Attack methods on privacy-preserving record linkage

Peter Christen[1], Rainer Schnell[2], Dinusha Vatsalan[1,3], Thilina Ranbaduge[1], and Anushka Vidanage[1]

[1] **Research School of Computer Science, The Australian National University, Canberra**

[2] **Methodology Research Group, University Duisburg-Essen, Germany**

[3] **CyberPhysical Systems Research Program, Data61, Sydney**

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

Outline

- A short introduction to record linkage and privacy-preserving record linkage (PPRL)
- Bloom filter encoding for PPRL
- Cryptanalysis attack methods on Bloom filter based PPRL
- Our novel efficient cryptanalysis attack methods
  1. Frequency based attack
  2. Pattern mining based attack
- Outlook and recommendations

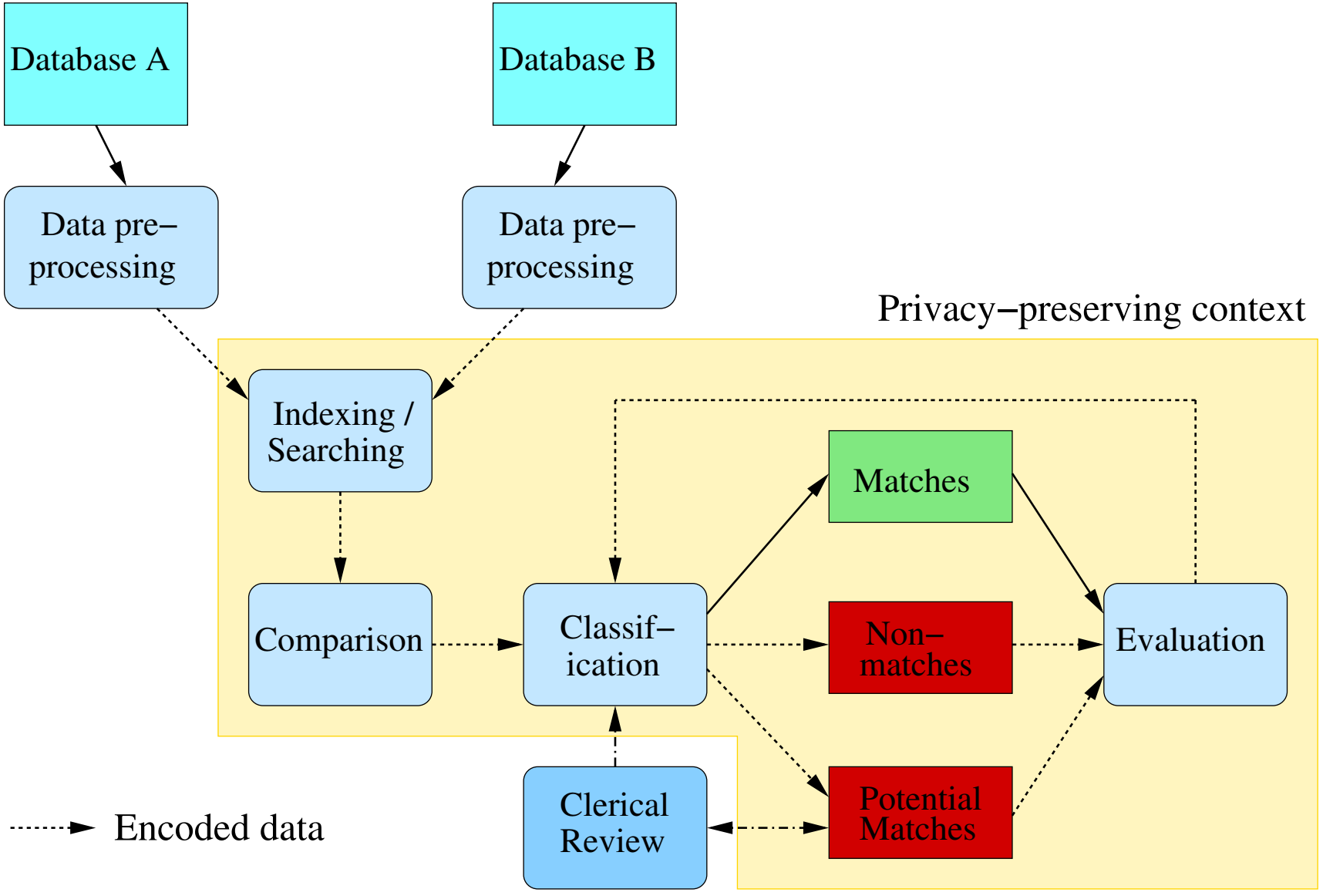
What is record linkage?

