac{2 \cdot \textbf{A.subl\_len}}{(\textbf{A.val\_len} + \textbf{B.val\_len})}$$

Q-gram based PPRL: Blindfolded record linkage (3)

- Simple example: *Alice* has (‘ra1’, ‘peter’) and *Bob* has (‘rb2’, ‘pete’) (and assume *q* = 2)
  - *Alice*’s quadruplets (shown unencoded):

    (‘ra1’, [‘pe’,‘et’,‘te’,‘er’], 4, 4),

    (‘ra1’, [‘et’,‘te’,‘er’], 3, 4),

    (‘ra1’, [‘pe’,‘te’,‘er’], 3, 4),

    (‘ra1’, [‘pe’,‘et’,‘er’], 3, 4), ↙ **A.subl_len** = 3

    (‘ra1’, **[‘pe’,‘et’,‘te’]**, 3, 4), etc. ← **A.val_len** = 4
  - *Bob*’s quadruplets:

    (‘rb2’, **[‘pe’,‘et’,‘te’]**, 3, 3), ← **B.subl_len** = 3

    (‘rb2’, [‘et’,‘te’], 2, 3), ↖ **B.val_len** = 3

    (‘rb2’, [‘pe’,‘te’], 2, 3),

    (‘rb2’, [‘pe’,‘et’], 2, 3), etc.

THE AUSTRALIAN NATIONAL UNIVERSITY

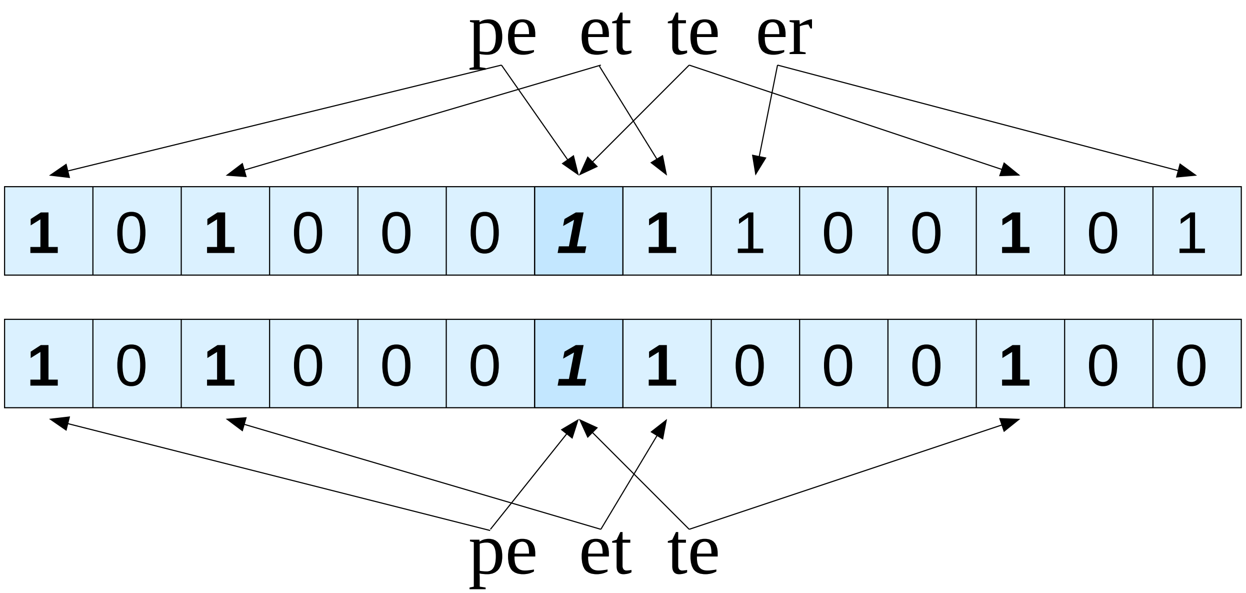
Q-gram based PPRL: Blindfolded record linkage (4)

- Several attributes can be compared independently (by different linkage units)
- These linkage units send their results to another party (*David*), who forms a (sparse) matrix by joining the results
- The final *matching weight* for a record pair is calculated by summing individual *sim*$_{Dice}$
- *David* arrives at a set of *blindly linked records* (triplets of [**A.id**, **B.id**, *sim*$_{total}$])
- Drawbacks: large communication overheads, *Carol* can mount a frequency attack (count how often certain hashed q-gram values appear)

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	0	**1**	0	0	0	***1***	**1**	1	0	0	**1**	0	1
1	0	**1**	0	0	0	***1***	**1**	0	0	0	**1**	0	0

- *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

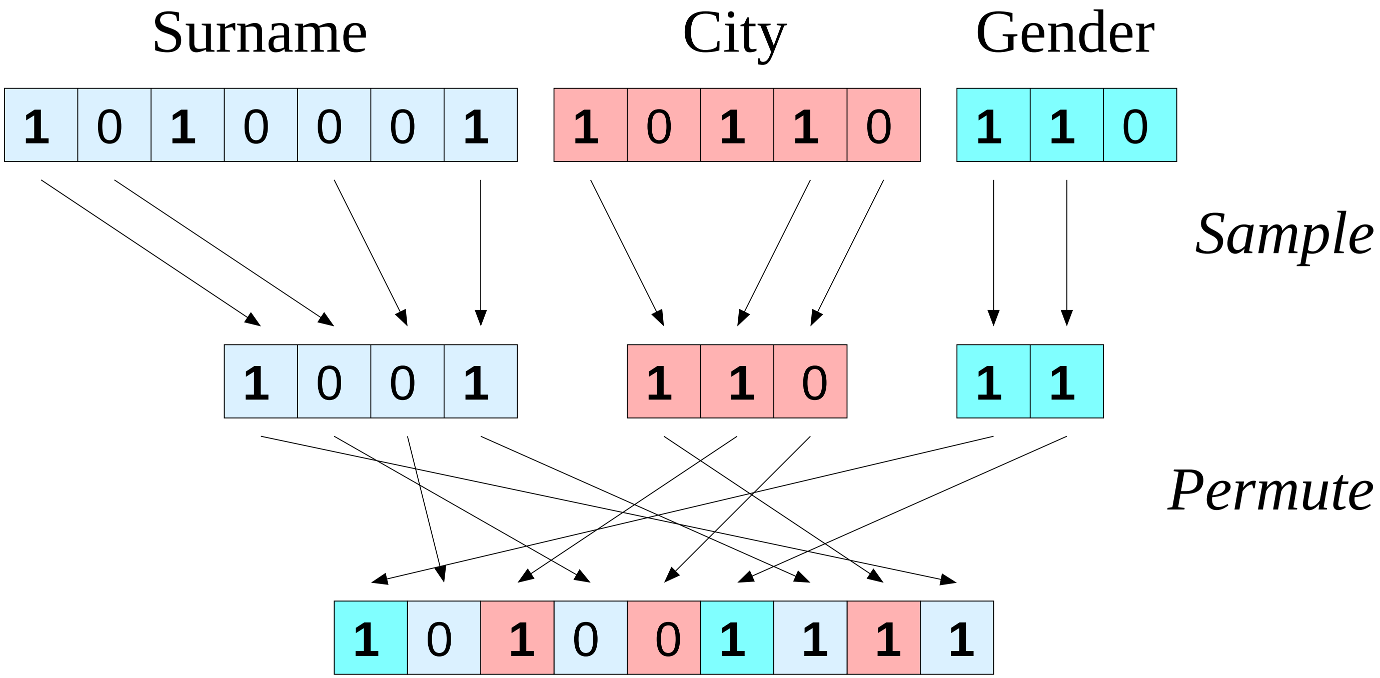
Bloom filter based PPRL (3)

