Data Fragmentation and Encryption

Pierangela Samarati
Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano
pierangela.samarati@unimi.it

Motivation (1)

- The management of large amount of sensitive information is quite expensive
- Database outsourcing is becoming increasingly popular (Database As a Service) [HIM-02, HIML-02]
  - \+ 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

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

Motivation (2)

- Encryption proposed in DAS makes query evaluation 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

Fragmentation and encryption

- Recent solutions for enforcing privacy requirements couple:
  - encryption together with
  - data fragmentation
- Privacy requirements are represented as a set of confidentiality constraints that capture sensitivity of attributes and associations

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

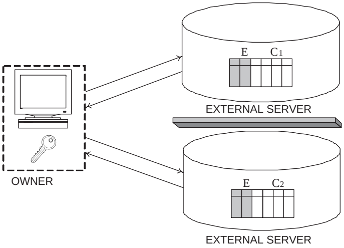
- Data fragmentation
  - Non-communicating pair of servers [ABGGKMSTX-05]
  - Multiple fragments [CDFJPS-07,CDFJPS-10]
  - Departing from encryption: Keep a few [CDFJPS-09]
- Publishing obfuscated associations
  - Anonymizing bipartite graph [CSYZ-08]
  - Fragments and loose associations [DFJPS-10]

Data Fragmentation

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 $c \in \mathscr{C}$, $c \not\subseteq (R_1 - E)$ and $c \not\subseteq (R_2 - E)$

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

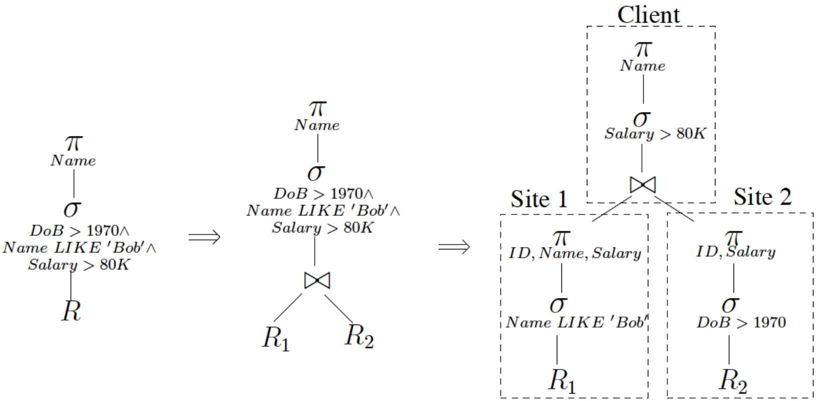
Query execution

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

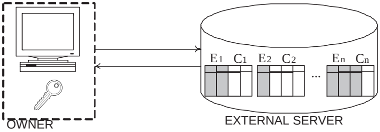
Multiple Fragments

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 [ESORICS'07]
  - affinity among attributes [ACM TISSEC'10]
  - query workload [ICDCS'09]

- 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

Minimal number of fragments

Basic principles:

- avoid excessive fragmentation $\Longrightarrow$ minimal number of fragments

Goal:

- determine a correct fragmentation with the minimal number of fragments
  $\Longrightarrow$ NP-hard problem (minimum hyper-graph coloring problem)

Basic idea of the heuristic:

- define a notion of minimality that can be used for efficiently computing a fragmentation
  - $\mathscr{F}$ is minimal if all the fragmentations that can be obtained from $\mathscr{F}$ by merging any two fragments in $\mathscr{F}$ violate at least one constraint
- iteratively select an attribute with the highest number of non-solved constraints and insert it in an existing fragment if no constraint is violated; create a new fragment otherwise

Minimal number of fragments – Example

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

Confidentiality constraints

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}

Minimal fragmentation $\mathscr{F}$

- F_1 = {Name}
- F_2 = {DoB,Zip}
- F_3 = {Illness,Physician}

Merging any two fragments would violate at least a constraint

Maximum affinity

Basic principles:

- preserve the associations among some attributes
  - e.g., association (Illness,DoB) should be preserved to explore the link between a specific illness and the age of patients
- affinity matrix for representing the advantage of having pairs of attributes in the same fragment

Goal:

- determine a correct fragmentation with maximum affinity (sum of fragments affinity computed as the sum of the affinity of the different pairs of attributes in the fragment)
  $\Longrightarrow$ NP-hard problem (minimum hitting set problem)

