Privacy Aspects in Big Data Integration: Challenges and Opportunities

Peter Christen

Research School of Computer Science,
The Australian National University,
Canberra, Australia

Contact: **peter.christen@anu.edu.au**

Based on work done in collaboration with Dinusha Vatsalan, Vassilios Verykios, Thilina Ranbaduge, and Dimitrios Karapiperis.

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

THE AUSTRALIAN NATIONAL UNIVERSITY

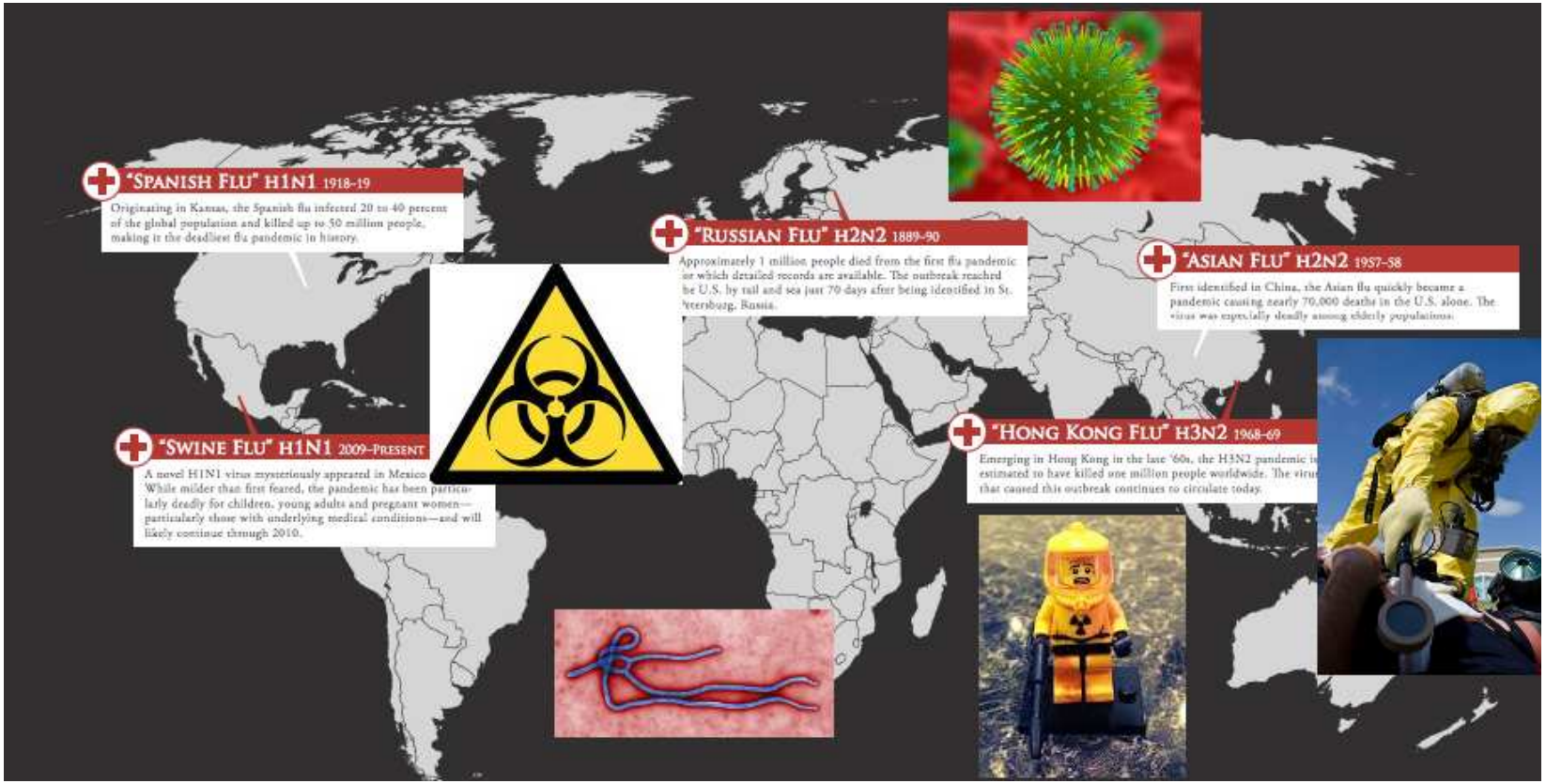
Motivation (1)