- 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
- Kuzu et al. (PET, 2011) proposed a constraint satisfaction cryptanalysis attack (certain number of hash functions and Bloom filter length are vulnerable)
- To improve privacy, create record-level Bloom filter from several attribute-level Bloom filters (proposed by Schnell et al. (2011) and further investigated by Durham (2012) and Durham et al. (TKDE, 2013))

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)

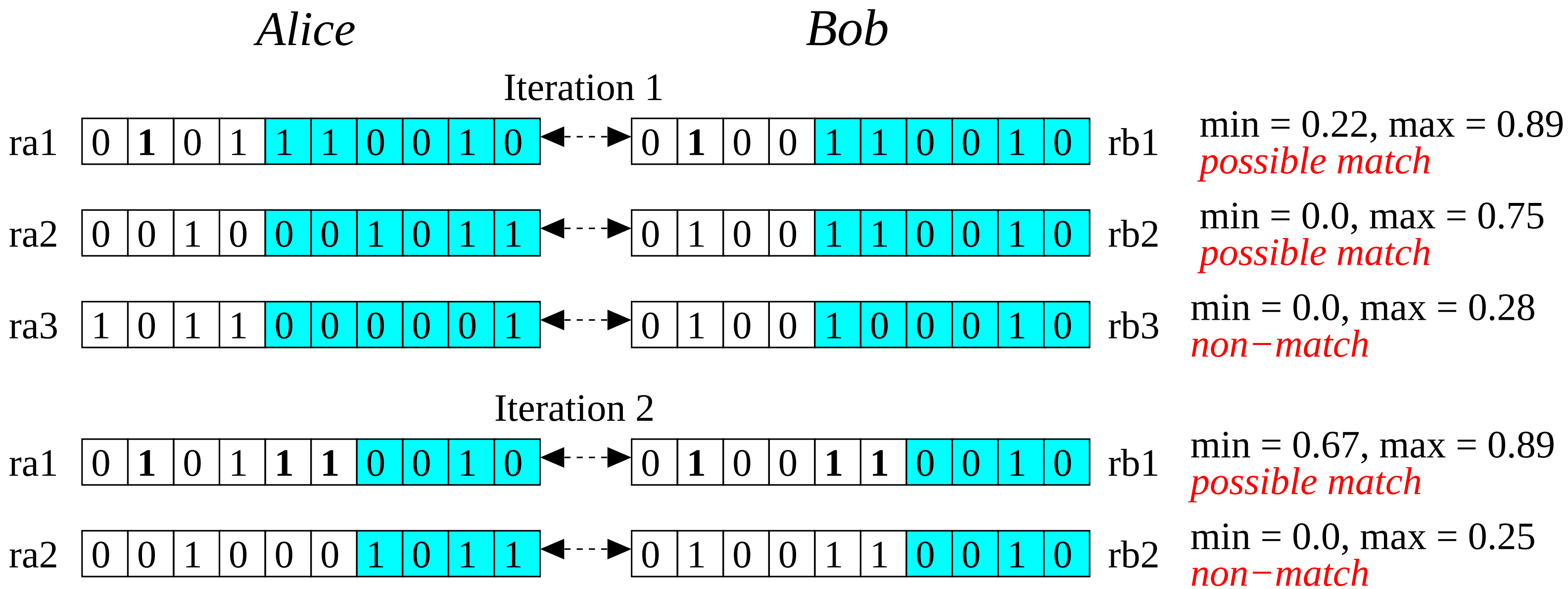


- Experimental results showed much improved security with regard to crypt-analysis attacks
- Scalability can be addressed by Locality Sensitive Hashing (LSH) based blocking → More in part 3

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)



Alice *Bob*

Iteration 1

ra1 0 **1** 0 1 1 1 0 0 1 0 ←- - -→ 0 **1** 0 0 1 1 0 0 1 0 rb1 min = 0.22, max = 0.89 *possible match*

ra2 0 0 1 0 0 0 1 0 1 1 ←- - -→ 0 1 0 0 1 1 0 0 1 0 rb2 min = 0.0, max = 0.75 *possible match*

ra3 1 0 1 1 0 0 0 0 0 1 ←- - -→ 0 1 0 0 1 0 0 0 1 0 rb3 min = 0.0, max = 0.28 *non−match*

Iteration 2

ra1 0 **1** 0 1 **1** **1** 0 0 1 0 ←- - -→ 0 **1** 0 0 **1** **1** 0 0 1 0 rb1 min = 0.67, max = 0.89 *possible match*

ra2 0 0 1 0 0 0 1 0 1 1 ←- - -→ 0 1 0 0 1 1 0 0 1 0 rb2 min = 0.0, max = 0.25 *non−match*

- 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*$_{Dice}(ra3, rb3))$ $=$ 2×1/(4+3)= 2/7 = 0.28

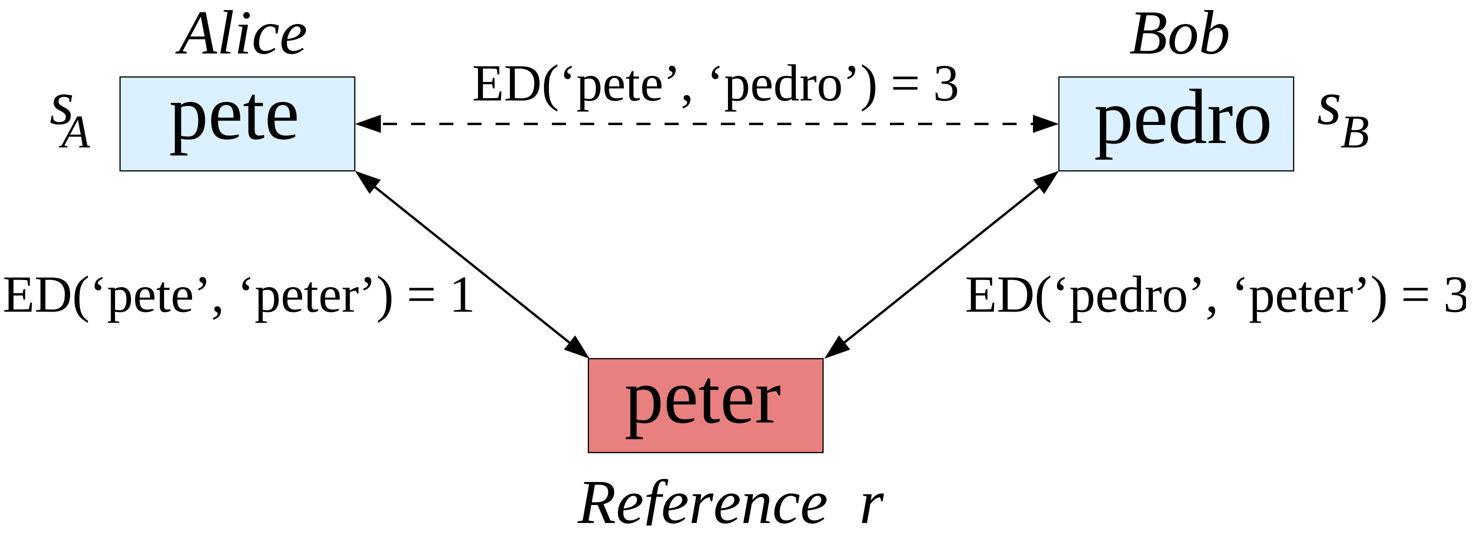
Reference value based PPRL (1)

- Proposed by Pang et al. (IPM, 2009)
- Basic idea: Use large public list of reference (string) values available to both *Alice* and *Bob*, and calculate distance estimates based on triangular inequality
- Assume reference value r and private values s_A held by *Alice* and s_B held by *Bob*, and edit-distance function ED(s_A, s_B):

$$\text{ED}(s_A, s_B) \leq \text{ED}(s_A, r) + \text{ED}(s_B, r)$$

- The third party calculates these distances based on encoded string and reference values

Reference value based PPRL (2)



- If s_A and s_B are compared with several reference values, the mean of distance estimates is used
- This approach can be employed with different (string) distance measures (but: not all are distance metrics!)
- A scalable approach if private values are only compared with 'similar' reference values (neighbourhood clustering)

THE AUSTRALIAN NATIONAL UNIVERSITY

Reference value based PPRL (3)

- Major drawback: Security issues, as third party can conduct analysis of string distances and size of cluster neighbourhoods (assuming the reference table is available to the third party)
- The size of clusters and the distribution of distances in a cluster can allow identification of rare names (for each reference value, there will be a specific distribution of how many other reference values there are with a distance of 1, 2, 3, etc. edits)

  For example:

  ‘new york’: [ed1=5, ed2=15, ed3=154, ed4=4371, . . .]
  ‘wollongong’: [ed1=0, ed2=0, ed3=4, ed4=5, . . .]