Basic idea of the heuristic:

- iteratively combine fragments that have the highest affinity and do not violate any confidentiality constraint

Maximum affinity – Example

MEDICALDATA

<u>SSN</u>	**Name**	**DoB**	**ZIP**	**Illness**	**Physician**
123-45-6789	A. Hellman	81/01/03	94142	hypertension	M. White
987-65-4321	B. Dooley	53/10/07	94141	obesity	D. Warren
246-89-1357	C. McKinley	52/02/12	94139	hypertension	M. White
135-79-2468	D. Ripley	81/01/03	94139	obesity	D. Warren

Confidentiality constraints

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={*n*}
F_2={*d*}
F_3={*z*}
F_4={*i*}
F_5={*p*}

	F_1	F_2	F_3	F_4	F_5
F_1		10	5	25	15
F_2			5	20	30
F_3				10	5
F_4					15
F_5					

	c_1	c_2	c_3	c_4	c_5	c_6
n	×	×	×	×		
d	×				×	×
z		×			×	×
i			×		×	
p				×		×

Maximum affinity – Example

MEDICALDATA

SSN	**Name**	**DoB**	**ZIP**	**Illness**	**Physician**
123-45-6789	A. Hellman	81/01/03	94142	hypertension	M. White
987-65-4321	B. Dooley	53/10/07	94141	obesity	D. Warren
246-89-1357	C. McKinley	52/02/12	94139	hypertension	M. White
135-79-2468	D. Ripley	81/01/03	94139	obesity	D. Warren

Confidentiality constraints

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={n}
F_2={d}
F_3={z}
F_4={i}
F_5={p}

	F_1	F_2	F_3	F_4	F_5
F_1		-1	-1	-1	-1
F_2			5	20	30
F_3				10	5
F_4					15
F_5					

	c_1	c_2	c_3	c_4	c_5	c_6
n	✓	✓	✓	✓		
d	✓				×	×
z		✓			×	×
i			✓		×	
p				✓		×

Maximum affinity – Example

MEDICALDATA

SSN	Name	DoB	ZIP	Illness	Physician
123-45-6789	A. Hellman	81/01/03	94142	hypertension	M. White
987-65-4321	B. Dooley	53/10/07	94141	obesity	D. Warren
246-89-1357	C. McKinley	52/02/12	94139	hypertension	M. White
135-79-2468	D. Ripley	81/01/03	94139	obesity	D. Warren

Confidentiality constraints

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={n}
F_2={d}
F_3={z}
F_4={i}
F_5={p}

	F_1	F_2	F_3	F_4	F_5
F_1		-1	-1	-1	-1
F_2			5	20	**30**
F_3				10	5
F_4					15
F_5					

	c_1	c_2	c_3	c_4	c_5	c_6
n	✓	✓	✓	✓		
d	✓				×	×
z		✓			×	×
i			✓		×	
p				✓		×

Maximum affinity – Example

MEDICALDATA

SSN	**Name**	**DoB**	**ZIP**	**Illness**	**Physician**
123-45-6789	A. Hellman	81/01/03	94142	hypertension	M. White
987-65-4321	B. Dooley	53/10/07	94141	obesity	D. Warren
246-89-1357	C. McKinley	52/02/12	94139	hypertension	M. White
135-79-2468	D. Ripley	81/01/03	94139	obesity	D. Warren

Confidentiality constraints

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={n}
F_2={d,p}
F_3={z}
F_4={i}

	F_1	F_2	F_3	F_4	F_5
F_1		-1	-1	-1	
F_2			-1	**35**	
F_3				10	
F_4					
F_5					

	c_1	c_2	c_3	c_4	c_5	c_6
n	✓	✓	✓	✓		
d	✓				×	✓
z		✓			×	✓
i			✓		×	
p				✓		✓

Maximum affinity – Example

MEDICALDATA

SSN	**Name**	**DoB**	**ZIP**	**Illness**	**Physician**
123-45-6789	A. Hellman	81/01/03	94142	hypertension	M. White
987-65-4321	B. Dooley	53/10/07	94141	obesity	D. Warren
246-89-1357	C. McKinley	52/02/12	94139	hypertension	M. White
135-79-2468	D. Ripley	81/01/03	94139	obesity	D. Warren