- Massive 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 records
  - Tax, social security, and census records
  - Emails, tweets, SMSs, blog posts, etc.
- Analysing (mining) such Big Data can provide huge benefits to businesses and governments

Motivation (2)

- 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 not possible on individual databases
- Lack of unique entity identifiers means that the integration is often based on personal information
- When sensitive (personal) data are integrated across organisations, preserving privacy and confidentiality becomes crucial

Motivating example: Health surveillance (1)



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 integrated at a central location
- Private patient data and confidential data from health care organisations must be kept secure, while still allowing integration and analysis

Outline

- Background to data integration
  - Importance of privacy preservation
  - Application scenarios
- Main concepts and techniques to facilitate privacy-preserving data integration
  - Focus on privacy-preserving record linkage (PPRL), where most work has been done so far
- Big data challenges to privacy-preserving data integration
  - Directions and opportunities for future research

Three main data integration aspects

- Schema matching and mapping
  (identify which attributes and tables contain the same information across several databases)
- Data matching (record linkage or entity resolution)
  (identify which records across organisations refer to the same real-world entities)
- Data fusion
  (merge records that refer to the same entity into consistent and coherent forms)
- Only limited work on privacy-preserving schema matching, no work (to our knowledge) on privacy-preserving data fusion

Data integration for Big Data

- Focus on scalability to very large databases
  (using for example MapReduce techniques)
- Less work on dynamic data, data streams, temporal data, data discovery, or complex data
  (increased work on Web data integration, mash-ups, etc. in recent years)
- Specific techniques have been developed in different domains
  (such as for business data or the life sciences)
- Meta-data integration across different domains is challenging
- Privacy-preserving techniques still assume static databases

Why preserving privacy and confidentiality?

- Basically, data integration assumes all data to be integrated are collected at one location
  (for integration and analysis)
- This makes sensitive data vulnerable
  (to both external as well as internal attacks – such as happened in recent NSA data leakages)
- Much better if sensitive data could be kept at their sources
  - While still allowing integration and analysis
  - Without revealing any private or confidential information
  - With well controlled access and usage limitations

Example scenario (1): Business collaboration

- Collaboration benefits businesses
  (for example in improving efficiency, cross marketing, and reducing the costs of their supply chains)
- They are not willing to share confidential data such as strategies and competitive knowledge
- Identifying which supplies and/or customers two businesses have in common must be done without revealing any other confidential knowledge
- Involvement of a third party to undertake the integration will be undesirable
  (due to the risk of collusion of the third party with either company, or potential security breaches at the third party)

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 from different sources
  (like law enforcement and tax agencies, Internet service providers, and financial institutions)
- These data are highly sensitive; and storage, retrieval and analysis must be tightly regulated
  (collecting such data in one place makes them vulnerable to both external and internal adversaries)
- Ideally, only integrated data are available to the unit (such as records of suspicious individuals)

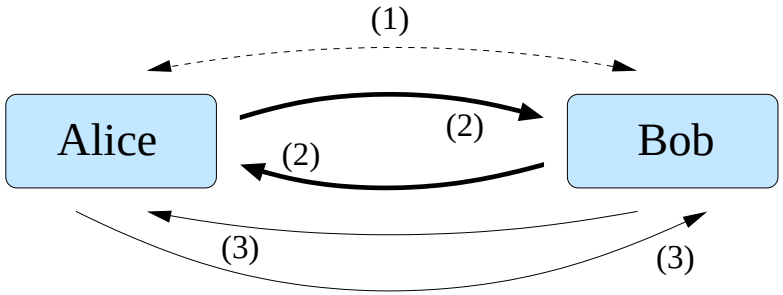
Example scenario (3): Monitoring drug usage

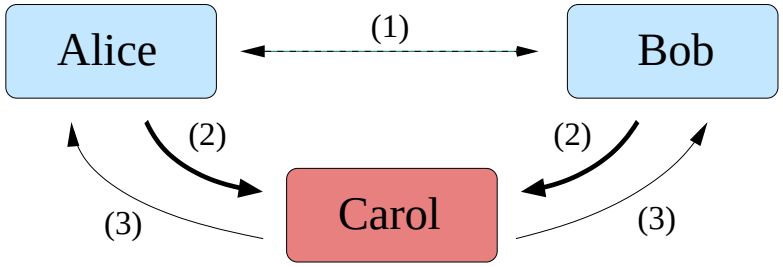
- Monitoring the usage of certain drugs across a country to identify potential disease outbreaks
- Needs collection of data from hospitals, doctors and pharmacies (who are reluctant to provide detailed drug usage and purchase data)
- We need a technique which allows anonymous collection of the usage of potentially a large number of drugs over a certain period of time
- Usage above a certain threshold over a certain time period and region should trigger an alarm (but no individual doctor, hospital, or pharmacy should be identifiable)
- Integration in (near-) real time is required