Reference value based PPRL (4)

- Security issues can be overcome by
  - Aiming to have all clusters being the same size
  - Use relative distances (add or subtract constant to all distances sent to the linkage unit)
- Recent, Vatsalan et al. proposed a two-party protocol based on reference values (AusDM, 2011)
  - Basic idea is to use binning of similarity values to hide actual values between the two database owners
  - Use of the reverse triangular inequality for similarities rather than distances (for classification of record pairs)
  - Scalability is achieved through the use of phonetic encoding to generate blocks (clusters)

Phonetic encoding based PPRL (1)

- Proposed by Karakasidis and Verykios (BCI, 2009)
- Use phonetic encoding functions (like Soundex, NYSIIS, Double-Metaphone, etc.) to generalise and obfuscate sensitive values

Soundex(‘peter’) = ‘p360’	Soundex(‘gail’) = ‘g400’
Soundex(‘pedro’) = ‘p360’	Soundex(‘gayle’) = ‘g400’

- Basic idea: Two database owners phonetically encode (and one-way hash-encode) their values, add ‘faked’ encoded phonetic values, and send these to a third party to conduct the linking
- The use of computationally fast phonetic algorithms make this an efficient approach

ANU
THE AUSTRALIAN NATIONAL UNIVERSITY

Phonetic encoding based PPRL (2)

- The quantitative measuring of privacy by means of Relative Information Gain (RIG) is used (Karakasidis et al., DPM, 2011)
  - Low RIG means no information can be gained from encoded phonetic values only
  - It is shown that phonetic codes do provide privacy
- Privacy is achieved in three ways:
  1. Generalisation properties of phonetic encoding (converting similar values into the same codes)
  2. Injection of fake codes (obfuscation), to maximise privacy in terms of RIG
  3. Secure hash encoding of all values communicated

Tutorial Outline

- Background to record linkage and PPRL
  - Applications, history, challenges, the record linkage and PPRL process
  - Scenarios, a definition, and a taxonomy for PPRL
- Exact and approximate PPRL techniques
  - Basic protocols for PPRL (two and three parties)
  - Hash-encoding for exact matching, and ↙ Tea break key techniques for approximate comparison
- Selected key techniques for scalable PPRL
  - Incl. private blocking; Bloom filters; hybrid, public reference, and differential privacy approaches, etc.
- Conclusions and challenges

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

Blocking aware private record linkage (1)

- Proposed by Al-Lawati et al. (IQIS, 2005)
- A three party protocol featuring the first attempt for private blocking to make PPRL scalable
- Basic idea: Private record linkage is achieved by using hash signatures based on TF-IDF vectors
- These vectors are built on tokens (unigrams) extracted from attribute values
- Three blocking approaches were presented, they provide a trade-off between performance and privacy achieved

Blocking aware private record linkage (2)

Database *A*

ID	Value
a1	{‘a’, ‘b’}
a2	{‘c’}

Database *B*

ID	Value
b1	{‘b’}
b2	{‘a’, ‘b’}

	F[0]	F[1]	F[2]	F[3]
HS(a1)	TF-IDF(a1,‘b’)	0	0	TF-IDF(a1,‘a’)
HS(a2)	0	0	TF-IDF(a2,‘c’)	0
HS(b1)	TF-IDF(b1,‘b’)	0	0	0
HS(b2)	TF-IDF(b2,‘b’)	0	0	TF-IDF(b2,‘a’)

(F is an array of floating-point numbers)

- Database owners can independently generate their TF-IDF weight vectors, and encode them into hash signatures (HS)
- Sent to a third party, which can calculate Cosine similarity

Blocking aware private record linkage (3)

- Three blocking approaches based on token intersection (Jaccard similarity): Records are only compared if their token intersection is non-empty
  - *Simple blocking*: a separate block is generated for each token in a record
  - *Record-aware blocking*: combines the hash signature of each record with a record ID so that duplicates appearing in simple blocking are eliminated
  - *Frugal third party blocking*: the database owners do a secure set intersection to identify common blocks
- All three blocking approaches are vulnerable to frequency attacks (database, block and vocabulary sizes, and record length)

Privacy-preserving schema and data matching (1)

- Proposed by Scannapieco et al. (SIGMOD, 2007)
- Schema matching is achieved by using an intermediate ‘global’ schema sent by the linkage unit (third party) to the database owners
  - The database owners assign each of their linkage attributes to the global schema
  - They send their hash-encoded attribute names to the linkage unit
- Basic idea of record linkage is to map attribute values into a multi-dimensional space such that distances are preserved (using the *SparseMap* algorithm)

THE AUSTRALIAN NATIONAL UNIVERSITY

Privacy-preserving schema and data matching (2)

- Three phases involving three parties
- Phase 1: Setting the embedding space
  - Database owners agree upon a set of (random) reference strings (known to both)
  - Each reference string is represented by a vector in the embedding space
- Phase 2: Embedding of database records into space using *SparseMap*
  - Essentially, vectors of the distances between reference and database values are calculated
  - Resulting vectors are sent to the third party

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

Privacy-preserving schema and data matching (3)

- Phase 3: Third party stores vectors in a multi-dimensional index and conducts a nearest-neighbour search (vectors close to each other are classified as matches)
- Major drawbacks:
  - Matching accuracy depends upon parameters used for the embedding (dimensionality and distance function)
  - Certain parameter settings give very low matching precision results
  - Multi-dimensional indexing becomes less efficient with higher dimensionality
  - Susceptible to frequency attacks (closeness of nearest neighbours in multi-dimensional index)