Confidentiality constraints

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={*n*}
F_2={*d*,*p*,*i*}
F_3={*z*}

	F_1	F_2	F_3	F_4	F_5
F_1		-1	-1		
F_2			-1		
F_3					
F_4					
F_5					

	c_1	c_2	c_3	c_4	c_5	c_6
n	✓	✓	✓	✓		
d	✓				✓	✓
z		✓			✓	✓
i			✓		✓	
p				✓		✓

Maximum affinity – Example

MEDICALDATA

SSN	Name	DoB	ZIP	Illness	Physician
123-45-6789	A. Hellman	81/01/03	94142	hypertension	M. White
987-65-4321	B. Dooley	53/10/07	94141	obesity	D. Warren
246-89-1357	C. McKinley	52/02/12	94139	hypertension	M. White
135-79-2468	D. Ripley	81/01/03	94139	obesity	D. Warren

Confidentiality constraints

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={n}
F_2={d,p,i}
F_3={z}

	F_1	F_2	F_3	F_4	F_5
F_1		-1	-1		
F_2			-1		
F_3					
F_4					
F_5					

	c_1	c_2	c_3	c_4	c_5	c_6
n	✓	✓	✓	✓		
d	✓				✓	✓
z		✓			✓	✓
i			✓		✓	
p				✓		✓

Maximum affinity fragmentation $\mathscr{F}$ (fragmentation affinity = 65)

Merging any two fragments would violate at least a constraint

Query workload

Basic principles:

- minimize the execution cost of queries
- representative queries (query workload) used as starting point
- query cost model: based on the selectivity of the conditions in queries and queries' frequencies

Goal:

- determine a fragmentation that minimizes the query workload cost $\Longrightarrow$ NP-hard problem (minimum hitting set problem)

Basic idea of the heuristic:

- exploit monotonicity of the query cost function with respect to a dominance relationship among fragmentations
- traversal (checking *ps* solutions at levels multiple of *d*) over a spanning tree of the fragmentation lattice

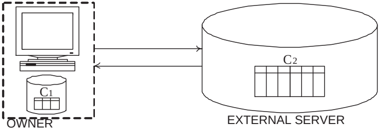
Departing from Encryption: Keep a Few

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}

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
  - $C_o = \{$Treatment = “antibiotic”$\}$
  - $C_s = \{$Job = “nurse”$\}$
  - $C_{so} = \{$Name = 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	⟨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⟩

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:
  $\mathscr{Q}=\{(q_1,freq(q_1),Attr(q_1)),\ldots,(q_l,freq(q_l)Attr(q_l))\}$
  - $q_1,\ldots,q_l$ 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
  - $w_q(\mathscr{F})=\sum_{q\in\mathscr{Q}} freq(q)$ $s.t.$ $Attr(q)\cap F_o\neq\emptyset$

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 (1)

- 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

Modeling of the minimization problems (2)

Problem	**Target** $\mathscr{T}$	$w(t)\ \forall t \in \mathscr{T}$
Min-Attr	$\{\{A\} \mid A \in R\}$	1
Min-Size	$\{\{A\} \mid A \in R\}$	$size(A)$ s.t. $\{A\}=t$
Min-Query	$\{attr \mid \exists q \in \mathscr{Q},\ Attr(q)=attr\}$	$\sum_{q \in \mathscr{Q}} freq(q)$ s.t. $Attr(q)=t$
Min-Cond	$\{cnd \mid \exists q \in \mathscr{Q},\ cnd \in Cond(q)\}$	$freq(cnd)$ s.t. $cnd=t$

Weighted Minimum Target Hitting Set Problem (WMTHSP). Given a finite set A, a set C of subsets of A, a set $\mathscr{T}$ (target) of subsets of A, and a weight function $w: \mathscr{T} \rightarrow \mathbb{R}^+$, determine a subset S of A such that:

1. S is a hitting set of A
2. $\nexists S'$ such that S' is a hitting set of A and $\sum_{t \in \mathscr{T}, t \cap S' \neq \emptyset} w(t) < \sum_{t \in \mathscr{T}, t \cap S \neq \emptyset} w(t)$

Modeling of the minimization problems (3)

