Privacy Preserving Techniques for Data Matching

Dinusha Vatsalan

Work done in collaboration with:
Prof. Peter Christen, A/Prof. Vassilios S. Verykios, Dr. Christine M. O'Keefe, Prof. Erhard Rahm, and Dr. Qing Wang

Funded by the Endeavour Postgraduate Research Award, ARC Discovery Projects DP130101801 and DP160101934, Australia-Germany Joint Research Cooperation Scheme, and GSoC

Research School of Computer Science, College of Engineering and Computer Science, The Australian National University, Australia

Outline

- An overview of privacy-preserving data matching (PPDM)
- A taxonomy of PPDM
- Two-source linkage (without a linkage unit)
- Multi-source linkage (MP-PPDM)
- Dynamic data matching
- Privacy-preserving similar patient matching (PPSPM)
- Privacy-preserving interactive record linkage
- Outlook to future research directions

Privacy-Preserving Data Matching (PPDM) – An Example

Health database

PID	Surname	Given_name	Age	Postcode	Sex	Pressure	Stress	Last_visited	Reason_of_visit
P1209	Robertt	Peter	41	2617	m	140/90	high	25 days ago	chest pain
P4204	Miller	Amelia	39	2415	f	120/80	high	61 days ago	headache
P4894	Siemen	Jeff	30	2602	m	110/80	normal	15 days ago	checkup

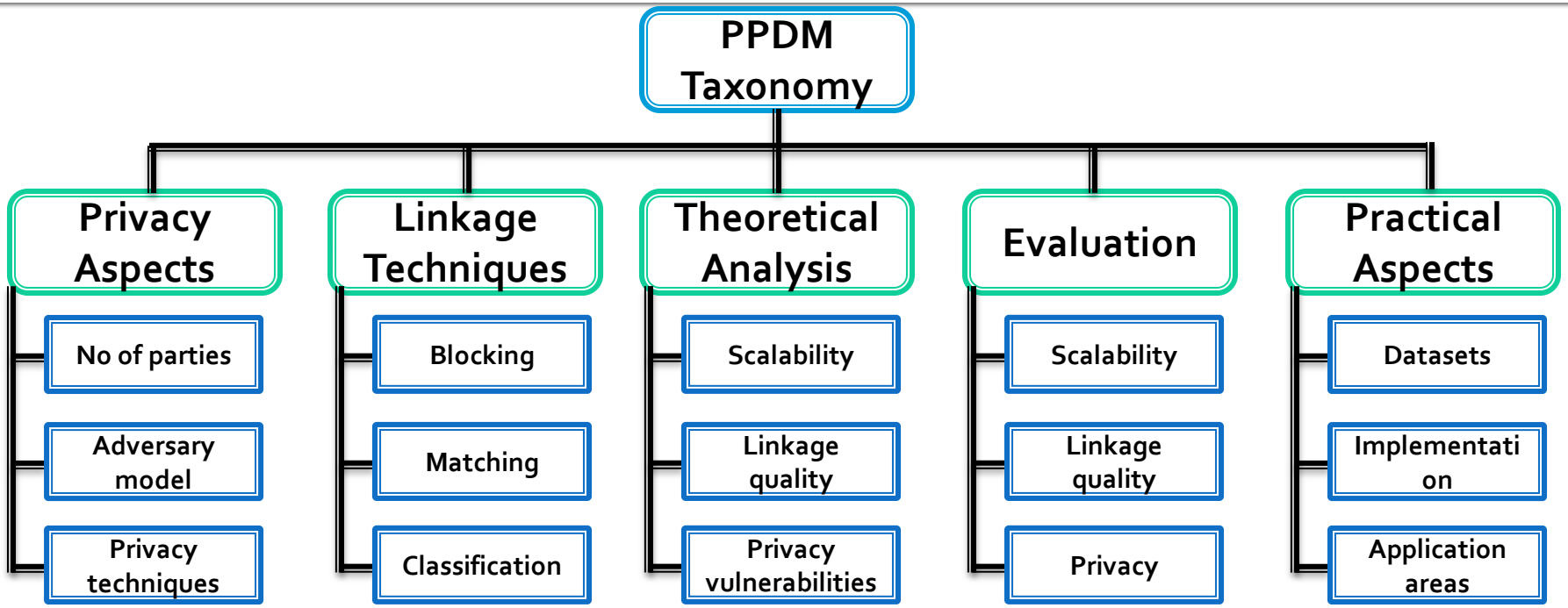
Social security database

ID	First_name	Last_name	DOB	Gender	Postcode	Loan_type	Period	Amount	Paid
6723	Peter	Robert	20.06.1972	M	2617	Mortgage	20	350,000	130,000
8345	Miller	Roberts	11.10.1979	M	2602	Personal	5	10,000	1,900
9241	Amelia	Millar	06.01.1974	F	2415	Mortgage	30	475,000	154,250

Bank database

SSN	Title	Last_name	First_name	Age	Postcode	Employment	Income	Benefits	Payment
490814	Mrs	Amilia	Smith	39	2642	Teacher	60,000	Child care	45,000
581233	Mr	Peter	Roberts	42	2627	Engineer	110,000	Family tax	50,000
932389	Mr	William	Smith	69	3205	Retired	-	Pension	35,000

A Taxonomy of PPDM Techniques



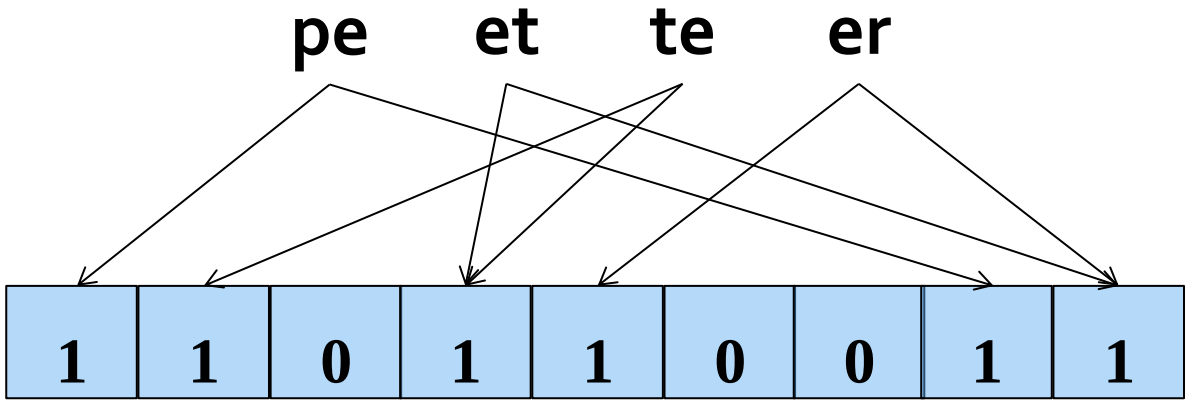
- Main gaps identified and addressed:
  - Non-Linkage unit (LU)-based protocols
  - Scalability
  - Privacy measures and evaluation
  - Comprehensive evaluation
  - Multi-source linkage
  - Data encoding for different attributes
  - Dynamic data matching

Vatsalan et al., A Taxonomy of PPRL techniques, JIS, 38(6), pp. 946-969, 2013

Bloom Filter (BF) Encoding

- *k* independent hash functions are used to hash-map each element in a set *S* into a Bloom filter (BF) of length *l* bits by setting the corresponding bits to 1

- E.g. hash-mapping bi-grams of string 'peter' (*S = ['pe', 'et', 'te', 'er']*) into a BF of *l=9* bits using *k=2* hash functions:

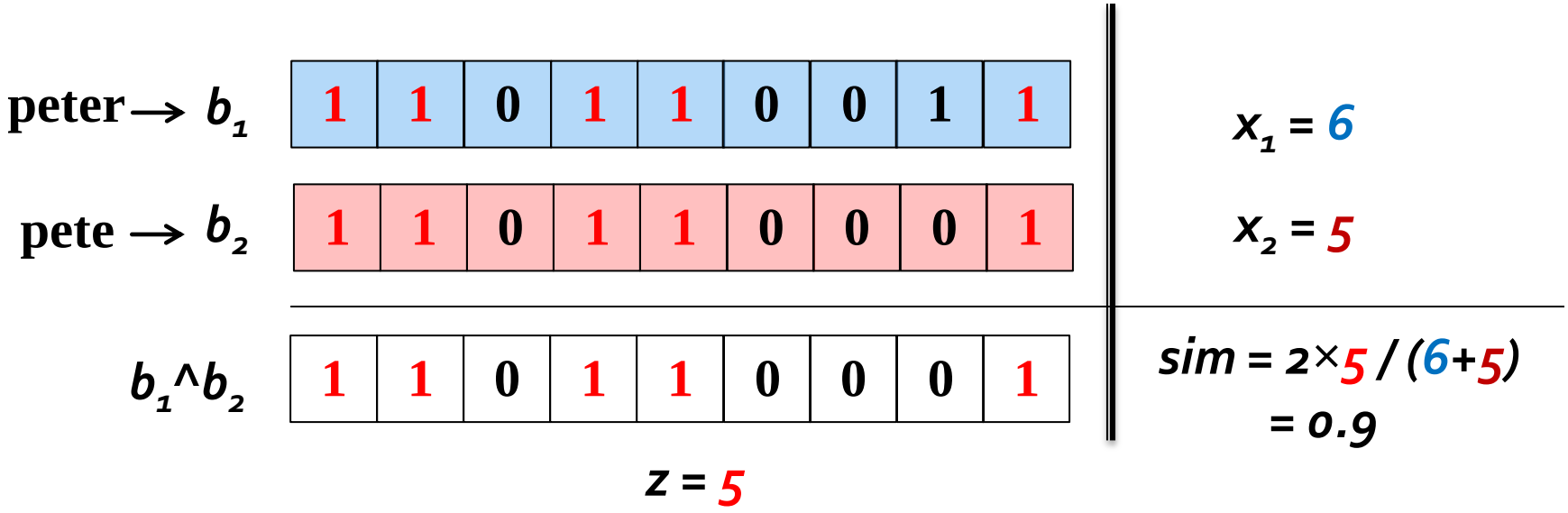


BF-based Approximate Matching

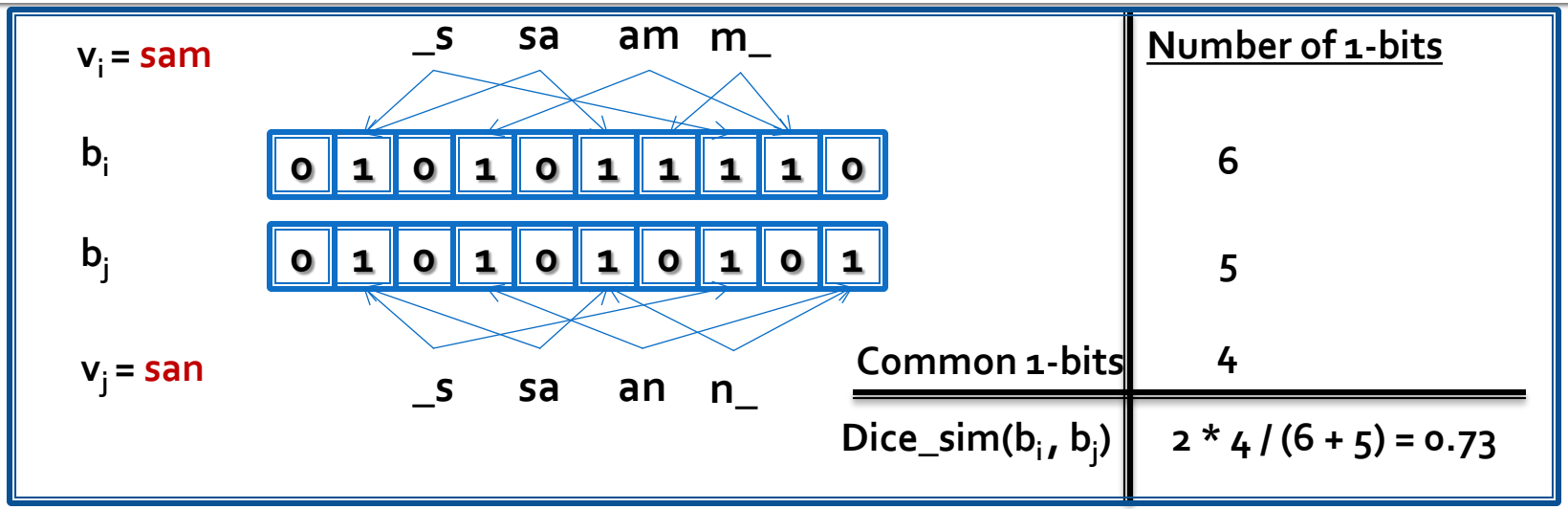
- **Dice coefficient similarity** of p BFs is calculated as:

$$Dice\_sim(b_1,\ldots,b_p) = \frac{p \times z}{\sum_i x_i},$$

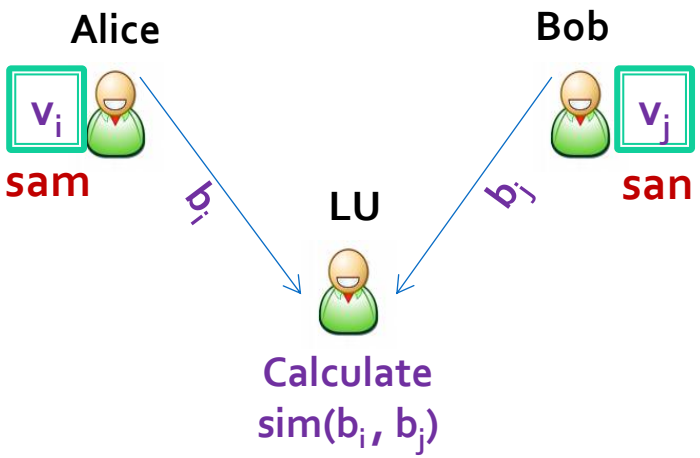
where z is the number of common 1-bits in p BFs and x_i is the number of 1-bits in BF b_i



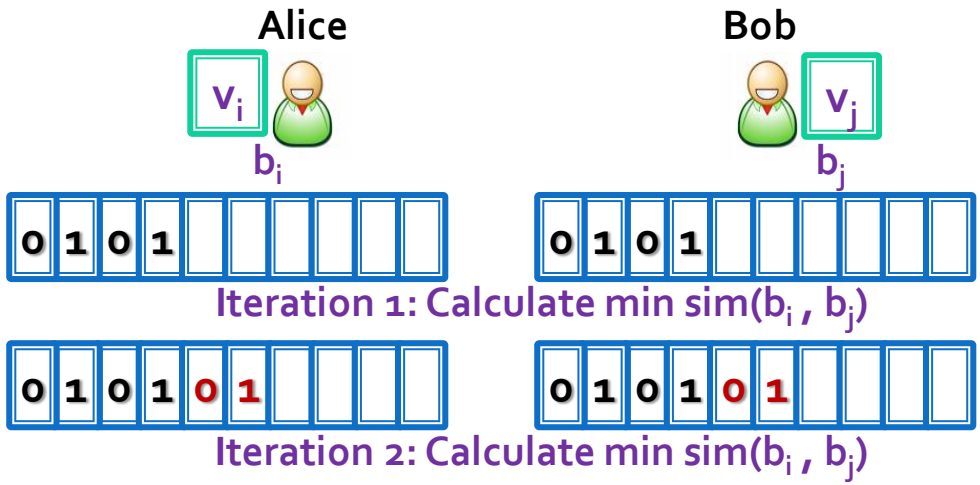
BF-based Two-Source Linkage



Bloom filters in three-party (Schnell et al. 2009)



Bloom filters in two-party



Vatsalan and Christen, An Iterative Two-Party Protocol for Scalable Privacy-Preserving Record Linkage, AusDM 2012

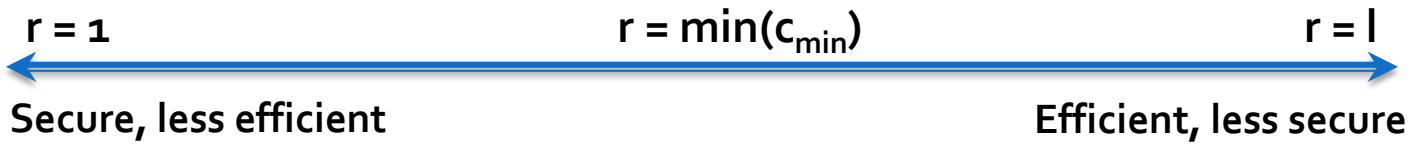
BF-based Two-Source Linkage (cont..)

- Minimum number of common 1-bits required – c_{min}

$$Dice\_sim(b^A, b^B) \geq s_t$$
$$2c/(x^A + x^B) \geq s_t$$
$$2\, c_{min} / (x^A + x^B) = s_t$$
$$c_{min} = [s_t\, (x^A + x^B)\, /2]$$

- Classify at each iteration i
  - c_{min} bits already found– match
  - c_{min} bits cannot be found in later iterations – non_match
  - c_{min} bits might be found in later iterations – possible match

- How many bits r_i to reveal in iteration i?



BF-based Two-Source Linkage (cont..)

Iteration 1

x^A	x^B	c_{min}	b^A	b^B	$x^A - x_i^A$	$x^B - x_i^B$	$C_{min} - c_i$	Class
6	5	5	0 1 1 1	1 1 0 1	3	2	3	NM
5	7	5	1 1 0 0	1 1 0 1	3	4	3	PM
5	5	4	1 1 0 0	1 1 0 1	3	2	2	PM

BF-based Two-Source Linkage (cont..)

Iteration 1

x^A	x^B	c_{min}	b^A	b^B	$x^A - x_i^A$	$x^B - x_i^B$	$C_{min} - c_i$	Class
~~6~~	~~5~~	~~5~~	~~0 1 1 1 □ □ □ □ □ □~~	~~1 1 0 1 □ □ □ □ □ □~~	~~3~~	~~2~~	~~3~~	~~NM~~
5	7	5	1 1 0 0 □ □ □ □ □ □	1 1 0 1 □ □ □ □ □ □	3	4	3	PM
5	5	(4)	1 1 0 0 □ □ □ □ □ □	1 1 0 1 □ □ □ □ □ □	3	2	2	PM

