Private Data Indexes for Selective Access to Outsourced Data

Sabrina De Capitani di Vimercati[1] Sara Foresti[1]
Sushil Jajodia[2] Stefano Paraboschi[3] Pierangela Samarati[1]

(1) DTI - Università degli Studi di Milano, Italy

(2) CSIS - George Mason University, USA

(3) DIIMM - Università degli Studi di Bergamo, Italy

10th Workshop on Privacy in the Electronic Society (WPES 2011)

October 17, 2011 – Chicago, IL, USA

Motivation (1)

- The management of large amount of sensitive information is quite expensive

- Database outsourcing is becoming increasingly popular (Database As a Service) [Hacigümüş et al., SIGMOD 2002]
  - $+$ 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

$\Longrightarrow$ data encryption provides both integrity and confidentiality

Motivation (2)

- The storage server can be honest-but-curious
- The server cannot decrypt the data for executing queries

  $\Longrightarrow$ indexes can be associated with encrypted data to allow the server to execute queries on them

- The data owner may want to provide different data views to different users

  $\Longrightarrow$ selective encryption uses different keys for different portions of the data

- The combination of the two solutions may open the door to inferences by users

Our contributions

- Characterize the exposure of confidential information due to indexes and access control enforcement
- Define a novel index function, depending on plaintext values and access control restrictions, that
  - supports efficient query evaluation
  - protects against inference exposure
- Translate queries formulated on the plaintext relation into equivalent queries on the encrypted relation using indexes

Encrypted relation

- Symmetric encryption is applied at the tuple-level
- The encrypted version of relation r over schema $R(\mathrm{A}_1, \ldots, \mathrm{A}_n)$ is a relation r^e over schema $R^e(\underline{\mathrm{tid}}, \mathrm{etuple}, \mathtt{I}_1, \ldots, \mathtt{I}_l)$:
  - tid: numerical attribute acting as primary key
  - etuple: ciphertext resulting from the encryption of a tuple
  - $\mathtt{I}_i$, i=1,. . . ,l: index over attribute $\mathrm{A}_{j_i} \in R$

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Indexing techniques

- Direct index (e.g., [Damiani et al., CCS 2003])
  each plaintext value is mapped to a different index value and viceversa

- Flattened index (e.g., [Wang and Lakshmanan, VLDB 2006])
  each plaintext value is mapped to a set of index values and each index value corresponds to a unique plaintext value

- Bucket/hash-based index (e.g., [Hacigümüş et al., SIGMOD 2002; Damiani et al., CCS 2003])
  different plaintext values are mapped to the same index value

User knowledge

Each user knows the:

- index functions used to define indexes in R^e
- plaintext tuples that she is authorized to access
- encrypted relation r^e in its entirety

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

User knowledge

Each user knows the:

- index functions used to define indexes in R^e
- plaintext tuples that she is authorized to access
- encrypted relation r^e in its entirety

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1				
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1				
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1				
t_2	002	Rome	**2010**	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι**(2010)**	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			**2010**	
t_2	002	Rome	**2010**	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι**(2010)**	ι(600)
2	β	ι(Rome)	ι**(2010)**	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	
t_2	002	Rome	2010	700
t_3	003	Rome	**2011**	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι**(2011)**	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	
t_2	002	Rome	2010	700
t_3	003	Rome	**2011**	600
t_4			**2011**	
t_5			**2011**	

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι**(2011)**	ι(600)
4	δ	ι(NY)	ι**(2011)**	ι(700)
5	ε	ι(Oslo)	ι**(2011)**	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	
t_2	002	Rome	2010	**700**
t_3	003	Rome	2011	600
t_4			2011	
t_5			2011	

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι**(700)**
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	
t_2	002	Rome	2010	**700**
t_3	003	Rome	2011	600
t_4			2011	**700**
t_5			2011	**700**

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι**(700)**
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι**(700)**
5	ε	ι(Oslo)	ι(2011)	ι**(700)**