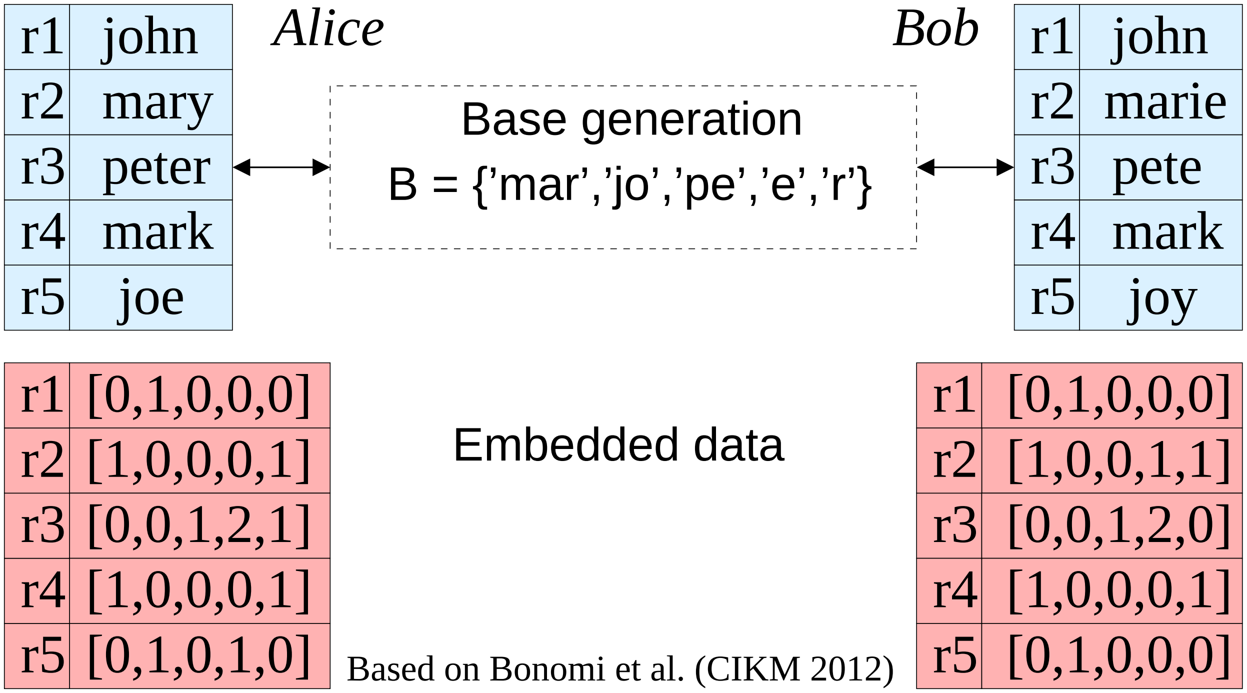
Efficient private record linkage

- Proposed by Yakout et al. (ICDE, 2009)
- Convert the three-party protocol by Scannapieco et al. into a two-party protocol
- Basic idea:
  - Embed records into a multi-dimensional space, then map them into complex numbers
  - Exchange these complex numbers between the database owners
  - Possible matching record pairs are those which have complex numbers within a certain maximum distance
  - Calculate actual distances between records using a secure scalar product based on random records

Frequent grams based embedding for PPRL

- Proposed by Bonomi et al. (CIKM, 2012)
- Embedding based on frequent q-grams mined from databases using prefix-tree pattern mining

  (counts of q-grams, which can have different lengths, are modified by differential privacy Laplace noise)



Alice

r1	john
r2	mary
r3	peter
r4	mark
r5	joe

Bob

r1	john
r2	marie
r3	pete
r4	mark
r5	joy

Base generation

B = {’mar’,’jo’,’pe’,’e’,’r’}

Embedded data

r1	[0,1,0,0,0]
r2	[1,0,0,0,1]
r3	[0,0,1,2,1]
r4	[1,0,0,0,1]
r5	[0,1,0,1,0]

r1	[0,1,0,0,0]
r2	[1,0,0,1,1]
r3	[0,0,1,2,0]
r4	[1,0,0,0,1]
r5	[0,1,0,0,0]

Based on Bonomi et al. (CIKM 2012)

A hybrid approach to PPRL (1)

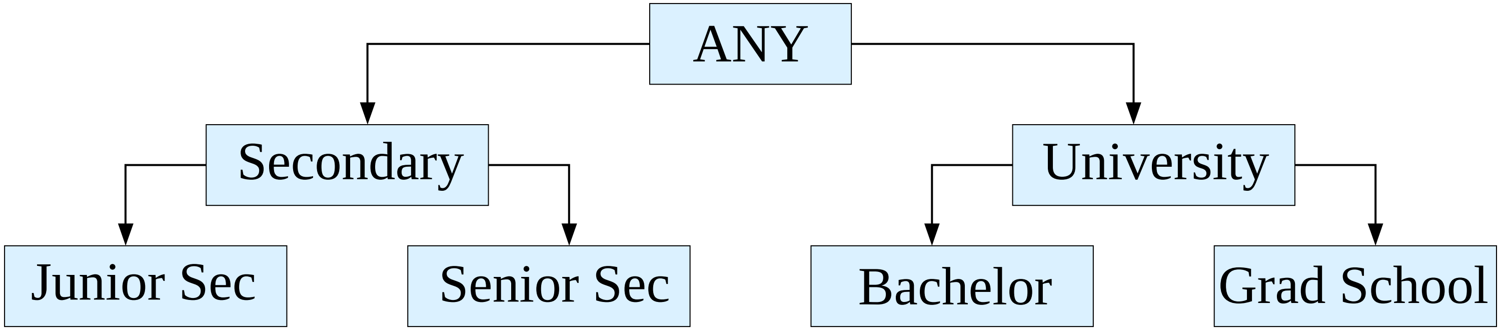
- Proposed by Inan et al. (ICDE, 2008)
- Use k-anonymity to generalise (sanitise) databases and find ‘blocks’ of possible matching record pairs
- Basic idea: In a first step, generate *value generalisation hierarchies* (VGH); in a second step calculate distances between records with same generalised values using a secure multi-party 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 alpha-numeric values (such as names) that do not have a VGH

PPRL using differential privacy (1)

- Proposed by Inan et al. (EDBT, 2010)
- A modification of their k-anonymity generalisation approach (improved security, and no third party required)
- Use a differential privacy based approach for blocking (differential privacy boils down to adding noise to aggregate queries in statistical database to avoid disclosure by combining results)
- Basic idea: the database owners disclose only the perturbed results of a set of statistical queries, and use special indexing techniques that are compliant with differential privacy

PPRL using differential privacy (2)

- Database owners partition their data into sub-sets, and exchange their size and extend
  - Spatial indexing techniques (BSP-, KD-, or R-Tree) are used to form sub-sets (hyper-rectangles)
  - Blocking phase filters out pairs of sub-sets that cannot contain matches
  - Construct transcripts that satisfy differential privacy (add output perturbation)
  - The way queries for the transcripts are generated is a crucial aspect of this approach
- SMC approach based on homomorphic encryption is used to calculate similarities for record pairs not removed by blocking

Hamming LSH blocking for Bloom filters

- Durham (2012) proposed to use Hamming based Locality Sensitive Hashing (LSH) to make the composite Bloom filter approach scalable
  - Hamming distance on Bloom filters: Number of bits where two Bloom filters differ
- Hamming LSH: Randomly select ϕ bits from composite Bloom filter, iterate μ times
- All records that have the same pattern in the ϕ selected bits are inserted into a block
- Because record pair are potentially compared up-to μ times, a hash-table or database is needed (scalability is sensitive to choice of parameter values)

Reference table based private blocking (1)

- Proposed by Karakasidis and Verykios (SAC, 2012)
- Based on the intuition that if two data elements are similar to a third one, they are very likely to be similar with each other
- Idea is to generate k-anonymous blocks using public reference values (blocks containing at least *k* values)
- May be combined with any private matching method
- Some information is leaked because clusters are likely of different sizes (depending upon distribution of database values)

Reference table based private blocking (2)

- The method consists of the following steps

1. Data holders agree on a common publicly available corpus of data, called reference table
2. They cluster the reference table data using the nearest neighbour clustering algorithm (with cluster size of *k* or more to assure k-anonymous blocks)
3. Each database attribute value is assigned to its closest cluster, and values in the same cluster form a block
4. The number of blocks formed is equal to the number of reference table clusters
5. The blocks are sent to a third party and records from corresponding blocks are privately matched using any private approximate matching algorithm

Hierarchical clustering based PPRL

- Proposed by Kuzu et al. (EDBT, 2013)
- In a three party protocol, public reference values are clustered using agglomerate hierarchical clustering (done by the third party)
- Then record values are placed in their closest clusters (using single link approach)
- Cluster sizes are perturbed using differential privacy (Laplace noise based addition of random records — no records are removed!)
- SMC-based detailed comparison of the record pairs in the same block (i.e. same cluster) using Paillier cryptosystem (so the third party does not learn similarities)