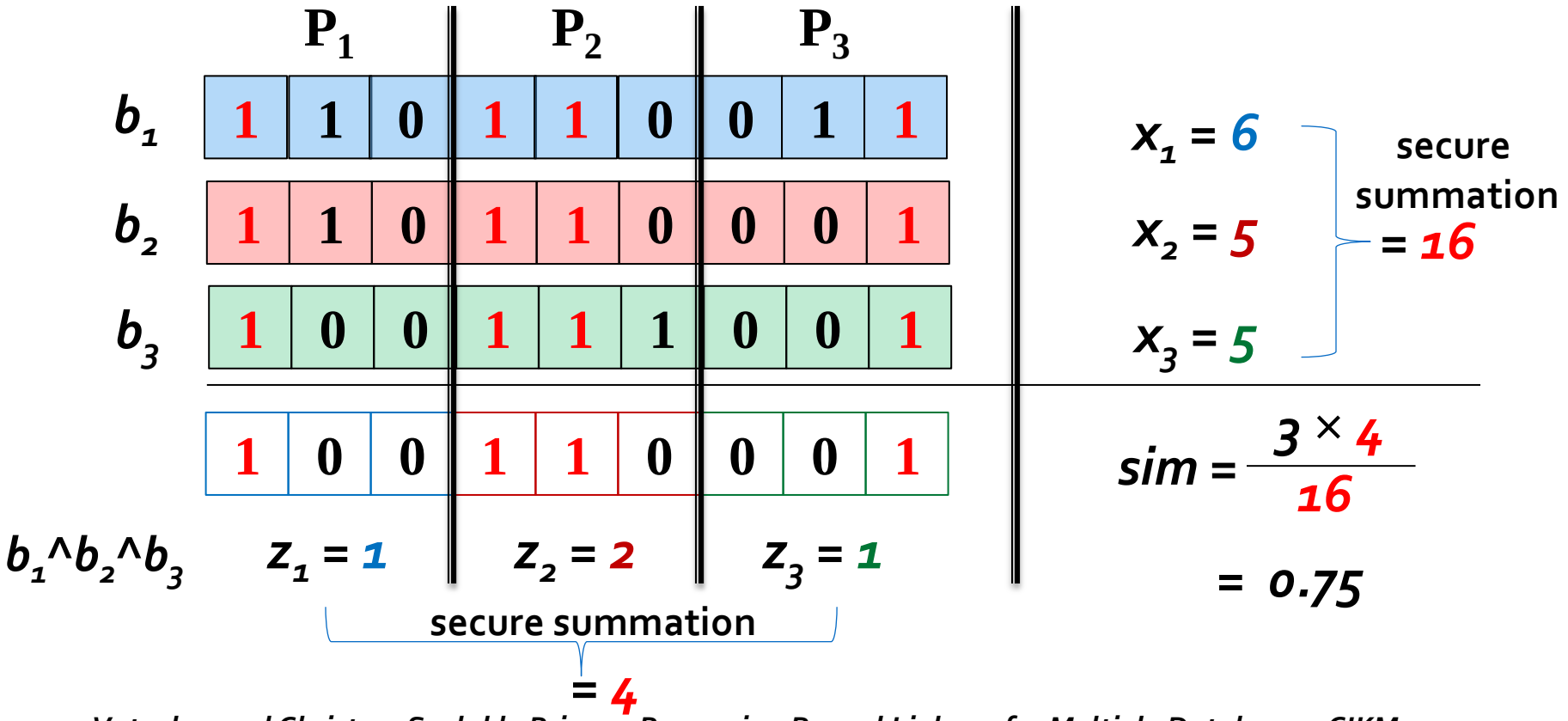
Iteration 2

x^A	x^B	c_{min}	b^A	b^B	$x^A - x_i^A$	$x^B - x_i^B$	$C_{min} - c_i$	Class
3	4	3	1 1 0 0 0 0 □ □ □ □	1 1 0 1 1 0 □ □ □ □	3	3	3	PM
3	2	(2)	1 1 0 0 1 1 □ □ □ □	1 1 0 1 1 1 □ □ □ □	1	0	0	M

BF-based Multi-Source Linkage

- The **similarity** calculation can be **distributed** among *p* parties:

$$Dice\_sim(b_1,\ldots,b_p) = \frac{p \times \sum z_i}{\sum_i x_i}$$



Vatsalan and Christen, Scalable Privacy-Preserving Record Linkage for Multiple Databases, CIKM 2014

Counting Bloom Filter (CBF) encoding

- An integer array of length *l* containing counts of values in each bit position β, *1* ≤β≤*l* over *p* BFs

- The similarity of *p* BFs can be calculated given a counting Bloom filter (CBF) *c*:

$$Dice\_sim(c) = \frac{p \times |\beta: 1 \leq \beta \leq l \; and \; c(\beta) = p|}{\sum_{\beta=1}^{l} c(\beta)}$$

	1	2	3	4	5	6	7	8	9
b_1	1	1	0	1	1	0	0	1	1
b_2	1	1	0	1	1	0	0	0	1
b_3	1	0	0	1	1	1	0	0	1
c	3	2	0	3	3	1	0	1	3

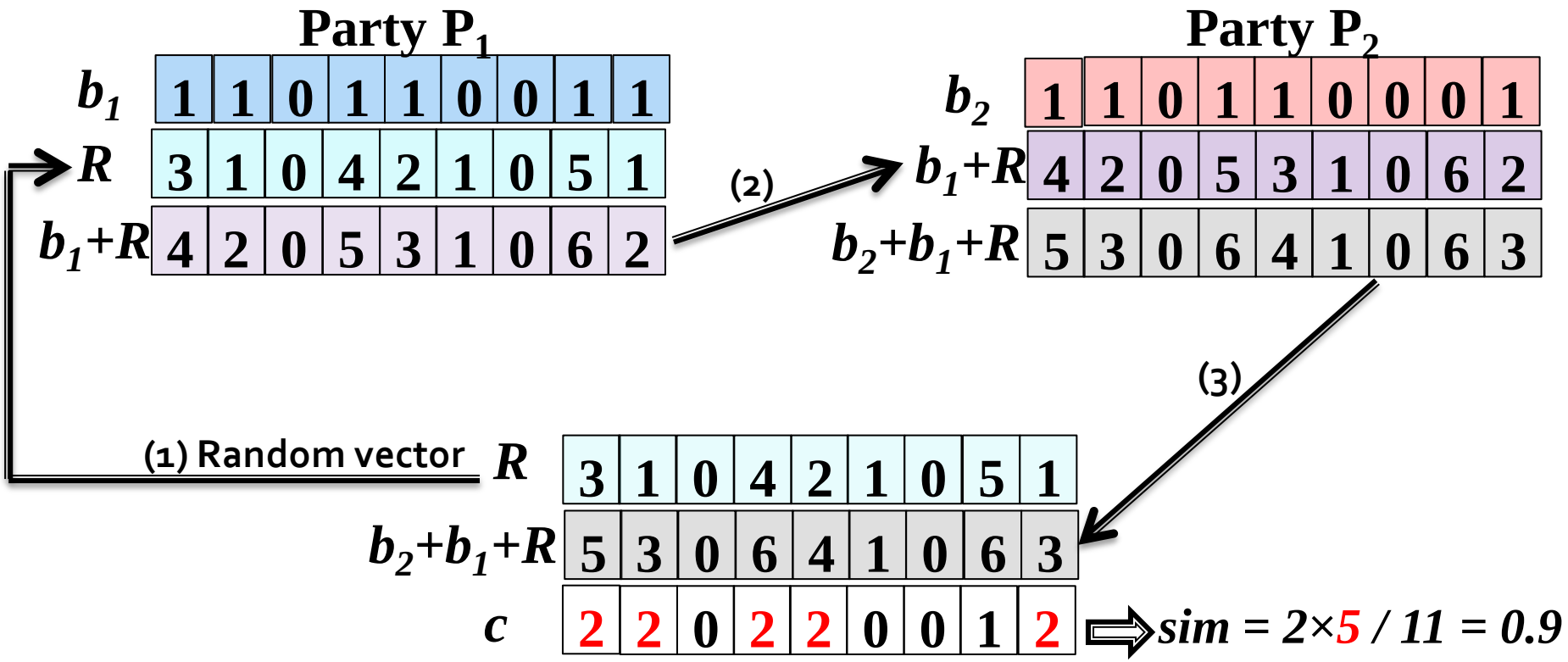
$$Dice\_sim = \frac{3 \times |\{1,4,5,9\}|}{(3+2+0+3+3+1+0+1+3)}$$

$$= \frac{3 \times 4}{16}$$

$$= 0.75$$

Vatsalan et al., Scalable privacy-preserving linking of multiple databases using counting Bloom filters, ICDMW 2016

CBF-based Multi-Source Linkage

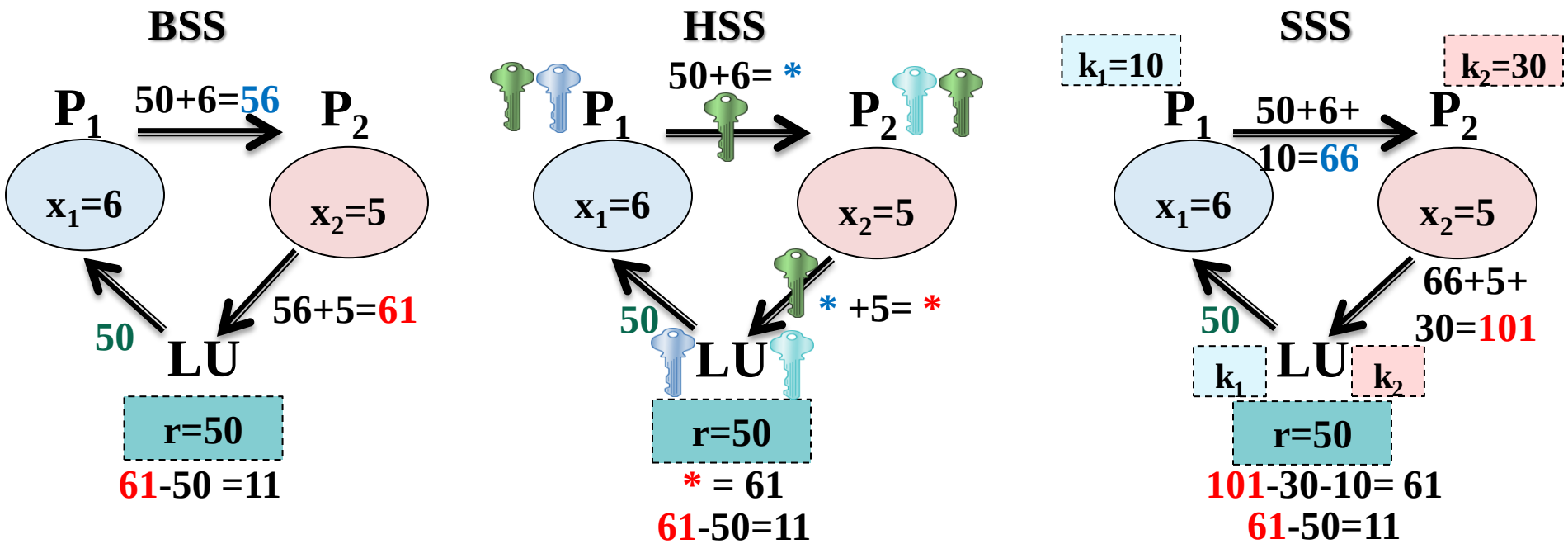


- A CBF *c* provides improved privacy as it contains only the summary information (count values), while a BF b_i contains individual bit values of records R_i *(1 ≤ i ≤ p) – proof is given in the paper*

Vatsalan et al., Scalable privacy-preserving linking of multiple databases using counting Bloom filters, ICDMW 2016

Extended Secure Summation

- The basic secure summation (BSS) protocol is susceptible to collusion risk by the database owners
- Two extended secure summation protocols:
  - Homomorphic-based secure summation (HSS)
  - Salting-based secure summation (SSS)