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	
t_2	002	Rome	2010	700
t_3	003	Rome	2011	**600**
t_4			2011	700
t_5			2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι**(600)**
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	**600**
t_2	002	Rome	2010	700
t_3	003	Rome	2011	**600**
t_4			2011	700
t_5			2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι**(600)**
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι**(600)**
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1			2010	600
t_2	002	**Rome**	2010	700
t_3	003	**Rome**	2011	600
t_4			2011	700
t_5			2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι**(Rome)**	ι(2010)	ι(700)
3	γ	ι**(Rome)**	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Direct index (1)

- Plaintext values are always represented by the same index value and viceversa

  $\Longrightarrow$ cells having the same plaintext values are exposed

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1		~~Rome~~	2010	600
t_2	002	**Rome**	2010	700
t_3	003	**Rome**	2011	600
t_4		~~Rome~~	2011	700
t_5		~~Rome~~	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι**(NY)**	ι(2010)	ι(600)
2	β	ι**(Rome)**	ι(2010)	ι(700)
3	γ	ι**(Rome)**	ι(2011)	ι(600)
4	δ	ι**(NY)**	ι(2011)	ι(700)
5	ε	ι**(Oslo)**	ι(2011)	ι(700)

Exposure risk: Direct index (2)

- Each user knows index function ι

  $\Longrightarrow$ all index-plaintext value correspondences are exposed to brute-force attacks

  $\Longrightarrow$ the whole outsourced relation is exposed to brute-force attacks

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1		NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4		NY	2011	700
t_5		Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι(NY)	ι(2010)	ι(600)
2	β	ι(Rome)	ι(2010)	ι(700)
3	γ	ι(Rome)	ι(2011)	ι(600)
4	δ	ι(NY)	ι(2011)	ι(700)
5	ε	ι(Oslo)	ι(2011)	ι(700)

Exposure risk: Flattened and bucket/hash-based index

- Flattened index: an index value always represents the same plaintext value and users know the index function

  $\Longrightarrow$ cells having the same plaintext values are exposed

  $\Longrightarrow$ all index-plaintext value correspondences are exposed to brute-force attacks

  $\Longrightarrow$ the whole outsourced relation is exposed to brute-force attacks

- Bucket/hash-based index: the same index value may represent different plaintext values

  $\Longrightarrow$ users can only infer with certainty that certain values do not correspond to given cells

Intuitive approach: ACL-based index

Index values directly depend on ACLs

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_{AB}(Rome)	ι_{AB}(2010)	ι_{AB}(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_{AC}(NY)	ι_{AC}(2011)	ι_{AC}(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

Intuitive approach: ACL-based index

Index values directly depend on ACLs

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_{AB}(Rome)	ι_{AB}(2010)	ι_{AB}(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_{AC}(NY)	ι_{AC}(2011)	ι_{AC}(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

\+ block inference exposure

– considerable burden at the client side for query translation

Intuitive approach: ACL-based index

Index values directly depend on ACLs

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1				
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_{AB}(Rome)	ι_{AB}(2010)	ι_{AB}(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_{AC}(NY)	ι_{AC}(2011)	ι_{AC}(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

\+ block inference exposure

– considerable burden at the client side for query translation

Ex: query submitted by user B with condition

`Year=2010`

Intuitive approach: ACL-based index

Index values directly depend on ACLs

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1				
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_{AB}**(Rome)**	ι_{AB}**(2010)**	ι_{AB}**(700)**
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_{AC}(NY)	ι_{AC}(2011)	ι_{AC}(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

\+ block inference exposure

– considerable burden at the client side for query translation

Ex: query submitted by user B with condition

`Year`=2010 $\Longrightarrow$ $\mathtt{I}_y$ IN $\{\iota_B(2010), \iota_{AB}(2010), \iota_{BC}(2010), \iota_{ABC}(2010)\}$

Intuitive approach: User-based index

- Each user u has an index function ι_u that depends on a private piece of information shared with the data owner

- For each cell t[A] in r and user u in $acl(t)$ there is index value $\iota_u(t[\mathrm{A}])$ in $t^e[\mathtt{I}_\mathrm{A}]$