Sorted neighbourhood clustering based private blocking (1)

- Proposed by Vatsalan et al. (PAKDD, 2013)
- Record values are clustered based on the records' sorting key values to generate k-anonymous clusters, each represented by one or several public reference values
- K-anonymous clusters (with encrypted record IDs and unencrypted reference values) are sent to a third party
- The third party sorts the clusters and merges neighbouring clusters from both database owners based on the common reference values to generate candidate record pairs

Sorted neighbourhood clustering based private blocking (2)

- Sorted neighbourhood clustering is more efficient compared to other blocking techniques in terms of number of candidate record pairs generated (experimental evaluation presented next)
- Also more secure due to more uniform block sizes generated (making frequency attacks more difficult)
- Converted the three-party sorted neighbourhood clustering into a two-party solution:

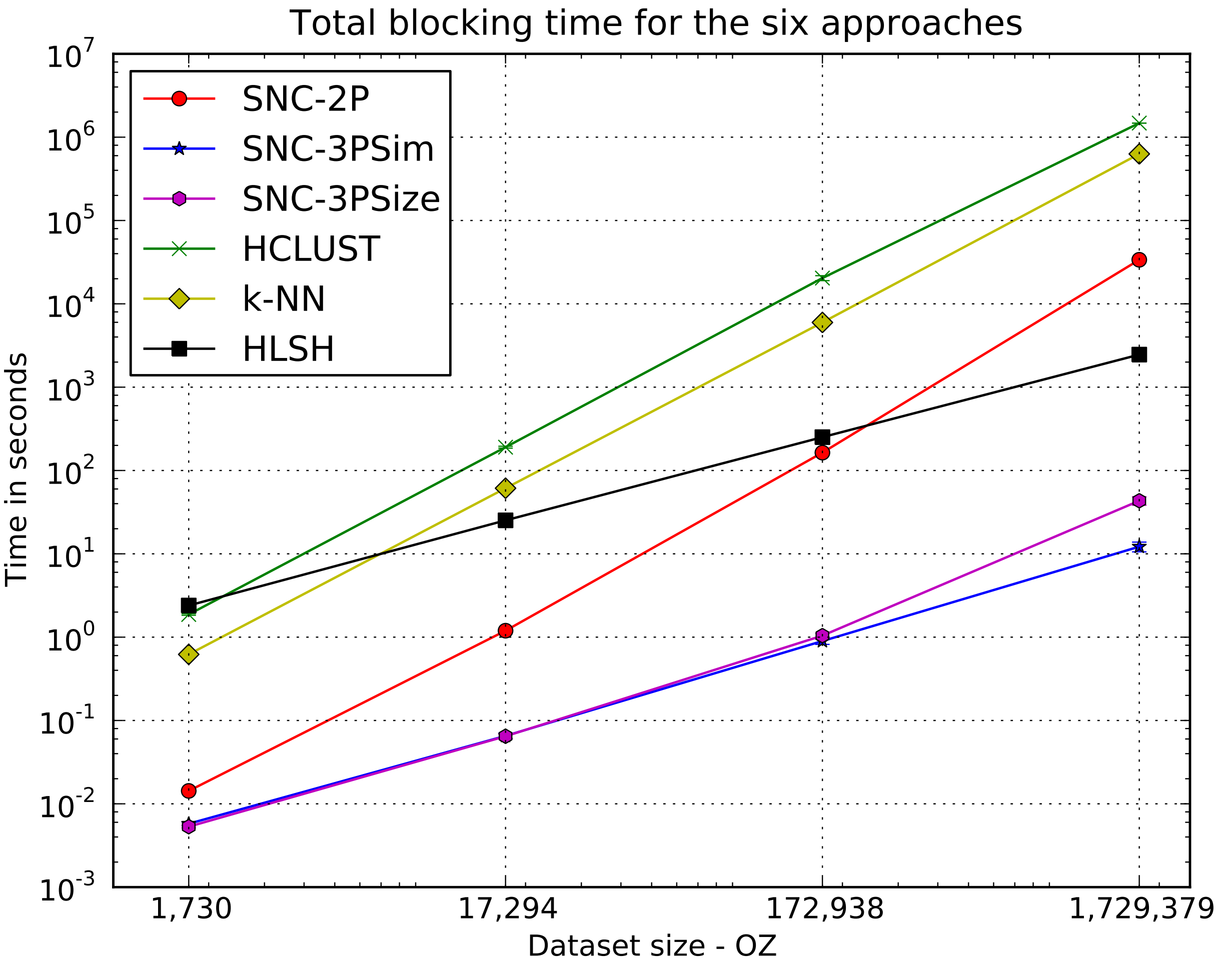
  Efficient two-party private blocking based on sorted nearest neighborhood clustering
  CIKM paper 636, Session 38, Thursday 9:45

Experimental comparison of scalable PPRL techniques (1)

- Experiments conducted on two real databases
  - Australian telephone database (OZ), 1,729,379 records
  - North Carolina voter database (NC), 629,362 records
- Used attributes like first and last name, street address, city, and zipcode
- For the OZ data we artificially added variations and typos (as the data set does not include duplicates)
- For the NC data, voter IDs are ‘ground truth’ (significant processing to remove exact duplicates, etc.)
- Data sets are available — talk to use after tutorial

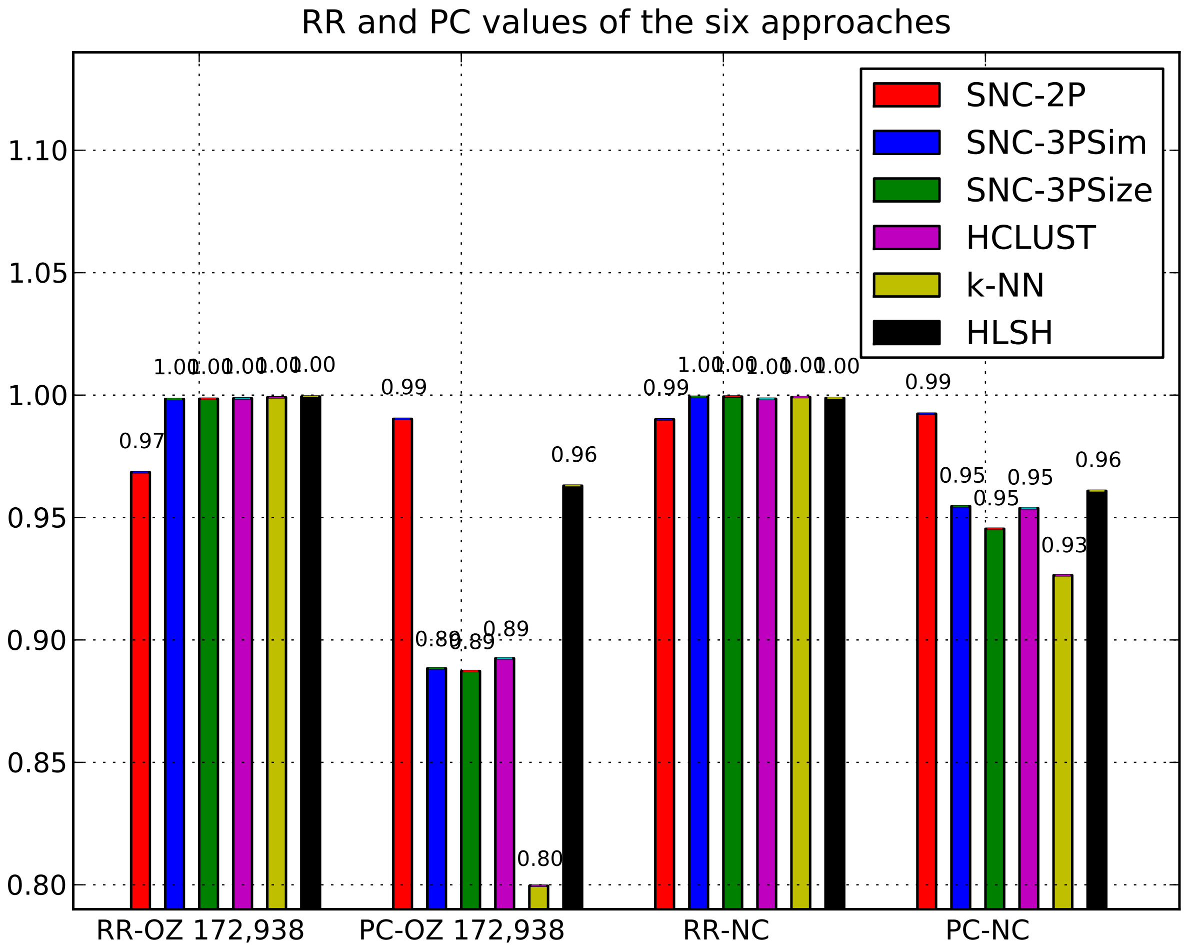
Experimental comparison of scalable PPRL techniques (2)