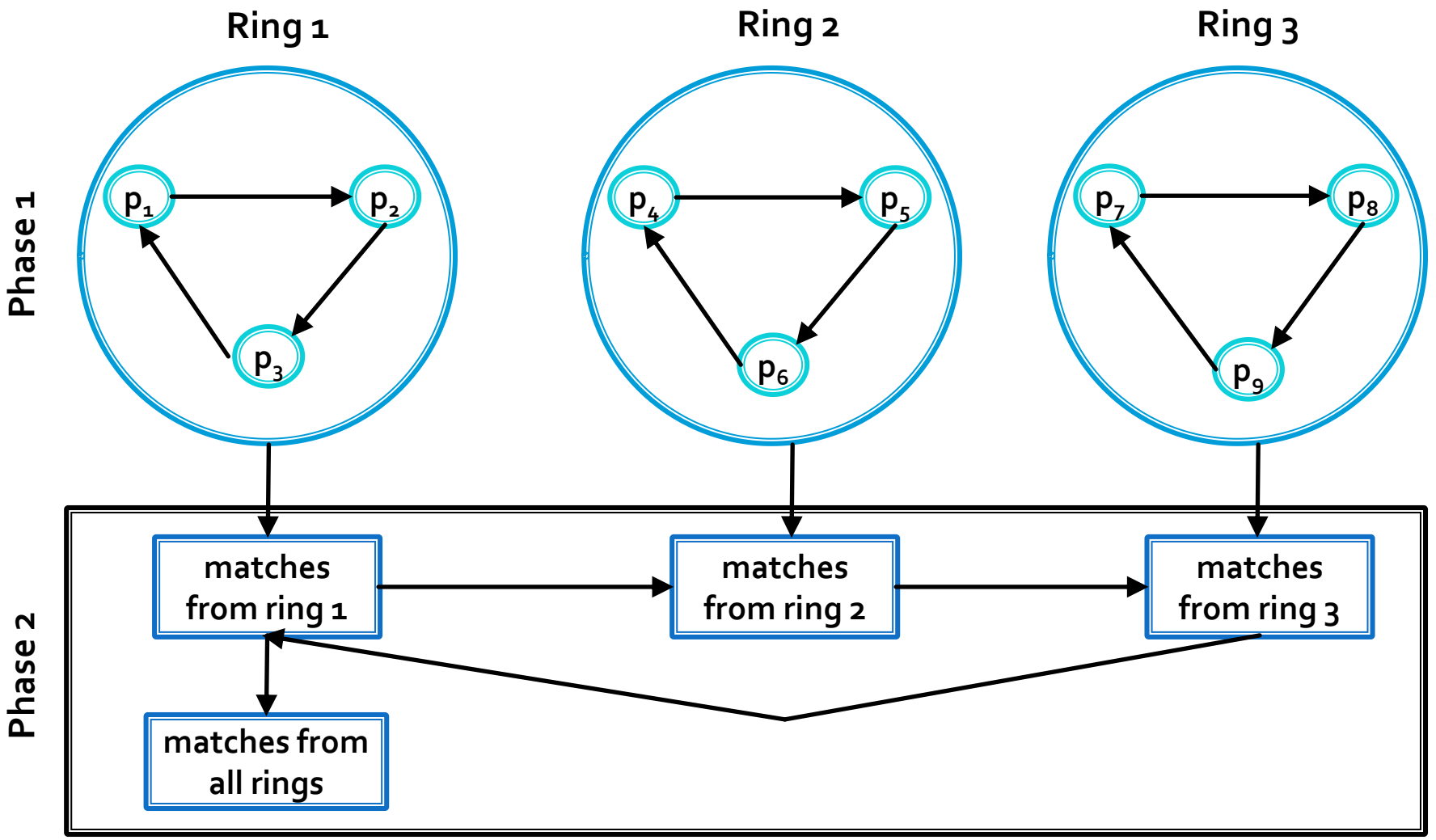
Improved Communication

- The naïve (NAI) comparison in multi-source linkage is exponential in the number of datasets, *p*, and the size of datasets even with a blocking technique

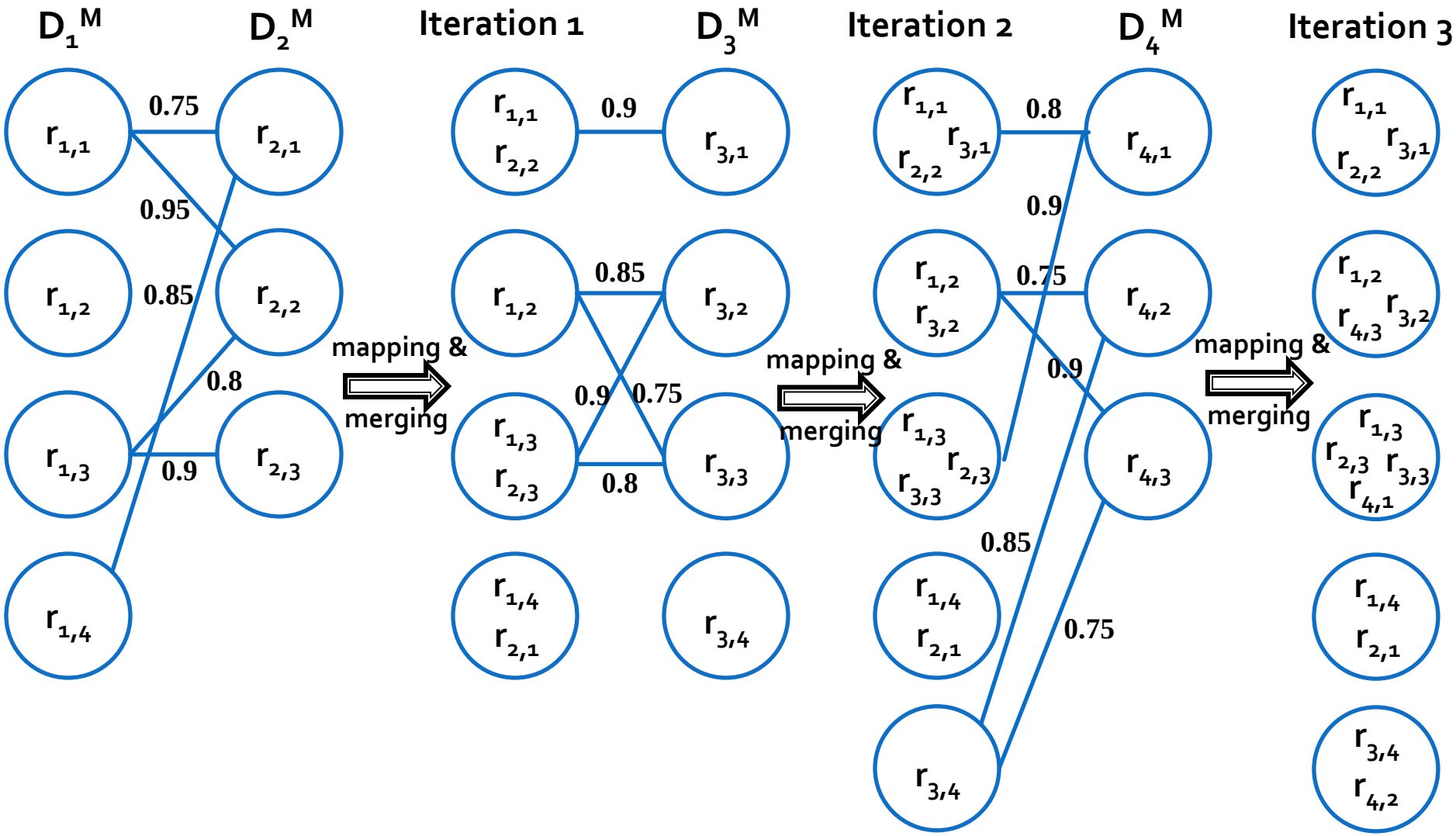
Dataset / block size	p=3	p=5	p=7	p=10
10,000 / 10	10^6	10^8	10^{10}	10^{13}
10,000 / 100	10^8	10^{12}	10^{16}	10^{22}
10,000 / 1,000	10^{10}	10^{16}	10^{22}	10^{31}
100,000 / 10	10^7	10^9	10^{11}	10^{14}
100,000 / 100	10^9	10^{13}	10^{17}	10^{23}
100,000 / 1,000	10^{11}	10^{17}	10^{23}	10^{32}

- However, most comparisons are between true non-matches (class imbalance problem) and a true matching record set must match between any subset of parties

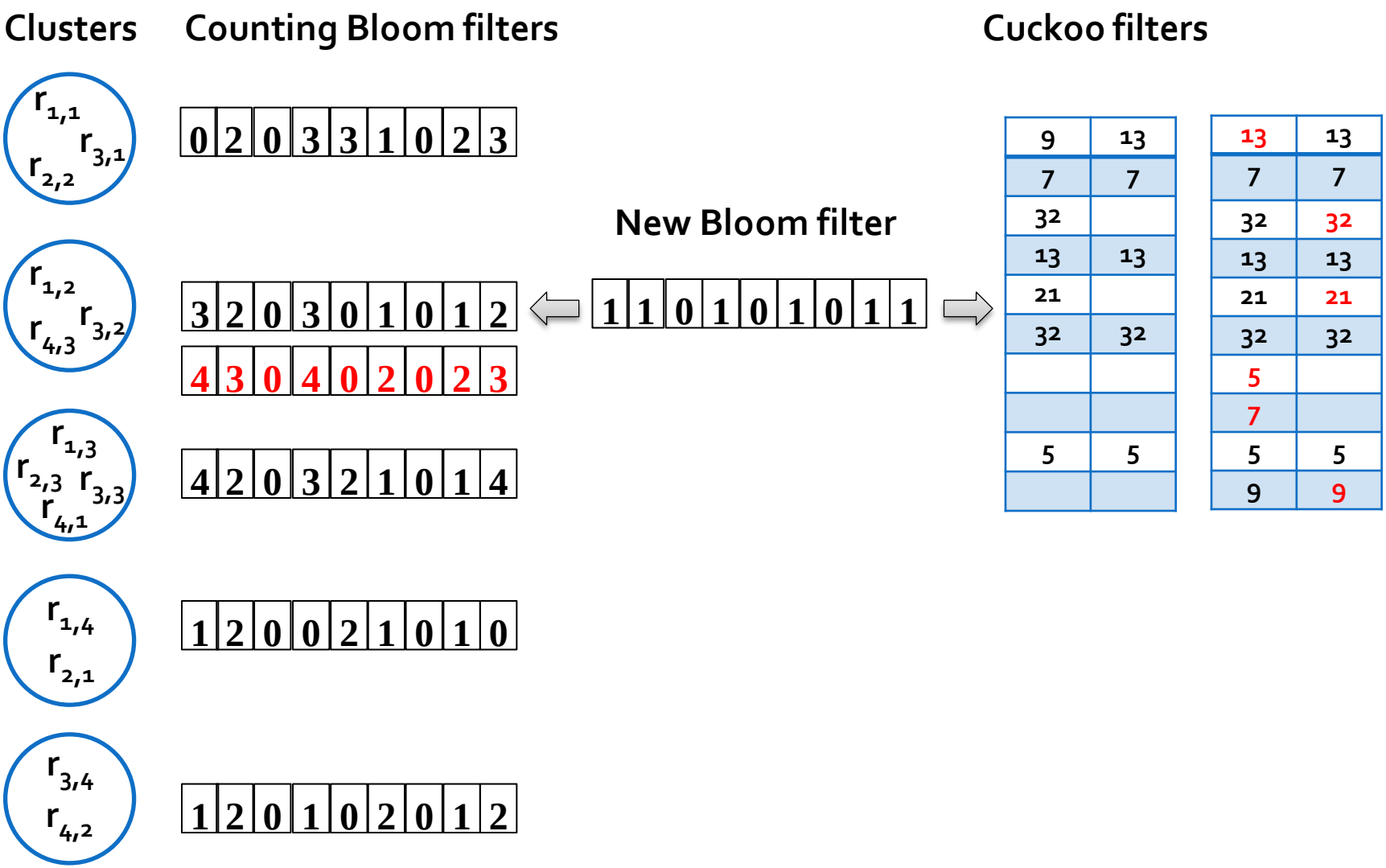
Improved Communication (contd..)