Intuitive approach: User-based index

- Each user u has an index function ι_u that depends on a private piece of information shared with the data owner

- For each cell t[A] in r and user u in $acl(t)$ there is index value $\iota_u(t[\mathrm{A}])$ in $t^e[\mathtt{I}_\mathrm{A}]$

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A(700)ι_B(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A(700)ι_C(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

Intuitive approach: User-based index

- Each user u has an index function ι_u that depends on a private piece of information shared with the data owner
- For each cell t[A] in r and user u in $acl(t)$ there is index value $\iota_u(t[\mathrm{A}])$ in $t^e[\mathtt{I}_\mathrm{A}]$

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A(700)ι_B(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A(700)ι_C(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

$\Longrightarrow$ remains vulnerable to inference

Intuitive approach: User-based index

- Each user u has an index function ι_u that depends on a private piece of information shared with the data owner
- For each cell t[A] in r and user u in $acl(t)$ there is index value $\iota_u(t[\mathrm{A}])$ in $t^e[\mathtt{I}_\mathrm{A}]$

SHOPS

	acl		Id	City	Year	Sales
t_1	A	t_1				
t_2	A,B	t_2	002	Rome	2010	700
t_3	B	t_3	003	Rome	2011	600
t_4	A,C	t_4				
t_5	C	t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A(700)ι_B(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A(700)ι_C(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

$\Longrightarrow$ remains vulnerable to inference

Intuitive approach: User-based index

- Each user u has an index function ι_u that depends on a private piece of information shared with the data owner
- For each cell t[A] in r and user u in $acl(t)$ there is index value $\iota_u(t[\mathrm{A}])$ in $t^e[\mathtt{I}_\mathrm{A}]$

SHOPS

	acl		Id	City	Year	Sales
t_1	A	t_1			**2010**	
t_2	A,B	t_2	002	Rome	**2010**	**700**
t_3	B	t_3	003	Rome	2011	600
t_4	A,C	t_4				**700**
t_5	C	t_5				**700**

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY)	ι_A**(2010)**	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A**(2010)**ι_B**(2010)**	ι_A**(700)**ι_B**(700)**
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A**(700)**ι_C**(700)**
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C**(700)**

$\Longrightarrow$ remains vulnerable to inference

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_{\mathsf{A}} t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$\sim_{\texttt{City}}$

$t_1 \sim_{\texttt{City}} t_4$

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$\sim$City

$t_1 \sim_{\texttt{City}} t_4$

$t_2 \sim_{\texttt{City}} t_3$

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$\sim_{\texttt{Year}}$

$t_1 \sim_{\texttt{City}} t_4$
$t_2 \sim_{\texttt{City}} t_3$
$t_1 \sim_{\texttt{Year}} t_2$

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales	
t_1	001	NY	2010	600	
t_2	002	Rome	2010	700	
t_3	003	Rome	2011	600	
t_4	004	NY	2011	700	$\sim_{\texttt{Year}}$
t_5	005	Oslo	2011	700	

$t_1 \sim_{\texttt{City}} t_4$
$t_2 \sim_{\texttt{City}} t_3$
$t_1 \sim_{\texttt{Year}} t_2$
$t_4 \sim_{\texttt{Year}} t_5$

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$\sim_{\texttt{Sales}}$

$t_1 \sim_{\texttt{City}} t_4$
$t_2 \sim_{\texttt{City}} t_3$
$t_1 \sim_{\texttt{Year}} t_2$
$t_4 \sim_{\texttt{Year}} t_5$
$t_2 \sim_{\texttt{Sales}} t_4$

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales	
t_1	001	NY	2010	600	
t_2	002	Rome	2010	700	
t_3	003	Rome	2011	600	
t_4	004	NY	2011	700	$\sim_{\texttt{Sales}}$
t_5	005	Oslo	2011	700	

