Privacy-Preserving Data Matching (Privacy-Preserving Record Linkage)

Peter Christen[1],
based on work done with Dinusha Vatsalan[1] and Vassilios Verykios[2]

[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

Contact: peter.christen@anu.edu.au

Outline

- Background about me and my research
- Record linkage and its challenges
  - The record linkage process
- Privacy-preserving record linkage (PPRL)
  - Two PPRL scenarios
  - Requirements for practical PPRL
- A scalable three-party protocol for PPRL
  - Overview of the protocol
  - Walking through the protocol with an example
  - Experiments and results
- Outlook and future work

Background - Short CV

- Born and grew up in Switzerland
  - Diploma in Computer Science, ETH Zürich in 1995
  - PhD in Parallel Computing, University of Basel in 1999
- Moved to Canberra / ANU in 1999
  - Postdoctoral Researcher from 1999–2000
  - Lecturer from 2001–2006
  - Senior Lecturer since 2007
  - Associate Dean (Higher Degree Research) 2009–2011

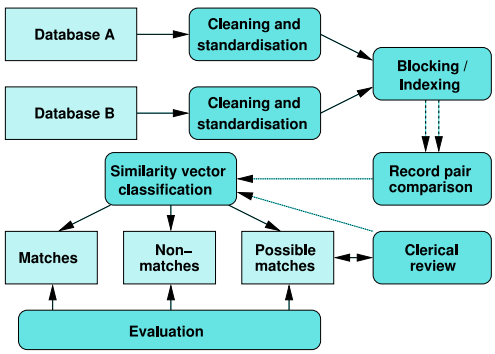
Background - My research

- Research in data mining since 1999
  - Parallel data mining algorithms
  - Visualisation of temporal cluster changes
- Research in record linkage since 2002
  - Collaboration with NSW Health, 2003–2008 (*Febrl* record linkage system, probabilistic geocoding techniques, privacy-preserving record linkage)
  - Collaboration with Veda Advantage, since 2009 (real-time entity resolution to detect identity fraud)

Record linkage and its challenges

- The process of linking and aggregating records that represent the same entity (such as a patient, a customer, a business, etc.)
  - Also called *data matching*, *entity resolution*, etc.
- Has several major challenges
  - Real world data is dirty (typographical errors and variations, missing and out-of-date values, etc.)
  - Scalability (naïve comparison of all record pairs is $O(n^2)$, so some form of blocking or indexing is required)
  - Privacy and confidentiality of the data to be linked (especially if data is linked across organisations)

The record linkage process



Privacy preserving record linkage

- Conduct record linkage between organisations such that:
  - The database owners do not need to give their full databases to the party undertaking the linkage
  - No sensitive information is revealed to any party or an external attacker
  - Only the identifiers of the records that match (i.e. have a similarity above a certain threshold) are either given back to the database owners or to another party
- Privacy is usually preserved through some form of encoding or encryption (commonly, hash-encoding: 'peter' ⇒ 'rgfg4@frg7566#4')

RL scenario 1@

- A researcher is interested in analysing t of car accidents upon the health system
  - *Most common types of injuries?*
  - *Financial burden upon the public health sys*
  - *General health of people after a serious ca*
- She needs access to data from hospital doctors, car and health insurers, and th
  - Currently, all identifying data has to be give researcher, or alternatively a trusted linkage
- This might prevent an organisation from or willing to participate (insurers or police)

RL scenario 2@

- Two pharmaceutical companies are inte collaborating on the development of nev
- The companies wish to identify how mu of confidential data there is in their data (without having to reveal any of that data to eac
- Techniques are required that allow com large amounts of data such that similar are found (while all other data is kept confide
- Involvement of a third party to undertak matching will be undesirable (due to the risk of collusion of the third party witl pany, or potential security breaches at the third

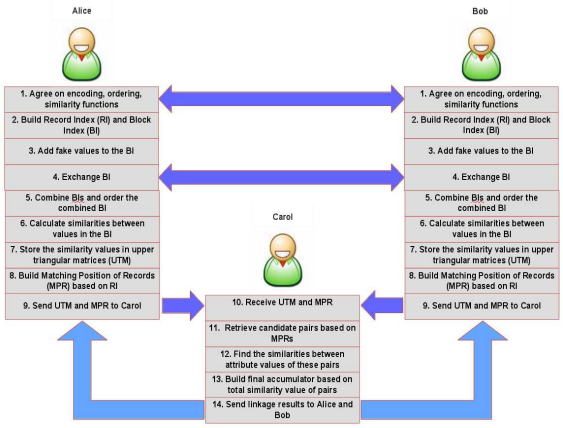
Requirements on practical PPRL

- Scalability to the linking of databases that contain millions of records
- Ability to match different types of data
  - Strings, numbers, dates, times, ages, location, etc.
  - Be able to calculate approximate matches
- Accurate classification of the compared record pairs into matches and non-matches
  - No training data in the form of known matches and non-matches will be available
- Assessment of the accuracy and completeness of the results of a PPRL project must be feasible