Large-scale Subset Matching

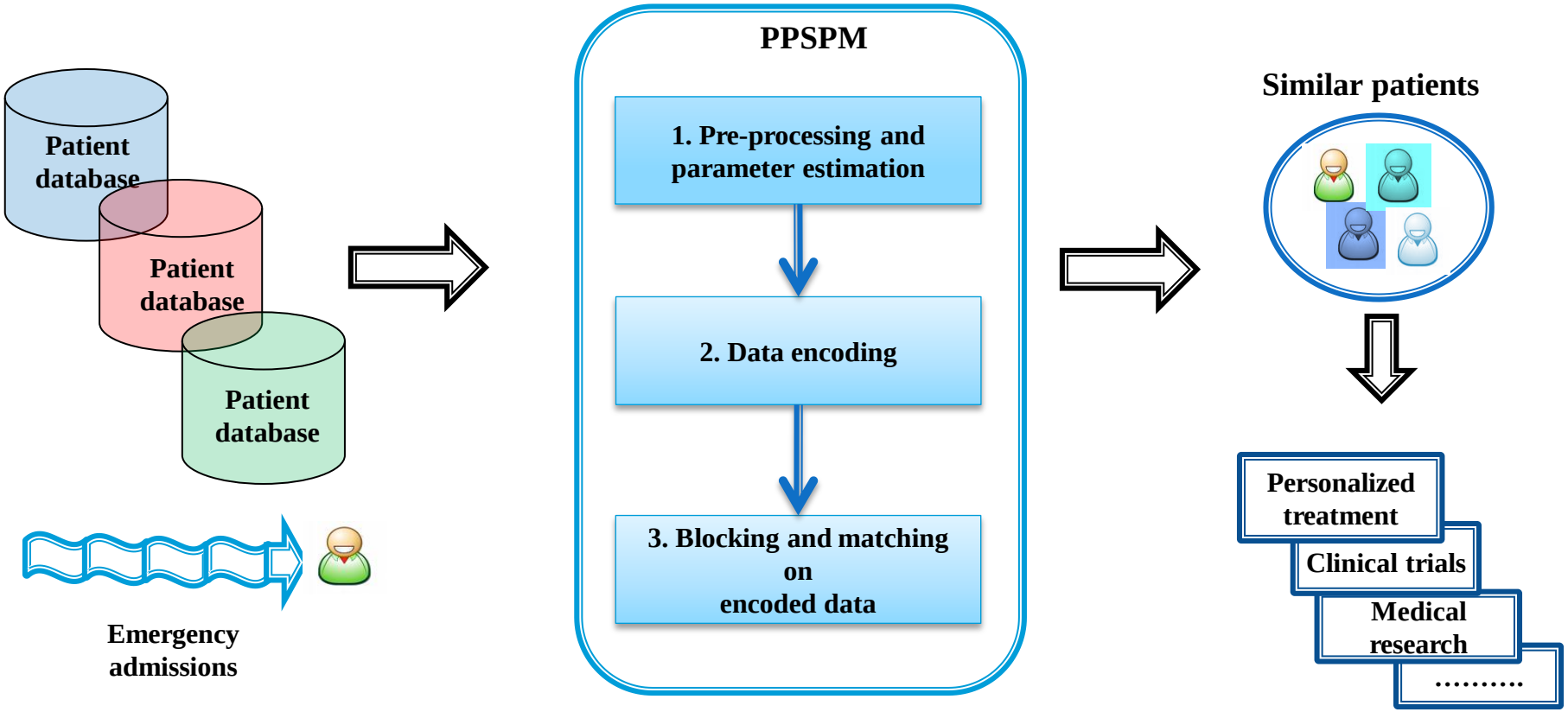


Dynamic Data Matching



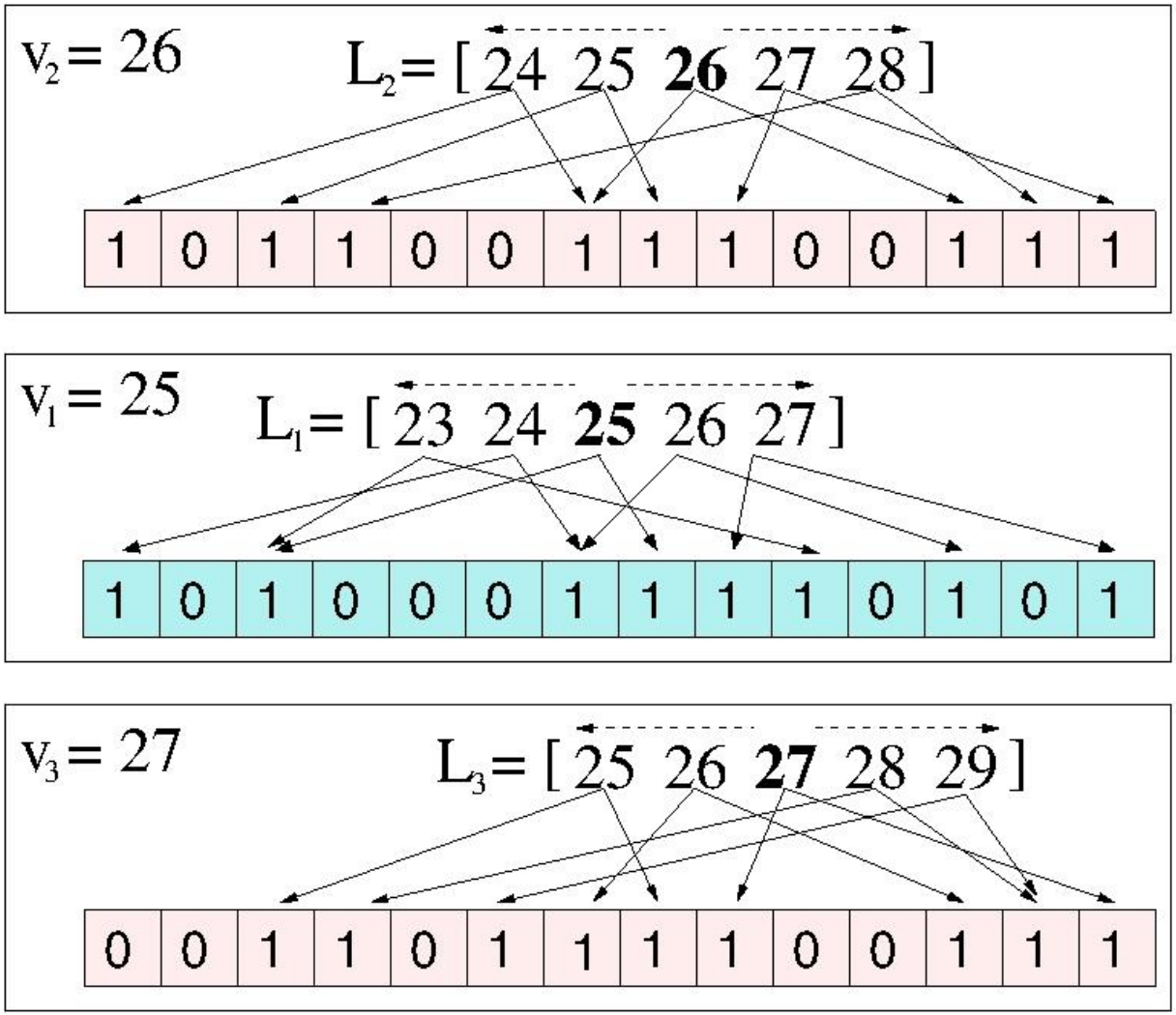
Privacy-Preserving Similar Patient Matching (PPSPM)

Healthcare applications –
Clinical trials, medical research, customized patient care



Vatsalan et al., Privacy-Preserving Matching of Similar Patients, JBI, vol. 59,pp. 285-298, 2016