- The Minimum Hitting Set Problem can be reduced to the WMTHSP
  - $\mathscr{T} = \{A_1, \ldots, A_n\}$; $w(\{A_i\}) = 1$, $i = 1, \ldots, n$
  - minimizing $\sum_{t \in \mathscr{T}, t \cap S \neq \emptyset} w(t)$ is equivalent to minimizing the cardinality of the hitting set S

  $\Longrightarrow$ WMTHSP is NP-hard

- We propose a heuristic algorithm for solving the WMTHSP that:
  - ensures minimality, that is, moving any attribute from F_o to F_s violates at least a constraint
  - has polynomial time complexity in the number of attributes (efficient execution time)
  - provides solutions close to the optimum (from experiments run: optimum was returned in many cases, 14% maximum error observed)

Heuristic algorithm – Input and output

- Input
  - $\mathscr{A}$: set of attributes not appearing in singleton constraints
  - $\mathscr{C}$: set of well defined constraints
  - $\mathscr{T}$: set of targets
  - w: weight function defined on $\mathscr{T}$
- Output
  - $\mathscr{H}$: set of attributes composing, together with those appearing in singleton constraints, F_o
  - F_s is computed as $R \setminus F_o$, obtaining a correct fragmentation

Heuristic algorithm – Data structure

- Priority-queue PQ with an element E for each attribute:
  - $E.A$: attribute
  - $E.C$: pointers to non-satisfied constraints that contain $E.A$
  - $E.T$: pointers to the targets non intersecting $\mathscr{H}$ that contain $E.A$
  - $E.n_c$: number of constraints pointed by $E.C$
  - $E.w$: total weight of targets pointed by $E.T$

  Priority dictated by $E.w/E.n_c$: elements with lower ratio have higher priority

Heuristic algorithm – Example of initialization (1)

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

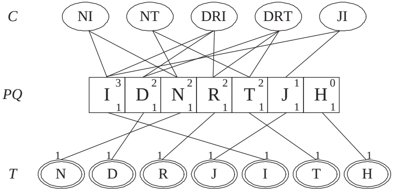
Confidentiality constraints

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}

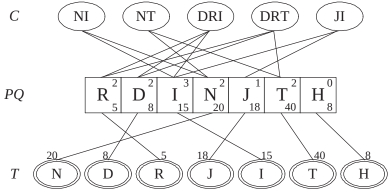
A	*size*(*A*)
SSN	9
Name	20
DoB	8
Race	5
Job	18
Illness	15
Treatment	40
HDate	8

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⟩

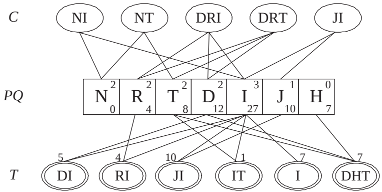
Heuristic algorithm – Example of initialization (2)



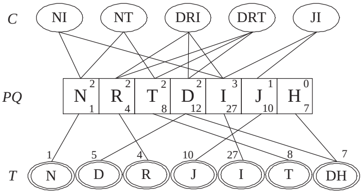
Min-Attr



Min-Size



Min-Query



Min-Cond

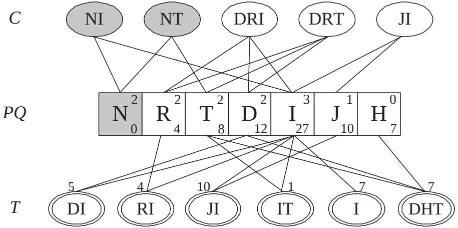
Heuristic algorithm – Working process

- **while** $PQ \neq \emptyset$ and $\exists E \in PQ$, $E.n_c \neq 0$
  - extract the element E with lowest $E.w/E.n_c$ from PQ
  - insert $E.A$ into $\mathscr{H}$
  - $\forall c$ pointed by $E.C$, remove the pointers to c from any element E' in PQ and update $E'.n_c$
  - $\forall t$ pointed by $E.T$, remove the pointers to t from any element E' in PQ and update $E'.w$
  - readjust PQ based on the new values for $E.w/E.n_c$ (*to_be_updated*)

- **for** each $A \in \mathscr{H}$
  - if $\mathscr{H} \setminus \{A\}$ is a hitting set for $\mathscr{C}$, remove A from $\mathscr{H}$

