UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

SCALABLE AND PRIVACY-PRESERVING DATA INTEGRATION - PART 2 -

ERHARD RAHM
ZIAD SEHILI
UNIVERSITY OF LEIPZIG

www.scads.de

ScaDS DRESDEN LEIPZIG AGENDA

- ScaDS Dresden/Leipzig
- Big Data Integration
  - Scalable entity resolution / link discovery
  - Large-scale schema/ontology matching
  - Holistic data integration
- Privacy-preserving record linkage (PPRL)
  - Privacy for Big Data
  - PPRL basics
  - Scalable PPRL

- Graph-based data integration and analytics
  - Introduction
  - Graph-based data integration / business intelligence (BIIIG)
  - Hadoop-based graph analytics (GRADOOP)

Privacy

- right of individuals to determine by themselves when, how and to what extent information about them is communicated to others (Agrawal 2002)

Privacy threats

- extensive collection of personal/private information / surveillance
- Information dissemination: disclosure of sensitive/confidential information
- Invasions of privacy: intrusion attacks to obtain access to private information
- Information aggregation: combining data, e.g. to enhance personal profiles or identify persons (de-anonymization)

ScaDS DRESDEN LEIPZIG

INFORMATION / DATA PRIVACY

- Protection especially for *personally identifiable information* (PID)
  - name, birthdate, address, email address etc
  - healthcare and genetic records, financial records
  - criminal justice investigations and proceedings ...
- Challenge: preserve privacy despite need to use person-related data for improved analysis / business success (advertisement, recommendations), website optimizations, clinical/health studies, identification of criminals ...
  - tracking and profiling of web / smartphone / social network users (different kinds of cookies, canvas fingerprinting ...)
  - often user agreement needed

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

A privacy reminder from Google

To be consistent with data protection laws, we're asking you to take a moment to review key points of Google's Privacy Policy. This is not about a change we've made - but please review the key points below. **Click "I agree" to agree to the terms set out below; you can also explore other options on this page.** You can revoke your consent at any time with effect for the future.

Usage and content data

- When you use Google services to do things like write a message in Gmail or comment on a YouTube video, we store the information you create.
- When you search for a restaurant on Google Maps or watch a video on YouTube, for example, we process information about that activity — including information like the video you watched, device IDs, IP addresses, cookie data, and location.
- Our Privacy Policy contains further descriptions of the data we process.
- We treat all of this as “personal information” when it’s associated with your Google Account.
- We also process the kinds of information described above when you use apps or sites that use Google services like ads, Analytics, and the YouTube video player.

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

Information that we collect

- **Information you give us.** For example, many of our services require you to sign up for a Google Account. When you do, we'll ask for personal information, like your name, email address, telephone number or credit card to store with your account. If you want to take full advantage of the sharing features we offer, we might also ask you to create a publicly visible Google Profile, which may include your name and photo.
- **Information we get from your use of our services.** We collect information about the services that
  - Device information

    We collect device-specific information (such as your hardware model, operating system version, unique device identifiers, and mobile network information including phone number). Google may associate your device identifiers or phone number with your Google Account.
  - **Log information**
  - **Location information**

    When you use Google services, we may collect and process information about your actual location. We use various technologies to determine location, including IP address, GPS and other sensors that may, for example, provide Google with information on nearby devices, Wi-Fi access points and mobile towers.
  - **Unique application numbers**

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

Purposes of the data processing

We process this data for the purposes described in our policy, including to:

- Help our services deliver more useful, customized content such as more relevant search results, based on your interests derived from such data;
- Improve the quality of our services and develop new ones;
- Deliver ads based on your interests, which we can determine based on this data, like ads that are related to things such as search queries or videos you've watched on YouTube;
- Improve security by protecting against fraud and abuse; and
- Conduct analytics and measurement to understand how our services are used.

Combining data

We also combine this data among our services and across your devices for these purposes. For example, we show you ads based on information about your interests, which we can derive from your use of Search and Gmail, and we use data from trillions of search queries to build spell-correction models that we use across all of our services.

ScaDS DRESDEN LEIPZIG

PRIVACY FOR BIG DATA

- Need for comprehensive privacy support (“privacy by design”)
- Privacy-preserving publishing of datasets
  - Anonymization of datasets
- Privacy-preserving data mining
  - analysis of anonymized data without re-identification
- Privacy-preserving record linkage
  - object matching with encoded data to preserve privacy
  - prerequisite for privacy-preserving data mining

ScaDS DRESDEN LEIPZIG

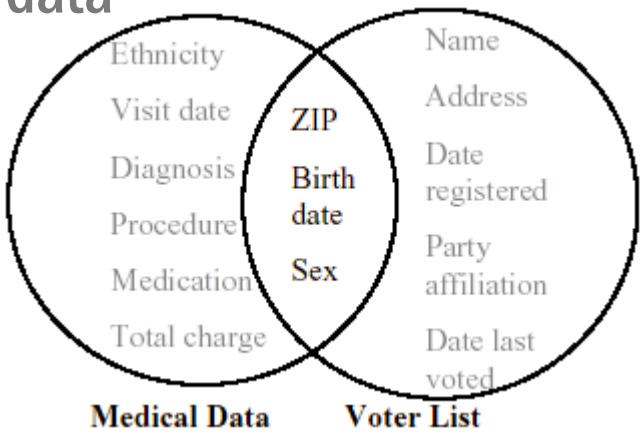
ANONYMIZATION VS PSEUDONYMIZATION

- Anonymization
  - removing, generalizing or changing personally identifying attributes so that people whom the data describe remain anonymous
  - no way to identify different records for same person (e.g., different points in time) or to match/combine records from different sources
- Pseudonymization
  - replace most identifying fields within a data record are replaced by one or more artificial identifiers, or pseudonyms
  - one-way pseudonymization (e.g. one-way hash functions) vs. re-identifiable pseudonymization
  - records with same pseudonym can be matched
  - improved potential for data analysis

ScaDS DRESDEN LEIPZIG

RE-IDENTIFICATION OF „ANONYMOUS DATA“ (SWEENEY 2001)

- US voter registration data
  - 69% unique on postal code (ZIP) and birth date
  - 87% US-wide with sex, postal code and birth data
- Solution approach: K-Anonymity
  - any combination of values appears at least k times
  - generalize values, e.g., on ZIP or birth date



ScaDS DRESDEN LEIPZIG

K-ANONYMITY EXAMPLE

ID	ZIP	AGE	DISEASE	TREATMENT
1	12345	23	Gastric ulcer	Antacid
2	12345	29	Gastritis	Acid-reducing drug
3	12363	41	Flu	Antipyretic drug
4	12361	43	Stomach cancer	Cytostatic drug
5	12362	59	Pneumonia	Antibiotics
6	12471	52	Bronchitis	Antibiotics
7	12473	55	Flu	Antipyretic drug

(a) Microdata-table

ID	ZIP	AGE	DISEASE	TREATMENT
1	123**	[20-29]	Gastric ulcer	Antacid
2	123**	[20-29]	Gastritis	Acid-reducing drug
3	123**	[40-49]	Flu	Antipyretic drug
4	123**	[40-49]	Stomach cancer	Cytostatic drug
5	123**	[50-59]	Pneumonia	Antibiotics
6	124**	[50-59]	Bronchitis	Antibiotics
7	124**	[50-59]	Flu	Antipyretic drug