Main concepts for privacy-preserving data integration

- We assume two or more database owners / data sources (most work so far has concentrated on two sources only)
- Adversary model (using models from cryptography: Honest-but-curious or malicious behaviour)
- Privacy technologies — many approaches (one-way hash-encoding, generalisation, differential privacy, secure multi-party computation, Bloom filters, public reference values, phonetic encoding, multi-dimensional mapping, hybrid approaches, etc.)
- Need to deal with data quality issues (real-world data are ‘dirty’: typos, missing, out-of date, etc.)

Basic protocols





- Two basic types of protocols
  - Two-party protocol: Only the two database owners who wish to integrate their data
  - Three-party protocols: Use a (trusted) third party to conduct the integration (this party will never see any unencoded values, but collusion is possible)

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*, *object identification*, *duplicate detection*, *identity uncertainty*, *merge-purge*, etc.
- Major challenge is that unique entity identifiers are not available in the databases to be linked
  (or if available, they are inconsistent or change over time)

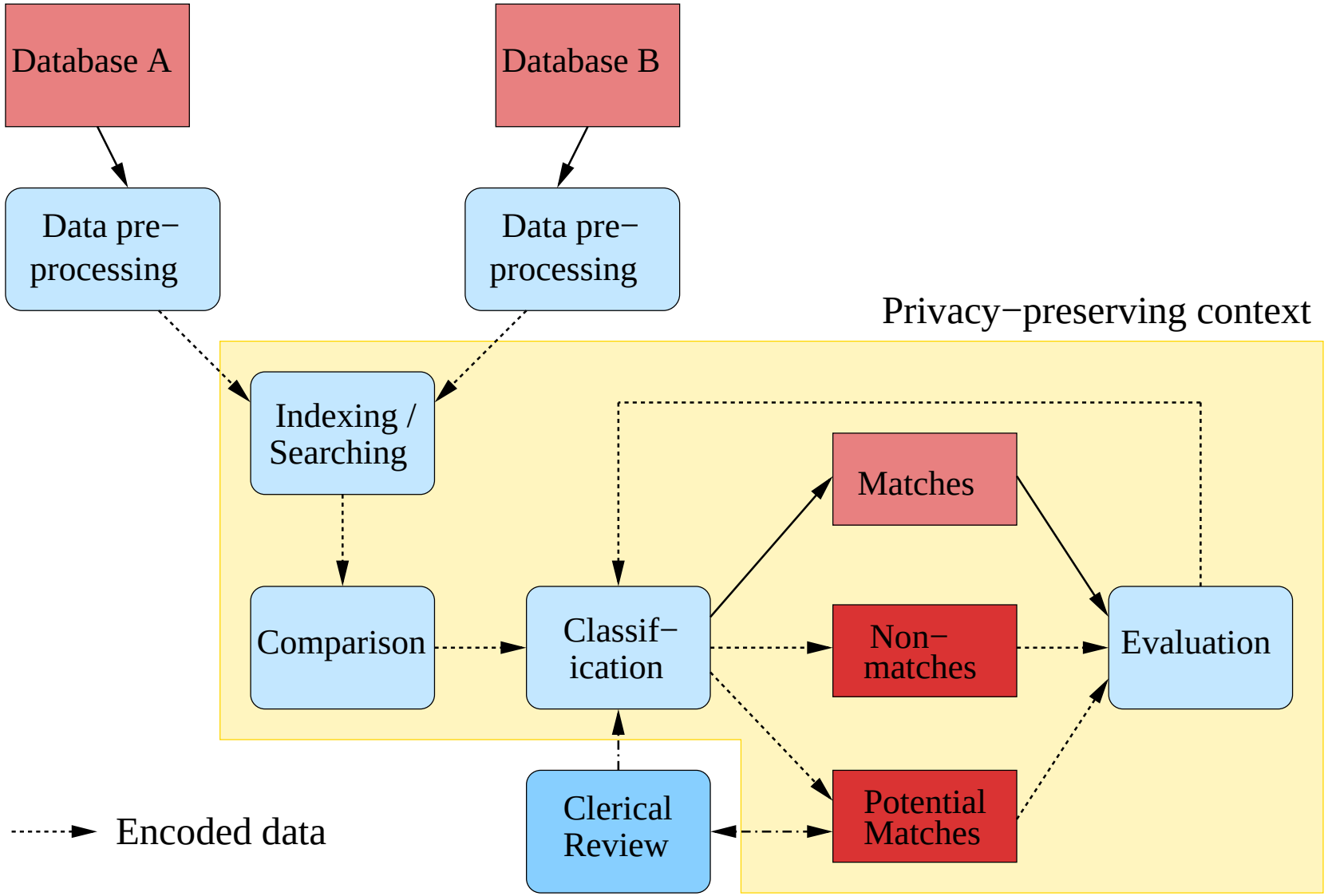
  Which of these records represent the same person?

