Privacy in Data Publication and Outsourcing Scenarios

Pierangela Samarati
Dipartimento di Informatica
Università degli Studi di Milano
pierangela.samarati@unimi.it

12th International School on Foundations of Security Analysis and Design
(FOSAD 2012)
Bertinoro, Italy - September 7-8, 2012

Motivation (1)

- Continuous growth of:
  - government and company databases
  - user-generated content delivered through collaborative Internet services such as YouTube, Facebook
  - personally identifiable information collected whenever a user creates an account, submits an application, signs up for newsletters, participates in a survey, ...

Motivation (2)

- Data sharing and dissemination, for e.g.:
  - study trends or to make useful statistical inference
  - share knowledge
  - access on-line services
- External data storage and computation:
  - cost saving and service benefits
  - higher availability and more effective disaster protection

$\Longrightarrow$ Need to ensure data privacy and integrity are properly protected

Outline

- Privacy in data publication

  $\Longrightarrow$ data release/dissemination

- Privacy in data outsourcing/cloud computing

  $\Longrightarrow$ third parties store and manage data

Privacy in Data Publication

V. Ciriani, S. De Capitani di Vimercati, S. Foresti, P. Samarati, “k-Anonymity,” in *Secure Data Management in Decentralized Systems*, T. Yu, and S. Jajodia (eds.), Springer, 2007

V. Ciriani, S. De Capitani di Vimercati, S. Foresti, P. Samarati, “Microdata Protection,” in *Secure Data Management in Decentralized Systems*, T. Yu, and S. Jajodia (eds.), Springer, 2007

Statistical DBMS vs statistical data

Release of data for statistical purpose

- statistical DBMS [AW-89]
  - the DBMS responds only to statistical queries
  - need run time checking to control information (indirectly) released
- statistical data [CDFS-07b]
  - publish statistics
  - control on indirect release performed before publication

Macrodata vs microdata

- In the past data were mainly released in tabular form (macrodata) and through statistical databases

- Today many situations require that the specific stored data themselves, called microdata, be released
  - increased flexibility and availability of information for the users

- Microdata are subject to a greater risk of privacy breaches (linking attacks)

Disclosure protection techniques for macrodata

The protection techniques include:

- sampling: data confidentiality is protected by conducting a sample survey rather than a census
- special rules: designed for specific tables, they impose restrictions on the level of detail that can be provided in a table
- threshold rule: rules that protect sensitive cells, for instance:
  - cell suppression
  - random rounding
  - controlled rounding
  - confidentiality edit

Disclosure protection techniques for microdata

The classical protection techniques (often applied to protect microdata before computing statistics) can be classified as follows:

- masking techniques: transform the original set of data by not releasing or perturbing their values
  - non-perturbative: the original data are not modified, but some data are suppressed and/or some details are removed (e.g., sampling, local suppression, generalization)
  - perturbative: the original data are modified (e.g., rounding, swapping)
- synthetic data generation techniques: release plausible but synthetic values instead of the real ones
  - fully synthetic: the released dataset contains synthetic data only
  - partially synthetic: the released dataset contains a mix of original and synthetic data

Restricted data and restricted access

- Some microdata include explicit identifiers (e.g., name, address, or Social Security number)
- Removing such identifiers is a first step in preparing for the release of microdata for which the confidentiality of individual information must be protected
- De-identification is not sufficient
- De-identification does not imply anonymity

  $\Longrightarrow$ de-identified data can be linked with other sources to re-identify individuals

The anonymity problem – Example

SSN	Name	Race	Date of birth	Sex	ZIP	Marital status	Disease
		asian	64/04/12	F	94142	divorced	hypertension
		asian	64/09/13	F	94141	divorced	obesity
		asian	64/04/15	F	94139	married	chest pain
		asian	63/03/13	M	94139	married	obesity
		asian	63/03/18	M	94139	married	short breath
		black	64/09/27	F	94138	single	short breath
		black	64/09/27	F	94139	single	obesity
		white	64/09/27	F	94139	single	chest pain
		white	64/09/27	F	94141	widow	short breath

Name	Address	City	ZIP	DOB	Sex	Status
....	....	....	....	....	....	....
....	....	....	....	....	....	....
Sue J. Doe	900 Market St.	San Francisco	94142	64/04/12	F	divorced
....	....	....	....	....	....	....

Classification of attributes in a microdata table

The attributes in the original microdata table can be classified as:

- identifiers. Attributes that uniquely identify a microdata respondent (e.g., SSN uniquely identifies the person with which is associated)
- quasi-identifiers. Attributes that, in combination, can be linked with external information to re-identify all or some of the respondents to whom information refers or reduce the uncertainty over their identities (e.g., DoB, ZIP, and Sex)
- confidential. Attributes of the microdata table that contain sensitive information (e.g., Disease)
- non confidential. Attributes that the respondents do not consider sensitive and whose release do not cause disclosure

Re-identification

A study of the 2000 census data [G-06] reported that the US population was uniquely identifiable by:

- year of birth, 5-digit ZIP code: 0,2%
- year of birth, county: 0,0%
- year and month of birth, 5-digit ZIP code: 4,2%
- year and month of birth, county: 0,2%
- year, month, and day of birth, 5-digit ZIP code: 63,3%
- year, month, and day of birth, county: 14,8%

Factors contributing to disclosure risk (1)

Possible sources of the disclosure risk of microdata

- Existence of high visibility records. Some records on the file may represent respondents with unique characteristics such as very unusual jobs (e.g., movie star) or very large incomes

- Possibility of matching the microdata with external information. There may be individuals in the population who possess a unique or peculiar combination of the characteristic variables on the microdata
  - if some of those individuals happen to be chosen in the sample of the population, there is a disclosure risk
  - note that the identity of the individuals that have been chosen should be kept secret

Factors contributing to disclosure risk (2)

The possibility of linking or its precision increases with:

- the existence of a high number of common attributes between the microdata table and the external sources
- the accuracy or resolution of the data
- the number of outside sources, not all of which may be known to the agency releasing the microdata

Factors contributing to decrease the disclosure risk (1)

- A microdata table often contains a subset of the whole population
  - this implies that the information of a specific respondent, which a malicious user may want to know, may not be included in the microdata table

- The information specified in microdata tables released to the public is not always up-to-date (often at least one or two-year old)
  - the values of the attributes of the corresponding respondents may have been changed in the meanwhile
  - the age of the external sources of information used for linking may be different from the age of the information contained in the microdata table

Factors contributing to decrease the disclosure risk (2)

- A microdata table and the external sources of information naturally contain noise that decreases the ability to link the information

- A microdata table and the external sources of information can contain data expressed in different forms thus decreasing the ability to link information

Measures of risk

The disclosure risk depends on:

- the probability that the respondent for whom an intruder is looking for is represented on both the microdata and some external source
- the probability that the matching variables are recorded in a linkable way on the microdata and on the external source
- the probability that the respondent for whom the intruder is looking for is unique (or peculiar) in the population of the external source

The percentage of records representing respondents who are unique in the population (population unique) plays a major role in the disclosure risk of microdata (with respect to the specific respondent)

Note that each population unique is a sample unique; the vice-versa is not true

k-anonymity [S-01] (1)

- k-anonymity, together with its enforcement via generalization and suppression, has been proposed as an approach to protect respondents' identities while releasing truthful information

- k-anonymity tries to capture the following requirement:
  - the released data should be indistinguishably related to no less than a certain number of respondents

- Quasi-identifier: set of attributes that can be exploited for linking (whose release must be controlled)

k-anonymity (2)

- Basic idea: translate the k-anonymity requirement on the released data
  - each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents
- In the released table the respondents must be indistinguishable (within a given set) with respect to a set of attributes
- k-anonymity requires that each quasi-identifier value appearing in the released table must have at least k occurrences
  - sufficient condition for the satisfaction of k-anonymity requirement

Generalization and suppression

- Generalization. The values of a given attribute are substituted by using more general values. Based on the definition of a generalization hierarchy
  - Example: consider attribute ZIP code and suppose that a step in the corresponding generalization hierarchy consists in suppressing the least significant digit in the ZIP code
    With one generalization step: 20222 and 20223 become 2022*; 20238 and 20239 become 2023*
- Suppression. Protect sensitive information by removing some values
  - the introduction of suppression can reduce the amount of generalization necessary to satisfy the k-anonymity constraint

Generalization hierarchy – Example

$$
\begin{tikzcd}
R_1 = \{\texttt{person}\} \\
R_0 = \{\texttt{asian,black,white}\} \arrow[u]
\end{tikzcd}
\qquad
\begin{tikzcd}
Z_2 = \{\texttt{941**}\} \\
Z_1 = \{\texttt{9413*,9414*}\} \arrow[u] \\
Z_0 = \{\texttt{94138,94139,94141,94142}\} \arrow[u]
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & \texttt{person} & \\
\texttt{asian} \arrow[ur] & \texttt{black} \arrow[u] & \texttt{white} \arrow[ul]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & & \texttt{941**} & & & \\
 & \texttt{9413*} \arrow[urr] & & & & \texttt{9414*} \arrow[ull] & \\
\texttt{94138} \arrow[ur] & & \texttt{94139} \arrow[ul] & & \texttt{94141} \arrow[ur] & & \texttt{94142} \arrow[ul]
\end{tikzcd}
$$

Generalized table with suppression – Example

Race	ZIP
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Race	ZIP
person	94141
person	94139
person	94139
person	94139
person	94139
person	94139
person	94141

GT

k-minimal table

- The solutions proposed for computing a k-anonymous table aim at finding a k-minimal table
- A k-minimal table does not generalize (or suppress) more than it is needed to reach the threshold k
- Different minimal generalizations may exist, preference criteria can be applied to determine which one to release
  - minimum absolute distance prefers the generalization(s) with the smallest total number of generalization steps
  - minimum relative distance prefers the generalization(s) with the smallest total number of relative generalization steps (a step is made relative by dividing it over the height of the domain hierarchy to which it refers)
  - maximum distribution prefers the generalization(s) with the greatest number of distinct tuples
  - minimum suppression prefers the generalization(s) that suppresses less tuples, that is, the one with the greatest cardinality

Examples of 2-minimal generalizations

Threshold of acceptable suppression=2

Race:R_0	**ZIP:**Z_0
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Race:R_1	**ZIP:**Z_0
person	94141
person	94139
person	94139
person	94139
person	94139
person	94139
person	94141

GT_1

Race:R_0	**ZIP:**Z_1
asian	9414*
asian	9414*
asian	9413*
asian	9413*
asian	9413*
black	9413*
black	9413*

GT_2

Classification of k-anonymity techniques (1)

Generalization and suppression can be applied at different levels of granularity

- Generalization can be applied at the level of single column (i.e., a generalization step generalizes all the values in the column) or single cell (i.e., for a specific column, the table may contain values at different generalization levels)

- Suppression can be applied at the level of row (i.e., a suppression operation removes a whole tuple), column (i.e., a suppression operation obscures all the values of a column), or single cells (i.e., a k-anonymized table may wipe out only certain cells of a given tuple/attribute)

Classification of k-anonymity techniques (2)

Generalization	**Suppression** *Tuple*	*Attribute*	*Cell*	*None*
Attribute	**AG_TS**	**AG_AS** $\equiv$ AG_	**AG_CS**	**AG_** $\equiv$ AG_AS
Cell	**CG_TS** not applicable	**CG_AS** not applicable	**CG_CS** $\equiv$ CG_	**CG_** $\equiv$ CG_CS
None	**_TS**	**_AS**	**_CS**	**_** not interesting

2-anonymized tables wrt different models (1)

Race	DOB	Sex	ZIP
asian	64/04/12	F	94142
asian	64/09/13	F	94141
asian	64/04/15	F	94139
asian	63/03/13	M	94139
asian	63/03/18	M	94139
black	64/09/27	F	94138
black	64/09/27	F	94139
white	64/09/27	F	94139
white	64/09/27	F	94141

PT

Race	DOB	Sex	ZIP
asian	64/04	F	941**
asian	64/04	F	941**
asian	63/03	M	941**
asian	63/03	M	941**
black	64/09	F	941**
black	64/09	F	941**
white	64/09	F	941**
white	64/09	F	941**

AG_TS

2-anonymized tables wrt different models (2)

Race	DOB	Sex	ZIP
asian	*	F	*
asian	*	F	*
asian	*	F	*
asian	63/03	M	9413*
asian	63/03	M	9413*
black	64/09	F	9413*
black	64/09	F	9413*
white	64/09	F	*
white	64/09	F	*

AG_CS

Race	DOB	Sex	ZIP
asian	64	F	941**
asian	64	F	941**
asian	64	F	941**
asian	63	M	941**
asian	63	M	941**
black	64	F	941**
black	64	F	941**
white	64	F	941**
white	64	F	941**

AG_ ≡AG_AS

2-anonymized tables wrt different models (3)

Race	DOB	Sex	ZIP
asian	64	F	941**
asian	64	F	941**
asian	64	F	941**
asian	63/03	M	94139
asian	63/03	M	94139
black	64/09/27	F	9413*
black	64/09/27	F	9413*
white	64/09/27	F	941**
white	64/09/27	F	941**

CG_ ≡ CG_CS

Race	DOB	Sex	ZIP

_TS

2-anonymized tables wrt different models (4)

Race	DOB	Sex	ZIP
asian	*	F	*
asian	*	F	*
asian	*	F	*
asian	*	M	*
asian	*	M	*
black	*	F	*
black	*	F	*
white	*	F	*
white	*	F	*

_AS

Race	DOB	Sex	ZIP
asian	*	F	*
asian	*	F	*
asian	*	F	*
asian	*	M	94139
asian	*	M	94139
*	64/09/27	F	*
*	64/09/27	F	94139
*	64/09/27	F	94139
*	64/09/27	F	*

_CS

Algorithms for computing a k-anonymous table (1)

- The problem of finding minimal k-anonymous tables is computationally hard (even in the case **AG_TS**)

- Many algorithms have been proposed:

  - exact (for **AG_TS**): computational time exponential in the number of the attributes composing the quasi-identifier

  - heuristic: based on genetic algorithms, simulated annealing, top-down heuristic; no bounds on efficiency and goodness, which are assessed via experimental results

  - approximation: for general and specific values of k (e.g., 1.5-approximation for 2-anonymity, and 2-approximation for 3-anonymity); also for **_CS** and **CG_**

Algorithms for computing a k-anonymous table (2)

Generalization-based algorithms can be partitioned into two classes depending on how the generalization is performed

- Hierarchy-based generalization based on the definition of a generalization hierarchy (pre-defined) for each attribute in QI (e.g., [S-01])
  - the most general value is at the root of the hierarchy
  - the leaves correspond to the values in the ground domain
- Recoding-based generalization based on the recoding into intervals protection method (e.g., [BA-05])
  - the ground domain of each attribute in QI is partitioned into possibly disjoint intervals (computed at run time) that are associated with a label
  - each value in the ground domain is mapped to the intervals they belong to

Mondrian [LDR-06] – Example (1)

Private table

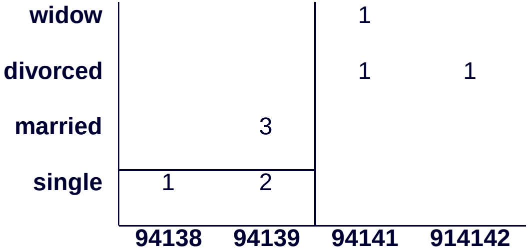
Marital status	**ZIP**
divorced	94142
divorced	94141
married	94139
married	94139
married	94139
single	94138
single	94139
single	94139
widow	94141

widow			1	
divorced			1	1
married		3		
single	1	2		
	94138	**94139**	**94141**	**914142**

Mondrian [LDR-06] – Example (2)

3-anonymous table

Marital status	ZIP
divorced or widow	9414*
divorced or widow	9414*
married	94139
married	94139
married	94139
single	9413*
single	9413*
single	9413*
divorced or widow	9414*



Minimal k-anonymization for cell generalization [GMT-08]

- k-anonymity requirement: Each release of data must be such that every combination of values of quasi-identifiers can be indistinctly matched to at least k respondents
- When generalization is performed at attribute level (**AG**) this is equivalent to require each quasi-identifier n-uple to have at least k occurrences
- When generalization is performed at cell level (**CG**) the existence of at least k occurrences is a sufficient but not necessary condition; a less stricter requirement would suffice
  1. For each sequence of values pt in PT[q] there are at least k tuples in $T[q]$ that contain a sequence of values generalizing pt
  2. For each sequence of values t in $T[q]$ there are at least k tuples in PT[q] that contain a sequence of values for which t is a generalization

Minimal k-anonymization for CG – Example

Race	ZIP
white	94138
black	94139
asian	94141
asian	94141
asian	94142

PT

Race	ZIP
person	9413*
person	9413*
asian	9414*
asian	9414*
asian	9414*

2-anonymity

Race	ZIP
person	9413*
person	9413*
asian	94141
asian	9414*
asian	9414*

2-anonymity (revisited)

Race	ZIP
person	9413*
person	9413*
asian	9414*
asian	9414*
asian	94142

Race	ZIP
person	9413*
person	9413*
asian	94141
asian	94141
asian	9414*

no 2-anonymity

Attribute Disclosure

2-anonymous table according to the **AG_** model

k-anonymity protects only identities not the association between generalized quasi-identifiers and sensitive information; it is then vulnerable to some attacks [MGK-06,S-01]

Race	DOB	Sex	ZIP	Disease
asian	64	F	941**	hypertension
asian	64	F	941**	obesity
asian	64	F	941**	chest pain
asian	63	M	941**	obesity
asian	63	M	941**	obesity
black	64	F	941**	short breath
black	64	F	941**	short breath
white	64	F	941**	chest pain
white	64	F	941**	short breath

Homogeneity of the sensitive attribute values

- All tuples with a quasi-identifier value in a k-anonymous table may have the same sensitive attribute value
  - an adversary knows that Carol is a black female and that her data are in the microdata table
  - the adversary can infer that Carol suffers from short breath

Race	DOB	Sex	ZIP	Disease
...	...	...	...	...
black	64	F	941**	short breath
black	64	F	941**	short breath
...	...	...	...	...