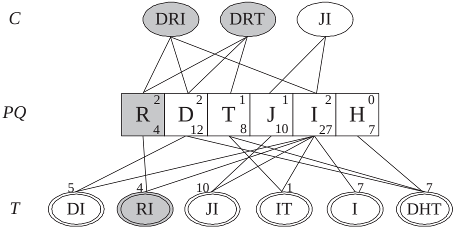
Heuristic algorithm – Example of Min-Query

$\mathscr{H} = \{\}$
$E.A = N$
$E.C = \{NI, NT\}$
$E.T = \{\}$
$to\_be\_updated = \{I, T\}$



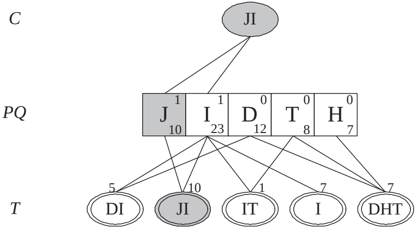
Heuristic algorithm – Example of Min-Query

$\mathscr{H} = \{N\}$
$E.A = R$
$E.C = \{DRI, DRT\}$
$E.T = \{RI\}$
$to\_be\_updated = \{D, I, T\}$



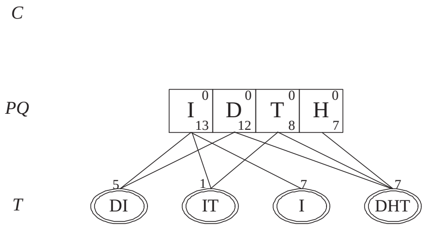
Heuristic algorithm – Example of Min-Query

$\mathscr{H} = \{N, R\}$
$E.A = J$
$E.C = \{JI\}$
$E.T = \{JI\}$
$to\_be\_updated = \{I\}$



Heuristic algorithm – Example of Min-Query

$\mathscr{H} = \{N, R, J\}$

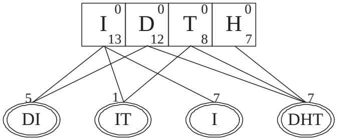


Heuristic algorithm – Example of Min-Query

C

$\mathscr{H} = \{N, R, J\}$

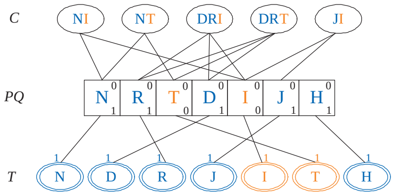
PQ



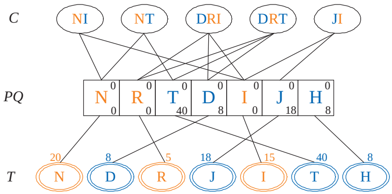
T

F_o={SSN,Name,Race,Job} F_s={Illness,DoB,Treatment,HDate}

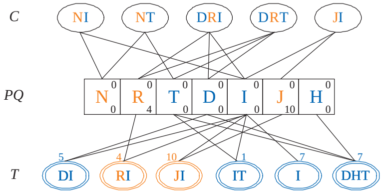
Example of solutions computed by the heuristic algorithm



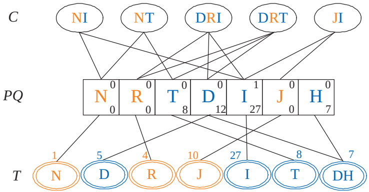
Min-Attr (2)



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-10] exploits fragmentation for enforcing confidentiality constraints and visibility requirements and publishes a sanitized form of associations

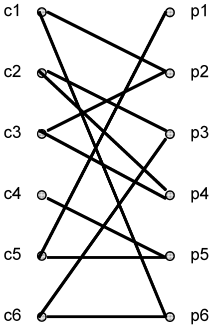
Anonymizing Bipartite Graph

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.)

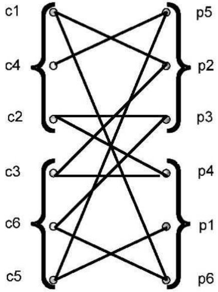
- Publish edges E' that are isomorphic to E, where mapping from E to E' is anonymized based on partitions of V and W

(3,3) grouping – Example (1)

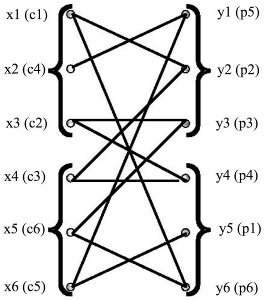
Customer	State
c1	NJ
c2	NC
c3	CA
c4	NJ
c5	NC
c6	CA