- Different sizes of OZ data sets generated to evaluate scalability (measured by total run time)



Experimental comparison of scalable PPRL techniques (3)

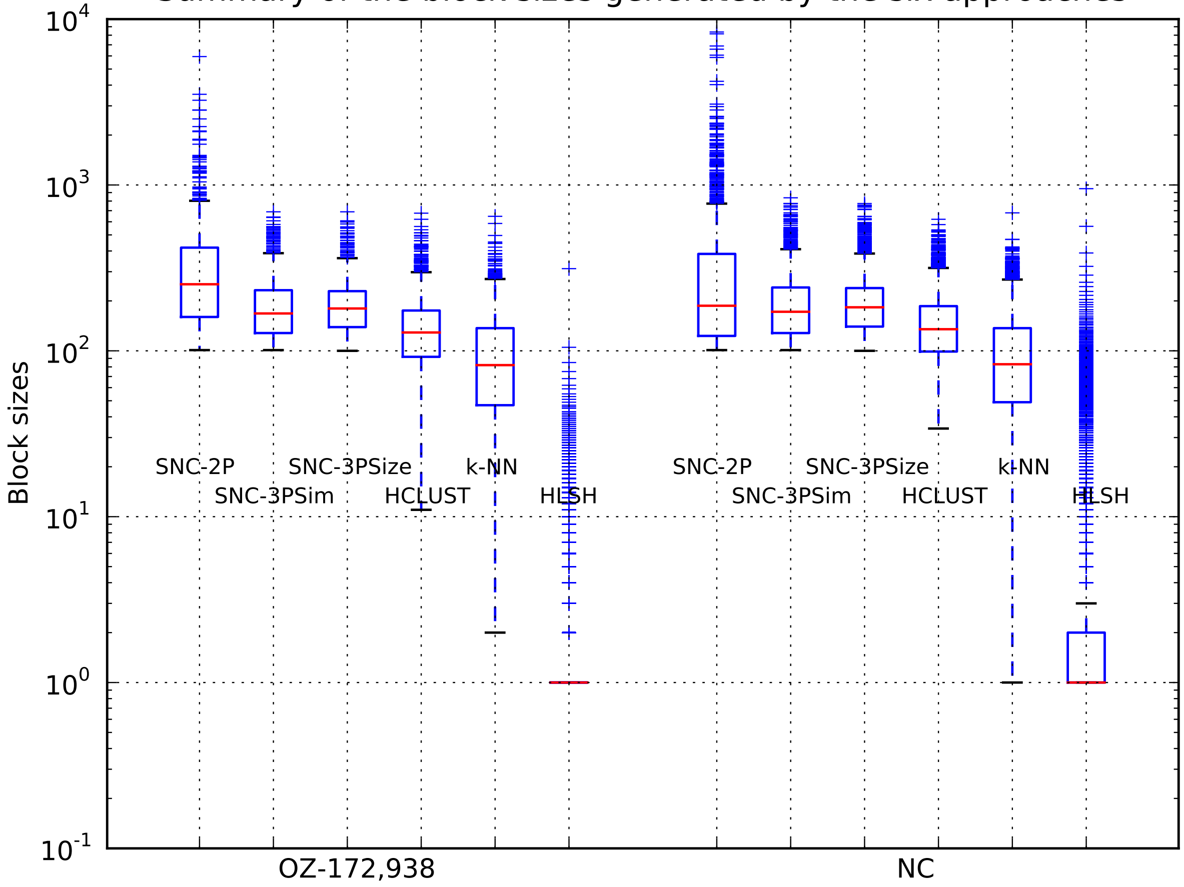
- Quality of blocking on the OZ-172,938 and NC data sets (measured by reduction ratio, RR, and pairs completeness, PC)



Experimental comparison of scalable PPRL techniques (4)

- Privacy of blocking on the OZ-172,938 and NC data sets (measured by block sizes generated - frequency attack)

Summary of the block sizes generated by the six approaches



Tutorial Outline

- Background to record linkage and PPRL
  - Applications, history, challenges, the record linkage and PPRL process
  - Scenarios, a definition, and a taxonomy for PPRL
- Exact and approximate PPRL techniques
  - Basic protocols for PPRL (two and three parties)
  - Hash-encoding for exact matching, and ↙ Tea break key techniques for approximate comparison
- Selected key techniques for scalable PPRL
  - Incl. private blocking; Bloom filters; hybrid, public reference, and differential privacy approaches, etc.
- Conclusions and challenges

Conclusions

- 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 (1)

- 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)

Challenges and future work (2)

- A framework for PPRL is needed
  - To facilitate comparative experimental evaluation of PPRL techniques
  - Needs to allow researchers to plug-in their techniques
  - Benchmark data sets are required (biggest challenge, as such data is sensitive!)
- 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

Thank you for attending our tutorial!

Enjoy the rest of CIKM and your stay in San Francisco...

For questions please contact:

peter.christen@anu.edu.au

verykios@eap.gr

dinusha.vatsalan@anu.edu.au

References (1)

- Agrawal R, Evfimievski A, and Srikant R: *Information sharing across private databases.* ACM SIGMOD, San Diego, 2005.
- Al-Lawati A, Lee D and McDaniel P: *Blocking-aware private record linkage.* IQIS, Baltimore, 2005.
- Atallah MJ, Kerschbaum F and Du W: *Secure and private sequence comparisons.* WPES, Washington DC, pp. 39–44, 2003.
- Bachteler T, Schnell R, and Reiher J: *An empirical comparison of approaches to approximate string matching in private record linkage.* Statistics Canada Symposium, 2010.
- Blakely T, Woodward A and Salmond C: *Anonymous linkage of New Zealand mortality and census data.* ANZ Journal of Public Health, 24(1), 2000.
- Barone D, Maurino A, Stella F, and Batini C: *A privacy-preserving framework for accuracy and completeness quality assessment.* Emerging Paradigms in Informatics, Systems and Communication, 2009.
- Bhattacharya, I and Getoor, L: *Collective entity resolution in relational data.* ACM TKDD, 2007.

References (2)

- Bloom, BH: *Space/time trade-offs in hash coding with allowable errors.* Communications of the ACM, 1970.
- Bonomi L, Xiong Li, Chen R, and Fung B: *Frequent grams based embedding for privacy preserving record linkage.* ACM Information and knowledge management, 2012.
- Bouzelat H, Quantin C, and Dusserre L: *Extraction and anonymity protocol of medical file.* AMIA Fall Symposium, 1996.
- Chaytor R, Brown E and Wareham T: *Privacy advisors for personal information management.* SIGIR workshop on Personal Information Management, Seattle, pp. 28–31, 2006.
- Christen P: *Privacy-preserving data linkage and geocoding: Current approaches and research directions.* PADM held at IEEE ICDM, Hong Kong, 2006.
- Christen P: *Geocode Matching and Privacy Preservation.* ACM PinKDD, 2009.
- Christen, P: *A survey of indexing techniques for scalable record linkage and deduplication.* IEEE TKDE, 2012.