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 and health research

The privacy-preserving record linkage (PPRL) process



A history of PPRL techniques

- First generation (mid 1990s): exact matching only using simple hash-encoding
- Second generation (early 2000s): approximate matching but not scalable (PP versions of edit distance and other string comparison functions)
- Third generation (mid 2000s): take scalability into account (often a compromise between PP and scalability, some information leakage accepted)
- Different approaches have been developed for PPRL, so far no clear best technique

  (for example, based on Bloom filters, phonetic encodings, generalisation, randomly added values, or secure multi-party computation)

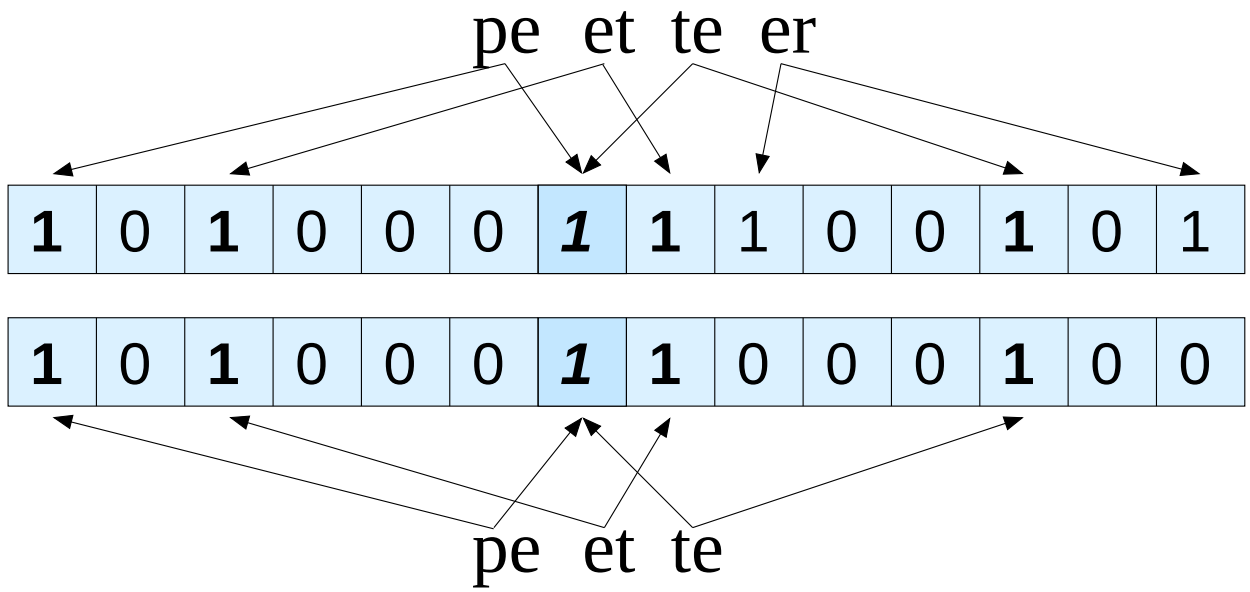
Hash-encoding for PPRL

- A basic building block of many PPRL protocols
- Idea: Use a one-way hash function (like SHA) to encode values, then compare hash-codes
  - Having only access to hash-codes will make it nearly impossible to learn their original input values
  - But dictionary and frequency attacks are possible
- Single character difference in input values results in completely different hash codes
  - For example:
    ‘peter’ $\rightarrow$ ‘101010...100101’ or ‘4R#x+Y4i9!e@t4o]’
    ‘pete’ $\rightarrow$ ‘011101...011010’ or ‘Z5%o-(7Tq1@?7iE/’
  - Only exact matching is possible