Background knowledge

- Based on prior knowledge of some additional external information
  - an adversary knows that Hellen is a white female and she is in the microdata table
  - the adversary can infer that the disease of Hellen is either chest pain or short breath
  - the adversary knows that the Hellen runs 2 hours a day and therefore that Hellen cannot suffer from short breath
    $\Longrightarrow$ the adversary infers that Hellen's disease is chest pain

Race	DOB	Sex	ZIP	Disease
...	...	...	...	...
white	64	F	941**	chest pain
white	64	F	941**	short breath

ℓ-diversity (1)

- A q-block (i.e., set of tuples with the same value for QI) in T is ℓ-diverse if it contains at least ℓ different well-represented values for the sensitive attribute in T
  - well-represented: different definitions based on entropy or recursion (e.g., a q-block is ℓ-diverse if removing a sensitive value it remains (ℓ-1)-diverse)
- ℓ-diversity: an adversary needs to eliminate at least ℓ-1 possible values to infer that a respondent has a given value

ℓ-diversity (2)

- T is ℓ-diverse if all its q-blocks are ℓ-*diverse*

  $\Longrightarrow$ the homogeneity attack is not possible anymore

  $\Longrightarrow$ the background knowledge attack becomes more difficult

- ℓ-diversity is monotonic with respect to the generalization hierarchies considered for k-anonymity purposes

- Any algorithm for k-anonymity can be extended to enforce the ℓ-diverse property

Skewness attack

ℓ-diversity leaves space to attacks based on the distribution of values inside q-blocks

- Skewness attack occurs when the distribution in a q-block is different from the distribution in the original population

- 20% of the population suffers from diabetes; 75% of tuples in a q-block have diabetes
  $\Longrightarrow$ people in the q-block have higher probability of suffering from diabetes

Race	**DOB**	**Sex**	**ZIP**	**Disease**
black	64	F	941**	diabetes
black	64	F	941**	short breath
black	64	F	941**	diabetes
black	64	F	941**	diabetes

Similarity attack

- Similarity attack happens when a q-block has different but semantically similar values for the sensitive attribute

Race	DOB	Sex	ZIP	Disease
black	64	F	941**	stomach ulcer
black	64	F	941**	stomach ulcer
black	64	F	941**	gastritis

Group closeness [LLV-07]

- A q-block respects t-closeness if the distance between the distribution of the values of the sensitive attribute in the q-block and in the considered population is lower than t
- T respects t-closeness if all its q-blocks respect t-closeness
- t-closeness is monotonic with respect to the generalization hierarchies considered for k-anonymity purposes
- Any algorithm for k-anonymity can be extended to enforce the t-closeness property, which however might be difficult to achieve

External knowledge [CLR-07,MKMGH-07] (1)

- The consideration of the adversary's background knowledge (or external knowledge) is necessary when reasoning about privacy in data publishing
- External knowledge can be exploited for inferring sensitive information about individuals with high confidence
- Positive inference
  - a respondent has a given value (or a value within a restricted set)
- Negative inference
  - a respondent does not have a given value
- Existing approaches have mostly focused on positive inference

External knowledge (2)

- External knowledge may include:
  - similar datasets released by different organizations
  - instance-level information
  - ...
- Not possible to know a-priori what external knowledge the adversary possesses
- It is necessary to provide the data owner with a means to specify adversarial knowledge

External knowledge modeling [CLR-07]

- An adversary has knowledge about an individual (target) represented in a released table and knows the individual's QI values
  $\Longrightarrow$ goal: predict whether the target has a target sensitive value
- External knowledge modeled through a logical expression
- Three basic classes of expressions, representing knowledge about:
  - the target individual: information that the adversary may know about the target individual
  - others: information about individuals other than the target
  - same-value families: knowledge that a group (or family) of individuals have the same sensitive value
- Other types of external knowledge may be identified......

External knowledge – Example (1)

Name	DOB	Sex	ZIP	Disease
Alice	74/04/12	F	94142	aids
Bob	74/04/13	M	94141	flu
Carol	74/09/15	F	94139	flu
David	74/03/13	M	94139	aids
Elen	64/03/18	F	94139	flu
Frank	64/09/27	M	94138	short breath
George	64/09/27	M	94139	flu
Harry	64/09/27	M	94139	aids

Original table

$\Longrightarrow$

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

Released table is 4-anonymized but ......

External knowledge – Example (2)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

An adversary knows that Harry, born in 64 and living in area 94139, is in the table

External knowledge – Example (2)

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

$\Longrightarrow$

DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

An adversary knows that Harry, born in 64 and living in area 94139, is in the table

$\Longrightarrow$ Harry belongs to the second group

$\Longrightarrow$ Harry has aids with confidence 1/4

External knowledge – Example (3)

DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From another dataset, the adversary knows that George (who is in the table, is born in 64, and leaves in area 941**) has flu

External knowledge – Example (3)

DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

$\Longrightarrow$

DOB	Sex	ZIP	Disease
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From another dataset, the adversary knows that George (who is in the table, is born in 64, and leaves in area 941**) has flu

$\Longrightarrow$ Harry has aids with confidence 1/3

External knowledge – Example (4)

DOB	Sex	ZIP	Disease
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From personal knowledge, the adversary knows that Harry does not have short breath

External knowledge – Example (4)

DOB	Sex	ZIP	Disease
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table

$\Longrightarrow$

DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	aids

4-anonymized table

From personal knowledge, the adversary knows that Harry does not have short breath

$\Longrightarrow$ Harry has aids with confidence 1/2

Multiple independent releases

- Data may be subject to frequent changes and may need to be published on regular basis

- The multiple release of a microdata table may cause information leakage since a malicious recipient can correlate the released datasets

Multiple independent releases – Example (1)

T_1

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table at time t_1

T_2

DOB	Sex	ZIP	Disease
[70-80]	F	9414*	hypertension
[70-80]	F	9414*	gastritis
[70-80]	F	9414*	aids
[70-80]	F	9414*	gastritis
[60-70]	M	9413*	flu
[60-70]	M	9413*	aids
[60-70]	M	9413*	flu
[60-70]	M	9413*	gastritis

4-anonymized table at time t_2

An adversary knows that Alice, born in 1974 and living in area 94142, is in both releases

Multiple independent releases – Example (1)

T_1

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids

4-anonymized table at time t_1

T_2

DOB	Sex	ZIP	Disease
[70-80]	F	9414*	hypertension
[70-80]	F	9414*	gastritis
[70-80]	F	9414*	aids
[70-80]	F	9414*	gastritis

4-anonymized table at time t_2

An adversary knows that Alice, born in 1974 and living in area 94142, is in both releases

$\Longrightarrow$ Alice belongs to the first group in T_1

$\Longrightarrow$ Alice belongs to the second group in T_2

Multiple independent releases – Example (1)

T_1

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids

4-anonymized table at time t_1

T_2

DOB	Sex	ZIP	Disease
[70-80]	F	9414*	hypertension
[70-80]	F	9414*	gastritis
[70-80]	F	9414*	aids
[70-80]	F	9414*	gastritis

4-anonymized table at time t_2

An adversary knows that Alice, born in 1974 and living in area 94142, is in both releases

$\Longrightarrow$ Alice belongs to the first group in T_1

$\Longrightarrow$ Alice belongs to the first group in T_2

Alice suffers from aids (it is the only illness common to both groups)

Multiple independent releases – Example (2)

T_1

DOB	Sex	ZIP	Disease
74	*	941**	aids
74	*	941**	flu
74	*	941**	flu
74	*	941**	aids
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table at time t_1

T_2

DOB	Sex	ZIP	Disease
[70-80]	F	9414*	hypertension
[70-80]	F	9414*	gastritis
[70-80]	F	9414*	aids
[70-80]	F	9414*	gastritis
[60-70]	M	9413*	flu
[60-70]	M	9413*	aids
[60-70]	M	9413*	flu
[60-70]	M	9413*	gastritis

4-anonymized table at time t_2

An adversary knows that Frank, born in 1964 and living in area 94132, is in T_1 but not in T_2

Multiple independent releases – Example (2)

T_1

DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table at time t_1

T_2

DOB	Sex	ZIP	Disease
[60-70]	M	9413*	flu
[60-70]	M	9413*	aids
[60-70]	M	9413*	flu
[60-70]	M	9413*	gastritis

4-anonymized table at time t_2

An adversary knows that Frank, born in 1964 and living in area 94132, is in T_1 but not in T_2

Multiple independent releases – Example (2)

T_1

DOB	Sex	ZIP	Disease
64	*	941**	flu
64	*	941**	short breath
64	*	941**	flu
64	*	941**	aids

4-anonymized table at time t_1

T_2

DOB	Sex	ZIP	Disease
[60-70]	M	9413*	flu
[60-70]	M	9413*	aids
[60-70]	M	9413*	flu
[60-70]	M	9413*	gastritis

4-anonymized table at time t_2

An adversary knows that Frank, born in 1964 and living in area 94132, is in T_1 but not in T_2

$\Longrightarrow$ Frank suffers from short breath
(it is the only illness that appears in T_1 and does not appear in T_2)

m-invariance [XT-07]

A sequence $T_1, \ldots, T_n$ of released microdata tables satisfies m-invariance iff

- each equivalence class includes at least m tuples
- no sensitive value appears more than once in each equivalence class
- for each tuple t, the equivalence classes to which t belongs in the sequence are characterized by the same set of sensitive values

$\Longrightarrow$ the correlation of the tuples in $T_1, \ldots, T_n$ does not permit a malicious recipient to associate less than m different sensitive values with each respondent

Extended scenarios (1)

k-anonymity, ℓ-diversity, and t-closeness are based on assumptions that make them not always applicable in specific scenarios

- Multiple tuples per respondent
  - (X,Y)-privacy [WF-06]
  - k^m-anonymity [TMK-08]
- Release of multiple tables, characterized by (functional) dependencies
  - (X,Y)-privacy [WF-06]
  - MultiR k-anonymity [NCN-07]
- Multiple quasi-identifiers
  - butterfly [PTLX-09]

Extended scenarios (2)

- Non-predefined quasi-identifiers
  - k^m-anonymity [TMK-08]
- Release of data streams
  - anonymize temporal data [WXWF-10]
  - k-anonymous data streams [ZHPJTJ-09]
- Fine-grained privacy preferences
  - (α_i, β_i)-closeness [FZ-08]
  - personalized anonymity [XT-06]
  - δ-presence [NAC-07]

k-anonymity in various applications

In addition to classical microdata release problem, the concept of k-anonymity and its extensions can be applied in different scenarios, e.g.:

- social networks (e.g.,[HMJTW-08])
- data mining (e.g.,[FWY-07, FWS-08])
- location data (e.g.,[GL-08])
- ...

Re-identification with any information

- Any information can be used to re-identify anonymous data

  $\Longrightarrow$ ensuring proper privacy protection is a difficult task since the amount and variety of data collected about individuals is increased

- Two examples:
  - AOL
  - Netflix

AOL data release (1)

- In 2006, to embrace the vision of an open research community, AOL (America OnLine) publicly posted to a web site 20 million search queries for 650,000 users of AOL's search engine summarizing three months of activity
- AOL suppressed any obviously identifying information such as AOL username and IP address
- AOL replaced these identifiers with unique identification numbers (this made searches by the same user linkable)

AOL data release (2)

- User 44117749:
  - "numb fingers", "60 single men", "dog that urinates on everything"
  - "hand tremors", "nicotine effects on the body", "dry mouth", and "bipolar"
  - "Arnold" (several people with this last name)
  - "landscapers in Lilburn, Ga", "homes sold in shadow lake subdivision Gwinnett county, Georgia"

  $\Longrightarrow$ Thelma Arnold, a 62-year-old widow who lives in Lilburn, Ga

- She was re-identified by two New York Times reporters

- She explained in an interview that she has three dogs and that she searched for medical conditions of some friends

AOL data release (3)

What about user 17556639?

- how to kill your wife
- how to kill your wife
- wife killer
- how to kill a wife
- poop
- dead people
- pictures of dead people
- killed people
- dead pictures
- dead pictures
- dead pictures
- murder photo
- steak and cheese
- photo of death
- photo of death
- death
- dead people photos
- photo of dead people
- www.murderdpeople.com
- decapatated photos
- decapatated photos
- car crashes3
- car crashes3
- car crash photo

Netflix prize data study (1)

- In 2006, Netlix (the world largest online movie rental service), launched the "Netflix Prize" (a challenge that lasted almost three years)
    - Prize of US $ 1 million to be awarded to those who could provide a movie recommendation algorithm that improved Netflix's algorithm by 10%
- Netflix provided 100 million records revealing how nearly 500,000 of its users had rated movies from Oct.'98 to Dec.'05
- In each record Netflix disclosed the movie rated, the rating assigned (1 to 5), and the date of the rating

Netflix prize data study (2)

- Only a sample (one tenth) of the database was released
- Some ratings were perturbed (but not much to not alter statistics)
- Identifying information (e.g., usernames was removed), but a unique user identifier was assigned to preserve rating-to-rating continuity
- Release was not k-anonymous for any $k > 1$

Netflix prize data study (3)

- De-identified Netflix data can be re-identified by linking with external sources (e.g., user ratings from IMDb users)
  - Knowing the precise ratings a person has assigned to six obscure (outside the top 500) movies, an adversary is able to uniquely identify that person 84% of the time
  - Knowing approximately when ($\pm$ 2 weeks) a person has rated six movies (whether or not obscure), an adversary is able to reidentify that person in 99% of the cases
  - Knowing two movies a user has rated, with precise ratings and rating dates ($\pm$ 3 days), an adversary is able to reidentify 68% of the users
- Movies may reveal your political orientation, religious views, or sexual orientations (Netflix was sued by a lesbian for breaching her privacy)

Differential privacy [D-06] (1)

- Differential privacy aims at preventing adversaries from being capable to detect the presence or absence of a given individual in a dataset. E.g.,:
  - the count of individuals with cancer from a medical database is produced with a release mechanism that when executed on datasets differing on one individual probably returns the same result
- It defines a property on the data release mechanism

Differential privacy [D-06] (2)

Informally:

- Differential privacy requires the probability distribution on the published results of an analysis to be "essentially the same" independent of whether an individual is represented or not in the dataset

Formally:

- A randomized function K gives ε-differential privacy if for all data sets D and D' differing on at most one row, and all $S \subseteq$ Range(K),
Pr[$K(D) \in S$] $\leq$ exp(ε) $\times$ Pr[$K(D') \in S$]

Differential privacy [D-06] (3)

- Applicable to two scenarios
  - non-interactive scenario: public release of a dataset
  - interactive scenario: evaluation of queries over a private dataset
- It is typically enforced by adding random noise
  $\Longrightarrow$ data truthfulness is not preserved
- ε-differentially private mechanisms compose automatically

Differential privacy variations and applications

- Variations of differential privacy to reduce the amount of noise in data/query result:
  - (ε, δ)-differential privacy [DS-09]: the ε bound on query answer probabilities may be violated with small probability (controlled by δ)
  - adversaries with polynomial time computational bounds (e.g., [MPRV-09])
  - use of wavelet transforms for improving data utility [XWG-11]
  - $\ldots$
- Similarly to k-anonymity, differentially private mechanisms have been developed for different domains:
  - social networks (e.g., [HLMJ-09, MW-09, RHMS-09])
  - data mining (e.g., [CMFDX-11, DWHL-11, MCFY-11])
  - location data (e.g., [HR-11])

Is differential privacy enough?

- Limiting the inference about the presence of a tuple if different from limiting the inference about the participation of the individual in the data generating process [KM-11, KM-12]
  - Bob's participation in a social network can cause links to form between Bob's friends (Bob's participation affects more than just the tuple marked "Bob")
- Differential privacy composes well with itself but not necessarily with other privacy definitions or data release mechanisms (which represent background knowledge that can cause privacy breaches)

Some open issues

- New privacy metrics
- New techniques to protect privacy
- External knowledge and adversarial attacks
- Evaluation of privacy vs utility

Privacy in Data Outsourcing/Cloud Computing

P. Samarati, S. De Capitani di Vimercati, “Data Protection in Outsourcing Scenarios: Issues and Directions,” in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2010),* Beijing, China, April, 2010.

Motivation

- The management of large amount of sensitive information is quite expensive
- Novel paradigms (e.g., data outsourcing, cloud computing) are emerging for enabling Internet-based access to data and applications shared among different clients [HIML-02,DFS-12]
- Data are typically stored at external data centers managed by parties different from the data owner
  - + significant cost savings and service benefits
  - + promises higher availability and more effective disaster protection than in-house operations
  - – sensitive data are not under the data owner's control
  - – servers may be honest-but-curious

$\Longrightarrow$ sensitive data have to be encrypted or kept separate from other PII

Issues to be addressed

- Data protection
- Query execution
- Private access
- Data integrity and correctness
- Access control enforcement
- Data publication and utility
- Collaborative query execution

Issues to be addressed

- Data protection: encryption and fragmentation
- Query execution: indexes
- Private access: [DFPPS-11]
- Data integrity and correctness
- Access control enforcement: encryption policy, over-encryption
- Data publication and utility: loose associations
- Collaborative query execution: [DFJPS-11]

Data Protection

P. Samarati, S. De Capitani di Vimercati, “ Data Protection in Outsourcing Scenarios: Issues and Directions,” in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2010)*, Beijing, China, Aprile 13-16, 2010.