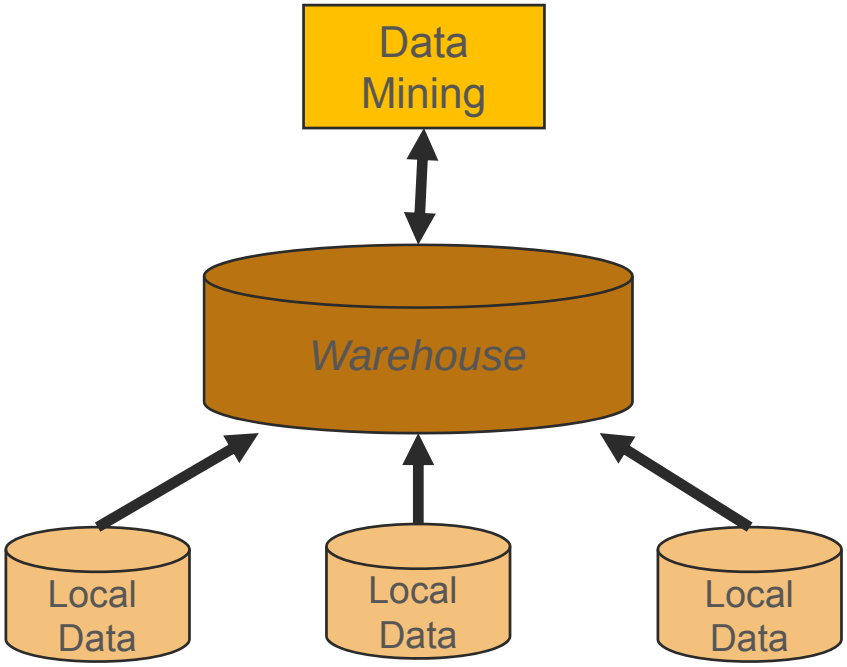
(b) 2-anonymous table

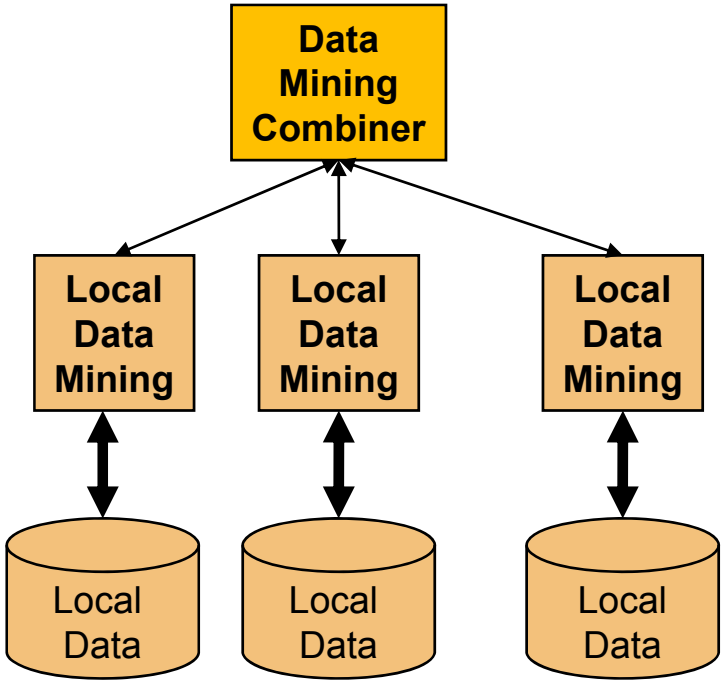
from: Nielsen et al: Proc BTW 2015

ScaDS DRESDEN LEIPZIG

PRIVACY-PRESERVING DATA MINING

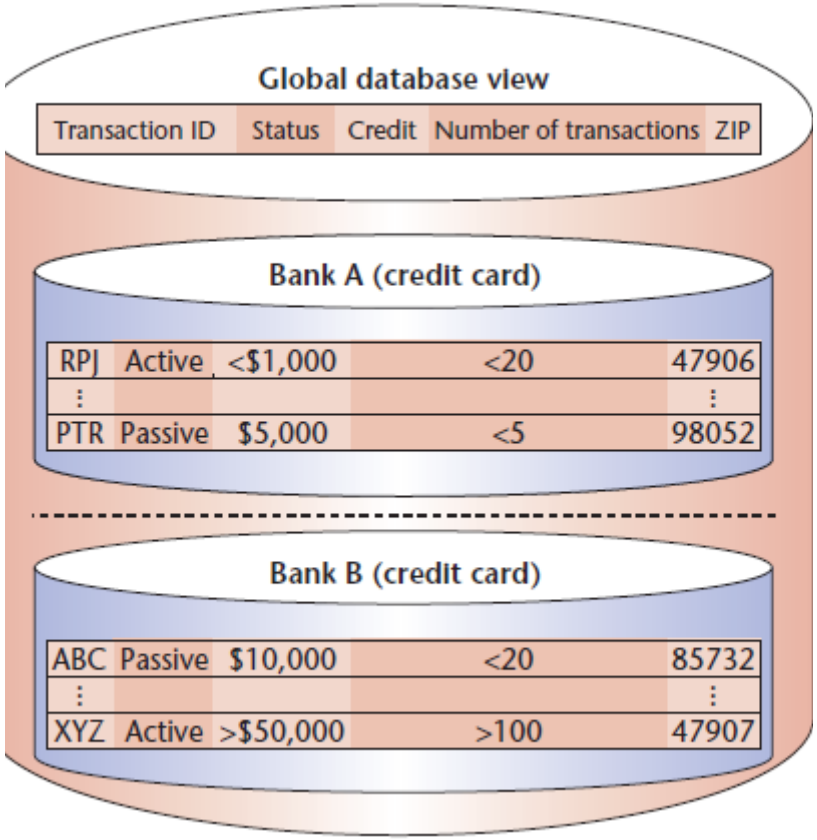
- Physically integrated data (e.g. data warehouse) about persons entails greatest privacy risks
- Data mining over distributed data can better protect personal data by limiting data exchange, e.g. using SMC methods