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 character 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 common *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

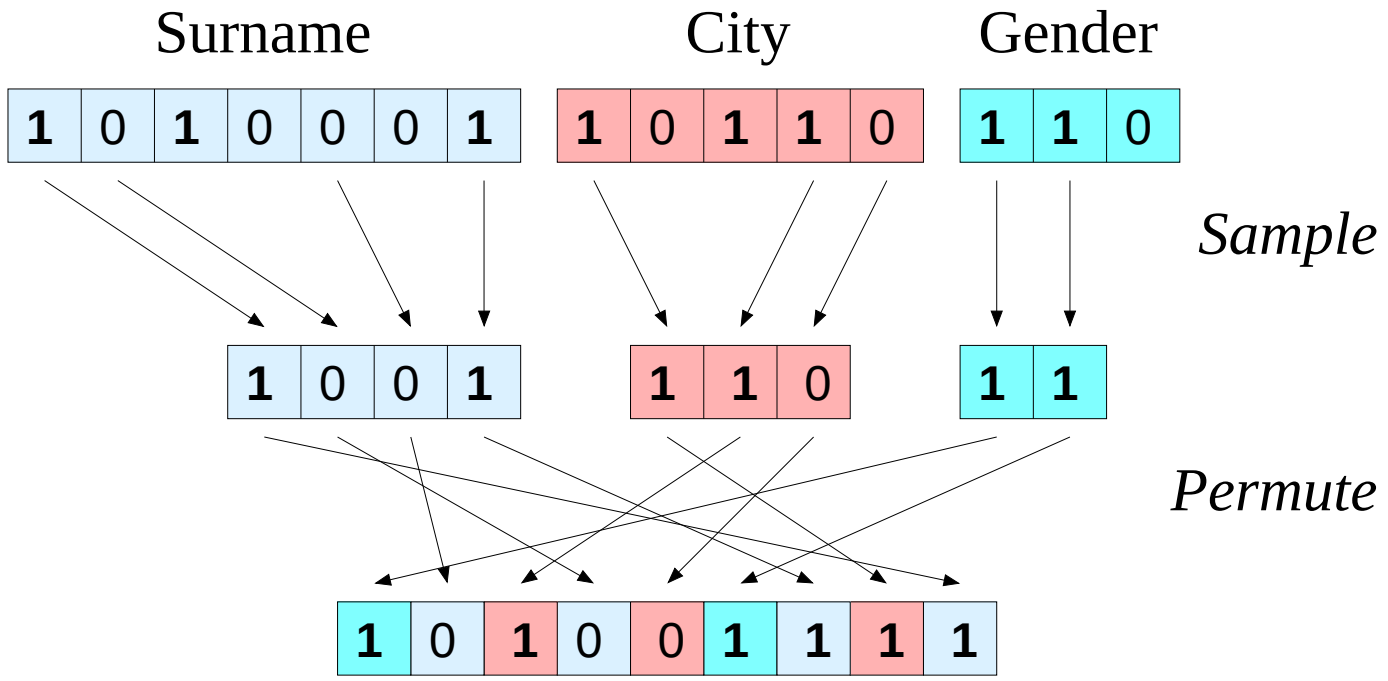
Bloom filter based PPRL (3)

- 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
- Kuzu et al. (PET, 2011) proposed a constraint satisfaction crypt-analysis 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)