Data protection: Solutions

- Solutions for protecting data can be based on
  - encryption
  - encryption and fragmentation
  - fragmentation

Encryption-Based Solutions and Indexes

Encryption and indexes (1)

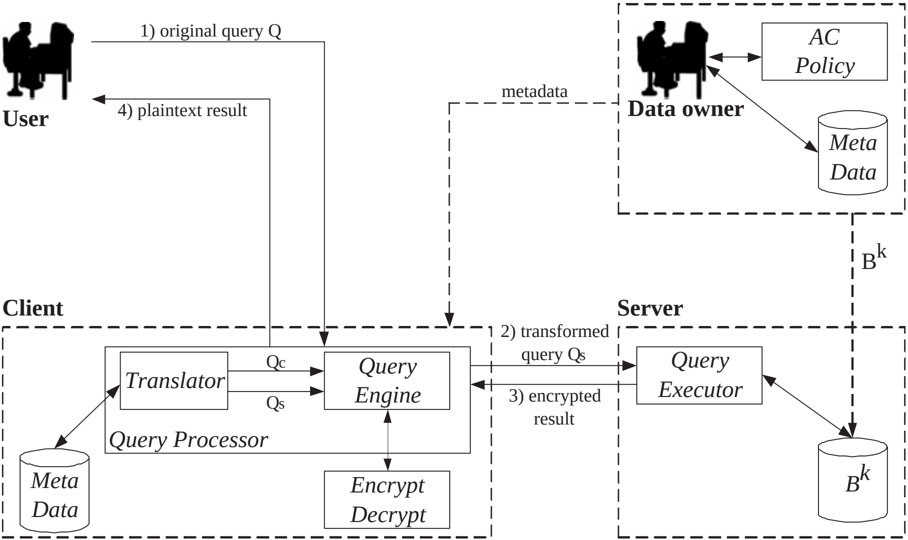
The granularity level at which database encryption is performed can depend on the data that need to be accessed. Encryption can be applied at the granularity of:

- table: each table in the plaintext database is represented through a single encrypted value in the encrypted database
- attribute: each column (attribute) in the plaintext table is represented by a single encrypted value in the encrypted table
- tuple: each tuple in the plaintext table is represented by a single encrypted value in the encrypted table
- cell: each cell (element) in the plaintext table is represented by a single encrypted value in the encrypted table

Encryption and indexes (2)

- For performance reasons, encryption is typically applied at the tuple level
- An index can be associated with each attribute on which conditions may need to be evaluated
- Indexes are used by the server to select data to be returned in response to a query
- A relation r over schema R(A_1,$A_2,\ldots,A_n$) is mapped onto a relation r^k over schema R^k(<u>Counter</u>, Etuple, $I_1,I_2,\ldots,I_n$):
  - Counter: primary key
  - Etuple: ciphertext for plaintext tuple t, Etuple=$E_k(t)$
  - I_j: index associated with attribute A_j

Entities involved in the outsourcing scenario



Indexing information

Different choices for indexing, e.g.:

- actual attribute value, $t[I_i] = t[A_i]$ (inapplicable)
- individual encrypted value, $t[I_i] = E_k(t[A_i])$
  - $+$ simple and precise for equality queries
  - $-$ preserves plaintext value distinguishability
- partition-based index, $t[I_i] = B$, with B the value associated with a partition containing $t[A_i]$
- secure hash function over the attribute values $t[I_i] = h(t[A_i])$

partition-based and secure hash function:

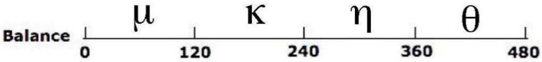
- $+$ support for equality queries
- $+$ collisions remove plaintext distinguishability
- $-$ result may contain spurious tuple that need to be eliminated (postprocessing query)

Partition-based index [HILM-02]

- Consider an arbitrary plaintext attribute A_i in relational schema R, with domain D_i
- D_i is partitioned in a number of non-overlapping subsets of values, called partitions, containing contiguous values
- Each partition is associated with an identifier
- The corresponding index value is the unique value associated with the partition to which the plaintext value $t[A_i]$ belongs
- The association partition-identifier can be order-preserving
  - + support for interval-based queries
  - – expose to inference (the comparison among the ordered sequences of plaintext and indexes would lead to reconstruct the correspondence)

Partition-based index – Example

Random mapping



Accounts

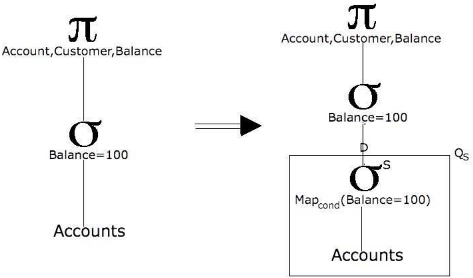
<u>Account</u>	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

$\mathbf{Accounts}_1^k$

<u>Counter</u>	Etuple	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	β	η
4	JpO8eLTVgwV1E	ρ	γ	κ
5	qctG6XnFNDTQc	ς	δ	θ
6	4QbqCeq3hxZHklU	ι	ε	κ

Query execution – Simple example

SELECT *
FROM Accounts
WHERE Balance = 100



Hash-based index [CDDJPS-05]

- Based on the concept of one-way hash function
- For each attribute A_i in R with domain D_i, a secure one-way hash function $h : D_i \rightarrow B_i$ is defined, where B_i is the domain of index I_i associated with A_i
- Given a plaintext tuple t in r, the index value corresponding to $t[A_i]$ is $h(t[A_i])$
- Important properties of any secure hash function h are:
  - $\forall x, y \in D_i : \; x = y \implies h(x) = h(y)$ (determinism)
  - given two values $x, y \in D_i$ with $x \neq y$, we may have that $h(x) = h(y)$ (collision)
  - given two distinct but near values x, y ($|\, x - y \,| < \varepsilon$) chosen randomly in D_i, the discrete probability distribution of the difference $h(x) - h(y)$ is uniform (strong mixing)

Hash-based index – Example

Accounts

<u>Account</u>	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts$^k_2$

<u>Counter</u>	<u>Etuple</u>	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	δ	θ
4	JpO8eLTVgwV1E	ρ	α	κ
5	qctG6XnFNDTQc	ς	β	κ
6	4QbqC3hxZHklU	ι	β	κ

- h_c(Alice)=h_c(Chris)=α
- h_c(Donna)=h_c(Elvis)=β
- h_c(Bob)=δ
- h_b(200)=h_b(400)=κ
- h_b(100)=μ
- h_b(300)=θ

Query execution

- Each query Q on the plaintext DB is translated into:
  - a query Q_s to be executed at the server
  - a query Q_c to be executed at client on the result
- Query Q_s is defined according to the index technique adopted
- Query Q_c is executed on the decrypted result of Q_s to filter out spurious tuples
- The translation should be performed in such a way that the server is responsible for the majority of the work

Query execution – Simple example

Accounts

<u>Account</u>	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts$_2^k$

<u>Counter</u>	<u>Etuple</u>	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	δ	θ
4	JpO8eLTVgwV1E	ρ	α	κ
5	qctG6XnFNDTQc	ς	β	κ
6	4QbqC3hxZHklU	ι	β	κ

Original query on Accounts	**Translation over Accounts$_2^k$**
Q := SELECT * FROM Accounts WHERE Balance = 200	Q_s := SELECT Etuple FROM Accounts$_2^k$ WHERE $I_B = \kappa$ Q_c := SELECT * FROM *Decrypt*(Q_s, *Key*) WHERE Balance = 200

Query execution – Simple example

Accounts

<u>Account</u>	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts_2^k

<u>Counter</u>	<u>Etuple</u>	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	δ	θ
4	JpO8eLTVgwV1E	ρ	α	κ
5	qctG6XnFNDTQc	ς	β	κ
6	4QbqC3hxZHklU	ι	β	κ

Original query on Accounts	Translation over Accounts_2^k
Q := SELECT * FROM Accounts WHERE Balance=200	Q_s := SELECT Etuple FROM Accounts_2^k WHERE $I_B=\kappa$ Q_c := SELECT * FROM *Decrypt*(Q_s, *Key*) WHERE Balance=200

Query execution – Simple example

Accounts

<u>Account</u>	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts$_2^k$

<u>Counter</u>	<u>Etuple</u>	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	δ	θ
4	JpO8eLTVgwV1E	ρ	α	κ
5	qctG6XnFNDTQc	ς	β	κ
6	4QbqC3hxZHklU	ι	β	κ

Original query on Accounts	Translation over Accounts$_2^k$
Q := SELECT * FROM Accounts WHERE Balance$=$200	Q_s := SELECT Etuple FROM Accounts$_2^k$ WHERE I_B=κ Q_c := SELECT * FROM *Decrypt*(Q_s, *Key*) WHERE Balance$=$200

Query execution – Simple example

Accounts

Account	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts$_2^k$

Counter	Etuple	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	δ	θ
4	JpO8eLTVgwV1E	ρ	α	κ
5	qctG6XnFNDTQc	ς	β	κ
6	4QbqC3hxZHklU	ι	β	κ

Original query on Accounts	Translation over Accounts$_2^k$
Q := SELECT * FROM Accounts WHERE Balance$=$200	Q$_s$:= SELECT Etuple FROM Accounts$_2^k$ WHERE $I_B=\kappa$ Q$_c$:= SELECT * FROM *Decrypt*(Q$_s$, *Key*) WHERE Balance$=$200

Query execution – Simple example

Accounts

Account	**Customer**	**Balance**
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts_2^k

Counter	**Etuple**	I_A	I_C	I_B
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	δ	θ
4	JpO8eLTVgwV1E	ρ	α	κ
5	qctG6XnFNDTQc	ς	β	κ
6	4QbqC3hxZHklU	ι	β	κ

Original query on Accounts	**Translation over Accounts_2^k**
Q := SELECT * FROM Accounts WHERE Balance=200	Q_s := SELECT Etuple FROM Accounts_2^k WHERE I_B=κ Q_c := SELECT * FROM *Decrypt*(Q_s, *Key*) WHERE Balance=200

Inference Exposure

Inference exposure

There are two conflicting requirements in indexing data:

- indexes should provide an effective query execution mechanism
- indexes should not open the door to inference and linking attacks

It is important to measure quantitatively the level of exposure due to the publication of indexes [CDDJPS-05]

Scenarios

The exposure due to indexes depends on:

- the indexing method adopted, e.g.,
  - direct encryption
  - hashing
- the a-priori knowledge of the intruder, e.g.,
  - Freq+DB^k:
    - the frequency distribution of plaintext values in the original database (Freq)
    - the encrypted database (DB^k)
  - DB+DB^k:
    - the plaintext database (DB)
    - the encrypted database (DB^k)

Possible inferences

Freq+DB^k

- *plaintext content*: determine the existence of a certain tuple (or *association* of values) in the original database
- *indexing function*: determine the correspondence between plaintext values and indexes

DB+DB^k

- *indexing function*: determine the correspondence between plaintext values and indexes

Freq+DB^k – Example

Knowledge

Account	Customer	Balance
Acc1	Alice	100
Acc2	Alice	200
Acc3	Bob	300
Acc4	Chris	200
Acc5	Donna	400
Acc6	Elvis	200

Accounts$_1^k$

Counter	Etuple	I$_A$	I$_C$	I$_B$
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	β	η
4	JpO8eLTVgwV1E	ρ	γ	κ
5	qctG6XnFNDTQc	ς	δ	θ
6	4QbqC3hxZHklU	ι	ε	κ

Inference

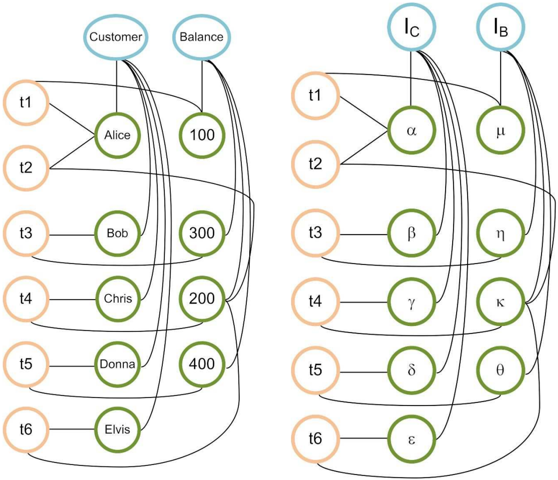
- $I_A =$ Account
- $I_C =$ Customer
- $I_B =$ Balance
- $\kappa = 200$ (indexing inference)
- $\alpha =$Alice (indexing inference)
- $\langle$Alice,200$\rangle$ is in the table (association inference)
- Alice is also associated with a value different from 200 (“100,300,400”, all equiprobable)

DB+DB^k – Example (1)

Customer	Balance
Alice	100
Alice	200
Bob	300
Chris	200
Donna	400
Elvis	200

I_C	I_B
α	μ
α	κ
β	η
γ	κ
δ	θ
ε	κ

DB+DB^k – Example (2)



Inference

- I_C = Customer
- I_B = Balance
- α = Alice
- μ = 100
- κ = 200
- $\{\gamma, \varepsilon\}$ = {Chris,Elvis}
- $\{\langle\beta,\eta\rangle,\langle\delta,\theta\rangle\}$= $\{\langle$Bob,300$\rangle$,$\langle$Donna,400$\rangle\}$

Searchable encryption

Order preserving encryption

- Order Preserving Encryption (OPES) is an encryption technique that takes as input a target distribution of index values and applies an order preserving transformation [AKSX-04]
  - \+ comparison can be directly applied on the encrypted data
  - \+ query evaluation does not produce spurious tuples
  - – vulnerable to inference attacks
- Order Preserving Encryption with Splitting and Scaling (OPESS) guarantees a flat distribution of the frequencies of index values [WL-06]
  - decreases exposure to inference attacks; remains vulnerable in dynamic scenarios

Fully homomorphic encryption [G-09]

Fully homomorphic encryption schema allows the execution of queries on encrypted data without decrypting them

- A query is sent to the server that transforms it as a function f
- The server homomorphically computes an encryption of f on the encrypted data
- The encrypted result of f is then sent to the requester that decrypts it and retrieves the relevant data

Still open problems:

- not practical for DBMSs
- vulnerable with respect to inference

Data Integrity

Integrity of outsourced data

Two aspects:

- Integrity in storage: data must be protected against improper modifications

  $\Longrightarrow$ unauthorized updates to the data must be detected

- Integrity in query computation: query results must be correct and complete

  $\Longrightarrow$ server's misbehavior in query evaluation must be detected

Integrity in storage

- Data integrity in storage typically relies on digital signatures
- Signatures are usually computed at tuple level
  - table and attribute level signatures can be verified only after downloading the whole table/column
  - cell level signature causes a high verification overhead
- The verification cost grows linearly with the number of tuples in the query result

  $\Longrightarrow$ the signature of a set of tuples can be combined in a unique signature [MNT-06]

Integrity in query computation

- Query result must be correct and complete
  - the result must not be tampered with
  - the result must include all data satisfying the query
- Two approaches:
  - authenticated data structures
  - probabilistic

Authenticated data structures approaches

- Based on the definition of appropriate data structures on the original data
  - signature chains (e.g., [NT-05])
  - Merkle hash trees (e.g., [DGMS-00])
  - skip lists (e.g., [PPP-10])
- Provide an absolute guarantee of query correctness and completeness but only for the attribute on which the data structure is built

Probabilistic approaches

- Based on the:
  - insertion of fake tuples in query results (e.g., [XWYM-07])
  - replication of a subset of the tuples in query results (e.g., [WYPY-08])
  - pre-computation of tokens associated with chosen query results (e.g., [S-05])
- Provide a probabilistic guarantee of completeness of query results
- More efficient than authenticated data structures approaches

Controlling Access to Outsourced Data

Access control

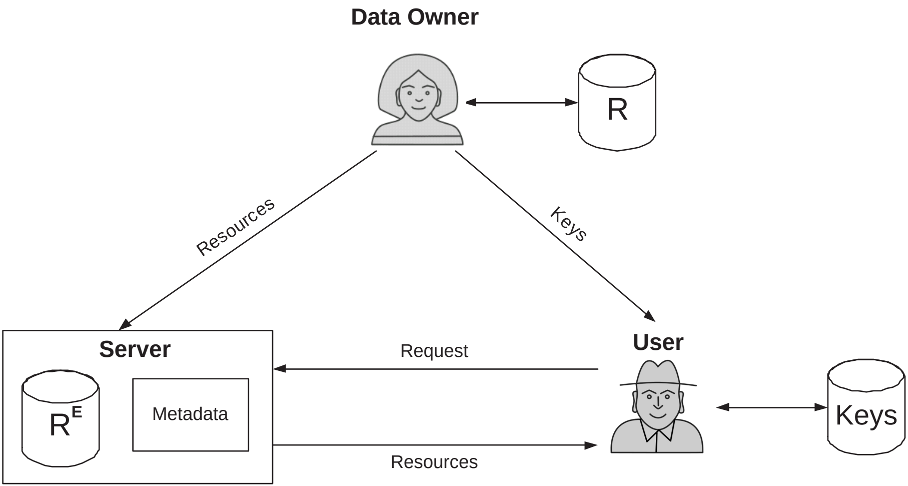
- Different users might need to enjoy different views on the outsourced data
- Enforcement of the access control policy requires the data owner to mediate access requests
- Existing approaches for data outsourcing can support the use of different keys for encrypting different data
  $\Longrightarrow$ selective encryption as a means to enforce selective access [DFJPS-10]

Selective encryption

Basic idea/desiderata:

- data themselves need to directly enforce access control
- different keys should be used for encrypting data
- authorization to access a resource translated into knowledge of the key with which the resource is encrypted
- each user is communicated the keys necessary to decrypt the resources she is entailed to access

Selective encryption – Scenario



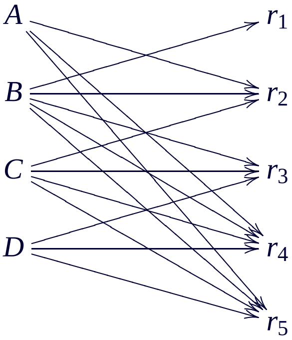
Authorization policy

- The data owner defines a discretionary access control (authorization) policy to regulate read access to the resources

- An authorization policy $\mathscr{A}$, is a set of permissions of the form $\langle$user,resource$\rangle$.
  It can be represented as:
  - an access matrix
  - a directed and bipartite graph having a vertex for each user u and for each resource r, and an edge from u to r for each permission $\langle u, r \rangle$

Authorization policy – Example

	r_1	r_2	r_3	r_4	r_5
A	0	1	0	1	1
B	1	1	1	1	1
C	0	1	1	1	1
D	0	0	1	1	1



Encryption policy

- The authorization policy defined by the data owner is translated into an equivalent encryption policy

- Possible solutions:
  - encrypt each resource with a different key and give users the keys for the resources they can access
    - requires each user to manage as many keys as the number of resources she is authorized to access
  - use a key derivation method for allowing users to derive from their user keys all the keys that they are entitled to access
    + allows limiting to one the key to be released to each user