Previous experimental work in PPRL

Publications	Data sets used	Timing results
Al-Lawati et al. 2005	BioMed, DLBP, e-Print, 10,000 rec each	Minimum 100 sec per data set
Scannapieco et al. 2007	BC voter's list (2000 rec) , Italian admin (20,000 rec)	Around 50–800 sec
Inan et al. 2008	UCI Adult data set, 20,098 rec	3.8 sec
Karakasidis et al. 2009	Synthetic data generated by Febrl, 10,000 / 100,000 rec	Minimum 105 sec
Schnell et al. 2009	Febrl (500,000 rec), German admin (15,000 rec)	No timing results
Yakout et al. 2009	British Columbia voter's list, 34,261 rec	50–30 sec for 15,000 rec
Inan et al. 2010	UCI Census-Income data set, 95,014 rec	No timing results

A scalable three-party PPRL protocol



Example databases and step 1

A

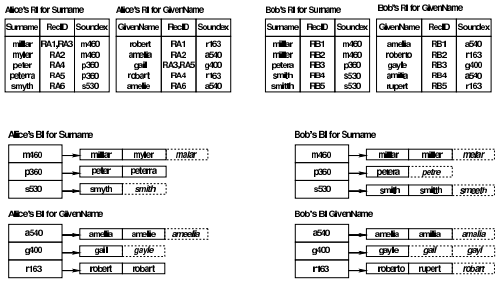
RecID	Surname	GivenName
RA1	millar	robert
RA2	myler	amelia
RA3	millar	gail
RA4	peter	robart
RA5	peterra	gail
RA6	smyth	amelie

B

RecID	Surname	GivenName
RB1	millar	amelia
RB2	miller	roberto
RB3	petera	gayle
RB4	smith	amilia
RB5	smitth	rupert

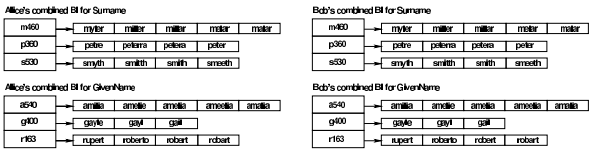
- The database owners, Alice and Bob, agree upon:
  - An encoding function, *encode(x)*, such as Soundex
  - A similarity function, *sim*(x_1,x_2), such as edit-distance
  - An ordering function *order(x)* to order attribute values
  - A hash-encoding function *hash(x,k)*, such as HMAC
  - Minimum similarity thresholds s_a and s_t

Step 2: Load databases and build Record and Block Index



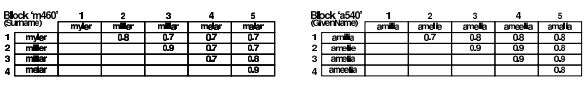
- Security of protocol can be improved by adding extra (faked) attribute values and records (step 3) (dashed list entries with italic names)

Steps 4 and 5: Exchange, combine and order Block Index



- Different ordering for (a) block lists and for (b) blocking key values (BKVs) or encodings
  - And different orderings for different attributes
- The orderings must not be known by the linkage unit, Carol

Step 6: Generate Similarity Index



- Any similarity function *sim*(x_1,x_2) can be used (for strings, numbers, dates, ages, times, locations, etc.)
- Assume that for exact matches *sim=1*, while for total dissimilarities *sim=0*
- Only *sim* $\geq s_a$ are stored in the SI
- Each block is stored as an upper triangular matrix (UTM) because similarities are symmetric