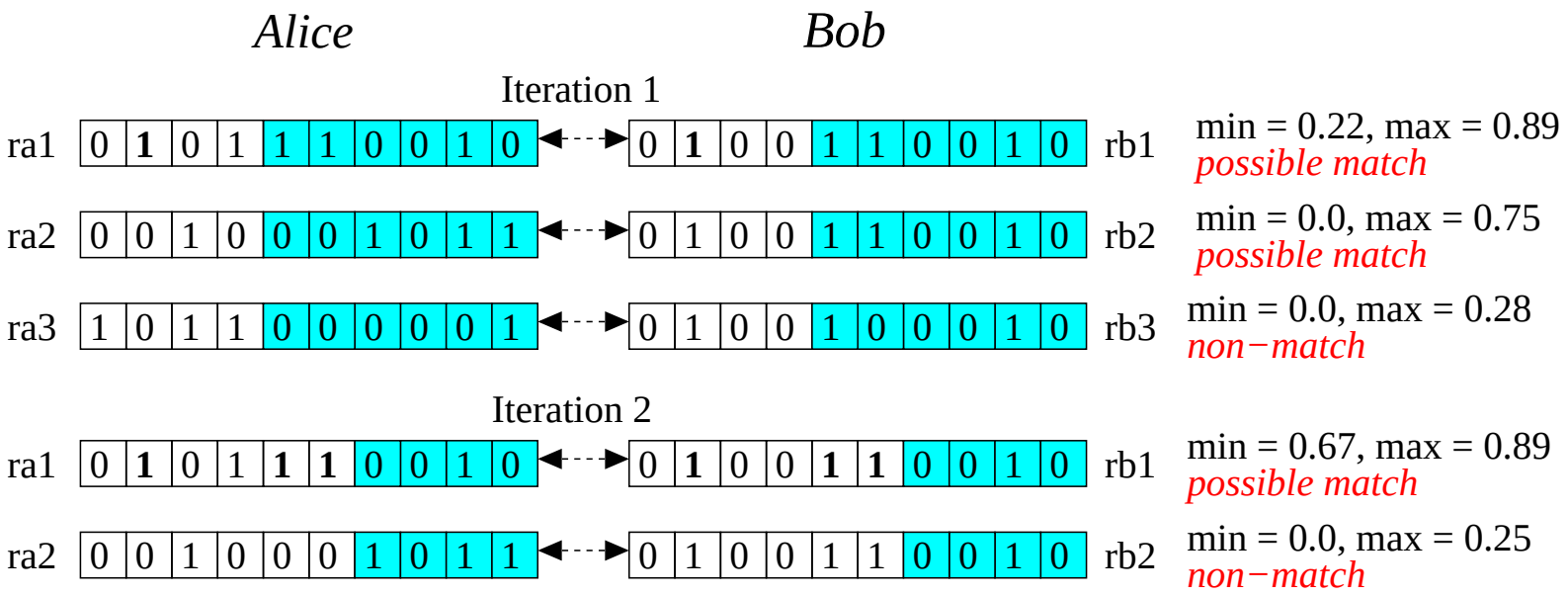


- Experimental results showed much improved security with regard to crypt-analysis attacks
- Scalability can be addressed by Locality Sensitive Hashing (LSH) based blocking

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 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: $max(sim_{Dice}(ra3, rb3)) = 2 \times 1/(4+3) = 2/7 = 0.28$

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 used 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)

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

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)

Challenges and future work (1)

- Limited scalability of current privacy-preserving techniques, especially PPRL
  - Most experimental results on databases of less than 10 million records
  - Some recent work has been using MapReduce to scale-up PPRL (individual techniques only)
  - Most techniques only consider two database owners
- Current techniques are only applicable in batch-mode on static and well defined relational data
- Required are approaches for dynamic data, real-time integration, data streams, and complex and ill-defined data

Challenges and future work (2)

- 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 PPRL 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 (3)

- Need to consider both personal as well as population privacy
  (possible discrimination against individuals and groups)
- Need to consider the trade-off between privacy, utility and costs (time, resources)
  (maximise or minimise one under certain constraints)
- Frameworks for privacy-preserving data integration are needed
  - To facilitate comparative evaluation of techniques
  - Need to allow researchers to plug-in their techniques
  - Benchmark data sets are required (biggest challenge, as such data are sensitive!)

Challenges and future work (4)

- Integration of several databases from multiple sources
  - Most work so far is limited to linking two databases
  - In many real applications data are required from several organistions (earlier example scenarios)
  - Pair-wise integration or PPRL does not scale-up
  - Computational efforts increase with more parties
  - Preventing collusion between (sub-groups of) parties becomes more difficult
- Do we need techniques for privacy-preserving data fusion? (facilitate fusion in same way across different organisations)

Challenges and future work (5)

Example number of candidate record sets generated with multiple parties for different sizes of data sets and blocks.

	Number of parties			
Data set / block sizes	3	5	7	10
100,000 / 10	10^7	10^9	10^{11}	10^{14}
100,000 / 100	10^9	10^{13}	10^{17}	10^{23}
100,000 / 1,000	10^{11}	10^{17}	10^{23}	10^{32}
1,000,000 / 10	10^8	10^{10}	10^{12}	10^{15}
1,000,000 / 100	10^{10}	10^{14}	10^{18}	10^{24}
1,000,000 / 1,000	10^{12}	10^{18}	10^{24}	10^{33}

Privacy Aspects in Big Data Integration

Thank you! Any questions?

Contact: **peter.christen@anu.edu.au**