- Increasingly, data from different sources need to be integrated and linked
  - To allow analytics not possible on individual databases
  - To improve data quality
  - To enrich data with additional information
- Record linkage is the process of linking records that represent the same entity in one or more databases (patients, customers, tax payers, etc.)
- Lack of unique *entity identifiers* means that linking is often based on sensitive personal information
- When databases are linked across organisations, it is crucial to ensure privacy and confidentiality

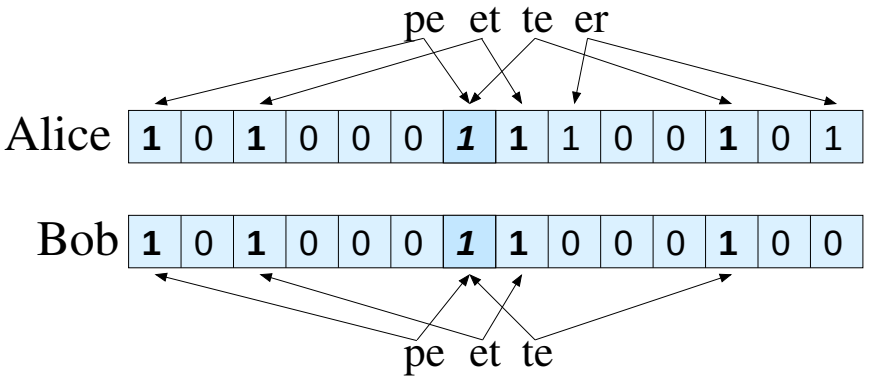
Privacy-preserving record linkage

- **Objective:** *To link data across organisations such that besides the linked records (the ones classified to refer to the same entities) no information about the sensitive source data can be learned by any party involved in the linkage, or any external party*
- Main challenges
  - Have techniques that are scalable to linking large databases across multiple parties
  - Allow for approximate linking of values
  - Being able to asses linkage quality and completeness
  - **Have techniques that are not vulnerable to any kind of attack** (frequency, dictionary, cryptanalysis, etc.)

The PPRL process



Bloom filter encoding *Schnell et al. (2009)*



'peter': x_1=*7*, 'pete': x_1=*5*, *c*=*5*, therefore sim_{Dice} = 2×5/(7+5)= 10/12 = 0.83

- Bloom filters are bit vectors initially all set to *0*
- Use $k \geq 1$ hash functions to hash-map a set of elements by setting corresponding *k* bit positions to *1*
- For PPRL, a set of *q*-grams (from strings) are hash-mapped to allow approximate matching
- Dice similarity of two Bloom filters b_1 and b_2 is: $sim_{Dice}(b_1, b_2) = \frac{2 \times c}{(x_1 + x_2)}$, with: $c = |b_1 \cap b_2|$, $x_i = |b_i|$
- Single or multiple attribute values can be encoded into one BF (known as ABF or RBF)

Attacks on Bloom filter based PPRL

Publication	Data set	Num BF	Correct	Knowledge
Kuzu et al. (2011)	NCVR first names	3,500	11%	*k*, fBF/PT
Kuzu et al. (2013)	Patient names	20	20%	*k*, fBF/PT
Niedermeyer et al. (2014)	German surnames	7,580	12%	*k*, DH, fBF/PT
Kroll and Steinmetzer ('15)	Names and locations	100K	44%	*k*, DH, fBF/PT
Mitchell et al. (2017)	NCVR first / last names	474K	77%	all!

- These cryptanalysis attacks mostly exploit the frequencies of 1-bit patterns within and between Bloom filters (only Mitchell et al. build a graph of possible q-grams encoded in a BF)
- They are feasible only for certain parameter settings and assumptions, and some of them require excessive computational resources (making them not really practical)

A novel efficient attack method

- Our novel cryptanalysis attack is based on the construction principle of Bloom filters of hashing elements of q-gram sets into bit positions
  - A 1-bit at a certain position means **at least one** of a set of q-grams was hashed to this position
  - A 0-bit at a certain bit position means **no q-gram** of a set of q-grams could have been hashed to this position
- The attack is independent of the hash encoding function and its parameters used
- It can correctly re-identify sensitive values even when certain hardening techniques have been applied (such as balancing or xor-folding)
- It runs in a few seconds instead of hours

Attack initialisation

Public database

First name	Freq
karen	231
mary	171
kate	109
mareo	42
...	...