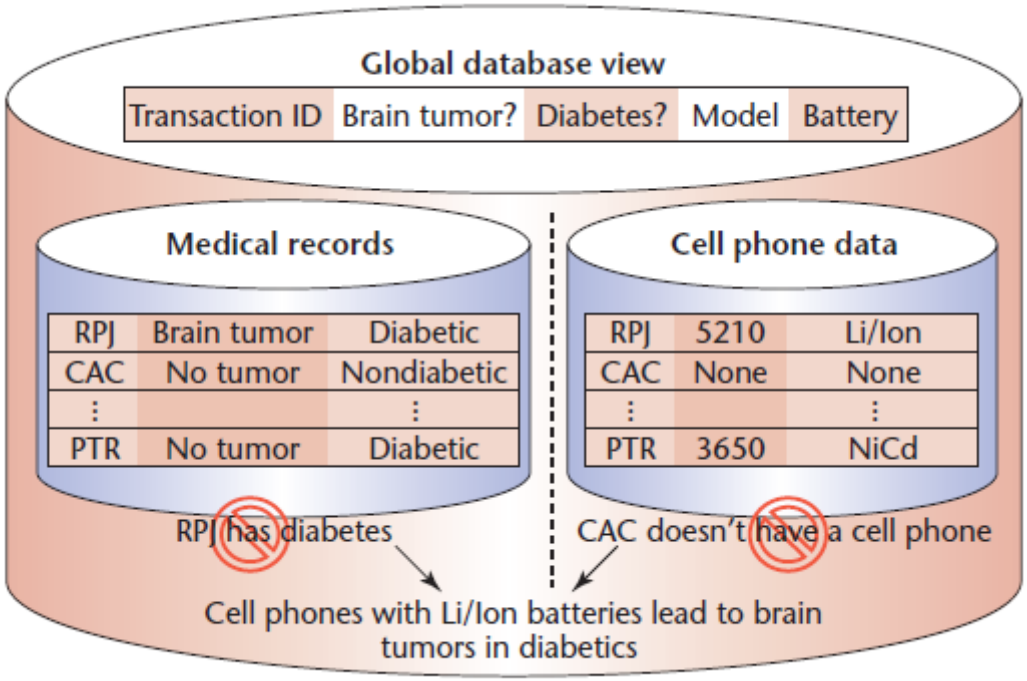




ScaDS DRESDEN LEIPZIG

HORIZONTAL VS VERTICAL DATA DISTRIBUTION





Source: J. Vaidya, C. Clifton: Privacy-Preserving data mining: Why, How, and When. IEEE Security&Privacy 2004

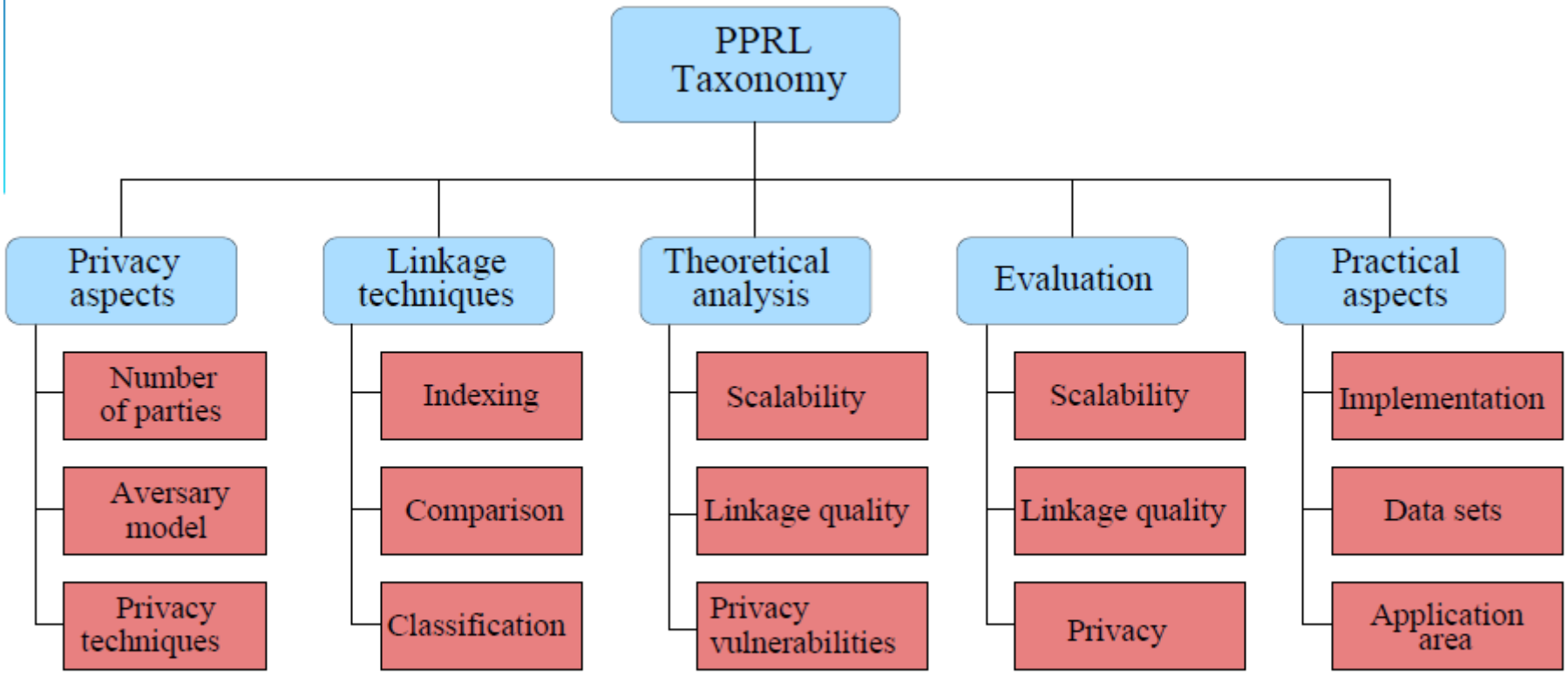
ScaDS DRESDEN LEIPZIG

PRIVACY-PRESERVING RECORD LINKAGE (PPRL)

- object matching with encoded data to preserve privacy
  - data exchange / integration of person-related data
  - many uses cases: medicine, sociology (“population informatics”), business, ...
- privacy aspects
  - need to support secure 1-way encoding (pseudonymization)
  - protection against attacks to identify persons
- conflicting requirements:
  - high privacy
  - match effectiveness (need to support fuzzy matches)
  - scalability to large datasets and many parties

ScaDS DRESDEN LEIPZIG

PRIVACY-PRESERVING RECORD LINKAGE



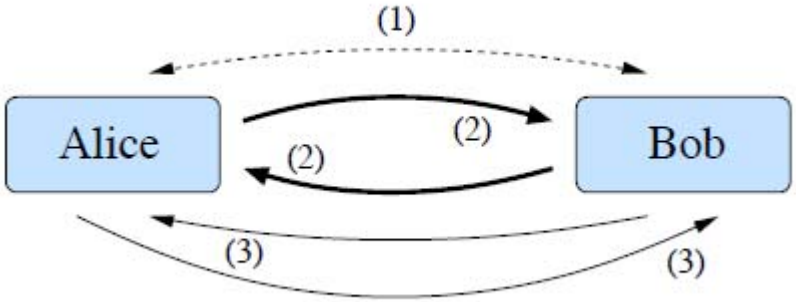
data encryption

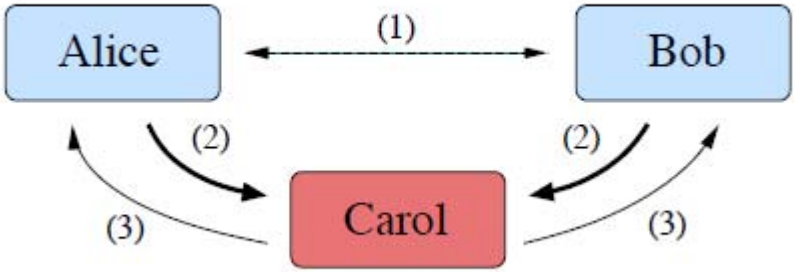
blocking
similarity functions

Vatasalan, Christen, Verykios: A taxonomy of privacy-preserving record linkage techniques. Information Systems 2013