Step 7: Store similarities into UTMs

Alice's UTM for Surname – BKV m460

	2	3	4	5
1	0.8	0.7	0.7	0.7
2		0.9	0.7	0.7
3			0.7	0.8
4				0.9

Alice's UTM for Surname – BKV s530

	2	3	4
1	0.8	0.9	0.8
2		0.9	0.8
3			0.8

Bob's UTM for Surname – BKV p360

	2	3	4
1	0.8	0.9	0.9
2		0.9	0.8
3			0.9

Alice's UTM for GivenName – BKV g400

	2	3
1	0.9	0.8
2		0.8

Bob's UTM for GivenName – BKV a540

	2	3	4	5
1	0.7	0.8	0.8	0.8
2		0.9	0.9	0.8
3			0.9	0.9
4				0.8

Bob's UTM for GivenName – BKV r163

	2	3	4
1	0.7	0.8	0.7
2		0.9	0.8
3			0.9

- Alice and Bob only need to calculate half the similarities (because they both have the full BI)
- UTMs are sent to Carol with BKV encoded (in step 9)
  (but position values and similarities are sent unencoded so Carol can use them to combine similarities)

Step 8: Generate the Matching Positions of Records

Alice's MPR for Surname

BKV	RecID	Position
m460	RA1,RA3	3
m460	RA2	1
p360	RA4	4
p360	RA5	2
s530	RA6	1

Alice's MPR for GivenName

BKV	RecID	Position
r163	RA1	3
a540	RA2	3
g400	RA3,RA5	3
r163	RA4	4
a540	RA6	2

Bob's MPR for Surname

BKV	RecID	Position
m460	RB1	3
m460	RB2	2
p360	RB3	3
s530	RB4	3
s530	RB5	2

Bob's MPR for GivenName

BKV	RecID	Position
a540	RB1	3
r163	RB2	2
g400	RB3	1
a540	RB4	1
r163	RB5	1

- MPRs are the Record Index with attribute values removed and positions added
- Position values correspond to positions in the UTMs of the SI
- BKVs are encoded, record identifiers are encrypted (independently by Alice and Bob)
- MPRs are sent to linkage unit, Carol (in step 9)

Steps 11 and 12: Combine similarities for record pairs

Accumulator for Surname

	RecIDs	Similarity
m460	RA1,RB1	1.0
	RA3,RB1	1.0
	RA2,RB1	0.7
	RA1,RB2	0.9
	RA3,RB2	0.9
	RA2,RB2	0.8
p360	RA4,RB3	0.9
	RA5,RB3	0.9
s530	RA6,RB4	0.9
	RA6,RB5	0.8

Accumulator for GivenName

	RecIDs	Similarity
r163	RA1,RB2	0.9
	RA1,RB5	0.8
	RA4,RB2	0.8
	RA4,RB5	0.7
a540	RA2,RB1	1.0
	RA2,RB4	0.8
	RA6,RB1	0.9
	RA6,RB4	0.7
g400	RA3,RB3	0.8
	RA5,RB3	0.8

Final summed accumulator

RecIDs	Similarity
RA1,RB2	1.8
RA2,RB1	1.7
RA5,RB3	1.7
RA6,RB4	1.6

- For each set of triplets in MPRs with same BKV:
  - Nested loop over record identifiers and positions
  - If position values are the same, then similarity *sim=1*
  - Otherwise, get similarity value from UTM with BKV at position *((min(p_a,p_b), max(p_a,p_b))*

Step 13: Classify the candidate record pairs

- Attribute similarities are summed for each record pair into one final accumulator
- Those record pairs with a total similarity $sim \geq s_t$ are classified as matches
- The record identifiers of the matches are sent back to Alice and Bob (step 14)
- Alice and Bob can now agree to share information about the matches, or send the matched records to another party (for example a researcher)

Security analysis

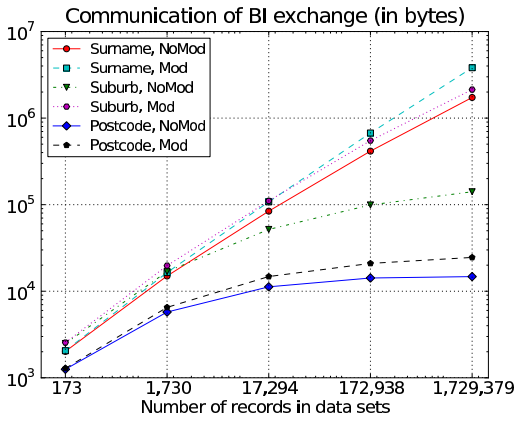
- We assume all parties follow the 'honest but curious' behaviour
- Alice and Bob learn each others attribute values in step 4
  - But not the content of individual records
  - Knowing a rare value can infer information about people with rare names for example
- Carol can compile frequency statistic about BKVs, size of blocks, matching positions, etc.
  (but not the content of individual records)
- If Alice and Carol collude they can learn everything about Bob's values (and vise-versa)