$t_1 \sim_{\texttt{City}} t_4$
$t_2 \sim_{\texttt{City}} t_3$
$t_1 \sim_{\texttt{Year}} t_2$
$t_4 \sim_{\texttt{Year}} t_5$
$t_2 \sim_{\texttt{Sales}} t_4$
$t_4 \sim_{\texttt{Sales}} t_5$

Conflicting tuples

- Tuples t_i and t_j are in conflict over attribute A, $t_i \sim_A t_j$, iff
  - have the same value for the attribute
  - can be accessed by different but overlapping sets of users

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$\not\sim_{\texttt{Sales}}$

$t_1 \sim_{\texttt{City}} t_4$
$t_2 \sim_{\texttt{City}} t_3$
$t_1 \sim_{\texttt{Year}} t_2$
$t_4 \sim_{\texttt{Year}} t_5$
$t_2 \sim_{\texttt{Sales}} t_4$
$t_4 \sim_{\texttt{Sales}} t_5$

Tuple exposure

$t_i \sim_A \ldots \sim_A t_j \Longrightarrow t_i[A]$ is exposed to all users in $acl(t_j) \setminus acl(t_i)$

$\Longrightarrow t_j[A]$ is exposed to all users in $acl(t_i) \setminus acl(t_j)$

Tuple exposure

$t_i \sim_A \ldots \sim_A t_j \Longrightarrow t_i[A]$ is exposed to all users in $acl(t_j) \setminus acl(t_i)$

$\Longrightarrow t_j[A]$ is exposed to all users in $acl(t_i) \setminus acl(t_j)$

SHOPS

	acl		Id	City	Year	Sales
t_1	A	t_1				
t_2	A,B	t_2	002	Rome	2010	700
t_3	B	t_3	003	Rome	2011	600
t_4	A,C	t_4				
t_5	C	t_5				

SHOPS^e

tid	etuple	I_c	I_y	I_s
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A(700)ι_B(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A(700)ι_C(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

Tuple exposure

$t_i \sim_A \ldots \sim_A t_j \Longrightarrow t_i[A]$ is exposed to all users in $acl(t_j) \setminus acl(t_i)$

$\Longrightarrow t_j[A]$ is exposed to all users in $acl(t_i) \setminus acl(t_j)$

SHOPS

	acl		Id	City	Year	Sales
t_1	A	t_1			2010	
t_2	A,B	t_2	002	Rome	2010	700
t_3	B	t_3	003	Rome	2011	600
t_4	A,C	t_4				
t_5	C	t_5				

SHOPS^e

tid	etuple	I_c	I_y	I_s
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A(700)ι_B(700)
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A(700)ι_C(700)
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

- $t_1 \sim_{\mathtt{Year}} t_2 \Longrightarrow t_1[\mathtt{Year}]$ is exposed to B

Tuple exposure

$t_i \sim_A \ldots \sim_A t_j \Longrightarrow t_i[A]$ is exposed to all users in $acl(t_j) \setminus acl(t_i)$

$\Longrightarrow t_j[A]$ is exposed to all users in $acl(t_i) \setminus acl(t_j)$

SHOPS

	acl		Id	City	Year	Sales
t_1	A	t_1			2010	
t_2	A,B	t_2	002	Rome	2010	**700**
t_3	B	t_3	003	Rome	2011	600
t_4	A,C	t_4				**700**
t_5	C	t_5				

SHOPS^e

tid	etuple	I_c	I_y	I_s
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A**(700)**ι_B**(700)**
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A**(700)**ι_C**(700)**
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C(700)

- $t_1 \sim_{\texttt{Year}} t_2 \Longrightarrow t_1[\texttt{Year}]$ is exposed to B
- $t_2 \sim_{\texttt{Sales}} t_4 \Longrightarrow t_4[\texttt{Sales}]$ is exposed to B

Tuple exposure

$t_i \sim_A \ldots \sim_A t_j \Longrightarrow t_i[A]$ is exposed to all users in $acl(t_j) \setminus acl(t_i)$

$\Longrightarrow t_j[A]$ is exposed to all users in $acl(t_i) \setminus acl(t_j)$