ScaDS DRESDEN LEIPZIG

BASIC PPRL CONFIGURATIONS



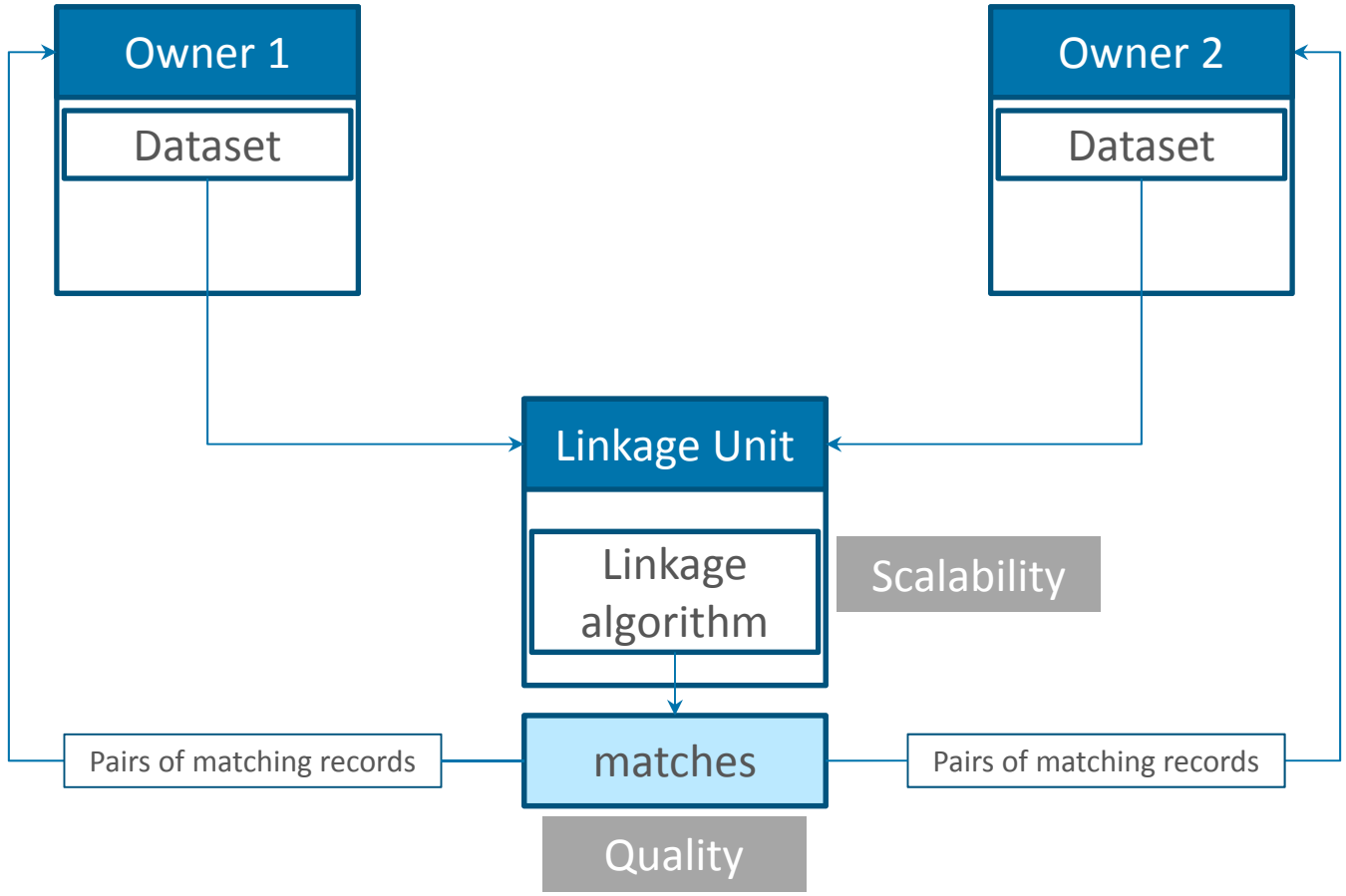


- Two-party protocols
  - only two data owners communicate who wish to link their data
- Three-party protocols
  - Use of a trusted third party (linkage unit, LU)
  - LU will never see unencoded data, but collusion is possible
- Multi-party protocols (> 2 data owners)
  - with or without linkage unit