References (3)

- Christen, P: *Data matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer, 2012.
- Christen, P: *Preparation of a real voter data set for record linkage and duplicate detection research.* Technical Report, The Australian National University, 2013.
- Christen P and Churches T: *Secure health data linkage and geocoding: Current approaches and research directions.* ehPASS, Brisbane, 2006.
- Christen, P and Goiser, K: *Quality and complexity measures for data linkage and deduplication.* In *Quality Measures in Data Mining.* Springer Studies in Computational Intelligence, vol. 43, 2007.
- Churches T: *A proposed architecture and method of operation for improving the protection of privacy and confidentiality in disease registers.* BMC Medical Research Methodology, 3(1), 2003.
- Churches T and Christen P: *Some methods for blindfolded record linkage.* BMC Medical Informatics and Decision Making, 4(9), 2004.
- Clifton C, Kantarcioglu M, Vaidya J, Lin X, and Zhu MY: *Tools for privacy preserving distributed data mining.* ACM SIGKDD Explorations, 2002.

References (4)

- Clifton C, Kantarcioglu M, Doan A, Schadow G, Vaidya J, Elmagarmid AK and Suciu D: *Privacy-preserving data integration and sharing.* SIGMOD workshop on Research Issues in Data Mining and Knowledge Discovery, Paris, 2004.
- Du W, Atallah MJ, and Kerschbaum F: *Protocols for secure remote database access with approximate matching.* ACM Workshop on Security and Privacy in E-Commerce, 2000.
- Dusserre L, Quantin C and Bouzelat H: *A one way public key cryptosystem for the linkage of nominal files in epidemiological studies.* Medinfo, 8:644-7, 1995.
- Durham, EA: *A framework for accurate, efficient private record linkage.* PhD Thesis, Vanderbilt University, 2012.
- Durham, EA, Toth C, Kuzu, M. Kantarcioglu M, and Malin B: *Composite Bloom for secure record linkage.* IEEE Transactions on Knowledge and Data Engineering, 2013.
- Durham, EA, Xue Y, Kantarcioglu M, and Malin B: *Private medical record linkage with approximate matching.* AMIA Annual Symposium, 2010.

References (5)

- Durham, EA, Xue Y, Kantarcioglu M, and Malin B: *Quantifying the correctness, computational complexity, and security of privacy-preserving string comparators for record linkage.* Information Fusion, 2012.
- Dwork, C: *Differential privacy.* International Colloquium on Automata, Languages and Programming, 2006.
- Elmagarmid AK, Ipeirotis PG and Verykios VS: *Duplicate record detection: A survey.* IEEE TKDE 19(1), pp. 1–16, 2007.
- Fienberg SE: *Privacy and confidentiality in an e-Commerce World: Data mining, data warehousing, matching and disclosure limitation.* Statistical Science, IMS Institute of Mathematical Statistics, 21(2), pp. 143–154, 2006.
- Hall R and Fienberg SE: *Privacy-preserving record linkage.* Privacy in Statistical Databases, Springer LNCS 6344, 2010.
- Herzog TN, Scheuren F, and Winkler WE: *Data quality and record linkage techniques.* Springer, 2007.
- Ibrahim A, Jin H, Yassin AA, and Zou D: *Approximate Keyword-based Search over Encrypted Cloud Data.* IEEE ICEBE, pp. 238–245, 2012.

References (6)

- Inan A, Kantarcioglu M, Bertino E and Scannapieco M: *A hybrid approach to private record linkage.* IEEE ICDE, Cancun, Mexico, pp. 496–505, 2008.
- Inan A, Kantarcioglu M, Ghinita G, and Bertino E: *Private record matching using differential privacy.* International Conference on Extending Database Technology, 2010.
- Jones JJ, Bond RM, Fariss CJ, Settle JE, Kramer ADI, Marlow C, and Fowler JH: *Yahtzee: An anonymized group level matching procedure.* PloS one, vol. 8, 2013.
- Kantarcioglu M, Jiang W, and Malin B: *A privacy-preserving framework for integrating person-specific databases.* Privacy in Statistical Databases, 2008.
- Kantarcioglu M, Inan A, Jiang W and Malin B: *Formal anonymity models for efficient privacy-preserving joins.* Data and Knowledge Engineering, 2009.
- Karakasidis A and Verykios VS: *Privacy preserving record linkage using phonetic codes.* IEEE Balkan Conference in Informatics, 2009.
- Karakasidis A and Verykios VS: *Advances in privacy preserving record linkage.* E-activity and Innovative Technology, Advances in Applied Intelligence Technologies Book Series, IGI Global, 2010.

References (7)

- Karakasidis A and Verykios VS: *Secure blocking+secure matching = Secure record linkage.* Journal of Computing Science and Engineering, 2011.
- Karakasidis A, Verykios VS, and Christen P: *Fake injection strategies for private phonetic matching.* International Workshop on Data Privacy Management, 2011.
- Karakasidis A and Verykios VS: *Reference table based k-anonymous private blocking.* Symposium on Applied Computing, 2012.
- Karakasidis A and Verykios VS: *A sorted neighborhood Approach to multidimensional privacy preserving blocking.* IEEE ICDM workshop, 2012.
- Karapiperis D and Verykios VS: *A distributed framework for scaling Up LSH-based computations in privacy preserving record linkage.* Balkan Conference in Informatics, 2013.
- Kelman CW, Bass AJ and Holman CDJ: *Research use of linked health data – A best practice protocol.* ANZ Journal of Public Health, 26(3), pp. 251–255, 2002.
- Kuzu M, Kantarcioglu M, Durham EA and Malin B: *A constraint satisfaction cryptanalysis of Bloom filters in private record linkage.* Privacy Enhancing Technologies, 2011.

References (8)

- Kuzu M, Kantarcioglu M, Inan A, Bertino E, Durham EA and Malin B: *Efficient privacy-aware record integration.* ACM Extending Database Technology, 2013.
- Kuzu M, Kantarcioglu M, Durham EA, Toth C, and Malin B: *A practical approach to achieve private medical record linkage in light of public resources.* Journal of the American Medical Informatics Association, vol. 20, pp. 285–292, 2013.
- Lai PK, Yiu SM, Chow KP, Chong CF, and Hui LC: *An efficient Bloom filter based solution for multiparty private matching.* International Conference on Security and Management, 2006.
- Li Y, Tygar JD and Hellerstein JM: *Private matching.* Computer Security in the 21st Century, Lee DT, Shieh SP and Tygar JD (editors), Springer, 2005.
- Li F, Chen Y, Luo B, Lee D, and Liu P: *Privacy preserving group linkage.* Scientific and Statistical Database Management, 2011.
- Malin B, Airoldi E, Edoho-Eket S and Li Y: *Configurable security protocols for multi-party data analysis with malicious participants.* IEEE ICDE, Tokyo, pp. 533–544, 2005.

References (9)

- Malin B and Sweeney L: *A secure protocol to distribute unlinkable health data.* American Medical Informatics Association 2005 Annual Symposium, Washington DC, pp. 485–489, 2005.
- Mohammed N, Fung BC and Debbabi M: *Anonymity meets game theory: secure data integration with malicious participants.* VLDB Journal, 2011.
- Murugesan M, Jiang W, Clifton C, Si L and Vaidya J: *Efficient privacy-preserving similar document detection.* VLDB Journal, 2010.
- Naumann F and Herschel M: *An introduction to duplicate detection.* Synthesis Lectures on Data Management, Morgan and Claypool Publishers, 2010.
- Navarro-Arribas G and Torra V: *Information fusion in data privacy: A survey.* Information fusion, 2012.
- O'Keefe CM, Yung M, Gu L and Baxter R: *Privacy-preserving data linkage protocols.* WPES, Washington DC, pp. 94–102, 2004.
- Pang C, Gu L, Hansen D and Maeder A: *Privacy-preserving fuzzy matching using a public reference table.* Intelligent Patient Management, 2009.