BF database

	Bloom filter	Freq
BF_1	[1,0,1,1,0,1]	242
BF_2	[1,1,0,0,1,0]	184
BF_3	[0,0,1,0,1,1]	115
BF_4	[0,1,0,1,1,1]	48
	...	...

p_1 p_2 p_3 ... p_6

- We assume the attacker has access to a set of encoded Bloom filters and attribute values, and their frequencies
- As with existing attack methods, we assume the attacker knows what attribute(s) are encoded in the Bloom filters
- We frequency-align attribute values and Bloom filters
- We only consider frequent attribute values and Bloom filters as long as they have unique counts

Attack step (1a)

Public database

First name	Freq
karen	231
mary	171
kate	109
mareo	42
...	...

BF database

	Bloom filter	Freq
BF_1	[1,0,1,1,0,1]	242
BF_2	[1,1,0,0,1,0]	184
BF_3	[0,0,1,0,1,1]	115
BF_4	[0,1,0,1,1,1]	48
	...	...

p_1 p_2 p_3 ... p_6

(1a) Position candidate sets

$\mathbf{c}^+[p_1] = \{ka,ar,re,en,ma,ry\}$
$\mathbf{c}^-[p_1] = \{ka,at,te,ma,ar,re,eo\}$

$\mathbf{c}^+[p_2] = \{ma,ar,ry,re,eo\}$
$\mathbf{c}^-[p_2] = \{ka,ar,re,en,at,te\}$

$\mathbf{c}^+[p_3] = \{ka,ar,re,en,at,te\}$
$\mathbf{c}^-[p_3] = \{ma,ar,ry,re,eo\}$
... ...

- For each bit position *p* in the Bloom filters, for all attribute values that have this bit set to **1** we add their q-grams to the set $c^+[p]$ of **possible** q-grams for that position (at least one q-gram of an attribute value was hashed to this position)
- For each bit position *p* in the Bloom filters, for all attribute values that have this bit set to **0** we add their q-grams to the set $c^-[p]$ of **not possible** q-grams for that position (no q-gram of an attribute value can be mapped to this position)

Attack step (1b)

Public database

First name	Freq
karen	231
mary	171
kate	109
mareo	42
...	...

BF database

	Bloom filter	Freq
BF_1	[1,0,1,1,0,1]	242
BF_2	[1,1,0,0,1,0]	184
BF_3	[0,0,1,0,1,1]	115
BF_4	[0,1,0,1,1,1]	48
	...	...

p_1 p_2 p_3 ... p_6

(1b) Position q−gram sets

$\mathbf{c}[p_1] = \mathbf{c}^+[p_1] \setminus \mathbf{c}^-[p_1] = \{en,ry\}$

$\mathbf{c}[p_2] = \mathbf{c}^+[p_2] \setminus \mathbf{c}^-[p_2] = \{ma,ry,eo\}$

$\mathbf{c}[p_3] = \mathbf{c}^+[p_3] \setminus \mathbf{c}^-[p_3] = \{ka,en,at,te\}$

... ...

- For each position *p* we obtain the set $c[p] = c^+[p] \setminus c^-[p]$
- Each *c[p]* is the set of possible q-grams that potentially have been hashed to position *p*
- We can now use the list $\boldsymbol{C} = [c[p_1], \ldots, c[p_l]]$ (where *l* is the length of the Bloom filters) to reconstruct attribute values mapped into a certain Bloom filter (based on the Bloom filter's 0 / 1 bit pattern)

Attack step (2)

Public database

First name	Freq
karen	231
mary	171
kate	109
mareo	42
...	...