ScaDS DRESDEN LEIPZIG

(PP)RL PROTOCOL

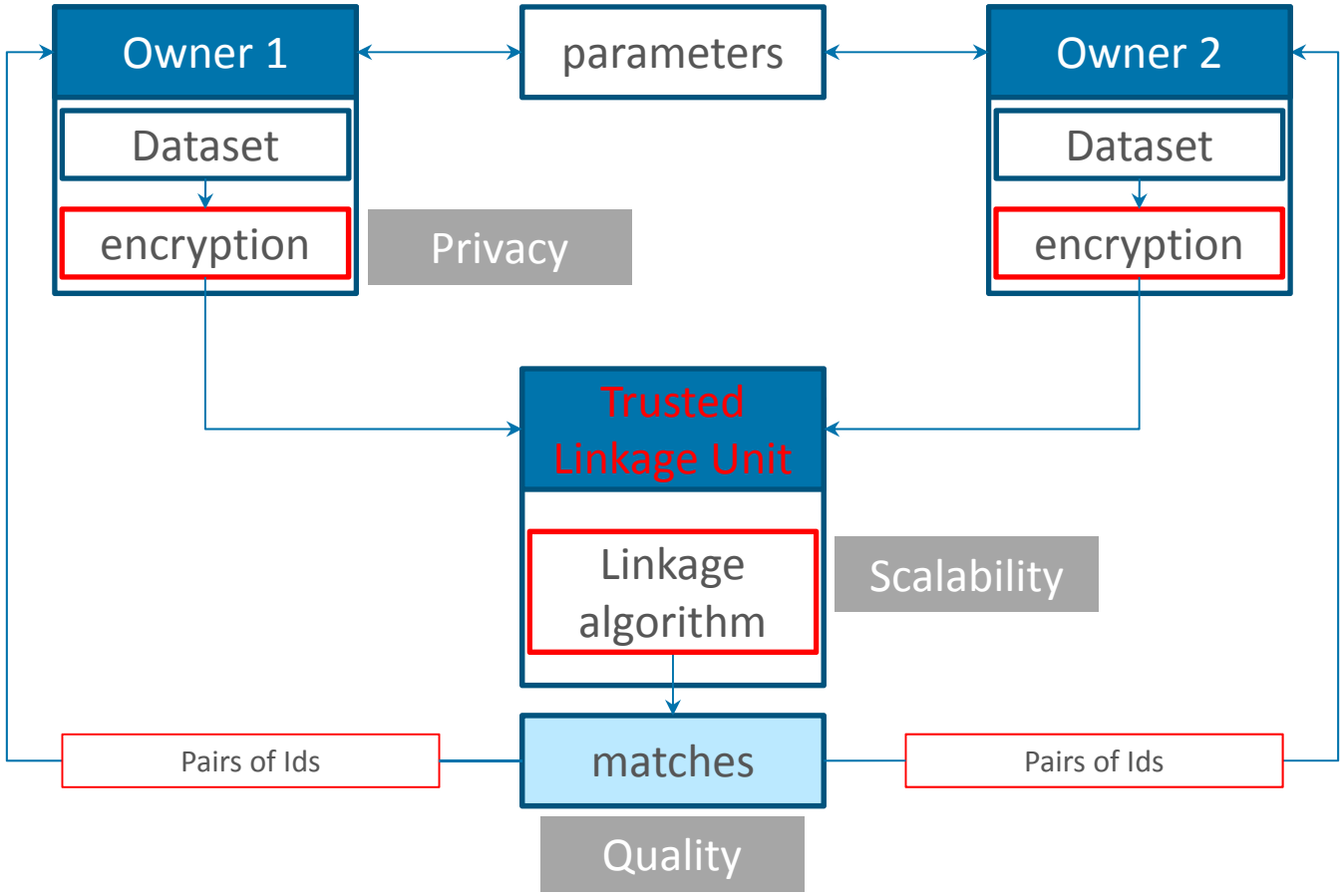
- Simple record linkage protocol



ScaDS DRESDEN LEIPZIG

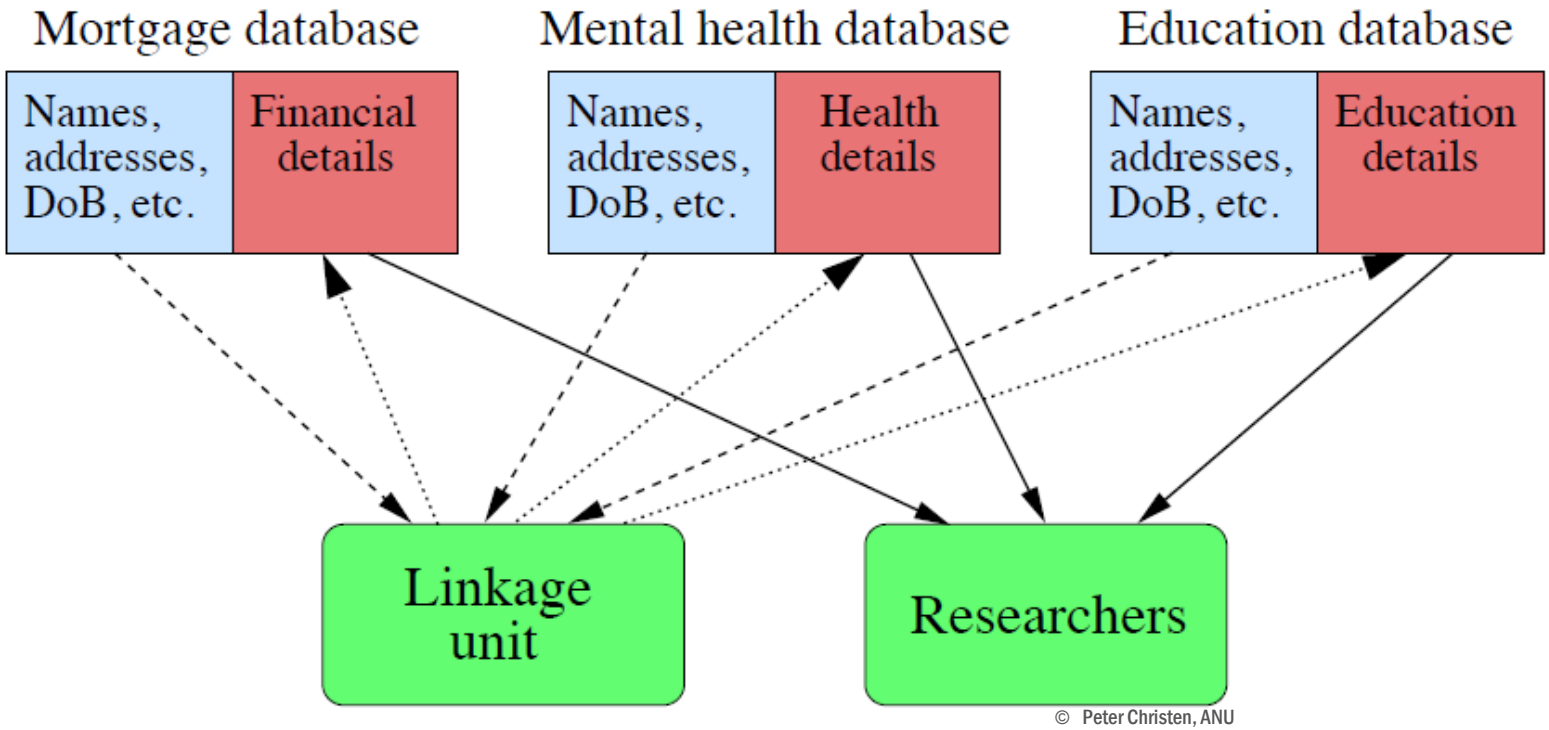
PPRL PROTOCOL

- Thee party protocol (Abstract)



ScaDS DRESDEN LEIPZIG

PPRL EXAMPLE IN HEALTH DOMAIN

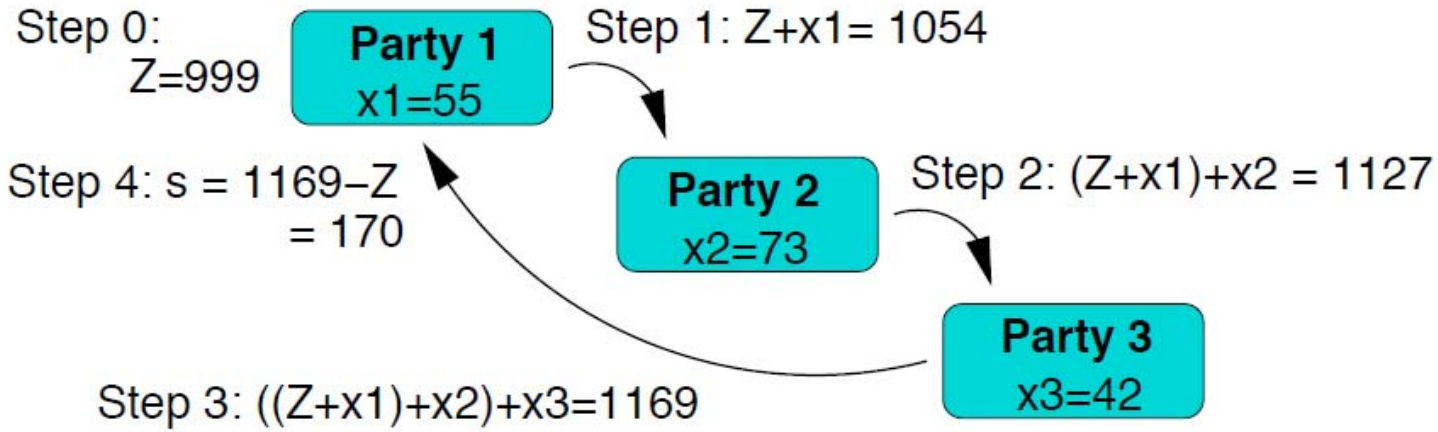


C.W. Kelman, A.J. Bass. C.D. Holman: Research use of linked health data--a best practice protocol. Aust NZ J Public Health. 2002

ScaDS DRESDEN LEIPZIG

SECURE MULTI-PARTY COMPUTATION (SMC)

- Compute a function across several parties, such as no party learns the information from the other parties, but all receive the final results
- Example 1: millionaire problem
  - two millionaires, Alice and Bob, are interested in knowing which of them is richer but without revealing their actual wealth.
- Example 2: secure summation



ScaDS DRESDEN LEIPZIG

PRIVACY ASPECTS

- Adversary models (from crytography)
  - Honest-But-Curious: parties follow agreed-on protocols
  - Malicious
- Privacy attacks
  - Crack data encoding based on background knowledge:
    - Frequency attack
    - Dictionary attack
  - Collusion between parties