Key derivation methods

- Based on a key derivation hierarchy $(\mathscr{K},\preceq)$
  - $\mathscr{K}$ is the set of keys in the system
  - $\preceq$ partial order relation defined on $\mathscr{K}$
- The knowledge of the key of vertex v_1 and of a piece of information publicly available allows the computation of the key of a lower level vertex v_2 such that $v_2 \preceq v_1$
- $(\mathscr{K},\preceq)$ can be graphically represented as a graph with a vertex for each $x \in \mathscr{K}$ and a path from x to y iff $y \preceq x$
- Depending on the partial order relation defined on $\mathscr{K}$, the key derivation hierarchy can be:
  - a chain [S-87]
  - a tree [G-80,S-87,S-88]
  - a DAG [AT-83,CMW-06,DFM-04,HL-90,HY-03,LWL-89,M-85,SC-02]

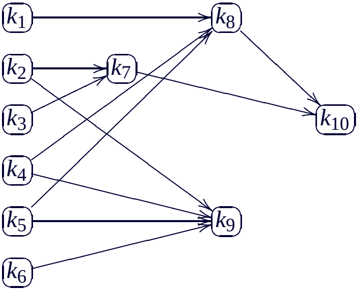
Token-based key derivation methods [AFB-05]

- Keys are arbitrarily assigned to vertices
- A public label l_i is associated with each key k_i
- A piece of public information $t_{i,j}$, called token, is associated with each edge in the hierarchy
- Given an edge (k_i,k_j), token $t_{i,j}$ is computed as $k_j \oplus h(k_i, l_j)$ where
  - $\oplus$ is the n-ary `xor` operator
  - h is a secure hash function
- Advantages of tokens:
  - they are public and allow users to derive multiple encryption keys, while having to worry about a single one
  - they can be stored on the remote server (just like the encrypted data), so any user can access them

Key and token graph

- Relationships between keys through tokens can be represented via a key and token graph
  - a vertex for each pair $\langle k, l\rangle$, where $k \in \mathscr{K}$ is a key and $l \in \mathscr{L}$ the corresponding label
  - an edge from a vertex $\langle k_i, l_i\rangle$ to vertex $\langle k_j, l_j\rangle$ if there exists a token $t_{i,j} \in \mathscr{T}$ allowing the derivation of k_j from k_i

Example



Key assignment and encryption schema

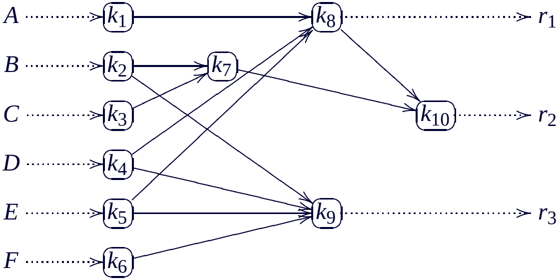
Translation of the authorization policy into an encryption policy:

- Starting assumptions (desiderata):
  - each user can be released only a single key
  - each resource is encrypted only once (with a single key)
- Function ϕ:$\mathscr{U} \cup \mathscr{R} \rightarrow \mathscr{L}$ describes:
  - the association between a user and (the label of) her key
  - the association between a resource and (the label of) the key used for encrypting it

Formal definition of encryption policy

- An encryption policy over users $\mathscr{U}$ and resources $\mathscr{R}$, denoted $\mathscr{E}$, is a 6-tuple $\langle \mathscr{U}, \mathscr{R}, \mathscr{K}, \mathscr{L}, \phi, \mathscr{T} \rangle$, where:
  - $\mathscr{K}$ is the set of keys defined in the system and $\mathscr{L}$ is the set of corresponding labels
  - ϕ is a key assignment and encryption schema
  - $\mathscr{T}$ is a set of tokens defined on $\mathscr{K}$ and $\mathscr{L}$
- The encryption policy can be represented via a graph by extending the key and token graph to include:
  - a vertex for each user and each resource
  - an edge from each user vertex u to the vertex $\langle k, l \rangle$ such that $\phi(u)$=l
  - an edge from each vertex $\langle k, l \rangle$ to each resource vertex r such that $\phi(r) = l$

Encryption policy graph – Example



- user A can access $\{r_1, r_2\}$
- user B can access $\{r_2, r_3\}$
- user C can access $\{r_2\}$
- user D can access $\{r_1, r_2, r_3\}$
- user E can access $\{r_1, r_2, r_3\}$
- user F can access $\{r_3\}$

Policy transformation

Goal: translate an authorization policy $\mathscr{A}$ into an equivalent encryption policy $\mathscr{E}$.

$\mathscr{A}$ and $\mathscr{E}$ are equivalent if they allow exactly the same accesses:

- $\forall u \in \mathscr{U}, r \in \mathscr{R} : u \xrightarrow{\mathscr{E}} r \implies u \xrightarrow{\mathscr{A}} r$
- $\forall u \in \mathscr{U}, r \in \mathscr{R} : u \xrightarrow{\mathscr{A}} r \implies u \xrightarrow{\mathscr{E}} r$

Translating $\mathscr{A}$ into $\mathscr{E}$ (1)

- Naive solution
  - each user is associated with a different key
  - each resource is encrypted with a different key
  - a token $t_{u,r}$ is generated and published for each permission $\langle u, r \rangle$

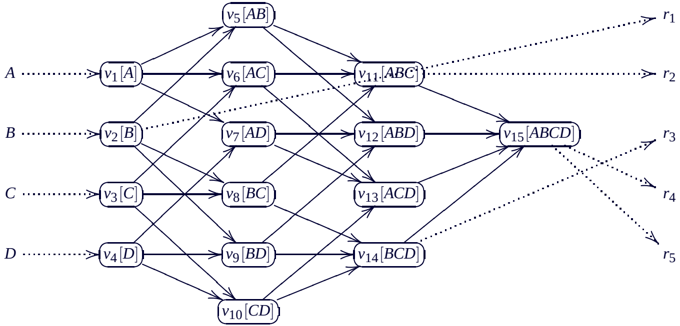
$\Longrightarrow$ producing and managing a token for each single permission can be unfeasible in practice

- Exploiting acls and user groups
  - group users with the same access privileges
  - encrypt each resource with the key associated with the set of users that can access it

Translating $\mathscr{A}$ into $\mathscr{E}$ (2)

- It is possible to create an encryption policy graph by exploiting the hierarchy among sets of users induced by the partial order relationship based on set containment ($\subseteq$)
- If the system has a large number of users, the encryption policy has a large number of tokens and keys ($2^{|\mathscr{U}|} - 1$)

$\Longrightarrow$ inefficient key derivation



Minimum encryption policy

- Observation: user groups that do not correspond to any acl do not need to have a key
- Goal: compute a minimum encryption policy, equivalent to a given authorization policy, that minimize the number of tokens to be maintained by the server
- Solution: heuristic algorithm based on the observation that:
  - only vertices associated with user groups corresponding to actual acls need to be associated with a key
  - the encryption policy graph may include only the vertices that are needed to enforce a given authorization policy, connecting them to ensure a correct key derivability
  - other vertices can be included if they are useful for reducing the size of the catalog

Construction of the key and token graph

Start from an authorization policy $\mathscr{A}$

1. Create a vertex/key for each user and for each non-singleton *acl* (initialization)

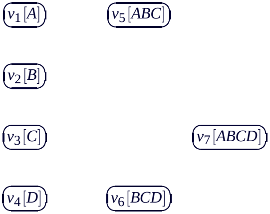
2. For each vertex v corresponding to a non-singleton *acl*, find a cover without redundancies (covering)
   - for each user u in v.*acl*, find an ancestor v' of v with $u \in v'$.*acl*

3. Factorize common ancestors (factorization)

An example of key and token graph

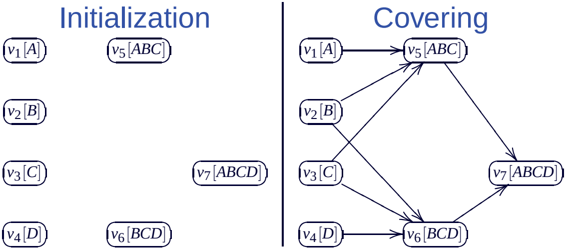
	r_1	r_2	r_3	r_4	r_5
A	0	1	0	1	1
B	1	1	1	1	1
C	0	1	1	1	1
D	0	0	1	1	1

Initialization



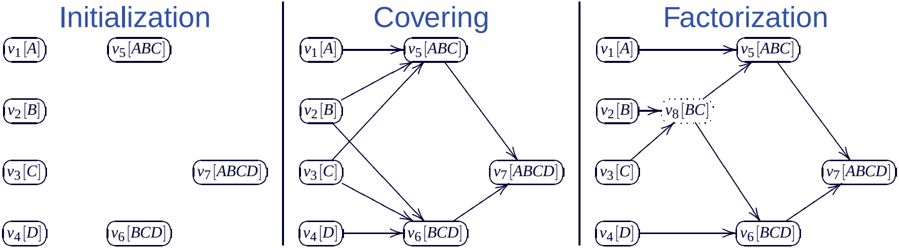
An example of key and token graph

	r_1	r_2	r_3	r_4	r_5
A	0	1	0	1	1
B	1	1	1	1	1
C	0	1	1	1	1
D	0	0	1	1	1

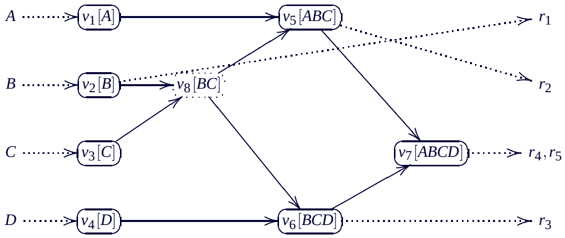


An example of key and token graph

	r_1	r_2	r_3	r_4	r_5
A	0	1	0	1	1
B	1	1	1	1	1
C	0	1	1	1	1
D	0	0	1	1	1



Key assignment and encryption schema ϕ and catalog



u	$\phi(u)$
A	$v_1.l$
B	$v_2.l$
C	$v_3.l$
D	$v_4.l$

r	$\phi(r)$
r_1	$v_2.l$
r_2	$v_5.l$
r_3	$v_6.l$
r_4,r_5	$v_7.l$

source	destination	token_value
$v_1.l$	$v_5.l$	$t_{1,5}$
$v_2.l$	$v_8.l$	$t_{2,8}$
$v_3.l$	$v_8.l$	$t_{3,8}$
$v_4.l$	$v_6.l$	$t_{4,6}$
$v_5.l$	$v_7.l$	$t_{5,7}$
$v_6.l$	$v_7.l$	$t_{6,7}$
$v_8.l$	$v_5.l$	$t_{8,5}$
$v_8.l$	$v_6.l$	$t_{8,6}$

Policy changes

- When authorizations dynamically change the data owner needs to:
  - download the resource from the server
  - create a new key for the resource
  - decrypt the resource with the old key
  - re-encrypt the resource with the new key
  - upload the resource to the server and communicate the public catalog updates

  $\Longrightarrow$ inefficient

- Possible solution: over-encryption

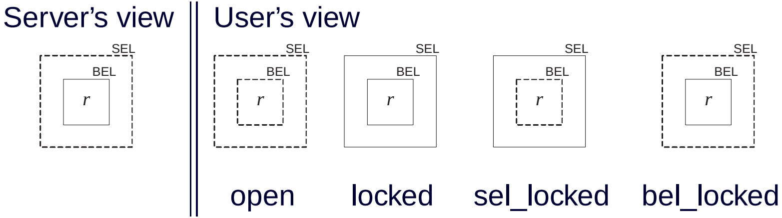
Over-encryption [DFJPS-07]

- Resources are encrypted twice
  - by the owner, with a key shared with the users and unknown to the server (Base Encryption Layer - BEL level)
  - by the server, with a key shared with authorized users (Surface Encryption Layer - SEL level)
- To access a resource a user must know both the corresponding BEL and SEL keys
- Grant and revoke operations may require
  - the addition of new tokens at the BEL level
  - the update of the SEL level according to the operations performed

Views on resource r (1)

- Four views:
  - open: the user knows the key at the BEL level as well as the key at the SEL level
  - locked: the user knows neither the key at the BEL level nor the key at the SEL level
  - sel_locked: the user knows only the key at the BEL level but does not know the key at the SEL level
  - bel_locked: the user knows only the key at the SEL level but does not know the one at the BEL level
- The server always has the bel_locked view

Views on resource r (2)



- Each layer is depicted as a fence
  - discontinuous, if the key is known
  - continuous, if the key is not known (protection cannot be passed)

Data Fragmentation

Fragmentation and encryption

- Encryption makes query evaluation and application execution more expensive or not always possible
- Often what is sensitive is the association between values of different attributes, rather than the values themselves
  - e.g., association between employee's names and salaries

  $\Longrightarrow$protect associations by breaking them, rather than encrypting
- Recent solutions for enforcing privacy requirements couple:
  - encryption
  - data fragmentation

Confidentiality constraints

- Sets of attributes such that the (joint) visibility of values of the attributes in the sets should be protected

- Sensitive attributes: the values of some attributes are considered sensitive and should not be visible

  $\Longrightarrow$ singleton constraints

- Sensitive associations: the associations among values of given attributes are sensitive and should not be visible

  $\Longrightarrow$ non-singleton constraints

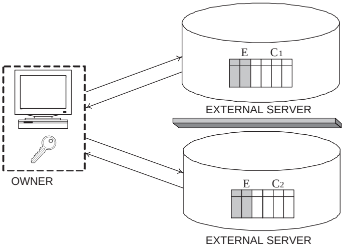
Outline

- Data fragmentation
  - Non-communicating pair of servers [ABGGKMSTX-05]
  - Multiple fragments [CDFJPS-07,CDFJPS-10]
  - Departing from encryption: Keep a few [CDFJPS-09b]
- Publishing obfuscated associations
  - Anonymizing bipartite graph [CSYZ-08]
  - Fragments and loose associations [DFJPS-10b]

Non-Communicating Pair of Servers

Non-communicating pair of servers

- Confidentiality constraints are enforced by splitting information over two independent servers that cannot communicate (need to be completely unaware of each other)
  - Sensitive associations are protected by distributing the involved attributes among the two servers
  - Encryption is applied only when explicitly demanded by the confidentiality constraints or when storing the attribute in any of the server would expose at least a sensitive association



- $E \cup C_1 \cup C_2 = R$
- $C_1 \cup C_2 \subseteq R$

Enforcing confidentiality constraints

- Confidentiality constraints $\mathscr{C}$ defined over a relation R are enforced by decomposing R as $\langle R_1, R_2, E \rangle$ where:
  - R_1 and R_2 include a unique tuple ID needed to ensure lossless decomposition
  - $R_1 \cup R_2 = R$
  - E is the set of encrypted attributes and $E \subseteq R_1$, $E \subseteq R_2$
  - for each $c \in \mathscr{C}$, $c \nsubseteq (R_1 - E)$ and $c \nsubseteq (R_2 - E)$

Confidentiality constraints – Example (1)

R = (Name,DoB,Gender,Zip,Position,Salary,Email,Telephone)

- {Telephone}, {Email}
  - attributes Telephone and Email are sensitive (cannot be stored in the clear)
- {Name,Salary}, {Name,Position}, {Name,DoB}
  - attributes Salary, Position, and DoB are private of an individual and cannot be stored in the clear in association with the name
- {DoB,Gender,Zip,Salary}, {DoB,Gender,Zip,Position}
  - attributes DoB, Gender, Zip can work as quasi-identifier
- {Position,Salary}, {Salary,DoB}
  - association rules between Position and Salary and between Salary and DoB need to be protected from an adversary

Enforcing confidentiality constraints – Example (2)

R = (Name,DoB,Gender,Zipcode,Position,Salary,Email,Telephone)

{Telephone}
{Email}
{Name,Salary}
{Name,Position}
{Name,DoB}
{DoB,Gender,Zipcode,Salary}
{DoB,Gender,Zipcode,Position}
{Position,Salary}
{Salary,DoB}

$\Longrightarrow R$ = (Name,DoB,Gender,Zipcode,Position,Salary,Email,Telephone)

- R_1: (ID,Name,Gender,Zipcode,Salary^e,Email^e,Telephone^e)
- R_2: (ID,Position,DoB,Salary^e,Email^e,Telephone^e)

Note that Salary is encrypted even if non sensitive per se since storing it in the clear in any of the two fragments would violate at least a constraint

Query execution

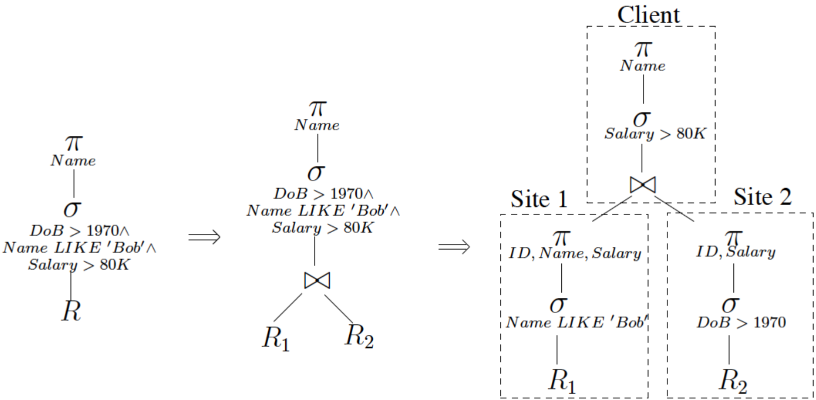
At the logical level: replace R with $R_1 \bowtie R_2$

Query plans:

- Fetch R_1 and R_2 from the servers and execute the query locally
  - extremely expensive
- Involve servers S_1 and S_2 in the query evaluation
  - can do the usual optimizations, e.g. push down selections and projections
  - selections cannot be pushed down on encrypted attributes
  - different options for executing queries:
    - send sub-queries to both S_1 and S_2 in parallel, and join the results at the client
    - send only one of the two sub-queries, say to S_1; the tuple IDs of the result from S_1 are then used to perform a semi-join with the result of the sub-query of S_2 to filter R_2