BF encoding for Numerical Data

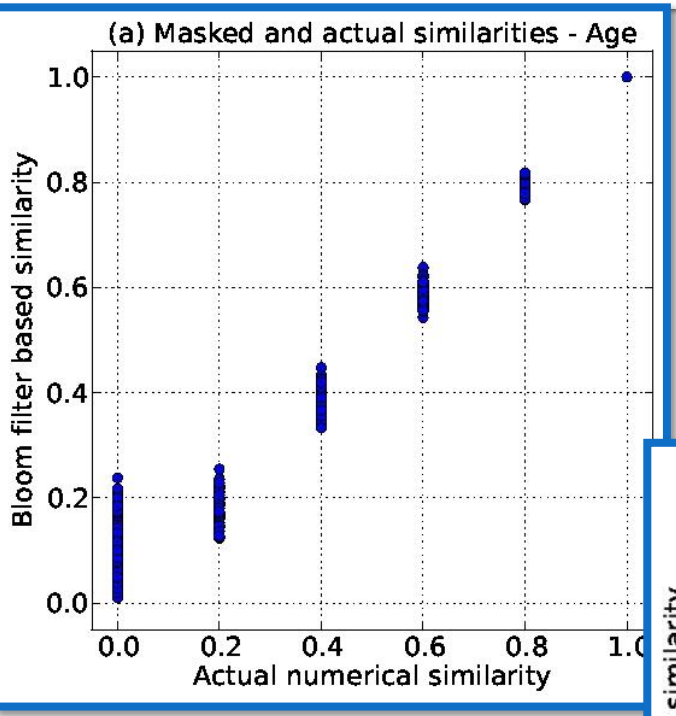


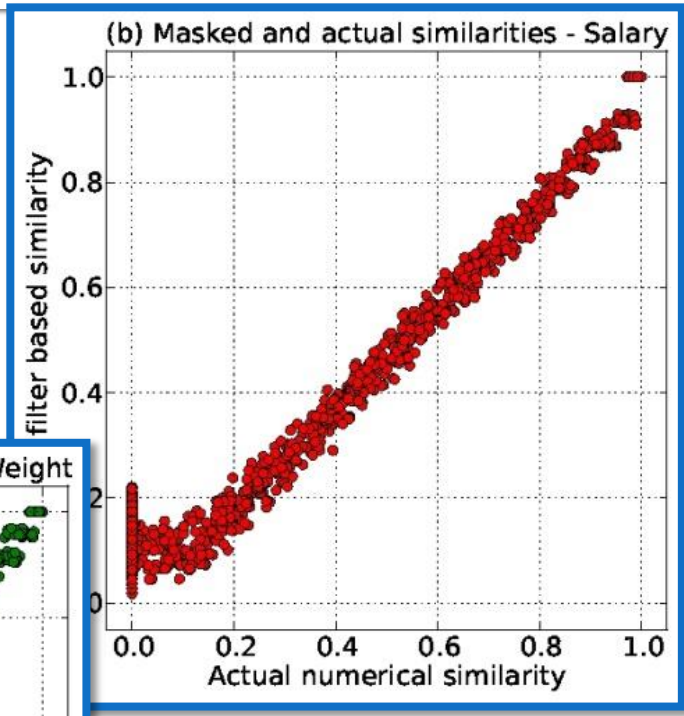
$$sim_M(v_1, v_2) = \frac{2 \times 7}{(8 + 9)} = 0.82$$

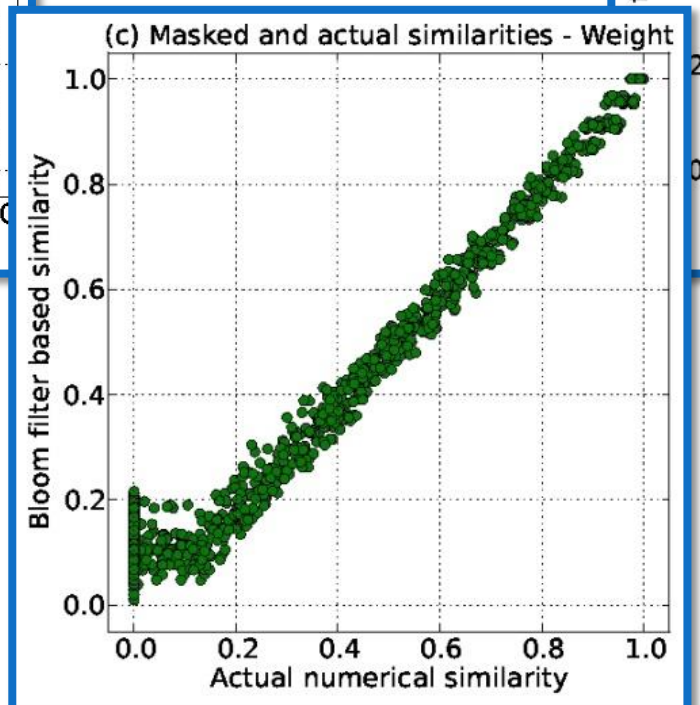
$$sim_M(v_1, v_3) = \frac{2 \times 6}{(8 + 9)} = 0.71$$

BF encoding for Numerical Data

(contd..)







Analysis

- **Complexity:**
  - Naïve (NAI) linkage (*p* databases of size *n*) – exponential growth with *p*
  - CBF-based linkage– exponential growth with ring size *r* (*r* ≤ *p/2*)
  - Clustering-based subset linkage –quadratic (2 <<< p) complexity

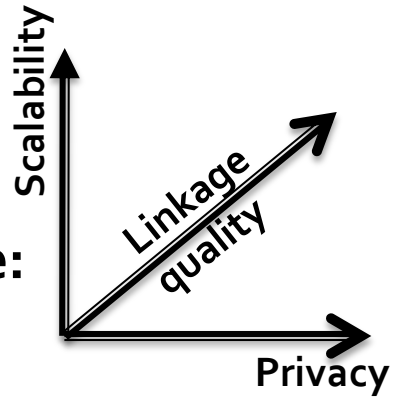
- **Privacy:**
  - Several hardening methods for BF encoding – RBF, random hashing
  - The probability of re-identifying values of records given a CBF is smaller than the probability of re-identifying record values using their BFs
  - Reduced collusion possibilities from *p(p-1)* to *p(r-1)*
  - Improved secure summation protocols against collusion attacks

- **Linkage quality:**
  - Support string and numerical in addition to categorical data matching
  - Allow approximate matching – $Dice\_sim(b_1,\ldots,b_p) = Dice\_sim(c)$
  - Subset matching and dynamic matching improve linkage quality

Evaluation Framework

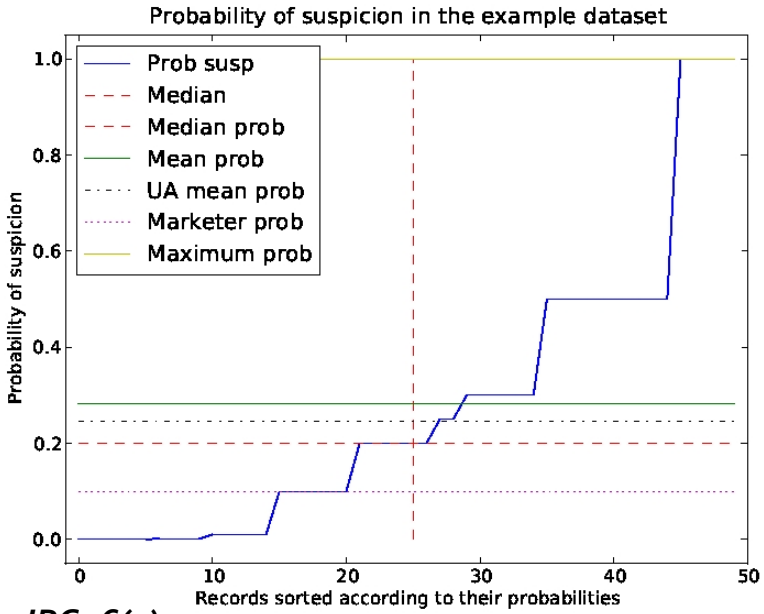
- Scalability – runtime, memory and communication
- Linkage quality – precision, recall, F-measure
- Privacy - Probability of suspicion of an encoded value:

  $P_s = 1/n_g$; Normalized $P_s = (1/n_g - 1/N) / (1 - 1/N)$



Sorted P_s values of encoded values in D^M of size n = 50.
G^M is of size N=1000.

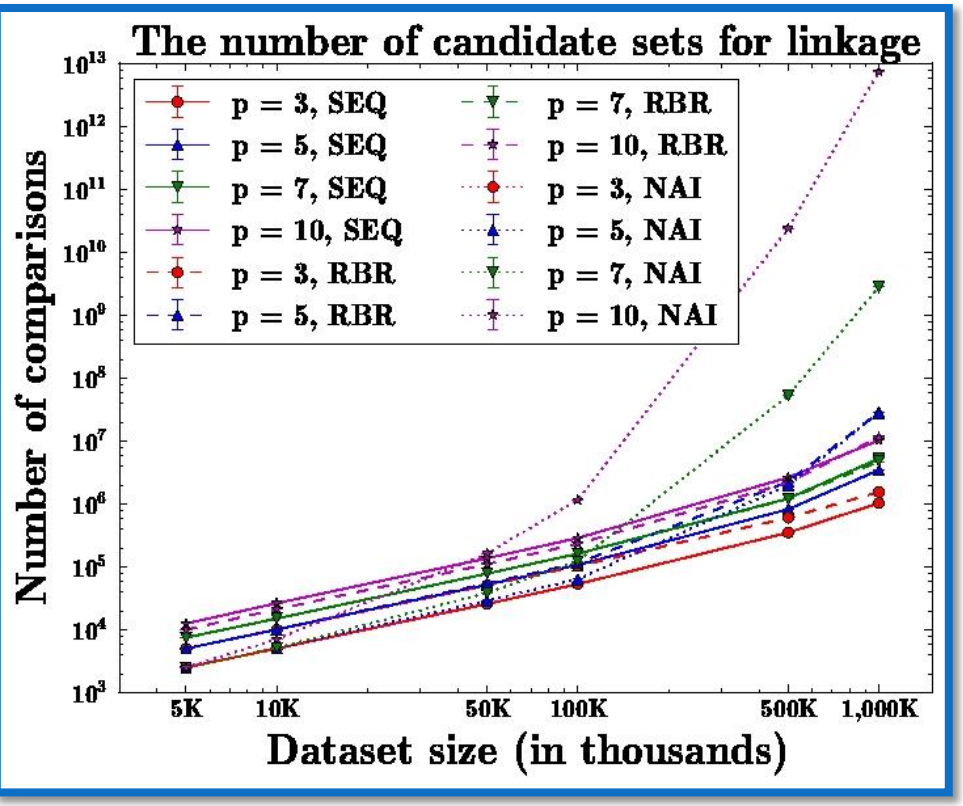
1.0	1.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	0.33
0.33	0.25	0.25	0.2	0.2	0.2	0.2	0.2	0.2	0.1
0.1	0.1	0.1	0.1	0.1	0.01	0.01	0.01	0.01	0.01
0.02	0.02	0.02	0.02	0.0	0.0	0.0	0.0	0.0	0.0