SHOPS

	acl		Id	City	Year	Sales
t_1	A	t_1			2010	
t_2	A,B	t_2	002	Rome	2010	**700**
t_3	B	t_3	003	Rome	2011	600
t_4	A,C	t_4				**700**
t_5	C	t_5				**700**

SHOPS^e

tid	etuple	I_c	I_y	I_s
1	α	ι_A(NY)	ι_A(2010)	ι_A(600)
2	β	ι_A(Rome)ι_B(Rome)	ι_A(2010)ι_B(2010)	ι_A**(700)**ι_B**(700)**
3	γ	ι_B(Rome)	ι_B(2011)	ι_B(600)
4	δ	ι_A(NY)ι_C(NY)	ι_A(2011)ι_C(2011)	ι_A**(700)**ι_C**(700)**
5	ε	ι_C(Oslo)	ι_C(2011)	ι_C**(700)**

- $t_1 \sim_{\texttt{Year}} t_2 \Longrightarrow t_1[\texttt{Year}]$ is exposed to B
- $t_2 \sim_{\texttt{Sales}} t_4 \Longrightarrow t_4[\texttt{Sales}]$ is exposed to B
- $t_2 \sim_{\texttt{Sales}} t_4 \sim_{\texttt{Sales}} t_5 \Longrightarrow t_5[\texttt{Sales}]$ is exposed to B

Safe index

- An index function is safe if conflicting tuples have different index values for all the users who can access them
- The index values computed by a safe index function cannot be exploited for inference purposes
- We define a safe index for attribute A by
  - safely partitioning tuples in clusters such that tuples in conflict over A do not belong to the same cluster
  - adopting a different salt for each cluster in the definition of the index function for A
- To minimize the burden at the client side for query translation, the number of salts (i.e., the number of clusters) must be minimized

Conflict graph

- Our minimization problem is equivalent to the minimum vertex coloring problem
- A conflict graph $G_A(V_A, E_A)$ is a non-directed graph with

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

Conflict graph

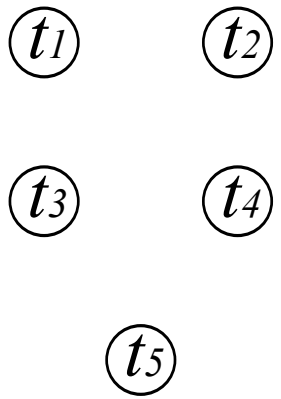
- Our minimization problem is equivalent to the minimum vertex coloring problem
- A conflict graph $G_A(V_A, E_A)$ is a non-directed graph with
  - a vertex in V_A for each tuple in r

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$G_{\texttt{City}}$



Conflict graph

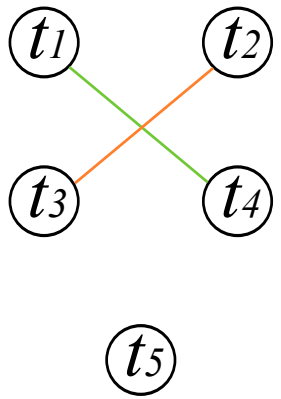
- Our minimization problem is equivalent to the minimum vertex coloring problem
- A conflict graph $G_A(V_A, E_A)$ is a non-directed graph with
  - a vertex in V_A for each tuple in r
  - an edge (t_i, t_j) in E_A iff $t_i \sim_A t_j$

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$G_{\texttt{City}}$



Conflict graph

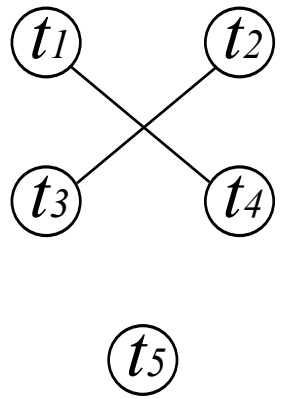
- Our minimization problem is equivalent to the minimum vertex coloring problem
- A conflict graph $G_A(V_A, E_A)$ is a non-directed graph with
  - a vertex in V_A for each tuple in r
  - an edge (t_i, t_j) in E_A iff $t_i \sim_A t_j$
- A minimum coloring of G_A is a minimum safe partitioning of r that solves conflicts w.r.t. A