Query execution – Example

- R_1: (ID,Name,Gender,Zipcode,Salary^e,Email^e,Telephone^e)
- R_2: (ID,Position,DoB,Salary^e,Email^e,Telephone^e)



Identifying the optimal decomposition (1)

Brute force approach for optimizing wrt workload W:

- For each possible safe decomposition of R:
  - optimize each query in W for the decomposition
  - estimate the total cost for executing the queries in W using the optimized query plans
- Select the decomposition that has the lowest overall query cost

Too expensive! $\Longrightarrow$ Exploit affinity matrix

Identifying the optimal decomposition (2)

Adapted affinity matrix M:

- $M_{i,j}$: 'cost' of placing cleartext attributes i and j in different fragments

- $M_{i,i}$: 'cost' of placing encrypted attribute i (across both fragments)

Goal: Minimize

$$\sum_{i,j:i \in (R_1 - E), j \in (R_2 - E)} M_{i,j} + \sum_{i \in E} M_{i,i}$$

Identifying the optimal decomposition (3)

Optimization problem equivalent to hypergraph coloring problem
Given relation R, define graph $G(R)$:

- attributes are vertexes
- affinity value $M_{i,j} \Longrightarrow$ weight of arc (i,j)
- affinity value $M_{i,i} \Longrightarrow$ weight of vertex i
- confidentiality constraints $\mathscr{C}$ represent a hypergraph $H(R,\mathscr{C})$ on the same vertexes

Identifying the optimal decomposition (4)

Find a 2-coloring of the vertexes such that:

- no hypergraph edge is monochromatic
- the weight of bichromatic edges is minimized
- a vertex can be deleted (i.e., encrypted) by paying the price equal to the vertex weight

Coloring a vertex is equivalent to place it in one of the two fragments.
The 2-coloring problem is NP-hard.
Different heuristics, all exploiting:

- approximate min-cuts
- approximate weighted set cover

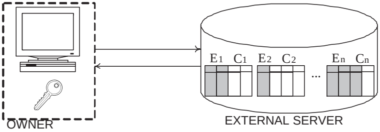
Multiple Fragments

Multiple fragments (1)

Coupling fragmentation and encryption interesting and promising, but, limitation to two servers:

- too strong and difficult to enforce in real environments
- limits the number of associations that can be solved by fragmenting data, often forcing the use of encryption

$\Longrightarrow$ allow for more than two non-linkable fragments



- $E_1 \cup C_1 = \ldots = E_n \cup C_n = R$
- $C_1 \cup \ldots \cup C_n \subseteq R$

Multiple fragments (2)

- A fragmentation of R is a set of fragments $\mathscr{F} = \{F_1, \ldots, F_m\}$, where $F_i \subseteq R$, for $i = 1, \ldots, m$

- A fragmentation $\mathscr{F}$ of R correctly enforces a set $\mathscr{C}$ of confidentiality constraints iff the following conditions are satisfied:
  - $\forall F \in \mathscr{F}, \forall c \in \mathscr{C} : c \not\subseteq F$ (each individual fragment satisfies the constraints)
  - $\forall F_i, F_j \in \mathscr{F}, i \neq j : F_i \cap F_j = \emptyset$ (fragments do not have attributes in common)

Multiple fragments (3)

- Each fragment F is mapped into a physical fragment containing:
  - all the attributes in F in the clear
  - all the other attributes of R encrypted (a salt is applied on each encryption)
- Fragment $F_i = \{A_{i_1}, \ldots, A_{i_n}\}$ of R mapped to physical fragment F_i^e(<u>salt</u>,enc,$A_{i_1}, \ldots, A_{i_n}$):
  - each $t \in r$ over R is mapped into a tuple $t^e \in f_i^e$ where f_i^e is a relation over F_i^e and:
    - t^e[*enc*] = $E_k(t[R - F_i] \otimes t^e$[*salt*])
    - $t^e[A_{i_j}] = t[A_{i_j}]$, for $j = 1, \ldots, n$

Multiple fragments – Example (1)

MEDICALDATA

SSN	**Name**	**DoB**	**Zip**	**Illness**	**Physician**
123-45-6789	Nancy	65/12/07	94142	hypertension	M. White
987-65-4321	Ned	73/01/05	94141	gastritis	D. Warren
963-85-2741	Nell	86/03/31	94139	flu	M. White
147-85-2369	Nick	90/07/19	94139	asthma	D. Warren

c_0= {SSN}
c_1= {Name, DoB}
c_2= {Name, Zip}
c_3= {Name, Illness}
c_4= {Name, Physician}
c_5= {DoB, Zip, Illness}
c_6= {DoB, Zip, Physician}

Multiple fragments – Example (1)

MEDICALDATA

SSN	Name	DoB	Zip	Illness	Physician
123-45-6789	Nancy	65/12/07	94142	hypertension	M. White
987-65-4321	Ned	73/01/05	94141	gastritis	D. Warren
963-85-2741	Nell	86/03/31	94139	flu	M. White
147-85-2369	Nick	90/07/19	94139	asthma	D. Warren

c_0= {SSN}
c_1= {Name, DoB}
c_2= {Name, Zip}
c_3= {Name, Illness}
c_4= {Name, Physician}
c_5= {DoB, Zip, Illness}
c_6= {DoB, Zip, Physician}

F_1

salt	enc	Name
s_1	α	Nancy
s_2	β	Ned
s_3	γ	Nell
s_4	δ	Nick

F_2

salt	enc	DoB	Zip
s_5	ε	65/12/07	94142
s_6	ζ	73/01/05	94141
s_7	η	86/03/31	94139
s_8	θ	90/07/19	94139

F_3

salt	enc	Illness	Physician
s_9	ι	hypertension	M. White
s_{10}	κ	gastritis	D. Warren
s_{11}	λ	flu	M. White
s_{12}	μ	asthma	D. Warren

Executing queries on fragments

- Every physical fragment of R contains all the attributes of R
  $\Longrightarrow$ no more than one fragment needs to be accessed to respond to a query
- If the query involves an encrypted attribute, an additional query may need to be executed by the client

Original query on R	**Translation over fragment F_3^e**
Q :=SELECT SSN, Name FROM MedicalData WHERE (Illness=‘gastritis’ OR Illness=‘asthma’) AND Physician=‘D. Warren’ AND Zip=‘94141’	Q^3 :=SELECT salt, enc FROM F_3^e WHERE (Illness=‘gastritis’ OR Illness=‘asthma’) AND Physician=‘D. Warren’ Q' := SELECT SSN, Name FROM *Decrypt*(Q^3, *Key*) WHERE Zip=‘94141’

Optimization criteria

- Goal: find a fragmentation that makes query execution efficient

- The fragmentation process can then take into consideration different optimization criteria:
  - number of fragments [CDFJPS-07]
  - affinity among attributes [CDFJPS-10]
  - query workload [CDFJPS-09a]

- All criteria obey maximal visibility
  - only attributes that appear in singleton constraints (sensitive attributes) are encrypted
  - all attributes that are not sensitive appear in the clear in one fragment

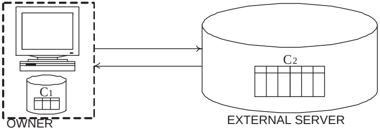
Departing from Encryption: Keep a Few

Keep a few

Basic idea:

- encryption makes query execution more expensive and not always possible
- encryption brings overhead of key management

$\Longrightarrow$ Depart from encryption by involving the owner as a trusted party to maintain a limited amount of data



- $C_1 \cup C_2 = R$

Fragmentation

Given:

- $R(A_1,\ldots,A_n)$: relation schema
- $\mathscr{C}=\{c_1,\ldots,c_m\}$: confidentiality constraints over R

Determine a fragmentation $\mathscr{F}=\langle F_o,F_s\rangle$ for R, where F_o is stored at the owner and F_s is stored at a storage server, and

- $F_o \cup F_s = R$ (completeness)
- $\forall c \in \mathscr{C}, c \not\subseteq F_s$ (confidentiality)
- $F_o \cap F_s = \emptyset$ (non-redundancy) /* can be relaxed */

At the physical level F_o and F_s have a common attribute (additional tid or non-sensitive key attribute) to guarantee lossless join

Fragmentation – Example

PATIENT

<u>**SSN**</u>	**Name**	**DoB**	**Race**	**Job**	**Illness**	**Treatment**	**HDate**
123-45-6789	Nancy	65/12/07	white	waiter	hypertension	ace	09/01/02
987-65-4321	Ned	73/01/05	black	nurse	gastritis	antibiotics	09/01/06
963-85-2741	Nell	86/03/31	red	banker	flu	aspirin	09/01/08
147-85-2369	Nick	90/07/19	asian	waiter	asthma	anti-inflammatory	09/01/10

c_0={SSN}
c_1={Name,Illness}
c_2={Name,Treatment}
c_3={DoB,Race,Illness}
c_4={DoB,Race,Treatment}
c_5={Job,Illness}

Fragmentation – Example

PATIENT

<u>SSN</u>	Name	DoB	Race	Job	Illness	Treatment	HDate
123-45-6789	Nancy	65/12/07	white	waiter	hypertension	ace	09/01/02
987-65-4321	Ned	73/01/05	black	nurse	gastritis	antibiotics	09/01/06
963-85-2741	Nell	86/03/31	red	banker	flu	aspirin	09/01/08
147-85-2369	Nick	90/07/19	asian	waiter	asthma	anti-inflammatory	09/01/10

c_0 ={SSN}
c_1 ={Name,Illness}
c_2 ={Name,Treatment}
c_3 ={DoB,Race,Illness}
c_4 ={DoB,Race,Treatment}
c_5 ={Job,Illness}

F_o

<u>tid</u>	SSN	Illness	Treatment
1	123-45-6789	hypertension	ace
2	987-65-4321	gastritis	antibiotics
3	963-85-2741	flu	aspirin
4	147-85-2369	asthma	anti-inflammatory

F_s

<u>tid</u>	Name	DoB	Race	Job	HDate
1	Nancy	65/12/07	white	waiter	09/01/02
2	Ned	73/01/05	black	nurse	09/01/06
3	Nell	86/03/31	red	banker	09/01/08
4	Nick	90/07/19	asian	waiter	09/01/10

Query evaluation

- Queries are formulated on R, therefore need to be translated into equivalent queries on F_o and/or F_s
- Queries of the form: SELECT A FROM R WHERE C where C is a conjunction of basic conditions
  - C_o: conditions that involve only attributes stored at the client
  - C_s: conditions that involve only attributes stored at the sever
  - C_{so}: conditions that involve attributes stored at the client and attributes stored at the server

Query evaluation – Example

- F_o={SSN,Illness,Treatment}, F_s={Name,DoB,Race,Job,HDate}
- q = SELECT SSN, DoB
  FROM Patient
  WHERE (Treatment="antibiotic")
  AND (Job="nurse")
  AND (Name=Illness)

- The conditions in the WHERE clause are split as follows
  - $C_o = \{\text{Treatment} = \text{"antibiotic"}\}$
  - $C_s = \{\text{Job} = \text{"nurse"}\}$
  - $C_{so} = \{\text{Name} = \text{Illness}\}$

Query evaluation strategies

Server-Client strategy

- server: evaluate C_s and return result to client
- client: receive result from server and join it with F_o
- client: evaluate C_o and C_{so} on the joined relation

Client-Server strategy

- client: evaluate C_o and send tid of tuples in result to server
- server: join input with F_s, evaluate C_s, and return result to client
- client: join result from server with F_o and evaluate C_{so}

Server-client strategy – Example

q = SELECT SSN, DoB
FROM Patient
WHERE (Treatment = "antibiotic")
AND (Job = "nurse")
AND (Name = Illness)

C_o={Treatment = "antibiotic"}
C_s={Job = "nurse"}
C_{so}={Name $=$ Illness}

q_s = SELECT tid,Name,DoB
FROM F_s
WHERE Job = "nurse"

q_{so} = SELECT SSN, DoB
FROM F_o JOIN r_s
ON F_o.tid=r_s.tid
WHERE (Treatment = "antibiotic") AND (Name = Illness)

Client-server strategy – Example

q = SELECT SSN, DoB
FROM Patient
WHERE (Treatment = “antibiotic”)
AND (Job = “nurse”)
AND (Name = Illness)

C_o={Treatment = “antibiotic”}
C_s={Job = “nurse”}
C_{so}={Name = Illness}

q_o = SELECT tid
FROM F_o
WHERE Treatment = “antibiotic”

q_s = SELECT tid,Name,DoB
FROM F_s JOIN r_o ON F_s.tid=r_o.tid
WHERE Job = “nurse”

q_{so} = SELECT SSN, DoB
FROM F_o JOIN r_s ON F_o.tid=r_s.tid
WHERE Name = Illness

Server-client vs client-server strategies

- If the storage server knows or can infer the query:
  - Client-Server leaks information: the server infers that some tuples are associated with values that satisfy C_o
- If the storage server does not know and cannot infer the query:
  - Server-Client and Client-Server strategies can be adopted without privacy violations
  - possible strategy based on performances: evaluate most selective conditions first

Minimal fragmentation

- The goal is to minimize the owner's workload due to the management of F_o
- Weight function w takes a pair $\langle F_o, F_s \rangle$ as input and returns the owner's workload (i.e., storage and/or computational load)
- A fragmentation $\mathscr{F} = \langle F_o, F_s \rangle$ is minimal iff:
  1. $\mathscr{F}$ is correct (i.e., it satisfies the completeness, confidentiality, and non-redundancy properties)
  2. $\nexists \mathscr{F}'$ such that $w(\mathscr{F}') < w(\mathscr{F})$ and $\mathscr{F}'$ is correct

Fragmentation metrics

Different metrics could be applied splitting the attributes between F_o and F_s, such as minimizing:

- storage
  - number of attributes in F_o (*Min-Attr*)
  - size of attributes in F_o (*Min-Size*)
- computation/traffic
  - number of queries in which the owner needs to be involved (*Min-Query*)
  - number of conditions within queries in which the owner needs to be involved (*Min-Cond*)

The metrics to be applied may depend on the information available

Data and workload information – Example

PATIENT(SSN,Name,DoB,Race,Job,Illness,Treatment,HDate)

A	$size(A)$
SSN	9
Name	20
DoB	8
Race	5
Job	18
Illness	15
Treatment	40
HDate	8

q	$freq(q)$	$Attr(q)$	$Cond(q)$
q_1	5	DoB, Illness	$\langle$Dob$\rangle$, $\langle$Illness$\rangle$
q_2	4	Race, Illness	$\langle$Race$\rangle$, $\langle$Illness$\rangle$
q_3	10	Job, Illness	$\langle$Job$\rangle$, $\langle$Illness$\rangle$
q_4	1	Illness, Treatment	$\langle$Illness$\rangle$, $\langle$Treatment$\rangle$
q_5	7	Illness	$\langle$Illness$\rangle$
q_6	7	DoB, HDate, Treatment	$\langle$DoB,HDate$\rangle$, $\langle$Treatment$\rangle$
q_7	1	SSN, Name	$\langle$SSN$\rangle$, $\langle$Name$\rangle$

Weight metrics and minimization problems (1)

- Min-Attr. Only the relation schema (set of attributes) and the confidentiality constraints are known

  $\Longrightarrow$ minimize the number of the attributes in F_o

  - $w_a(\mathscr{F})=card(F_o)$

- Min-Size. The relation schema (set of attributes), the confidentiality constraints, and the size of each attribute are known

  $\Longrightarrow$ minimize the physical size of F_o

  - $w_s(\mathscr{F})=\sum_{A\in F_o} size(A)$

Weight metrics and minimization problems (2)

- Min-Query. The relation schema (set of attributes), the confidentiality constraints, and a representative profile of the expected query workload are known

  Query workload profile:
  $\mathcal{Q}=\{(q_1, freq(q_1), Attr(q_1)), \ldots, (q_l, freq(q_l) Attr(q_l))\}$
  - $q_1, \ldots, q_l$ queries to be executed
  - $freq(q_i)$ expected execution frequency of q_i
  - $Attr(q_i)$ attributes appearing in the WHERE clause of q_i

  $\Longrightarrow$ minimize the number of query executions that require processing at the owner
  - $w_q(\mathcal{F})=\sum_{q\in\mathcal{Q}} freq(q)$ $s.t.$ $Attr(q)\cap F_o \neq \emptyset$

Weight metrics and minimization problems (3)

- Min-Cond. The relation schema (set of attributes), the confidentiality constraints, and a complete profile (conditions in each query of the form a_i op v or a_i op a_j) of the expected query workload are known

  Query workload profile:
  $\mathcal{Q}$={$(q_1, freq(q_1), Cond(q_1)), \ldots, (q_l, freq(q_l) Cond(q_l))$}
  - $q_1, \ldots, q_l$ queries to be executed
  - $freq(q_i)$ expected execution frequency of q_i
  - $Cond(q_i)$ set of conditions in the WHERE clause of query q_i; each condition is represented as a single attribute or a pair of attributes

  $\Longrightarrow$ minimize the number of conditions that require processing at the owner
  - $w_c(\mathcal{F})=\sum_{cnd \in Cond(\mathcal{Q})} freq(cnd)$ s.t. $cnd \cap F_o \neq \emptyset$, where $Cond(\mathcal{Q})$ denotes the set of all conditions of queries in $\mathcal{Q}$, and $freq(cnd)$ is the overall frequency of *cnd*

Modeling of the minimization problems

- All the problems of minimizing storage or computation/traffic aim at identifying a hitting set
  - F_o must contain at least an attribute for each constraint
- Different metrics correspond to different criteria according to which the hitting set should be minimized
- We represent all criteria with a uniform model based on:
  - target set: elements (i.e., attributes, queries, or conditions) with respect to which the minimization problem is defined
  - weight function: function that associates a weight with each target element
  - weight of a set of attributes: sum of the weights of the targets intersecting with the set

$\Longrightarrow$ compute the hitting set of attributes with minimum weight

Example (1)

PATIENT(SSN,Name,DoB,Race,Job,Illness,Treatment,HDate)