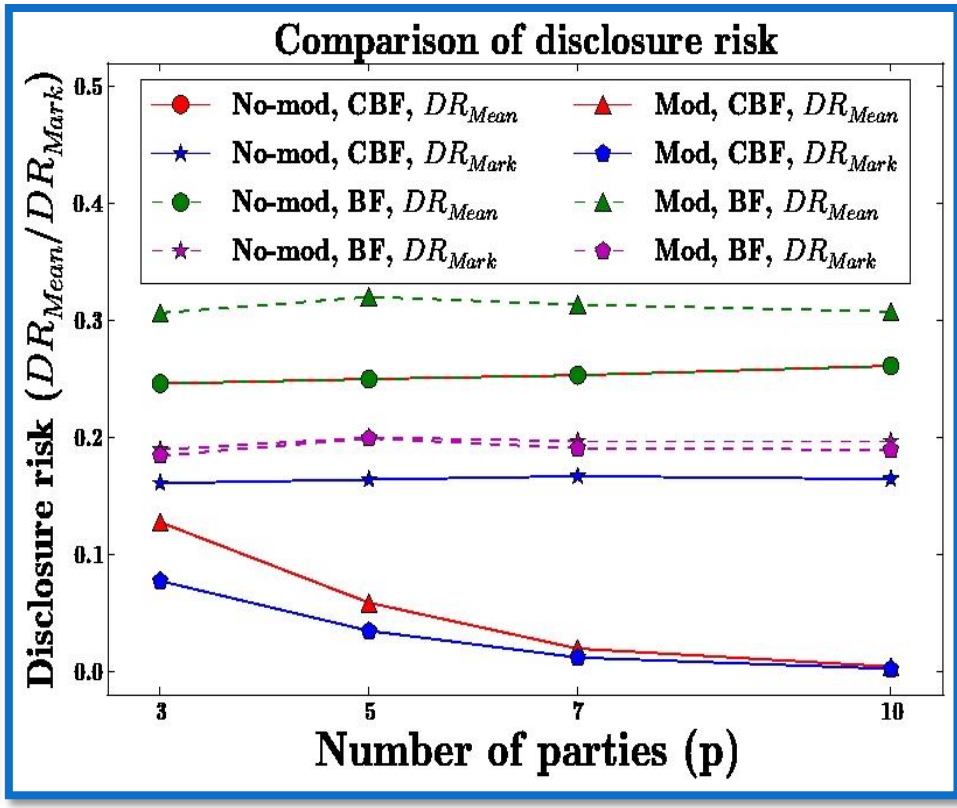
Vatsalan et al., An Evaluation Framework for PPRL, JPC, 6(1), pp. 35-75, 2014

Experimental Evaluation

Scalability

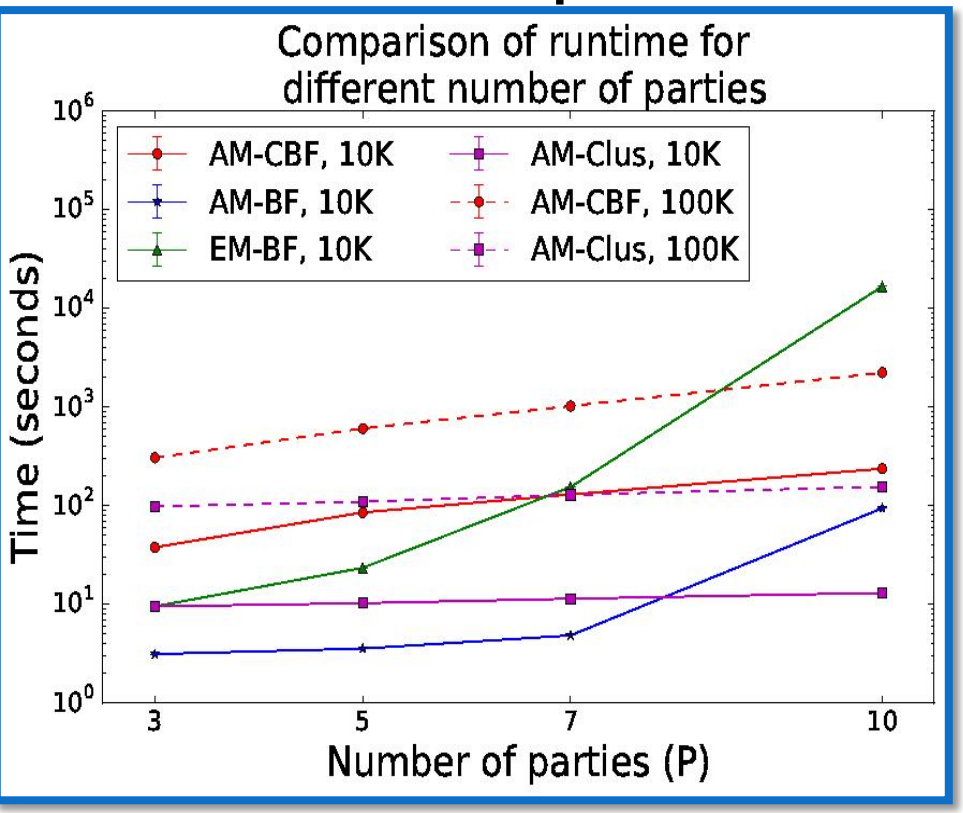


Privacy

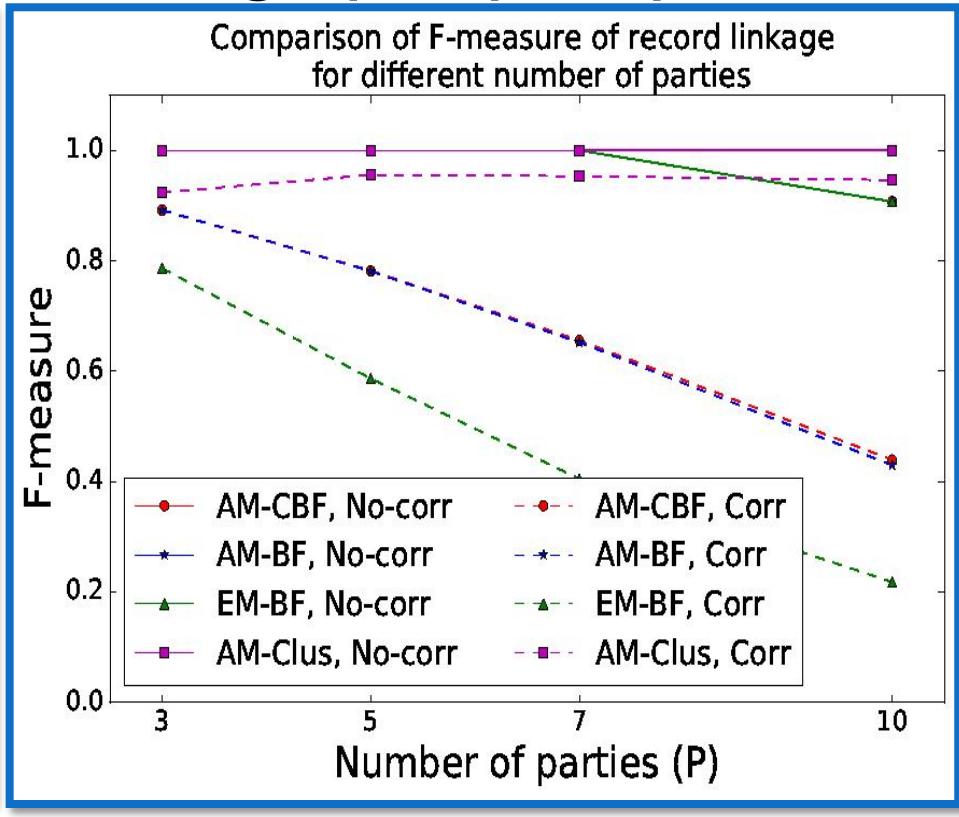


Experimental Evaluation (contd..)