BF database

	Bloom filter	Freq
BF_1	[1,0,1,1,0,1]	242
BF_2	[1,1,0,0,1,0]	184
BF_3	[0,0,1,0,1,1]	115
BF_4	[0,1,0,1,1,1]	48
	...	...

p_1 p_2 p_3 ... p_6

(2) Re−identify attribute values (BF_1)

$\mathbf{G}$ = {karen, mary, kate, mareo}

$\mathbf{g}_{p_1} = \mathbf{G} \odot \mathbf{c}[p_1]$ = {karen, mary}

$\mathbf{g}_{p_3} = \mathbf{g}_{p_1} \odot \mathbf{c}[p_3]$ = { karen}

- Given a set ***G*** of attribute values which we aim to map to Bloom filters (i.e. aim to re-identify)
- We analyse each frequent Bloom filter, and remove from ***G*** those attribute values that are not possible matches according to ***C*** because they do not contain any q-grams that would have been hashed to a certain 1-bit
- For example, *kate* is not possible for BF_1 because for the 1-bit in position p_1 it would need to contain either *'en'* or *'ry'* (from *c[1] = {en, ry}*)

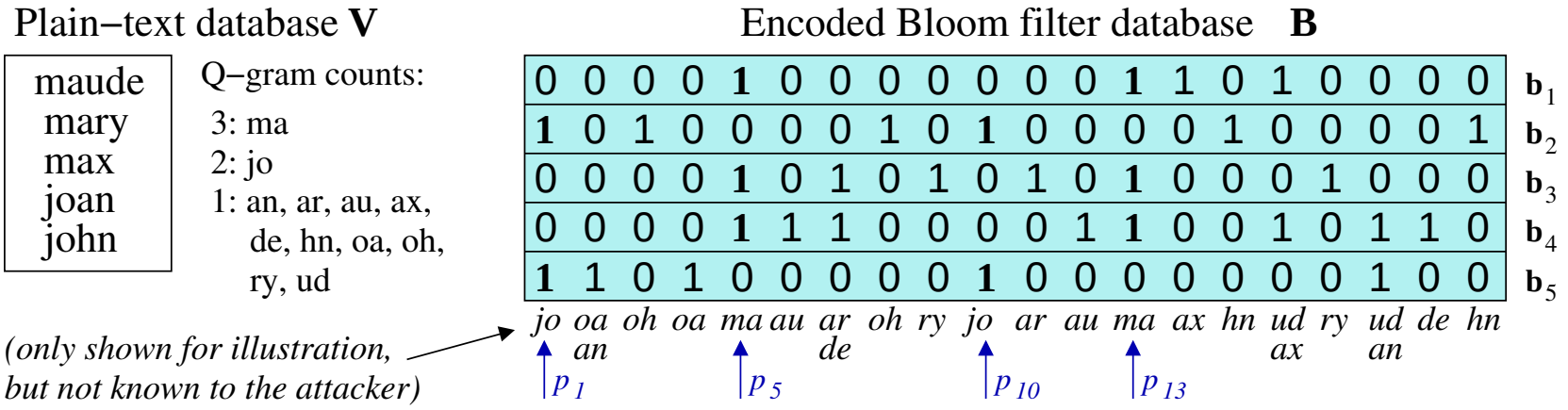
A pattern mining based attack

- Based on the following observation:

  Assuming a q-gram q occurs in $n_q < n$ records in a plain-text database $\mathbf{V}$ that contains $n = |\mathbf{V}|$ records, and $k \geq 1$ independent hash functions are used to encode q-grams from $\mathbf{V}$ into the encoded database $\mathbf{B}$ of n BFs, i.e. $|\mathbf{V}| = |\mathbf{B}|$. Then:

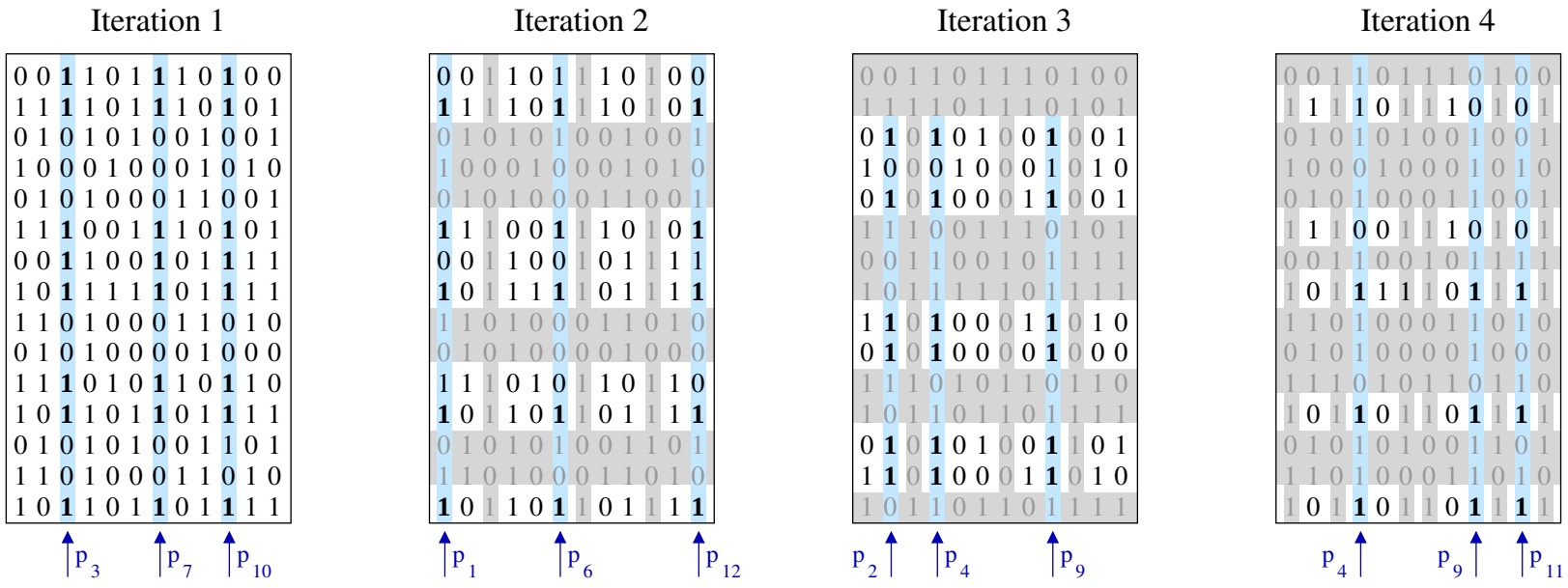
  1. each BF bit position that can encode q must contain a 1-bit in at least n_q BFs in $\mathbf{B}$, and
  2. if $k > 1$ then up to k bit positions must contain a 1-bit in the same subset of BFs $\mathbf{B}_q \subseteq \mathbf{B}$, with $n_q = |\mathbf{B}_q|$, that encode q.

Attack example (1)



- Using frequent itemset mining, we first find bit positions p_5 and p_{13} have co-occurring 1-bits in the *same three BFs* ($\mathbf{b}_1$, $\mathbf{b}_3$, and $\mathbf{b}_4$) and therefore must encode 'ma' which is the only q-gram that occurs in *three plain-text values*.
- Next, we find that p_1 and p_{10} must encode 'jo' because they have co-occurring 1-bits in the *same two BFs* ($\mathbf{b}_2$ and $\mathbf{b}_5$) and 'jo' is the only q-gram that occurs in *two plain-text values*.

Attack example (2)



1. We apply pattern mining on all Bloom filters and all positions
2. Consider only those BFs with a 1-bit in positions p_1, p_7, and p_{10} (partition encoding the frequent q_f from Iteration 1)
3. Consider only those BFs with a 0-bit in positions p_1, p_7, and p_{10} (partition not encoding the frequent q_f from Iteration 1)
4. And so on until a given minimum partition size reached

Experimental evaluation

- We have run a variety of experiments on different data sets (UK census and North Carolina voter)
- Both attacks can correctly identify q-grams and also re-identify encoded values
- The frequency based attack even works with certain hardening technique (balancing and XOR folding)
- The pattern mining attack can identify q-gram positions with very high precision even when each BF in a database is unique!
- The larger the data sets the more successful these attacks are

Recommendations

- First Bloom filter based PPRL systems are now being employed in real-world record linkage applications in the health domain
  (including in Australia, Germany, Wales, Canada, Brazil and Switzerland)
- To limit the vulnerability of such PPRL systems to known attack methods we recommend to:
  1. Use record-level Bloom filter encoding
  2. Apply advanced Bloom filter hardening methods
  3. Reduce the frequency of bit patterns by, for example, salting to prevent any frequency based analysis

Key insight and outlook

- Bloom filters are one single hash step from q-gram to bit array, therefore bit patterns contain information directly relating to q-grams
- Some form of two-step hashing, or further hardening, obfuscation, encoding, or encryption is required
- Future attack ideas: Similarity graph matching, language models, correlation clustering
- Ideally we can attack (identify vulnerabilities) for any PPRL method (for different scenarios)