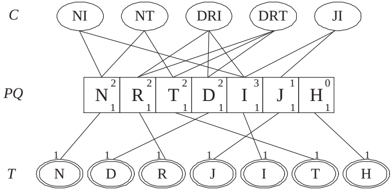
Confidentiality constraints

c_0={SSN}
c_1={Name,Illness}
c_2={Name,Treatment}
c_3={DoB,Race,Illness}
c_4={DoB,Race,Treatment}
c_5={Job,Illness}

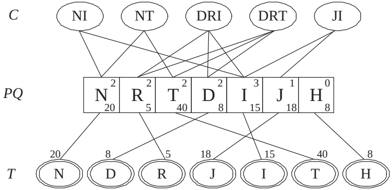
A	*size*(A)
SSN	9
Name	20
DoB	8
Race	5
Job	18
Illness	15
Treatment	40
HDate	8

q	*freq*(q)	*Attr*(q)	*Cond*(q)
q_1	5	DoB, Illness	⟨Dob⟩, ⟨Illness⟩
q_2	4	Race, Illness	⟨Race⟩, ⟨Illness⟩
q_3	10	Job, Illness	⟨Job⟩, ⟨Illness⟩
q_4	1	Illness, Treatment	⟨Illness⟩, ⟨Treatment⟩
q_5	7	Illness	⟨Illness⟩
q_6	7	DoB, HDate, Treatment	⟨DoB,HDate⟩, ⟨Treatment⟩
q_7	1	SSN, Name	⟨SSN⟩, ⟨Name⟩

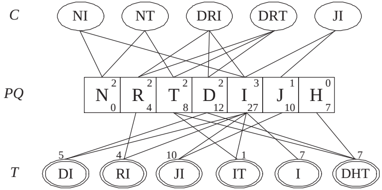
Example (2)



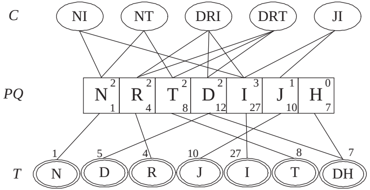
Min-Attr



Min-Size

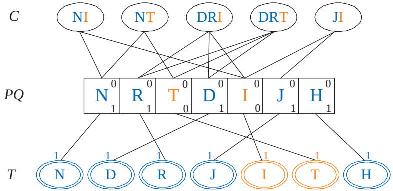


Min-Query

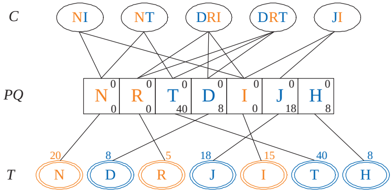


Min-Cond

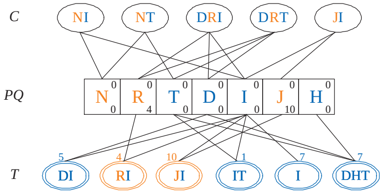
Example (3)



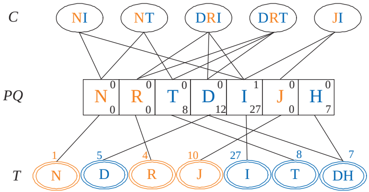
Min-Attr (2)



Min-Size (40)



Min-Query (14)



Min-Cond (15)

Publishing obfuscated associations

Motivation

- Sensitive associations among data may need to be protected, while allowing execution of certain queries
  - e.g., the set of products available in a pharmacy and the set of customers may be of public knowledge; allow retrieving the average number of products purchased by customers while protecting the association between a particular customer and a particular product

- Possible solutions:
  - [CSYZ-08] exploits a graphical representation of sensitive associations and masks the mapping from entities to nodes of the graph while preserving the graph structure
  - [DFJPS-10b] exploits fragmentation for enforcing confidentiality constraints and visibility requirements and publishes a sanitized form of associations

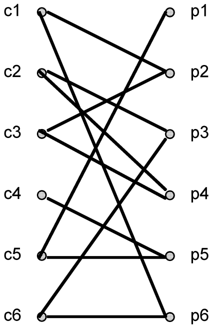
Anonymizing Bipartite Graph

Private associations – Example [CSYZ-08]

Customer	State
c1	NJ
c2	NC
c3	CA
c4	NJ
c5	NC
c6	CA

Product	Avail
p1	Rx
p2	OTC
p3	OTC
p4	OTC
p5	Rx
p6	OTC

Customer	Product
c1	p2
c1	p6
c2	p3
c2	p4
c3	p2
c3	p4
c4	p5
c5	p1
c5	p5
c6	p3
c6	p6



Problem statement

Publish anonymized and useful version of bipartite graph in such a way that:

- a broad class of queries can be answered accurately
  - Type 0 - Graph structure only. E.g., what is the average number of products purchased by customers?
  - Type 1 - Attribute predicate on one side only. E.g., what is the average number of products purchased by NJ customers?
  - Type 2 - Attribute predicate on both side. E.g., what is the average number of OTC products purchased by NJ customers?
- privacy of the specific associations is preserved

(k,l) grouping

Basic idea: preserve the graph structure but permute mapping from entities to nodes

(k,l) grouping of bipartite graph $G = (V, W, E)$

- Partition V (W, resp.) into non-intersecting subsets of size $\geq$ k (l, resp.)

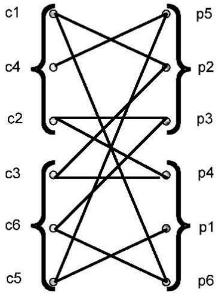
- Publish edges E' that are isomorphic to E, where mapping from E to E' is anonymized based on partitions of V and W

(3,3) grouping – Example (1)

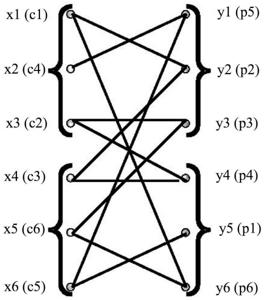
Customer	State
c1	NJ
c2	NC
c3	CA
c4	NJ
c5	NC
c6	CA

Product	Avail
p1	Rx
p2	OTC
p3	OTC
p4	OTC
p5	Rx
p6	OTC

Customer	Product
c1	p2
c1	p6
c2	p3
c2	p4
c3	p2
c3	p4
c4	p5
c5	p1
c5	p5
c6	p3
c6	p6



(3,3) grouping – Example (2)



x1	y2
x1	y6
x2	y1
x3	y3
x3	y4
x4	y2
x4	y4
x5	y3
x5	y6
x6	y1
x6	y5

E'

Customer	**Group**
c1	CG1
c2	CG1
c3	CG2
c4	CG1
c5	CG2
c6	CG2

H_V

Product	**Group**
p1	PG2
p2	PG1
p3	PG1
p4	PG2
p5	PG1
p6	PG2

H_W

X-node	**Group**
x1	CG1
x2	CG1
x3	CG1
x4	CG2
x5	CG2
x6	CG2

R_V

Y-node	**Group**
y1	PG1
y2	PG1
y3	PG1
y4	PG2
y5	PG2
y6	PG2

R_W

Safe groupings

- There are different ways of creating a (k,l) grouping but not all the resulting groupings offer the same level of privacy (e.g., local clique)

  $\Longrightarrow$ safe (k,l) groupings: nodes in the same group of V are not connected to a same node in W

- The computation of a safe grouping can be hard even for small values of k and l
  - The computation of a safe, strict (3,3)-grouping is NP-hard (reduction from partitioning a graph into triangles)

- Greedy algorithm that iteratively adds a node to a group with fewer than k nodes, if it is safe (it creates a new group if such insertion is not possible)

- The algorithm works when bipartite graph is sparse enough

Fragments and Loose Associations

Data publication [DFJPS-10b]

- Fragmentation can also be used to protect sensitive associations in data publishing
  $\Longrightarrow$ publish/release to external parties only views (fragments) that do not expose sensitive associations

- To increase the utility of published information fragments could be coupled with some associations in sanitized form
  $\Longrightarrow$ loose associations: associations among groups of values (in contrast to specific values)

Confidentiality constraints

As already discussed....

- Sets of attributes such that the (joint) visibility of values of the attributes in the sets should be protected
- They permit to express different requirements
  - sensitive attributes: the values of some attributes are considered sensitive and should not be visible
  - sensitive associations: the associations among values of given attributes are sensitive and should not be visible

Confidentiality constraints – Example

SSN	Patient	Birth	City	Illness	Doctor
123-45-6789	Page	56/12/9	Rome	diabetes	David
987-65-4321	Patrick	53/3/19	Paris	gastritis	Daisy
963-85-2741	Patty	58/5/18	Oslo	flu	Damian
147-85-2369	Paul	53/12/9	Oslo	asthma	Daniel
782-90-5280	Pearl	56/12/9	Rome	gastritis	Dorothy
816-52-7272	Philip	57/6/25	Paris	obesity	Drew
872-62-5178	Phoebe	53/12/1	NY	measles	Dennis
712-81-7618	Piers	60/7/25	Rome	diabetes	Daisy

- SSN is sensitive
  - {SSN}
- Illness and Doctor are private of an individual and cannot be stored in association with the name of the patient
  - {Patient, Illness}, {Patient, Doctor}
- {Birth,City} can work as quasi-identifier
  - {Birth, City, Illness}, {Birth, City, Doctor}

Visibility requirements

- Monotonic Boolean formulas over attributes, representing views over data (negations are captured by confidentiality constraints)
- They permit to express different requirements
  - visible attributes: some attributes should be visible
  - visible associations: the association among values of given attributes should be visible
  - alternative views: at least one of the specified views should be visible

Visibility requirements – Example

SSN	Patient	Birth	City	Illness	Doctor
123-45-6789	Page	56/12/9	Rome	diabetes	David
987-65-4321	Patrick	53/3/19	Paris	gastritis	Daisy
963-85-2741	Patty	58/5/18	Oslo	flu	Damian
147-85-2369	Paul	53/12/9	Oslo	asthma	Daniel
782-90-5280	Pearl	56/12/9	Rome	gastritis	Dorothy
816-52-7272	Philip	57/6/25	Paris	obesity	Drew
872-62-5178	Phoebe	53/12/1	NY	measles	Dennis
712-81-7618	Piers	60/7/25	Rome	diabetes	Daisy

- Either names of Patients or their Cities should be released
  - Patient $\vee$ City
- Either Birth dates and Cities of patients in association should be released or the SSN of patients should be released
  - (Birth $\wedge$ City)$\vee$ SSN
- Illnesses and Doctors, as well as their association, should be released
  - Illness $\wedge$ Doctor

Fragmentation

Fragmentation can be applied to satisfy both confidentiality constraints and visibility requirements

- Publish/release to external parties only fragments that
  - do not include sensitive attributes and sensitive associations
  - include the requested attributes and/or associations (all the requirements should be satisfied, not necessarily by a single fragment)

Fragmentation – Example

SSN	Patient	Birth	City	Illness	Doctor
123-45-6789	Page	56/12/9	Rome	diabetes	David
987-65-4321	Patrick	53/3/19	Paris	gastritis	Daisy
963-85-2741	Patty	58/5/18	Oslo	flu	Damian
147-85-2369	Paul	53/12/9	Oslo	asthma	Daniel
782-90-5280	Pearl	56/12/9	Rome	gastritis	Dorothy
816-52-7272	Philip	57/6/25	Paris	obesity	Drew
872-62-5178	Phoebe	53/12/1	NY	measles	Dennis
712-81-7618	Piers	60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

v_1=Patient $\vee$ City
v_2=(Birth $\wedge$ City)$\vee$ SSN
v_3=Illness $\wedge$ Doctor

Fragmentation – Example

SSN	Patient	Birth	City	Illness	Doctor
123-45-6789	Page	56/12/9	Rome	diabetes	David
987-65-4321	Patrick	53/3/19	Paris	gastritis	Daisy
963-85-2741	Patty	58/5/18	Oslo	flu	Damian
147-85-2369	Paul	53/12/9	Oslo	asthma	Daniel
782-90-5280	Pearl	56/12/9	Rome	gastritis	Dorothy
816-52-7272	Philip	57/6/25	Paris	obesity	Drew
872-62-5178	Phoebe	53/12/1	NY	measles	Dennis
712-81-7618	Piers	60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

v_1=Patient $\vee$ City
v_2=(Birth $\wedge$ City)$\vee$ SSN
v_3=Illness $\wedge$ Doctor

F_l

Birth	City
56/12/9	Rome
53/3/19	Paris
58/5/18	Oslo
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
diabetes	David
gastritis	Daisy
flu	Damian
asthma	Daniel
gastritis	Dorothy
obesity	Drew
measles	Dennis
diabetes	Daisy

Correct and minimal fragmentation

- A fragmentation is correct if
  - each confidentiality constraint is satisfied by all fragments
  - each visibility requirement is satisfied by at least a fragment
  - fragments do not have attributes in common (to prevent joins on fragments to retrieve associations)
- A correct fragmentation is minimal if
  - the number of fragments is minimum (i.e., any other correct fragmentation has an equal or greater number of fragments)
- The Min-CF problem of computing a correct and minimal fragmentation is NP-hard

Computing a correct and minimal fragmentation

A SAT solver can efficiently solve the Min-CF problem

- An instance of the Min-CF problem is translated into an instance of the SAT problem

- The inputs to the Min-CF problem are interpreted as boolean formulas
  - visibility requirements are already represented as boolean formulas
  - each confidentiality constraint is represented via a boolean formula as a conjunction of the attributes appearing in the constraint

- Iterate the evaluation of a SAT solver, starting with one fragment and increasing fragments by one at each iteration, until a solution is found (solution is guaranteed to be minimal)

Publishing loose associations (1)

- Fragmentation breaks associations among attributes
- To increase utility of published information, fragments can be coupled with some associations in sanitized form
- A given privacy degree of the association must be guaranteed

$\Longrightarrow$ loose associations: associations among groups of values (in contrast to specific values)

Publishing loose associations (2)

Given two fragments F_l and F_r, a loose association between F_l and F_r

- partitions tuples in the fragments in groups
- provides information on the associations at the group level
- does not permit to exactly reconstruct the original associations among the tuples in the fragments
- provides enriched utility of the published data

Grouping

- Given fragment F_i and its instance f_i, a k-grouping over f_i partitions the tuples in f_i in groups of size greater than or equal to k

  $\Longrightarrow$ each tuple t in f_i is associated with a group identifier $G_i(t)$

- A k-grouping is minimal if it maximizes the number of groups (intuitively, it minimizes the size of the groups)

- (k_l,k_r)-grouping denotes the groupings over two instances f_l and f_r of F_l and F_r

- A (k_l,k_r)-grouping is minimal if both the k_l-grouping and the k_r-grouping are minimal

Minimal (2,2)-grouping – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
56/12/9	Rome
53/3/19	Paris
58/5/18	Oslo
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
diabetes	David
gastritis	Daisy
flu	Damian
asthma	Daniel
gastritis	Dorothy
obesity	Drew
measles	Dennis
diabetes	Daisy

Minimal (2,2)-grouping – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association

- A (k_l,k_r)-grouping induces a group association A among the groups in f_l and f_r

- A group association A over f_l and f_r is a set of pairs of group identifiers such that:
  - A has the same cardinality as the original relation
  - there is a bijective mapping between the original relation and A that associates each tuple in the original relation with a pair $(G_l(l),G_r(r))$ in A, with $l \in f_l$ and $r \in f_r$

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Group association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City	G
53/3/19	Paris	bc1
53/12/9	Oslo	bc1
56/12/9	Rome	bc2
57/6/25	Paris	bc2
58/5/18	Oslo	bc3
56/12/9	Rome	bc3
53/12/1	NY	bc4
60/7/25	Rome	bc4

G_l	G_r
bc1	id1
bc1	id2
bc2	id1
bc2	id3
bc3	id2
bc3	id4
bc4	id3
bc4	id4

F_r

G	Illness	Doctor
id1	gastritis	Daisy
id1	diabetes	David
id2	asthma	Daniel
id2	flu	Damian
id3	obesity	Drew
id3	measles	Dennis
id4	gastritis	Dorothy
id4	diabetes	Daisy

Group association protection

- Duplicates in fragments are maintained (all fragments have the same cardinality as the original relation)
    - fragments may contain tuples that are equal

- Even tuples that are different may have the same values for attributes involved in a confidentiality constraint

- The looseness protection offered by grouping can be compromised

  $\Longrightarrow$ need to control occurrences of the same values

Alikeness

- Two tuples l_i, l_j in f_l (r_i, r_j in f_r) are alike w.r.t. a constraint c, denoted $l_i \simeq_c l_j$ ($r_i \simeq_c r_j$), if
  - $c \subseteq (F_l \cup F_r)$ (c is covered by F_l and F_r)
  - $l_i[c \cap F_l] = l_j[c \cap F_l]$ ($r_i[c \cap F_r] = r_j[c \cap F_r]$)
- Two tuples l_i, l_j in f_l (r_i, r_j in f_r) are alike $l_i \simeq l_j$ ($r_i \simeq r_j$) if they are alike w.r.t. at least a constraint $c \subseteq (F_l \cup F_r)$
- $\simeq_c$ is transitive for any constraint c
- $\simeq$ is not transitive if there are at least two constraints covered by F_l and F_r

Alikeness – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
56/12/9	Rome
53/3/19	Paris
58/5/18	Oslo
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
diabetes	David
gastritis	Daisy
flu	Damian
asthma	Daniel
gastritis	Dorothy
obesity	Drew
measles	Dennis
diabetes	Daisy

Alikeness – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
56/12/9	Rome
53/3/19	Paris
58/5/18	Oslo
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
diabetes	David
gastritis	Daisy
flu	Damian
asthma	Daniel
gastritis	Dorothy
obesity	Drew
measles	Dennis
diabetes	Daisy

$\simeq_{c_4}$

Alikeness – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
56/12/9	Rome
53/3/19	Paris
58/5/18	Oslo
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
diabetes	David
gastritis	Daisy
flu	Damian
asthma	Daniel
gastritis	Dorothy
obesity	Drew
measles	Dennis
diabetes	Daisy

$\simeq_{c_4}$ $\simeq_{c_3}$

Alikeness – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
56/12/9	Rome
53/3/19	Paris
58/5/18	Oslo
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
diabetes	David
gastritis	Daisy
flu	Damian
asthma	Daniel
gastritis	Dorothy
obesity	Drew
measles	Dennis
diabetes	Daisy