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 for 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)
- The authors propose a 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

- 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 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 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

$\not\simeq$

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

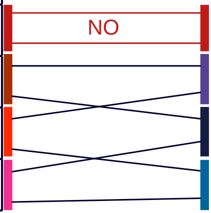
No group can be associated twice with another group (the group association cannot contain any duplicate)

- 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



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
  - it makes it possible to evaluate queries more precisely than if only the fragments were published
- Increased utility corresponds to a greater exposure of information (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
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

$\simeq_{c_3}$

c_3={Birth,City,Illness}

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

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

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

$\simeq_{c_3}$ (over columns gastritis through measles)

		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}

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

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

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

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

P(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

$\simeq_{c_3}$ (over columns diabetes through diabetes)

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

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

P(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	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

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

$\simeq_{c_3}$ (rows 56/12/9 Rome through 56/12/9 Rome)

		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

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

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

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

P(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
			$\simeq_{c_3}$					
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(53/3/19,Paris,diabetes) = P(53/12/9,Oslo,diabetes) = … = 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) = ... = 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

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

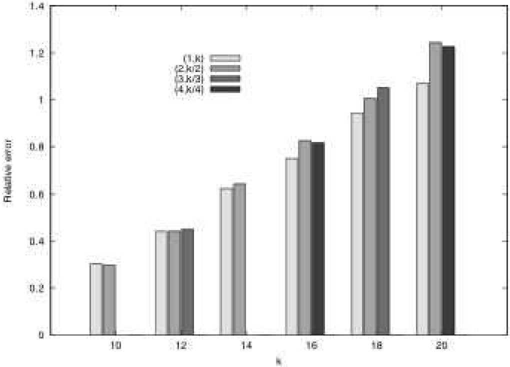
		gastritis	diabetes	asthma	flu	obesity	measles
53/3/19	Paris	1/64	1/64	1/8	1/8	-1/8	-1/8
53/12/9	Oslo	1/64	1/64	1/8	1/8	-1/8	-1/8
56/12/9	Rome	97/4096	97/4096	1/64	1/64	1/64	1/64
57/6/25	Paris	1/64	1/64	-1/8	-1/8	1/8	1/8
58/5/18	Oslo	1/64	1/64	1/8	1/8	-1/8	-1/8
53/12/1	NY	1/64	1/64	-1/8	-1/8	1/8	1/8
60/7/25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

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

Experimental evaluation

- Considered Census data (IPUMS-USA, http://www.ipums.org)
- Evaluated queries of the form
  - SELECT FROM WHERE returning a COUNT aggregation function
  - WHERE condition $\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{m} a_i = v_{i_j})$
- Evaluated precision of queries
- Evaluated impact of k, k_l, and k_r on query precision

Experimental evaluation – results



- Precision in query evaluation progressively decreases as k increases

- The critical parameter in the configuration is the overall privacy degree k, rather than individual values of k_l and k_r

Summary of contributions

- Novel approach to the problem of protecting privacy when publishing data

- Generic setting of the privacy problem that explicitly takes into consideration both privacy needs and visibility requirements

- Definition of loose associations for increasing data utility while preserving a given degree of privacy

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 the 2nd Conference on Innovative Data Systems Research (CIDR 2005)*, Asilomar, California, USA, January 2005.
- [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 the 12th European Symposium On Research In Computer Security*, Dresden, Germany, September 24-26, 2007.
- [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 the 29th International Conference on Distributed Computing Systems (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 the 14th European Symposium On Research In Computer Security*, Saint Malo, France, September 21-25, 2009.

References (2)

- [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* , 2010 (to appear).
- [CSYZ-08] G. Cormode, D. Srivastava, T. YU, Q. Zhang, “Anonymizing bipartite graph data using safe groupings,” in *Proc. of the 34th International Conference on Very Large*, Auckland, New Zealand, August 23-28, 2008.
- [DFJPS-10] 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*, September, 2010.
- [HIM-02] H. Hacigümüs, B. Iyer, and S. Mehrotra, “Providing database as a service,” in *Proc. of 18th ICDE*, San Jose, CA, USA, Feb. 2002.
- [HIML-02] H. Hacigümüs, B. Iyer, S. Mehrotra, 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.