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

$G_{\texttt{City}}$



Conflict graph

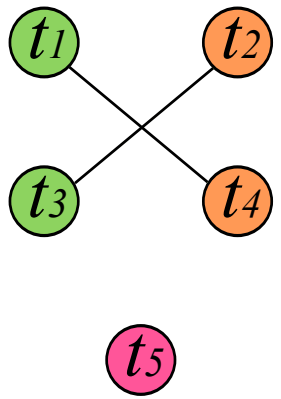
- Our minimization problem is equivalent to the minimum vertex coloring problem
- A conflict graph $G_A(V_A, E_A)$ is a non-directed graph with
  - a vertex in V_A for each tuple in r
  - an edge (t_i, t_j) in E_A iff $t_i \sim_A t_j$
- A minimum coloring of G_A is a minimum safe partitioning of r that solves conflicts w.r.t. A

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

G_{City}



Safe but not minimum coloring

Conflict graph

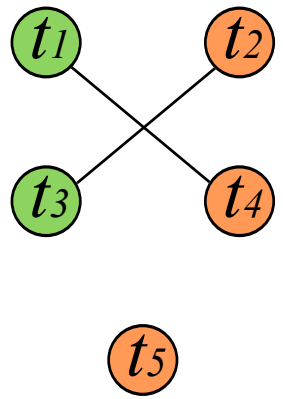
- Our minimization problem is equivalent to the minimum vertex coloring problem
- A conflict graph $G_A(V_A, E_A)$ is a non-directed graph with
  - a vertex in V_A for each tuple in r
  - an edge (t_i, t_j) in E_A iff $t_i \sim_A t_j$
- A minimum coloring of G_A is a minimum safe partitioning of r that solves conflicts w.r.t. A

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

G_{City}



Safe and minimum coloring

Computing a safe index

Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

Computing a safe index

Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α			
2	β			
3	γ			
4	δ			
5	ε			

Computing a safe index

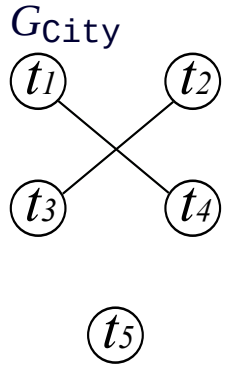
Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700



SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α			
2	β			
3	γ			
4	δ			
5	ε			

Computing a safe index

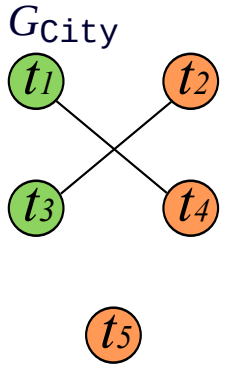
Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700



SHOPS^e

tid	etuple	I_c	I_y	I_s
1	α			
2	β			
3	γ			
4	δ			
5	ε			

Computing a safe index

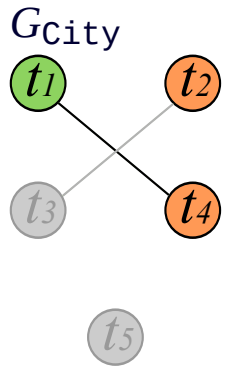
Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700



SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY,s_A)		
2	β	ι_A(Rome,s'_A)		
3	γ			
4	δ	ι_A(NY,s'_A)		
5	ε			

Computing a safe index

Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

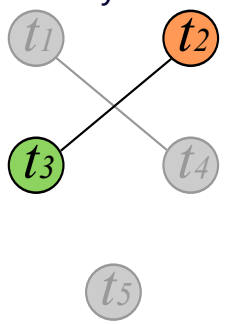
- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

G_{City}



SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY,s_A)		
2	β	ι_A(Rome,s'_A)ι_B(Rome,s_B)		
3	γ	ι_B(Rome,s'_B)		
4	δ	ι_A(NY,s'_A)		
5	ε			