$\not\approx$

k-loose association

- A group association is k-loose if every tuple in the group association A indistinguishably corresponds to at least k distinct associations among tuples in the fragments

- A k-loose association is also k'-loose for any $k' \leq k$

- A (k_l, k_r)-grouping induces a minimal group association A if
  - A is k-loose
  - $\nexists$ a (k'_l, k'_r)-grouping inducing a k-loose association s.t. $k'_l \cdot k'_r < k_l \cdot k_r$

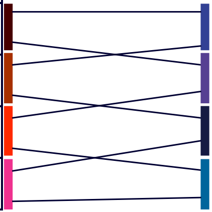
4-loose association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome



F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

4-loose association – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City	G
53/3/19	Paris	bc1
53/12/9	Oslo	bc1
56/12/9	Rome	bc2
57/6/25	Paris	bc2
58/5/18	Oslo	bc3
56/12/9	Rome	bc3
53/12/1	NY	bc4
60/7/25	Rome	bc4

G_l	G_r
bc1	id1
bc1	id2
bc2	id1
bc2	id3
bc3	id2
bc3	id4
bc4	id3
bc4	id4

F_r

G	Illness	Doctor
id1	gastritis	Daisy
id1	diabetes	David
id2	asthma	Daniel
id2	flu	Damian
id3	obesity	Drew
id3	measles	Dennis
id4	gastritis	Dorothy
id4	diabetes	Daisy

Heterogeneity properties

- There is a correspondence between k_l, k_r of the groupings and the degree of k-looseness of the induced group association
  - a (k_l,k_r)-grouping cannot induce a k-loose association for a $k > k_l \cdot k_r$
  - the value $k \leq k_l \cdot k_r$ depends on how groups are defined
- If a (k_l,k_r)-grouping satisfies given heterogeneity properties, the induced group association is k-loose with $k = k_l \cdot k_r$
  - group heterogeneity
  - association heterogeneity
  - deep heterogeneity

Group heterogeneity

No group can contain tuples that are alike with respect to the constraints covered by F_l and F_r

- it ensures diversity of tuples within groups

c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor	
gastritis	Daisy	NO
gastritis	Dorothy	
asthma	Daniel	
flu	Damian	
obesity	Drew	
measles	Dennis	
diabetes	David	NO
diabetes	Daisy	

Group heterogeneity

No group can contain tuples that are alike with respect to the constraints covered by F_l and F_r

- it ensures diversity of tuples within groups

c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Association heterogeneity

No group can be associated twice with another group (the group association cannot contain any duplicate)

- it ensures that for each real tuple in the original relation there are at least $k_l \cdot k_r$ pairs in the group association that may correspond to it

c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

NO

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Association heterogeneity

No group can be associated twice with another group (the group association cannot contain any duplicate)

- it ensures that for each real tuple in the original relation there are at least $k_l \cdot k_r$ pairs in the group association that may correspond to it

c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Deep heterogeneity

No group can be associated with two groups that contain alike tuples

- it ensures that all $k_l \cdot k_r$ pairs in the group association to which each tuple could correspond to contain diverse values for attributes involved in constraints

c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
56/12/9	Rome
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
53/12/9	Oslo
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
gastritis	Dorothy
flu	Damian
obesity	Drew
measles	Dennis
asthma	Daniel
diabetes	Daisy

NO

Deep heterogeneity

No group can be associated with two groups that contain alike tuples

- it ensures that all $k_l \cdot k_r$ pairs in the group association to which each tuple could correspond to contain diverse values for attributes involved in constraints

c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Flat grouping vs sparse grouping

- A (k_l,k_r)-grouping is
  - flat if either k_l or k_r is equal to 1
  - sparse if both k_l and k_r are different from 1
- Flat grouping resembles k-anonymity and captures at the same time the ℓ-diversity property, but it works on associations and attributes' values are not generalized
- Sparse grouping guarantees larger applicability than flat grouping, with the same level of protection
  (there may exist a sparse grouping providing k-looseness but not a flat grouping)

Flat grouping – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
58/5/18	Oslo
53/12/1	NY
56/12/9	Rome
57/6/25	Paris
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
asthma	Daniel
diabetes	David
flu	Damian
measles	Dennis
gastritis	Dorothy
obesity	Drew
diabetes	Daisy

Sparse grouping – Example

Birth	City	Illness	Doctor
56/12/9	Rome	diabetes	David
53/3/19	Paris	gastritis	Daisy
58/5/18	Oslo	flu	Damian
53/12/9	Oslo	asthma	Daniel
56/12/9	Rome	gastritis	Dorothy
57/6/25	Paris	obesity	Drew
53/12/1	NY	measles	Dennis
60/7/25	Rome	diabetes	Daisy

c_0={SSN}
c_1={Patient,Illness}
c_2={Patient,Doctor}
c_3={Birth,City,Illness}
c_4={Birth,City,Doctor}

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Privacy vs utility

- The publication of loose associations increases data utility
  - makes it possible to evaluate queries more precisely than if only the fragments were published

- Increased utility corresponds to a lower privacy degree

Association exposure

- The exposure of a sensitive association $\langle l[c \cap F_l], r[c \cap F_r]\rangle$, with c a constraint covered by F_l, F_r, can be expressed as the probability of the association to hold in the original relation (given the published information)

- The increased exposure due to the publication of loose associations can be measured as the difference between
  - the probability $P^A(l[c \cap F_l], r[c \cap F_r])$ that the sensitive association $\langle l[c \cap F_l], r[c \cap F_r]\rangle$ appears in the original relation, given f_l, f_r, and A
  - the probability $P(l[c \cap F_l], r[c \cap F_r])$ that the sensitive association $\langle l[c \cap F_l], r[c \cap F_r]\rangle$ appears in the original relation, given f_l and f_r

Exposure without loose association (1)

- Given $l \in f_l$ and $r \in f_r$ the probability $P(l,r)$ that tuple $\langle l,r \rangle$ belongs to the original relation is $1/|f_l| = 1/|f_r|$

Exposure without loose association (1)

- Given $l \in f_l$ and $r \in f_r$ the probability $P(l,r)$ that tuple $\langle l,r \rangle$ belongs to the original relation is $1/|f_l| = 1/|f_r|$

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
		Daisy	David	Daniel	Damian	Drew	Dennis	Dorothy	Daisy
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

Exposure without loose association (2)

- Exposure ($P(l[c \cap F_l], r[c \cap F_r])$) depends on the presence of alike tuples
- Let l_i,l_j be two tuples in f_l s.t. $l_i \simeq_c l_j$, $P(l_i[c \cap F_l], r[c \cap F_r])$ is the composition of the probability that
  - l_i is associated with r
  - l_j is associated with r

$$P(l_i,r) + P(l_j,r) - (P(l_i,r) \cdot P(l_j,r))$$

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
		Daisy	David	Daniel	Damian	Drew	Dennis	Dorothy	Daisy
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
		Daisy	David	Daniel	Damian	Drew	Dennis	Dorothy	Daisy
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

Exposure without loose association – Example

			gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
	53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
$\simeq_{c_3}$	57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
	60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

$$P(\text{56/12/9,Rome,gastritis}) = P(\text{56/12/9,Rome,diabetes}) = \ldots = P(\text{56/12/9,Rome,diabetes}) = \frac{1}{8} + \frac{1}{8} - \left(\frac{1}{8} \cdot \frac{1}{8}\right)$$

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	15/64	15/64	15/64	15/64	15/64	15/64	15/64	15/64
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

P(56/12/9,Rome,gastritis) = P(56/12/9,Rome,diabetes) = $\ldots$ = P(56/12/9,Rome,diabetes) =

$$\frac{1}{8} + \frac{1}{8} - \left(\frac{1}{8} \cdot \frac{1}{8}\right) = \frac{15}{64}$$

Exposure without loose association – Example

$\simeq_{c_3}$

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	15/64	15/64	15/64	15/64	15/64	15/64	15/64	15/64
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

P(53/3/19,Paris,gastritis) = P(53/12/9,Oslo,gastritis) = $\ldots$ = P(60/7/25,Rome,gastritis) =

$$\frac{1}{8}+\frac{1}{8}-\left(\frac{1}{8}\cdot\frac{1}{8}\right)$$

P(56/12/9,Rome,gastritis) = $\frac{15}{64}+\frac{15}{64}-\left(\frac{15}{64}\cdot\frac{15}{64}\right)$

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	diabetes
53/3/19	Paris	15/64	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	15/64	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1695/4096	15/64	15/64	15/64	15/64	15/64	15/64
57/6/25	Paris	15/64	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	15/64	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	15/64	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	15/64	1/8	1/8	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

$$P(\text{53/3/19,Paris,gastritis}) = P(\text{53/12/9,Oslo,gastritis}) = \ldots = P(\text{60/7/25,Rome,gastritis}) =$$
$$\frac{1}{8} + \frac{1}{8} - \left(\frac{1}{8} \cdot \frac{1}{8}\right) = \frac{15}{64}$$
$$P(\text{56/12/9,Rome,gastritis}) = \frac{15}{64} + \frac{15}{64} - \left(\frac{15}{64} \cdot \frac{15}{64}\right) = \frac{1695}{4096}$$

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	diabetes
53/3/19	Paris	15/64	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	15/64	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1695/4096	15/64	15/64	15/64	15/64	15/64	15/64
57/6/25	Paris	15/64	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	15/64	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	15/64	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	15/64	1/8	1/8	1/8	1/8	1/8	1/8

$\simeq_{c_3}$

c_3={Birth,City,Illness}

$$P(\text{53/3/19,Paris,diabetes}) = P(\text{53/12/9,Oslo,diabetes}) = \ldots = P(\text{60/7/25,Rome,diabetes}) = \frac{1}{8}+\frac{1}{8}-\left(\frac{1}{8}\cdot\frac{1}{8}\right)$$

$$P(\text{56/12/9,Rome,diabetes}) = \frac{15}{64}+\frac{15}{64}-\left(\frac{15}{64}\cdot\frac{15}{64}\right)$$

Exposure without loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	15/64	15/64	1/8	1/8	1/8	1/8
53/12/9	Oslo	15/64	15/64	1/8	1/8	1/8	1/8
56/12/9	Rome	1695/4096	1695/4096	15/64	15/64	15/64	15/64
57/6/25	Paris	15/64	15/64	1/8	1/8	1/8	1/8
58/5/18	Oslo	15/64	15/64	1/8	1/8	1/8	1/8
53/12/1	NY	15/64	15/64	1/8	1/8	1/8	1/8
60/7/25	Rome	15/64	15/64	1/8	1/8	1/8	1/8

c_3={Birth,City,Illness}

P(53/3/19,Paris,diabetes) = P(53/12/9,Oslo,diabetes) = $\ldots$ = P(60/7/25,Rome,diabetes) =

$$\frac{1}{8}+\frac{1}{8}-\left(\frac{1}{8}\cdot\frac{1}{8}\right)=\frac{15}{64}$$

P(56/12/9,Rome,diabetes) = $\frac{15}{64}+\frac{15}{64}-\left(\frac{15}{64}\cdot\frac{15}{64}\right)=\frac{1695}{4096}$

Exposure with loose association

- Given $l \in f_l$ and $r \in f_r$ the probability $P^A(l,r)$ that tuple $\langle l,r \rangle$ belongs to the original relation is at most $1/k$

- $P^A(l[c \cap F_l], r[c \cap F_r])$ is evaluated considering the alike $\simeq_c$ relationship
  - let l_i,l_j in f_l s.t. $l_i \simeq_c l_j$, $P^A(l_i[c \cap F_l], r[c \cap F_r])$ is the composition of the probability that
    - l_i is associated with r
    - l_j is associated with r

$$P^A(l_i,r) + P^A(l_j,r) - (P^A(l_i,r) \cdot P^A(l_j,r))$$

Exposure with loose association – Example

		gastritis Daisy	diabetes David	asthma Daniel	flu Damian	obesity Drew	measles Dennis	gastritis Dorothy	diabetes Daisy
53/3/19	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/9	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
57/6/25	Paris	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
58/5/18	Oslo	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
56/12/9	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
53/12/1	NY	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
60/7/25	Rome	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
		Daisy	David	Daniel	Damian	Drew	Dennis	Dorothy	Daisy
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–	–
56/12/9	Rome	1/4	1/4	–	–	1/4	1/4	–	–
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–	–
58/5/18	Oslo	–	–	1/4	1/4	–	–	1/4	1/4
56/12/9	Rome	–	–	1/4	1/4	–	–	1/4	1/4
53/12/1	NY	–	–	–	–	1/4	1/4	1/4	1/4
60/7/25	Rome	–	–	–	–	1/4	1/4	1/4	1/4

F_l

Birth	City
53/3/19	Paris
53/12/9	Oslo
56/12/9	Rome
57/6/25	Paris
58/5/18	Oslo
56/12/9	Rome
53/12/1	NY
60/7/25	Rome

F_r

Illness	Doctor
gastritis	Daisy
diabetes	David
asthma	Daniel
flu	Damian
obesity	Drew
measles	Dennis
gastritis	Dorothy
diabetes	Daisy

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
		Daisy	David	Daniel	Damian	Drew	Dennis	Dorothy	Daisy
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–	–
56/12/9	Rome	1/4	1/4	–	–	1/4	1/4	–	–
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–	–
58/5/18	Oslo	–	–	1/4	1/4	–	–	1/4	1/4
56/12/9	Rome	–	–	1/4	1/4	–	–	1/4	1/4
53/12/1	NY	–	–	–	–	1/4	1/4	1/4	1/4
60/7/25	Rome	–	–	–	–	1/4	1/4	1/4	1/4

c_3={Birth,City,Illness}

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–	–
56/12/9	Rome	1/4	1/4	–	–	1/4	1/4	–	–
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–	–
58/5/18	Oslo	–	–	1/4	1/4	–	–	1/4	1/4
56/12/9	Rome	–	–	1/4	1/4	–	–	1/4	1/4
53/12/1	NY	–	–	–	–	1/4	1/4	1/4	1/4
60/7/25	Rome	–	–	–	–	1/4	1/4	1/4	1/4

$\simeq_{c_3}$

c_3={Birth,City,Illness}

P(56/12/9,Rome,gastritis) = P(56/12/9,Rome,diabetes) = $\ldots$ = P(56/12/9,Rome,diabetes) =

$$\frac{1}{4} + 0 - \left(\frac{1}{4} \cdot 0\right)$$

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–	–
56/12/9	Rome	1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/4
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–	–
58/5/18	Oslo	–	–	1/4	1/4	–	–	1/4	1/4
53/12/1	NY	–	–	–	–	1/4	1/4	1/4	1/4
60/7/25	Rome	–	–	–	–	1/4	1/4	1/4	1/4

c_3={Birth,City,Illness}

P(56/12/9,Rome,gastritis) = P(56/12/9,Rome,diabetes) = $\ldots$ = P(56/12/9,Rome,diabetes) =

$$\frac{1}{4} + 0 - \left(\frac{1}{4} \cdot 0\right) = \frac{1}{4}$$

Exposure with loose association – Example

$\simeq_{c_3}$

		gastritis	diabetes	asthma	flu	obesity	measles	gastritis	diabetes
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–	–
56/12/9	Rome	1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/4
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–	–
58/5/18	Oslo	–	–	1/4	1/4	–	–	1/4	1/4
53/12/1	NY	–	–	–	–	1/4	1/4	1/4	1/4
60/7/25	Rome	–	–	–	–	1/4	1/4	1/4	1/4

c_3={Birth,City,Illness}

P(53/3/19,Paris,gastritis) = P(53/12/9,Oslo,gastritis) = $\ldots$ = P(60/7/25,Rome,gastritis) =

$$\frac{1}{4} + 0 - \left(\frac{1}{4} \cdot 0\right)$$

P(56/12/9,Rome,gastritis) = $\frac{1}{4} + \frac{1}{4} - \left(\frac{1}{4} \cdot \frac{1}{4}\right)$

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	diabetes
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–
56/12/9	Rome	7/16	1/4	1/4	1/4	1/4	1/4	1/4
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–
58/5/18	Oslo	1/4	–	1/4	1/4	–	–	1/4
53/12/1	NY	1/4	–	–	–	1/4	1/4	1/4
60/7/25	Rome	1/4	–	–	–	1/4	1/4	1/4

c_3={Birth,City,Illness}

$$P(\text{53/3/19,Paris,gastritis}) = P(\text{53/12/9,Oslo,gastritis}) = \ldots = P(\text{60/7/25,Rome,gastritis}) = \frac{1}{4} + 0 - \left(\frac{1}{4} \cdot 0\right) = \frac{1}{4}$$

$$P(\text{56/12/9,Rome,gastritis}) = \frac{1}{4} + \frac{1}{4} - \left(\frac{1}{4} \cdot \frac{1}{4}\right) = \frac{7}{16}$$

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles	diabetes
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–	–
56/12/9	Rome	7/16	1/4	1/4	1/4	1/4	1/4	1/4
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4	–
58/5/18	Oslo	1/4	–	1/4	1/4	–	–	1/4
53/12/1	NY	1/4	–	–	–	1/4	1/4	1/4
60/7/25	Rome	1/4	–	–	–	1/4	1/4	1/4

$\simeq_{c_3}$

c_3={Birth,City,Illness}

P(53/3/19,Paris,diabetes) = P(53/12/9,Oslo,diabetes) = $\ldots$ = P(60/7/25,Rome,diabetes) =

$$\frac{1}{4} + 0 - \left(\frac{1}{4} \cdot 0\right)$$

P(56/12/9,Rome,diabetes) = $\frac{1}{4} + \frac{1}{4} - \left(\frac{1}{4} \cdot \frac{1}{4}\right)$

Exposure with loose association – Example

		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–
56/12/9	Rome	7/16	7/16	1/4	1/4	1/4	1/4
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4
58/5/18	Oslo	1/4	1/4	1/4	1/4	–	–
53/12/1	NY	1/4	1/4	–	–	1/4	1/4
60/7/25	Rome	1/4	1/4	–	–	1/4	1/4

c_3={Birth,City,Illness}

P(53/3/19,Paris,diabetes) = P(53/12/9,Oslo,diabetes) = $\ldots$ = P(60/7/25,Rome,diabetes) =

$$\frac{1}{4} + 0 - \left(\frac{1}{4} \cdot 0\right) = \frac{1}{4}$$

P(56/12/9,Rome,diabetes) = $\frac{1}{4} + \frac{1}{4} - \left(\frac{1}{4} \cdot \frac{1}{4}\right) = \frac{7}{16}$