References (10)

- Quantin C, Bouzelat H and Dusserre L: *Irreversible encryption method by generation of polynomials.* Medical Informatics and The Internet in Medicine, Informa Healthcare, 21(2), pp. 113–121, 1996.
- Quantin C, Bouzelat H, Allaert FAA, Benhamiche AM, Faivre J and Dusserre L: *How to ensure data quality of an epidemiological follow-up: Quality assessment of an anonymous record linkage procedure.* International Journal of Medical Informatics, 49, pp. 117–122, 1998.
- Quantin C, Bouzelat H, Allaert FAA, Benhamiche AM, Faivre J and Dusserre L: *Automatic record hash coding and linkage for epidemiological follow-up data confidentiality.* Methods of Information in Medicine, Schattauer, 37(3), pp. 271–277, 1998.
- Ravikumar P, Cohen WW and Fienberg SE: *A secure protocol for computing string distance metrics.* PSDM held at IEEE ICDM, Brighton, UK, 2004.
- Scannapieco M, Figotin I, Bertino E and Elmagarmid AK: *Privacy preserving schema and data matching.* ACM SIGMOD, 2007.

References (11)

- Schadow G, Grannis SJ and McDonald CJ: *Discussion paper: Privacy-preserving distributed queries for a clinical case research network.* CRPIT'14: Proceedings of the IEEE international Conference on Privacy, Security and Data Mining, Maebashi City, Japan, pp. 55–65, 2002.
- Schnell R, Bachteler T and Reiher J: *Privacy-preserving record linkage using Bloom filters.* BMC Medical Informatics and Decision Making, 9(1), 2009.
- Schnell R, Bachteler T and Reiher J: *A novel error-tolerant anonymous linking code.* German record linkage center working paper series, 2011.
- Schnell R: *Privacy-preserving record linkage and privacy-preserving blocking for large files with cryptographic keys using multibit trees.* ASA JSM Proceedings, Alexandria, VA, 2013.
- Sweeney L: *Privacy-enhanced linking.* ACM SIGKDD Explorations, 7(2), 2005.
- Trepetin S: *Privacy-preserving string comparisons in record linkage systems: a review.* Information Security Journal: A Global Perspective, 2008.
- Vatsalan D, Christen P and Verykios VS: *An efficient two-party protocol for approximate matching in private record linkage.* AusDM, CRPIT, 2011.

References (12)

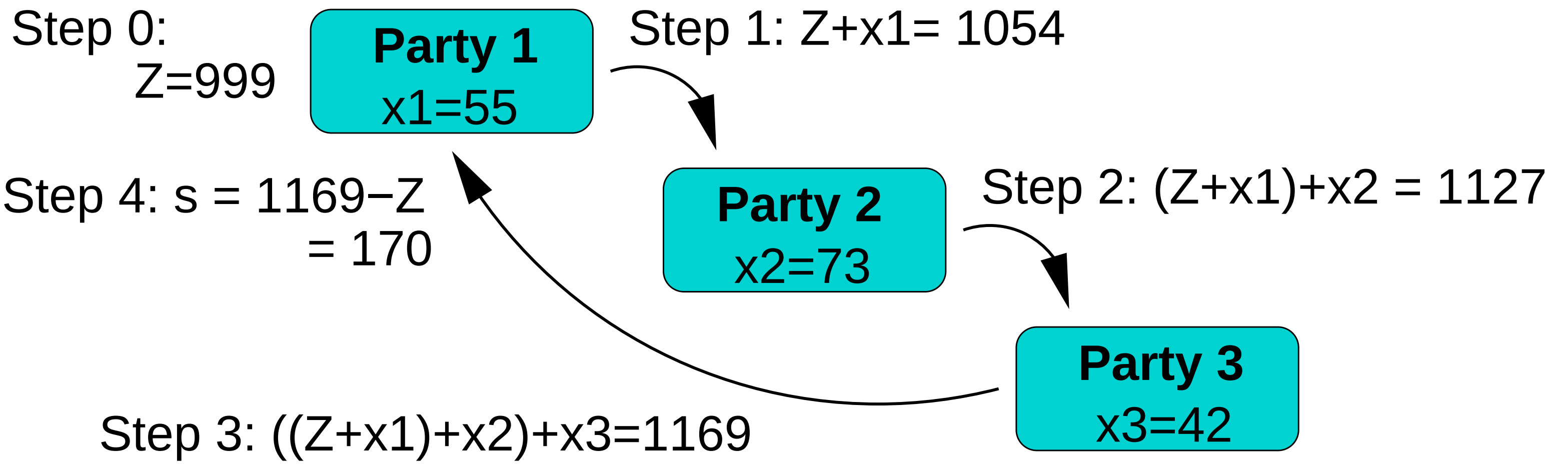
- Vatsalan D and Christen P: *An iterative two-party protocol for scalable privacy-preserving record linkage.* AusDM, CRPIT, vol. 134, 2012.
- Vatsalan D and Christen P: *Sorted nearest neighborhood clustering for efficient private blocking.* PAKDD, Gold Coast, Australia, Springer LNCS vol. 7819, 2013.
- Vatsalan D, Christen P and Verykios VS: *A taxonomy of privacy-preserving record linkage techniques.* Journal of Information Systems, 2013.
- Vatsalan D, Christen P and Verykios VS: *Efficient two-party private-blocking based on sorted nearest neighborhood clustering.* CIKM, 2013.
- Vaidya J and Clifton C: *Secure set intersection cardinality with application to association rule mining.* Journal of Computer Security, 2005.
- Verykios VS, Karakasidis A and Mitrogiannis VK: *Privacy preserving record linkage approaches.* International Journal of Data Mining, Modelling and Management, 2009.
- Wartell J and McEwen T: *Privacy in the information age: A Guide for sharing crime maps and spatial data.* Institute for Law and Justice, National Institute of Justice, 188739, 2001.

References (13)

- Weber SC, Lowe H, Das A and Ferris T: *A simple heuristic for blindfolded record linkage.* Journal of the American Medical Informatics Association, 2012.
- Winkler WE: *Masking and re-identification methods for public-use microdata: Overview and research problems.* Privacy in Statistical Databases, Barcelona, Springer LNCS 3050, pp. 216–230, 2004.
- Winkler WE: *Overview of record linkage and current research directions.* RR 2006/02, US Census Bureau, 2006.
- Yakout M, Atallah MJ and Elmagarmid AK: *Efficient private record linkage.* IEEE ICDE, 2009.
- Yao, AC: *How to generate and exchange secrets.* Annual Symposium on Foundations of Computer Science, 1986.
- Zhang Q and Hansen D: *Approximate processing for medical record linking and multidatabase analysis.* International Journal of Healthcare Information Systems and Informatics, 2(4), pp. 59–72, 2007.

Secure multi-party computation

- Compute a function across several parties, such that no party learns the information from the other parties, but all receive the final results
  [Yao 1982; Goldreich 1998/2002]
- Simple example: Secure summation $s = \sum_i x_i$.



Step 0:
Z=999

Party 1
x1=55

Step 1: Z+x1= 1054

Party 2
x2=73

Step 2: (Z+x1)+x2 = 1127

Party 3
x3=42

Step 3: ((Z+x1)+x2)+x3=1169

Step 4: s = 1169−Z
= 170