Runtime comparison

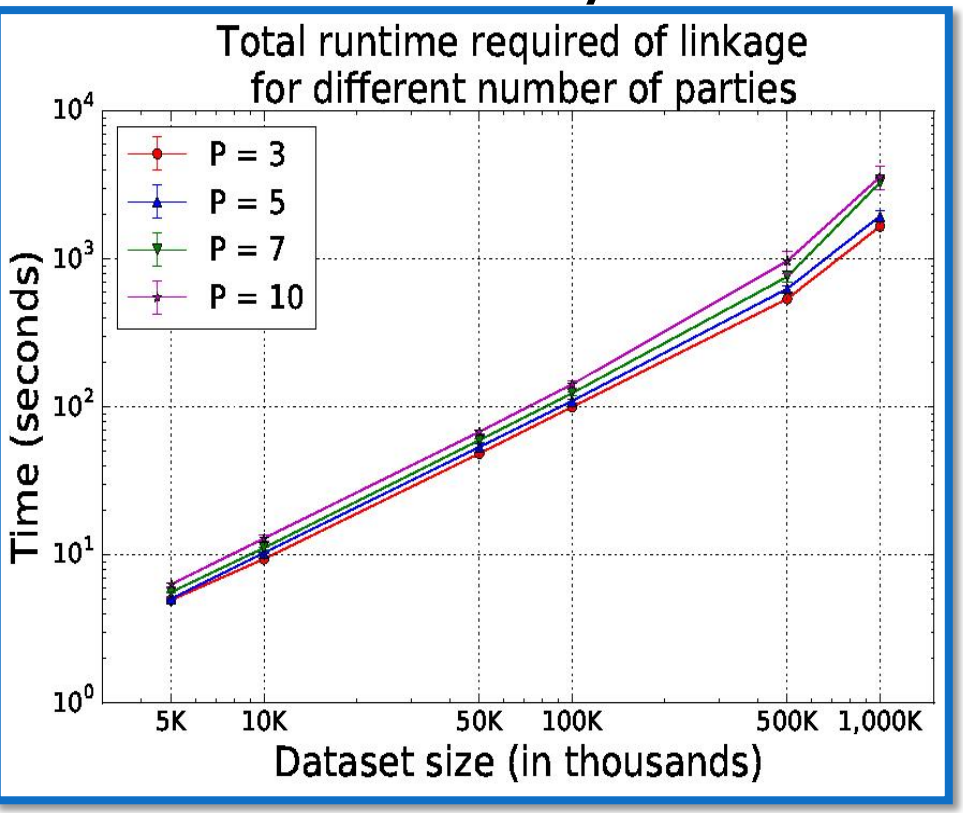


Linkage quality comparison

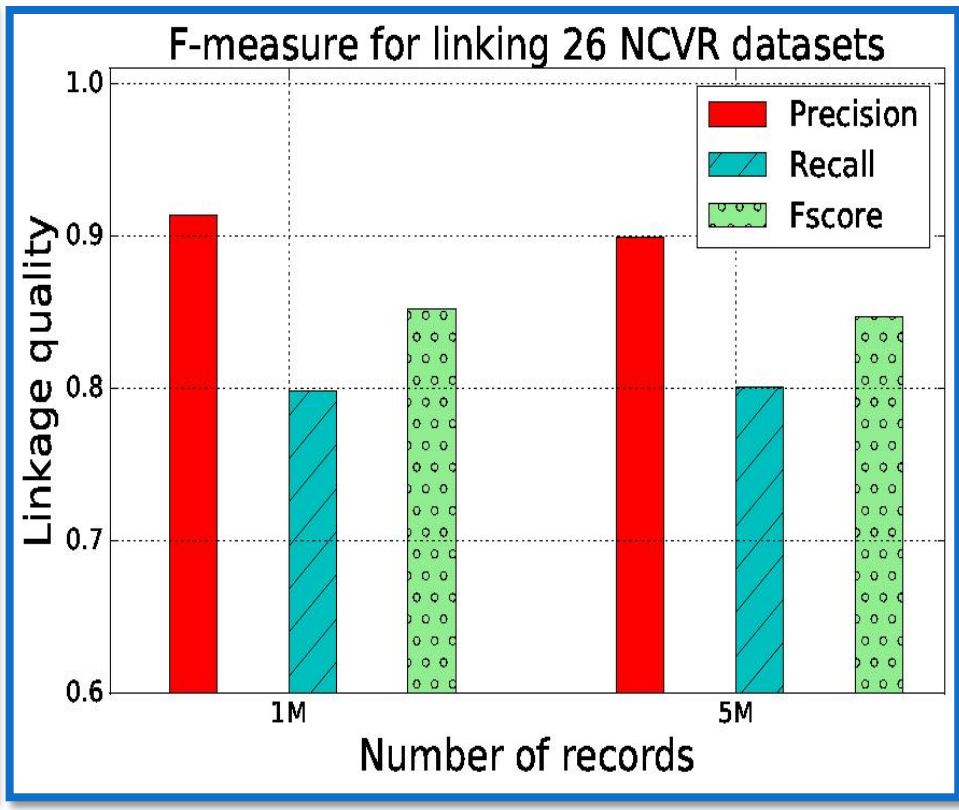


Experimental Evaluation (contd..)

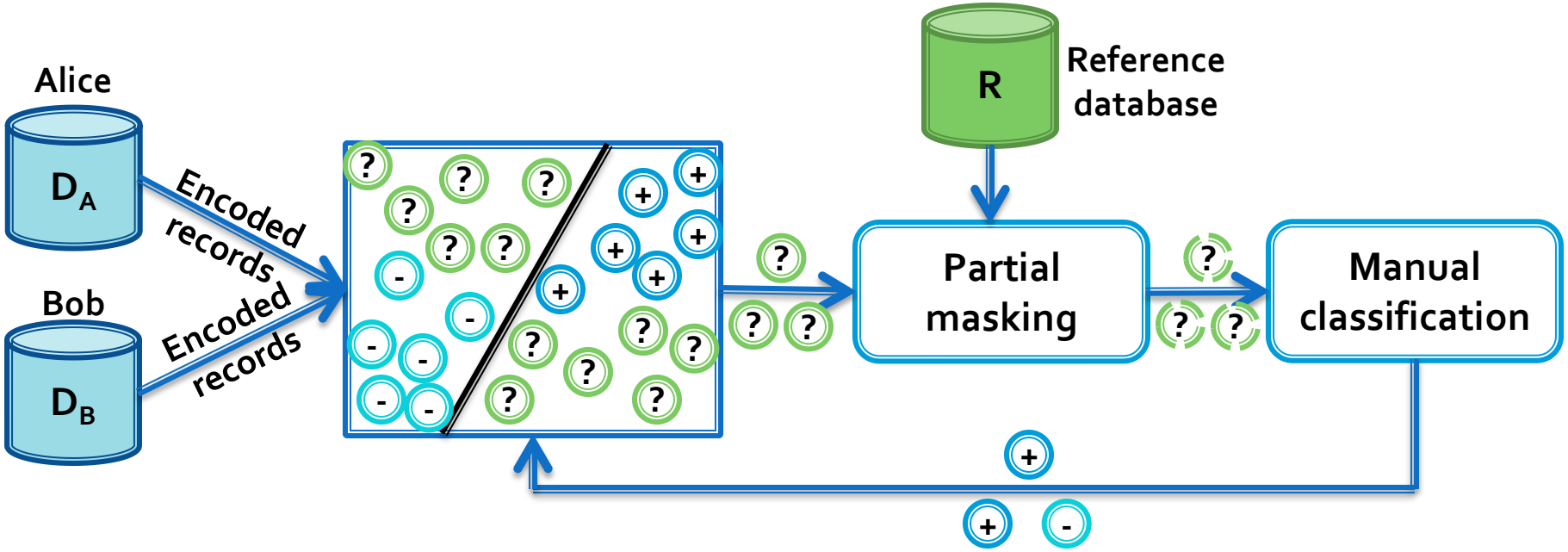
Scalability



Results for linking large datasets



Privacy-Preserving Interactive Record Linkage

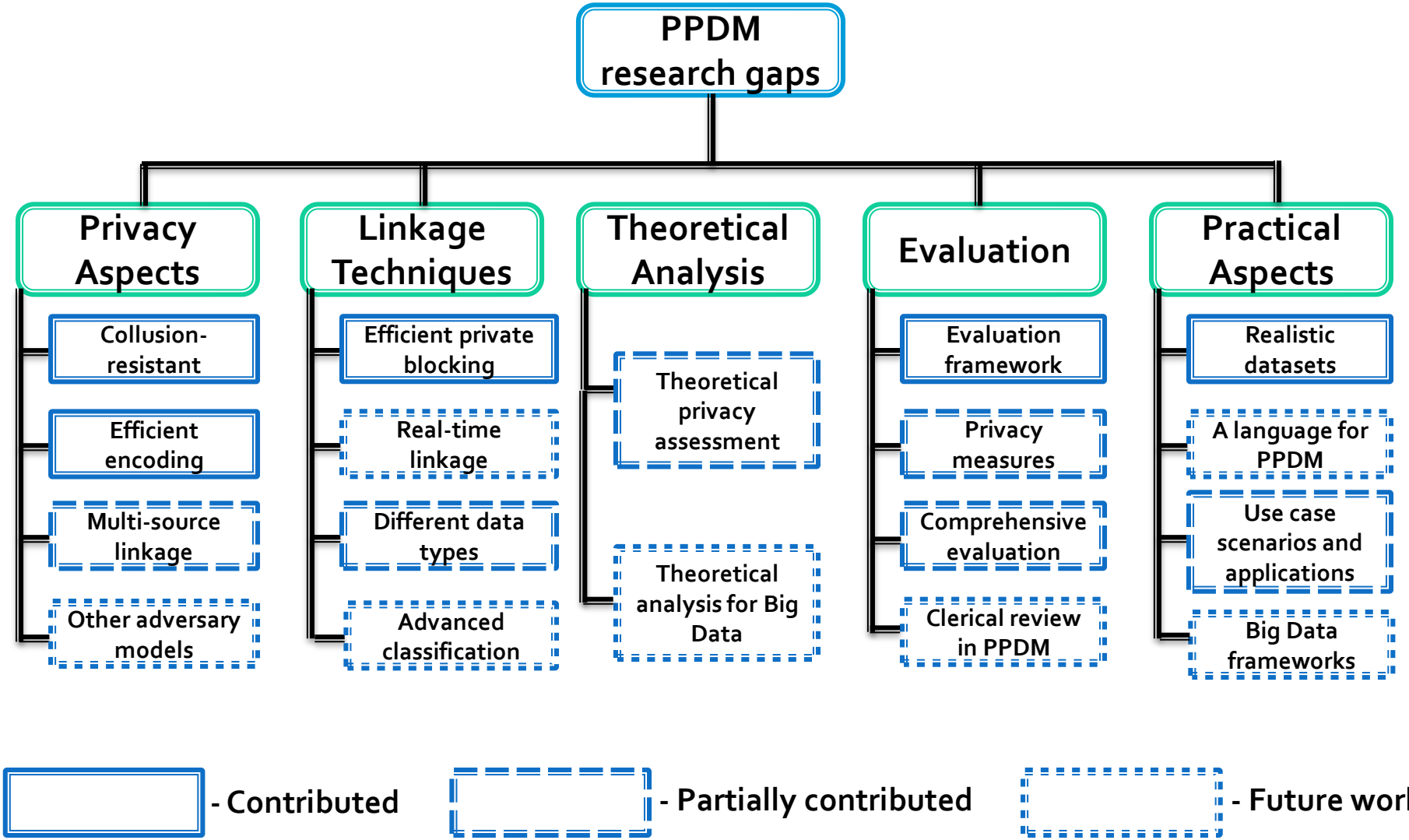


- **Partial masking** – that allows the oracle to make better decision while not revealing any sensitive information
  - Identifying and revealing which characters are matching and which are not in a pair of strings (QIDs)
  - K-anonymous masking – reveal sub-stings in QIDs that meet k-anonymous privacy while masking other characters

Privacy-Preserving Interactive Record Linkage (contd..)

- **Private comparison function** for identifying and replacing non-matching characters in a pair of strings (from Alice and Bob) with '^', for example, and matching characters with '#' – ***allows similarity calculation on masked values***
- **Initial results and example output:**
  - christine / kristine -> ^^######## / ^########
  - peter / petre -> ###^^ / ###^^
  - smith / william -> ^^^^^ / ^^^^^^^
  - richardson / robinson -> #^^^^^^### / #^^^^###
  - elizabeth / liza -> ^#####^^^^ / ####
  - miller / miller -> ###### / ######

Conclusions and Future Work



Thank You !