Measuring privacy and utility

- Utility: average over the variation of probability $|P^A(l[c \cap F_l], r[c \cap F_r]) - P(l[c \cap F_l], r[c \cap F_r])|$ for each sensitive association $\langle l[c \cap F_l], r[c \cap F_r] \rangle$
  - measured also in terms of the precision in responding to queries
- Privacy: in addition to the k-loose degree, an exposure threshold $\delta_{\max}$ could be specified
  - given a threshold $\delta_{\max}$, A can be published if $\delta_{\max} \geq (P^A(l[c \cap F_l], r[c \cap F_r]) - P(l[c \cap F_l], r[c \cap F_r]))$ for all sensitive associations $\langle l[c \cap F_l], r[c \cap F_r] \rangle$

Measuring utility – Example

P^A

		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	1/4	1/4	1/4	1/4	–	–
53/12/9	Oslo	1/4	1/4	1/4	1/4	–	–
56/12/9	Rome	7/16	7/16	1/4	1/4	1/4	1/4
57/6/25	Paris	1/4	1/4	–	–	1/4	1/4
58/5/18	Oslo	1/4	1/4	1/4	1/4	–	–
53/12/1	NY	1/4	1/4	–	–	1/4	1/4
60/7/25	Rome	1/4	1/4	–	–	1/4	1/4

P

		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	15/64	15/64	1/8	1/8	1/8	1/8
53/12/9	Oslo	15/64	15/64	1/8	1/8	1/8	1/8
56/12/9	Rome	1695/4096	1695/4096	15/64	15/64	15/64	15/64
57/6/25	Paris	15/64	15/64	1/8	1/8	1/8	1/8
58/5/18	Oslo	15/64	15/64	1/8	1/8	1/8	1/8
53/12/1	NY	15/64	15/64	1/8	1/8	1/8	1/8
60/7/25	Rome	15/64	15/64	1/8	1/8	1/8	1/8

$P^A(l[\text{Birth,City}], r[\text{Illness}]) - P(l[\text{Birth,City}], r[\text{Illness}])$

Measuring utility – Example

$P^A(l[\text{Birth,City}], r[\text{Illness}]) - P(l[\text{Birth,City}], r[\text{Illness}])$

		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	1/64	1/64	1/8	1/8	-1/8	-1/8
53/12/9	Oslo	1/64	1/64	1/8	1/8	-1/8	-1/8
56/12/9	Rome	97/4096	97/4096	1/64	1/64	1/64	1/64
57/6/25	Paris	1/64	1/64	-1/8	-1/8	1/8	1/8
58/5/18	Oslo	1/64	1/64	1/8	1/8	-1/8	-1/8
53/12/1	NY	1/64	1/64	-1/8	-1/8	1/8	1/8
60/7/25	Rome	1/64	1/64	-1/8	-1/8	1/8	1/8

Measuring utility – Example

$P^A(l[\text{Birth,City}],\ r[\text{Illness}]) - P(l[\text{Birth,City}],\ r[\text{Illness}])$

		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	1/64	1/64	1/8	1/8	-1/8	-1/8
53/12/9	Oslo	1/64	1/64	1/8	1/8	-1/8	-1/8
56/12/9	Rome	97/4096	97/4096	1/64	1/64	1/64	1/64
57/6/25	Paris	1/64	1/64	-1/8	-1/8	1/8	1/8
58/5/18	Oslo	1/64	1/64	1/8	1/8	-1/8	-1/8
53/12/1	NY	1/64	1/64	-1/8	-1/8	1/8	1/8
60/7/25	Rome	1/64	1/64	-1/8	-1/8	1/8	1/8

$$\text{Utility} = \frac{\sum_{l,r} |P^A(l[\text{Birth,City}],r[\text{Illness}]) - P(l[\text{Birth,City}],r[\text{Illness}])|}{42} = \frac{13506}{172032}$$

Future directions

- Schema vs. instance constraints and visibility requirements
- Data dependencies not captured by confidentiality constraints
- External knowledge
- Support for different kinds of queries
- Different metrics to measure privacy and utility

References (1)

- [ABGGKMSTX-05] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu, "Two can keep a secret: A distributed architecture for secure database services," in *Proc. of CIDR 2005* Asilomar, CA, USA, January 4-7, 2005.
- [AKSX-04] R. Agrawal, J. Kierman, R. Srikant, Y. Xu, "Order preserving encryption for numeric data," in *Proc. of ACM SIGMOD 2004*, Paris, France, 2004.
- [AT-83] S. Akl, P. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer System*, 1(3):239–248, 1983.
- [AW-89] N.R. Adam, J.C. Wortmann, "Security-control methods for statistical databases: A comparative study," in *ACM Computing Survey*, vol. 21, n. 4, December 1989, pp. 515-556.
- [BA-05] R.J. Bayardo, R. Agrawal, "Data privacy through optimal k-anonymization," in *Proc. of ICDE 2005)*, pp. 217–228, Tokyo, Japan, 2005.
- [CM-08] A. Calì, D. Martinenghi, "Querying data under access limitations," in *IEEE ICDE 2008*, Cancun, Mexico, April 7-12, 2008.

References (2)

- [CDDJPS-05] A. Ceselli, E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, P. Samarati, "Modeling and assessing inference exposure in encrypted databases," in *ACM Transactions on Information and System Security (TISSEC)*, February, 2005.
- [CMW-06] J. Crampton, K. Martin, P. Wild, "On key assignment for hierarchical access control," in *Proc. of CSFW 2006*, Los Alamitos, CA, USA, 2006.
- [CDFS-07a] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, P. Samarati, "k-Anonymity," in *Secure Data Management in Decentralized Systems*, T. Yu and S. Jajodia (eds), Springer-Verlag, 2007.
- [CDFS-07b] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, P. Samarati, "Microdata protection," in *Secure Data Management in Decentralized Systems*, T. Yu, and S. Jajodia (eds.), Springer, 2007.
- [CDFJPS-07] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, "Fragmentation and encryption to enforce privacy in data storage," in *Proc. of ESORICS 2007*, Dresden, Germany, September 24-26, 2007.

References (3)

- [CDFJPS-10] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Combining fragmentation and encryption to protect privacy in data storage,” in *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 3, July, 2010.
- [CDFJPS-09a] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Fragmentation design for efficient query execution over sensitive distributed databases,” in *Proc. of ICDCS 2009*, Montreal, Quebec, Canada, June 22-26, 2009.
- [CDFJPS-09b] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Keep a few: outsourcing data while maintaining confidentiality,” in *Proc. of ESORICS 2009)*, Saint Malo, France, September 21-25, 2009.
- [CLR-07] B-C. Chen, K. LeFevre, R. Ramakrishnan, “Privacy skyline: Privacy with multidimensional adversarial knowledge,” in *Proc. of VLDB 2007*, Vienna, Austria, September 23-28 2007.
- [CMFDX-11] R. Chen, N. Mohammed, B.C.M. Fung, B.C. Desai, L. Xiong, “Publishing set-valued data via differential privacy,” in *PVLDB,* 4(11):1087-1098, September 2011.

References (4)

- [CSYZ-08] G. Cormode, D. Srivastava, T. YU, Q. Zhang, “Anonymizing bipartite graph data using safe groupings,” in *Proc. of VLDB 2008*, Auckland, New Zealand, August 23-28, 2008.
- [DFJPS-12] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Support for write privileges on outsourced data,” in *Proc. of SEC 2012,* Heraklion, Crete, Greece, June 4-6, 2012.
- [DFJPS-10a] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Encryption policies for regulating access to outsourced data,” in *ACM Transactions on Database Systems (TODS),* vol. 35, no. 2, April, 2010.
- [DFJPS-10b] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Fragments and loose associations: Respecting privacy in data publishing,” in *Proc. of the VLDB Endowment*, vol. 3, no. 1, 2010.
- [DFPPS-11] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, P. Samarati, “Efficient and private access to outsourced data,” in *Proc. of ICDCS 2011,* Minneapolis, Minnesota, USA, June 20-24, 2011.
- [DFJPS-11] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Authorization enforcement in distributed query evaluation,” in *Journal of Computer Security*, 2011.

References (5)

- [DFS-12] S. De Capitani di Vimercati, S. Foresti, P. Samarati, “Protecting data in outsourcing scenarios,” in *Handbook on Securing Cyber-Physical Critical Infrastructure*, S.K Das, K. Kant, and N. Zhang (eds.), Morgan Kaufmann, 2012.
- [DFM-04] A. De Santis, A.L. Ferrara, B. Masucci, “Cryptographic key assignment schemes for any access control policy,” *Inf. Process. Lett.*, 92(4):199–205, 2004.
- [DGMS-00] P.T. Devanbu, M. Gertz, C.U. Martel, S.G. Stubblebine, “Authentic third-party data publication,” in *Proc. of DBSec 2000*, Schoorl, The Netherlands, 2000.
- [DWHL-11] B. Ding, M. Winslett, J. Han, Z. Li, “Differentially private data cubes: Optimizing noise sources and consistency,” in *Proc. of SIGMOD 2011*, Athens, Greece, June 2011.
- [D-06] C. Dwork, “Differential privacy,” in *Proc. of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, Venice, Italy, July 2006.
- [D-11] C. Dwork, “Differential privacy,” in *Encyclopedia on Cryptography and Security,* H.C.A. van Tilborg, and S. Jajodia (eds.), Springer, 2011.
- [FWS-08] A. Friedman, R. Wolff, A. Schuster, “Providing k-anonymity in data mining,” in *The VLDB Journal,* 17(4):789-804, July 2008.

References (6)

- [FZ-08] K.B. Frikken, Y. Zhang, “Yet another privacy metric for publishing micro-data,” In *Proc. of WPES 2008*, Alexandria, VA, USA, 2008.
- [FWY-07] B.C.M. Fung, K. Wang, P.S. Yu, “Anonymizing classification data for privacy preservation,” in *IEEE TKDE,* 19(5):711-725, May 2007.
- [GL-08] B. Gedik, L. Liu, “Protecting location privacy with personalized k-anonymity: Architecture and algorithms,” in *IEEE TMC,* 7(1):1-18, January 2008.
- [G-09] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proc. of STOC 2009*, Bethesda, MA, USA, 2009.
- [G-80] E. Gudes, “The design of a cryptography based secure file system,” *IEEE Transactions on Software Engineering*, 6(5):411–420, 1980.
- [GMT-08] A. Gionis, A. Mazza and T. Tassa, “k-Anonymization revisited,” in *Proc. of ICDE 2008,* Cancun, Mexico, 2008.
- [G-06] P. Golle, “Revisiting the uniqueness of simple demographics in the US population,” in *Proc. of WPES 2006,* Alexandria, VA, USA, October 30, 2006.
- [HIML-02] H. Hacigümüs, B. Iyer, S. Mehrotra,S-87 and C. Li, “Executing SQL over encrypted data in the database-service-provider model,” in *Proc. of the ACM SIGMOD 2002*, Madison, Wisconsin, USA, June 2002.

References (7)

- [HMJTW-08] M. Hay, G. Miklau, D. Jensen, D. Towsley, P. Weis, “Resisting structural re-identification in anonymized social networks,” in *PVLDB,* 1(1):102-114, August 2008.
- [HL-90] L. Harn, H. Lin, “A cryptographic key generation scheme for multilevel data security,” *Computers and Security*, 9(6):539–546, 1990.
- [HLMJ-09] M. Hay, C. Li, G. Miklau, D. Jensen, “Accurate estimation of the degree distribution of private networks,” in *Proc. of ICDM 2009,* Miami, FL, USA, December 2009.
- [HY-03] M. Hwang, W. Yang, “Controlling access in large partially ordered hierarchies using cryptographic keys,” *The Journal of Systems and Software*, 67(2):99–107, 2003.
- [HR-11] S.-S. Ho, S. Ruan, “Differential privacy for location pattern mining,” in *Proc. of SPRINGL 2011,* Chicago, IL, USA, November 2011.
- [KM-11] D. Kifer, A. Machanavajjhala, “No free lunch in data privacy,” in *Proc. of SIGMOD 2011,* Athens, Greece, June 2011.
- [KM-12] D. Kifer, A. Machanavajjhala, “A rigorous and customizable framework for privacy,” in *Proc. of PODS 2012,* Scottsdale, AZ, USA, May 2012.

References (8)

- [LDR-06] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proc. of the International Conference on Data Engineering (ICDE'06)*, Atlanta, GA, USA, 2006.
- [LLV-07] N. Li, T. Li, and S. Venkatasubramanian. "t-closeness: Privacy beyond k-anonymity and ℓ-diversity," In *Proc. of ICDE 2007)*, Istanbul, Turkey, April 2007.
- [LWL-89] H. Liaw, S. Wang, C. Lei, "On the design of a single-key-lock mechanism based on newton's interpolating polynomial," *IEEE Transaction on Software Engineering*, 15(9):1135–1137, 1989.
- [M-85] S. MacKinnon et al., "An optimal algorithm for assigning cryptographic keys to control access in a hierarchy," *IEEE Transactions on Computers,* 34(9):797–802, 1985.
- [MW-09] D.J. Mir, R.N. Wright, "A differentially private graph estimator," in *Proc. of ICDMW 2009,* Miami, FL, USA, December 2009.
- [MCFY-11] N. Mohammed, R. Chen, B.C.M. Fung, P.S. Yu, "Differentially private data release for data mining," in *Proc. of KDD 2011,* San Diego, CA, USA, August 2011.
- [MGK-06] A. Machanavajjhala, J. Gehrke, D. Kifer "ℓ-diversity: Privacy beyond k-anonymity," in *Proc. of the International Conference on Data Engineering (ICDE'06)*, Atlanta, GA, USA, 2006.

References (9)

- [MKMGH-07] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, J.Y. Halpern, “Worst-case background knowledge for privacy-preserving data publishing,” in *Proc. of ICDE 2007)*, Istanbul, Turkey, April 2007.
- [NAC-07] M.E. Nergiz, M. Atzori, C. Clifton, “Hiding the presence of individuals from shared databases,” in *Proc. of SIGMOD 2007*, Beijing, China, 2007.
- [MNT-06] E. Mykletun, M. Narasimha, G. Tsudik, “Authentication and integrity in outsourced databases,” *ACM Transactions on Storage*, 2(2):107–138, 2006.
- [NT-05] M. Narasimha, G. Tsudik, “DSAC: Integrity for outsourced databases with signature aggregation and chaining,” in *Proc. CIKM 2005*, Bremen, Germany, 2005.
- [PPP-10] B. Palazzi, M. Pizzonia, S. Pucacco, “Query racing: Fast completeness certification of query results,” in *Proc. DBSEC 2010*, Rome, Italy, 2010.
- [RHMS-09] V. Rastogi, M. Hay, G. Miklau, D. Suciu, “Relationship privacy: Output perturbation for queries with joins,” in *Proc. of PODS 2009,* Providence, RI, USA, June-July 2009.
- [S-01] P. Samarati, “Protecting respondents’ identities in microdata release,” in *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, n. 6, November/December 2001, pp. 1010-1027.

References (10)

- [S-05] R. Sion, “Query execution assurance for outsourced databases,” in *Proc. of VLDB 2005*, Trondheim, Norway, 2005.
- [SD-10] P. Samarati, S. De Capitani di Vimercati, “Data protection in outsourcing scenarios: Issues and directions,” in *Proc. of ASIACCS 2010,* Beijing, China, April, 2010.
- [S-87] R. Sandhu, “On some cryptographic solutions for access control in a tree hierarchy,” in *Proc. of the 1987 Fall Joint Computer Conference on Exploring Technology: Today and Tomorrow*, Dallas, TX, USA, 1987.
- [S-88] R. Sandhu, “Cryptographic implementation of a tree hierarchy for access control,” *Information Processing Letters*, 27(2):95–98, 1988.
- [SC-02] V. Shen, T. Chen, “A novel key management scheme based on discrete logarithms and polynomial interpolations,” *Computer and Security*, 21(2):164–171, 2002.
- [TMK-08] M. Terrovitis, N. Mamoulis, P. Kalnis, “Privacy-preserving anonymization of set-valued data,” *Proc. of the VLDB Endowment*, vol. 1, August 2008, pp. 115-125.
- [WF-06] K. Wang, B. Fung, “Anonymizing sequential releases,” in *Proc. of KDD 2006*, Philadelphia, PA, USA, 2006.

References (11)

- [WL-06] H. Wang, Laks V. S. Lakshmanan, "Efficient secure query evaluation over encrypted XML databases," in *Proc. of VLDB 2006*, Seoul, Korea, 2006.
- [WXWF-10] K. Wang, Y. Xu, R. Wong, A. Fu, "Anonymizing temporal data," in *Proc. of ICDM 2010*, Sydney, Australia, 2010.
- [XWG-11] X. Xiao, G.Wang, J. Gehrke, "Differential privacy via wavelet transforms," in *IEEE Trans. on Knowl. Data Eng.,* 23(8): 1200-1214, 2011.
- [XT-06] X. Xiao, Y. Tao, "Personalized privacy preservation," in *Proc. of SIGMOD 2006*, Chicago, IL, USA, 2006.
- [XT-07] X. Xiao, Y. Tao, "m-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proc. of SIGMOD 2007*, Beijing, China, 2007.
- [XWYM-07] M. Xie, H. Wang, J. Yin, X. Meng, "Integrity auditing of outsourced data," in *Proc. VLDB 2007*, Vienna, Austria, 2007.
- [WYPY-08] H. Wang, J. Yin, C. Perng, P.S. Yu, "Dual encryption for query integrity assurance," in *Proc. CIKM 2008*, Napa Valley, USA, 2008.
- [ZHPJTJ-09] B. Zhou, Y. Han, J. Pei, B. Jiang, Y. Tao, Y. Jia, "Continuous privacy preserving publishing of data streams," in *Proc. of EDBT 2009*, Saint Petersburg, Russia, 2009.