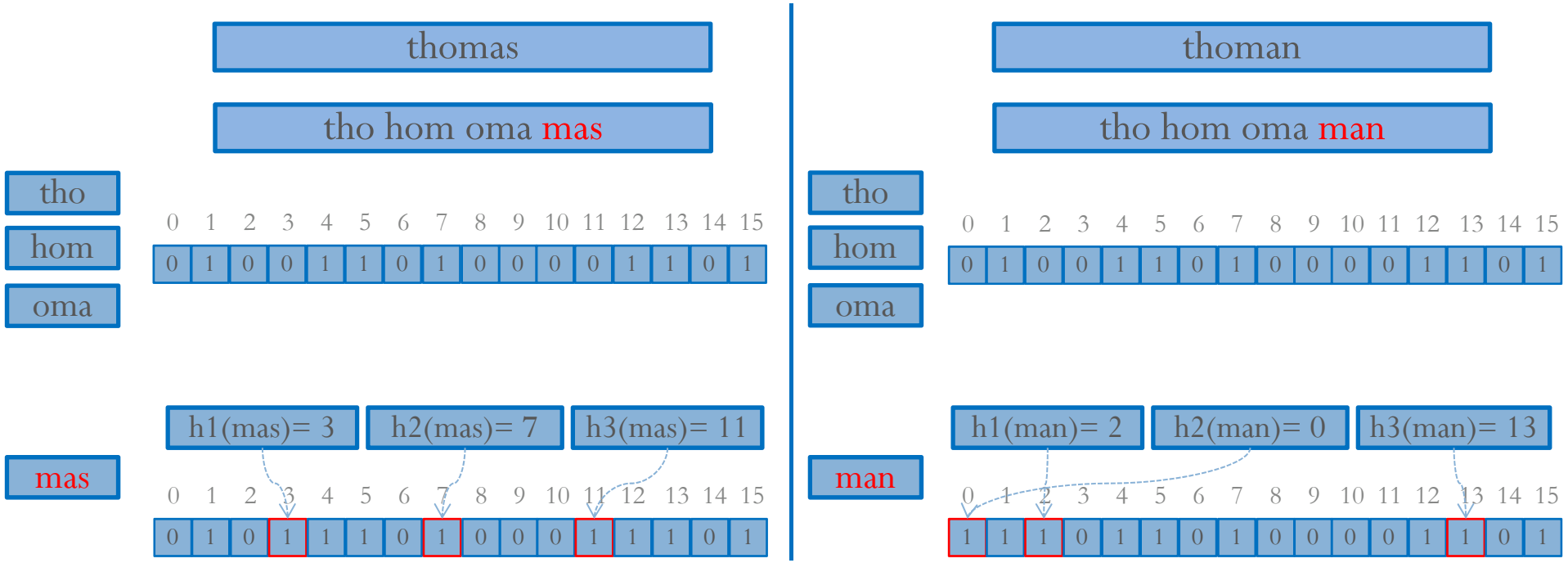
ScaDS DRESDEN LEIPZIG

PPRL WITH BLOOM FILTERS

- effective and simple encoding uses cryptographic bloom filters (Schnell et al., 2009)
- tokenize all match-relevant attribute values, e.g. using bigrams or trigrams
  - typical attributes: first name, last name (at birth), sex, date of birth, country of birth, place of birth
- map each token with a family of one-way hash functions to fixed-size bit vector (fingerprint)
  - original data cannot be reconstructed
- match of bit vectors (Jaccard similarity) is good approximation of true match result

ScaDS DRESDEN LEIPZIG

SIMILARITY COMPUTATION - EXAMPLE



$$Sim_{Jaccard}(r1, r2) = (r1 \wedge r2) / (r1 \vee r2)$$

$$Sim_{Jaccard}(r1, r2) = 7/11$$

ScaDS DRESDEN LEIPZIG

SCALABLE PPRL

- same optimization techniques than for regular object matching apply
- (private) blocking approaches
- filtering for specific similarity metrics / thresholds to reduce number of comparisons
  - privacy-preserving PPJoin (P4Join)
  - metric space: utilize triangular inequality
- parallel linkage (Hadoop, GPUs, ...)

ScaDS DRESDEN LEIPZIG

PRIVATE BLOCKING

- Many possibilities: standard blocking, sorted-neighborhood, Locality-sensitive Hashing (LSH) ...
- Simple approach for standard blocking
  - Each database performs agreed-upon standard blocking, e.g. using soundex or another function on selected attributes (e.g., names)
  - Encoded records are transferred blockwise to LU, possibly with added noise records for improved security
  - LU only matches records per block
- different blocks could be processed in parallel

ScaDS DRESDEN LEIPZIG

PP-JOIN: POSITION PREFIX JOIN (XIAO ET AL, 2008)

- one of the most efficient ***similarity join*** algorithms
  - determine all pairs of records with $sim_{Jaccard}(x,y) \geq t$
- use of filter techniques to reduce search space
  - length, prefix, and position filter
- relatively easy to run in parallel
- good candidate to improve scalability for PPRL
- evaluate set bit positions instead of (string) tokens

ScaDS DRESDEN LEIPZIG

LENGTH FILTER

- matching records pairs must have similar lengths

$$\text{Sim}_{\text{Jaccard}}(x, y) \geq t \Rightarrow |x| \geq |y| * t$$

- length / cardinality: number of set bits in bit vector
- Example for minimal similarity $t = 0{,}8$:

ID	Bit vector	card.
B	1 0 1 0 0 0 0 0 1 1 0 0 0	4
C	0 0 0 1 1 1 1 1 1 1 0 0 0	7
A	0 1 0 1 1 1 1 1 1 1 0 0 0	8

length filter
$7 * 0.8 = 5.6 > 4$

  - record B of length 4 cannot match with C and all records with greater length (number of set positions), e.g., A

ScaDS DRESDEN LEIPZIG

PREFIX FILTER

- Similar records must have a minimal overlap α in their sets of tokens (or set bit positions)

$$\text{Sim}_{\text{Jaccard}}(x, y) \geq t \quad \Leftrightarrow \quad Overlap(x, y) \geq \alpha = \lceil \left(\frac{t}{1+t} * (|x|) + |y|) \right) \rceil$$

- Prefix filter approximates this test
  - reorder bit positions for all fingerprints according to their overall frequency from infrequent to frequent
  - exclude pairs of records without any overlap in their prefixes with

$$\text{prefix\_length}(x) = \lceil \left((1-t) * |x| \right) + 1 \rceil$$

- Example (t = 0.8)

ID	reordered fingerprint	card.	prefix fingerprint
B	1 0 1 0 0 0 0 0 1 1 0 0 0 0	4	1 0 1
C	0 0 0 1 1 1 1 1 1 1 0 0 0 0	7	0 0 0 1 1 1
A	0 1 0 1 1 1 1 1 1 1 0 0 0 0	8	0 1 0 1 1

AND operation on prefixes shows non-zero result for C and A so that these records still need to be considered for matching

ScaDS DRESDEN LEIPZIG

P4JOIN: POSITION FILTER

- improvement of prefix filter to avoid matches even for overlapping prefixes
  - estimate maximally possible overlap and checking whether it is below the *minimal overlap α* to meet threshold t
  - *original position filter* considers the position of the last common prefix token
- revised position filter
  - record x, prefix 1 1 0 1 length 9
  - record y, prefix 1 1 1 length 8
  - highest prefix position (here fourth pos. in x) limits possible overlap with other record: the third position in y prefix cannot have an overlap with x
  - maximal possible overlap = #shared prefix tokens (2) + min (9-3, 8-3)= 7
    < minimal overlap α = 8