Experiments

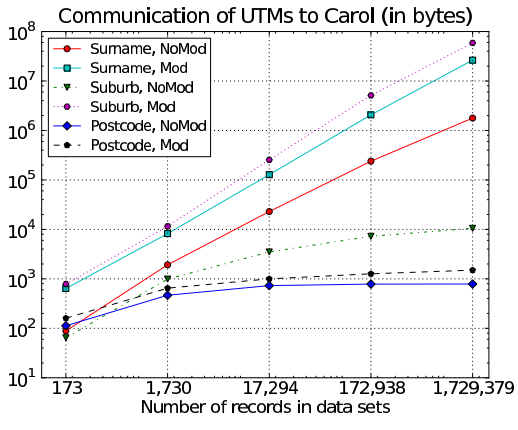
- Data from Australian telephone directory with Surnames (404,651 values), Suburb names (13,109 values and postcodes (2,632 values)
- We sampled data sets from 100%, 10%, 1%, 0.1% and 0.01% of the total 6,917,514 records
- One series of Bob's data sets were modified (one character edit per attribute value)

Data set sizes	25% overlap	50% overlap	75% overlap
173 / 173	38	86	130
1730 / 1730	446	897	1310
17,290 / 17,290	4365	8611	12,973
172,938 / 172,938	42,980	86,363	129,542
1,729,379 / 1,729,379	432,538	86,487	1,297,029

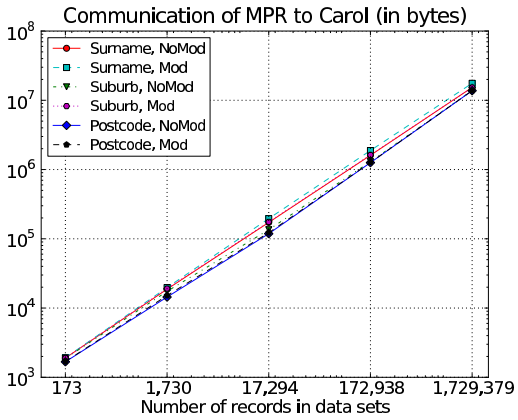
Communication – Exchange Block Index between Database Owners



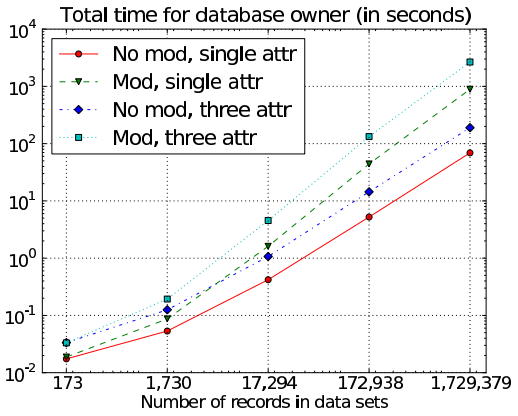
Communication – Send Similarity Index to Linkage Unit



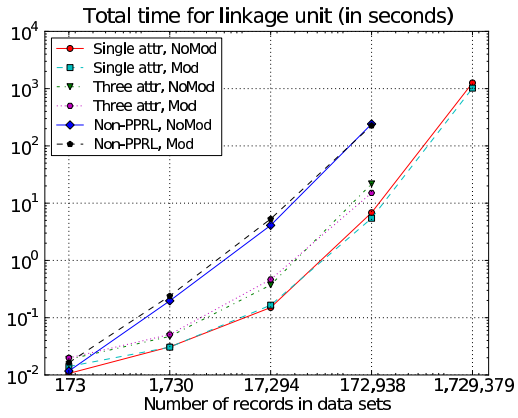
Communication – Send Matching Positions of Records to Linkage Unit



Timing – Steps at Database Owner



Timing – Steps at Linkage Unit



Conclusions and future work

- First approach to PPRL that is experimentally evaluated on more than 1 million records
- Our approach is scalable in size of data sets, but has a quadratic computational complexity of the linking step
- Several security drawbacks
  - We will investigate how adding extra attribute values and records will improve security
  - We plan to develop parallel versions of protocol
  - We plan to develop a two-party version of our protocol (with no linkage unit)