ScaDS DRESDEN LEIPZIG

EVALUATION

- comparison between NestedLoop, P4Join, MultiBitTree
  - MultiBitTree: best filter approach in previous work by Schnell
    - applies length filter and organizes fingerprints within a binary tree so that fingerprints with the same set bits are grouped within sub-trees
    - can be used to filter out many fingerprints from comparison
- two input datasets R, S
  - determined with FEBRL data generator
    N=[100.000, 200.000, ..., 500.000]. |R|=1/5·N, |S|=4/5·N
  - bit vector length: 1000
  - similarity threshold 0.8

ScaDS DRESDEN LEIPZIG

EVALUATION RESULTS

- runtime in minutes on standard PC

Approach	Dataset size N				
	100 000	200 000	300 000	400 000	500 000
NestedLoop	6.1	27.7	66.1	122.0	194.8
MultiBitTree	4.7	19.0	40.6	78.2	119.7
P4Join	2.3	15.5	40.1	77.8	125.5

- similar results for P4Join and Multibit Tree
- relatively small improvements compared to NestedLoop

ScaDS DRESDEN LEIPZIG

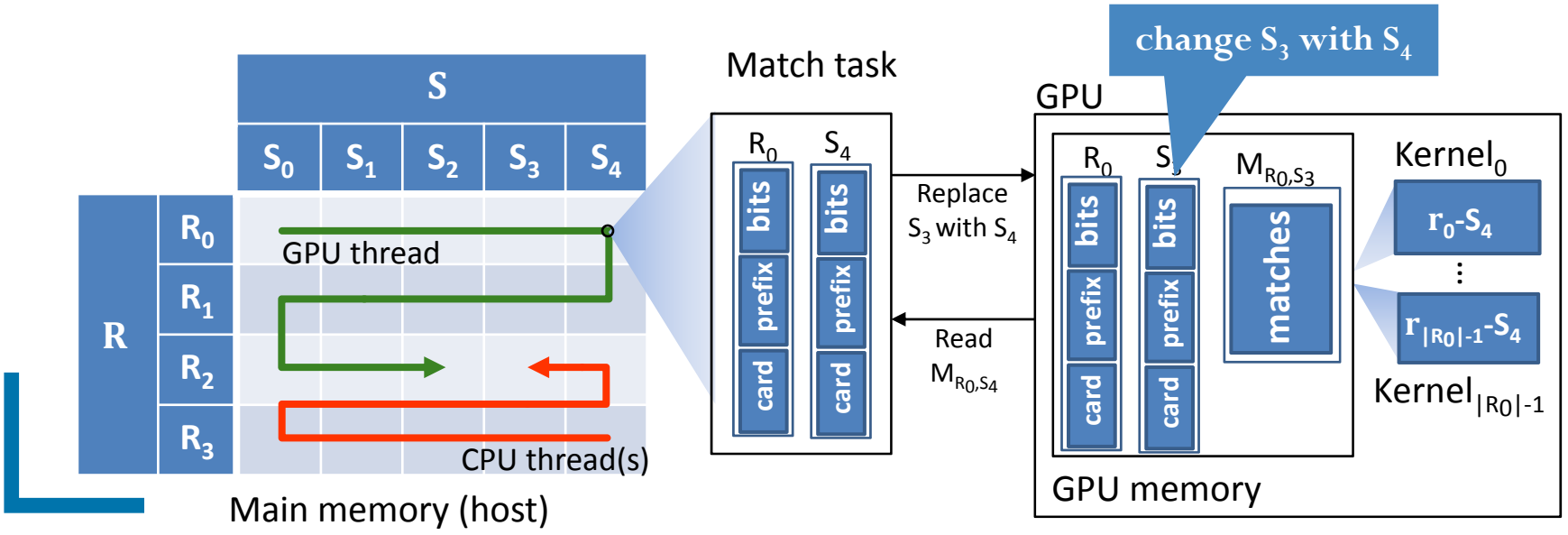
GPU-BASED PPRL

- Operations on bit vectors easy to compute on GPUs
  - Length and prefix filters
  - Jaccard similarity
- Frameworks CUDA und **OpenCL** support data-parallel execution of general computations on GPUs
  - program („kernel“) written in C dialect
  - limited to base data types (float, long, int, short, arrays)
  - no dynamic memory allocation (programmer controls memory management)
  - important to minimize data transfer between main memory and GPU memory

ScaDS DRESDEN LEIPZIG

EXECUTION SCHEME

- partition inputs R and S (fingerprints sorted by length) into equally-sized partitions that fit into GPU memory
  - generate match tasks per pair of partition
  - only transfer to GPU if length intervals per partition meet length filter
  - optional use of CPU thread to additionally match on CPU

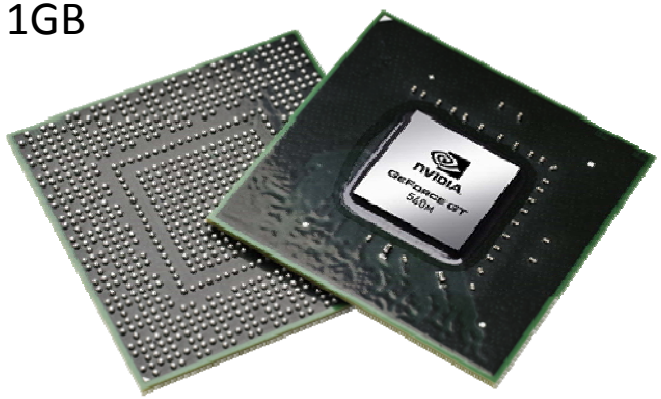


ScaDS DRESDEN LEIPZIG

GPU-BASED EVALUATION RESULTS

GeForce GT 610
- 48 Cuda Cores@810MHz
- 1GB
- 35€

GeForce GT 540M
- 96 Cuda Cores@672MHz
- 1GB

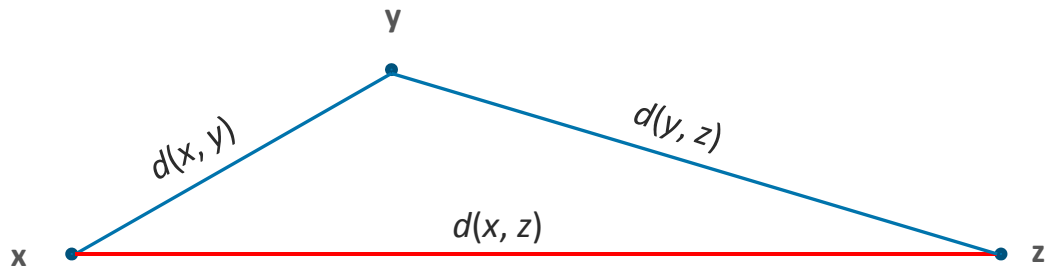


	100 000	200 000	300 000	400 000	500 000
GForce GT 610	0.33	1.32	2.95	5.23	8.15
GeForce GT 540M	0.28	1.08	2.41	4.28	6.67

- improvements by up to a factor of 20, despite low-profile graphic cards
- still non-linear increase in execution time with growing data volume

METRIC SPACE-BASED DISTANCE FUNCTIONS

- Characteristics of distance functions d for metric spaces:
  - $d(x, x) = 0$ reflexivity
  - $d(x, y) \geq 0$ positiveness
  - $d(x, y) = d(y, x)$ symmetry
  - $d(x, y) + d(y, z) \geq d(x, z)$ triangular inequality

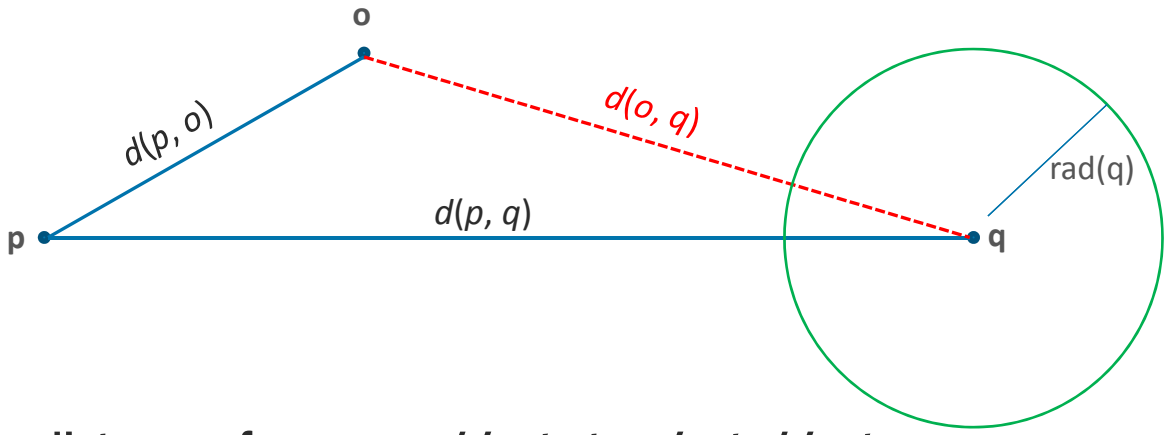


- Sample metric-space distance functions: Euclidean distance, edit distance, Hamming distance, Jaccard coefficient
  - distance = 1 – similarity

ScaDS DRESDEN LEIPZIG

REDUCTION OF SEARCH SPACE

- utilize triangle equality to avoid similarity computations



- consider distances for *query object q* to *pivot object p*
  - check whether we need to determine distance **d(o,q)** to *object o* for given similarity threshold (maximal distance max-dist=***rad(q)***)
- From triangle inequality we know:
  - $d(p,o) + d(o,q) \geq d(p,q) \Rightarrow d(p,q) - d(p,o) <= d(o,q)$
- we can safely exclude object o if $d(p,q) - d(p,o) > \text{max-dist} = \text{rad}(q)$

ScaDS DRESDEN LEIPZIG

LINKAGE APPROACH

- **Preprocessing:**
  - Given: a set of objects (fingerprints) S1 to be matched
  - choose a subset $P=\{p_1,\ldots, p_j\}$ of objects to be the pivots
    - different possibilities, e.g., random subset
  - assign each object o_i to the nearest pivot p_j and save the distance $d(p_j,o_i)$, determine radius(p) as maximal distance from p to its assigned objects
- **Matching:**
  - Given: set of query objects (fingerprints) S2, threshold *max-dist*
  - for each q of S2 and each pivot p do:
    - determine distance d(p,q)
    - If d(p,q) <= rad(p)+max-dist:
      for each object assigned to p
      - check whether it can be skipped due to triangular inequality (*d(p,q)-d(p,o)> max-dist*)
      - otherwise match o with p (o matches q if *d(o,q) <= max-dist*)

ScaDS DRESDEN LEIPZIG

EVALUATION

- Comparison with previous approaches using the same datasets
- Runtime in minutes (using faster PC than in previous evaluation)

Algorithms	Datasets				
	100 000	200 000	300 000	400 000	500 000
NestedLoop	3.8	20.8	52.1	96.8	152.6
MultiBitTree	2.6	11.3	26.5	50.0	75.9
P4Join	1.4	7.4	24.1	52.3	87.9
Pivots (metric space)	0.2	0.4	0.9	1.3	1.7

- Pivot-based approach shows the best results and is up to 40X faster than other algorithms

ScaDS DRESDEN LEIPZIG

SUMMARY

- Privacy for Big Data
  - privacy-preserving publishing / record linkage / data mining
  - tradeoff between protection of personal/sensitive data and data utility for analysis
  - complete anonymization prevents record linkage -> 1-way pseudonymization of sensitive attributes good compromise
- Scalable Privacy-Preserving Record Linkage
  - bloom filters allow simple, effective and relatively efficient match approach
  - performance improvements by blocking / filtering / parallel PPRL
  - effective filtering by P4Join and utilizing metric-space distance functions
  - GPU usage achieves significant speedup

ScaDS DRESDEN LEIPZIG

OUTLOOK / CHALLENGES

- High PPRL match quality for real, dirty data
- Quantitative evaluation of privacy characteristics
- Efficient SCM approaches for multiple sources without linkage unit
- Combined study of PPRL + data mining

ScaDS DRESDEN LEIPZIG

REFERENCES

- E.A. Durham, M. Kantarcioglu, Y. Xue, C. Tóth, M.Kuzu, B.Malin: *Composite Bloom Filters for Secure Record Linkage.* IEEE Trans. Knowl. Data Eng. 26(12): 2956-2968 (2014)
- B. Fung, K. Wang, R.Chen, P.S Yu: *Privacy-preserving data publishing: A survey of recent developments*. ACM CSUR 2010
- R. Hall, S. E. Fienberg: *Privacy-Preserving Record Linkage.* Privacy in Statistical Databases 2010: 269-283
- D.Karapiperis, V. S. Verykios: *A distributed near-optimal LSH-based framework for privacy-preserving record linkage.* Comput. Sci. Inf. Syst. 11(2): 745-763 (2014)
- C.W. Kelman, A.J. Bass. C.D. Holman: *Research use of linked health data--a best practice protocol*. Aust NZ J Public Health. 2002
- R.Schnell, T. Bachteler, J. Reiher: *Privacy-preserving record linkage using Bloom filters*. BMC Med. Inf. & Decision Making 9: 41 (2009)
- Schnell, R.: *Privacy-preserving Record Linkage and Privacy-preserving Blocking for Large Files with Cryptographic Keys Using Multibit Trees*. Proc. Joint Statistical Meetings, American Statistical Association, 2013
- Z. Sehili, L. Kolb, C. Borgs, R. Schnell, E. Rahm: *Privacy Preserving Record Linkage with PPJoin*. Proc. BTW Conf. 2015
- Z. Sehili, E. Rahm: Speeding Up Privacy Preserving Record Linkage for Metric Space Similarity Measures. TR, Univ. Leipzig, 2016
- J. Vaidya, C. Clifton: *Privacy-Preserving data mining: Why, How, and When*. IEEE Security&Privacy, 2004
- D. Vatasalan, P- Christen, V.S. Verykios: *An evaluation framework for privacy-preserving record linkage*. Journal of Privacy and Confidentialty 2014
- D. Vatasalan, P- Christen, C.M. O'Keefe, V.S. Verykios: *A taxonomy of privacy-preserving record linkage techniques*. Information Systems 2013
- C. Xiao, W. Wang, X. Lin, J.X. Yu: *Efficient Similarity Joins for Near Duplicate Detection*. Proc. WWW 2008