Computing a safe index

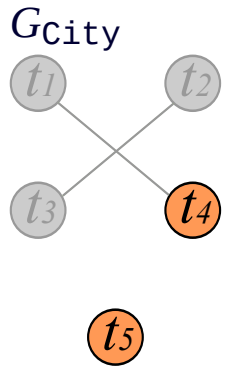
Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700



SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	$\iota_A(\text{NY},s_A)$		
2	β	$\iota_A(\text{Rome},s'_A)\iota_B(\text{Rome},s_B)$		
3	γ	$\iota_B(\text{Rome},s'_B)$		
4	δ	$\iota_A(\text{NY},s'_A)\iota_C(\text{NY},s_C)$		
5	ε	$\iota_C(\text{Oslo},s_C)$		

Computing a safe index

Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

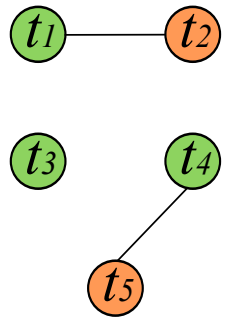
- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

G_{Year}



SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	ι_A(NY,s_A)	ι_A(2010,s_A)	
2	β	ι_A(Rome,s'_A)ι_B(Rome,s_B)	ι_A(2010,s'_A)ι_B(2010,s_B)	
3	γ	ι_B(Rome,s'_B)	ι_B(2011,s'_B)	
4	δ	ι_A(NY,s'_A)ι_C(NY,s_C)	ι_A(2011,s_A)ι_C(2011,s_C)	
5	ε	ι_C(Oslo,s_C)	ι_C(2011,s'_C)	

Computing a safe index

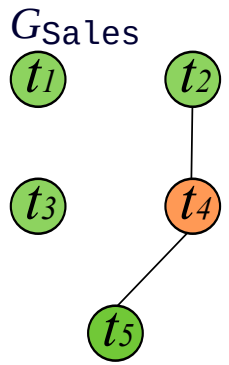
Index function ι_u for user u over attribute A is defined applying randomly generated salts to tuples

- tuples in different clusters are assigned different salts
- tuples in the same cluster are assigned the same salt

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700



SHOPS^e

tid	etuple	I_c	I_y	I_s
1	α	ι_A(NY,s_A)	ι_A(2010,s_A)	ι_A(600,s_A)
2	β	ι_A(Rome,s'_A)ι_B(Rome,s_B)	ι_A(2010,s'_A)ι_B(2010,s_B)	ι_A(700,s_A)ι_B(700,s_B)
3	γ	ι_B(Rome,s'_B)	ι_B(2011,s'_B)	ι_B(600,s_B)
4	δ	ι_A(NY,s'_A)ι_C(NY,s_C)	ι_A(2011,s_A)ι_C(2011,s_C)	ι_A(700,s'_A)ι_C(700,s_C)
5	ε	ι_C(Oslo,s_C)	ι_C(2011,s'_C)	ι_C(700,s'_C)

Relation level approach (1)

- The conflict graph can also be defined over the whole schema of the outsourced relation, defining a unique partitioning of r
- Each tuple t is associated with a unique salt, used to compute all the index values associated with t
- Conflict graph $G_R(V_R, E_R)$ is a non-directed graph with

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

Relation level approach (1)

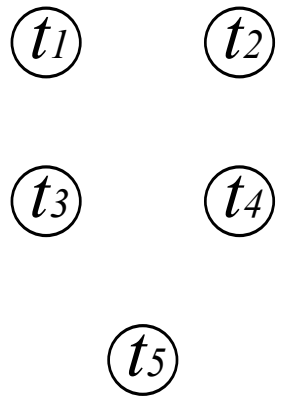
- The conflict graph can also be defined over the whole schema of the outsourced relation, defining a unique partitioning of r
- Each tuple t is associated with a unique salt, used to compute all the index values associated with t
- Conflict graph $G_R(V_R, E_R)$ is a non-directed graph with
  - a vertex in V_R for each tuple in r

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

G_{SHOPS}



Relation level approach (1)

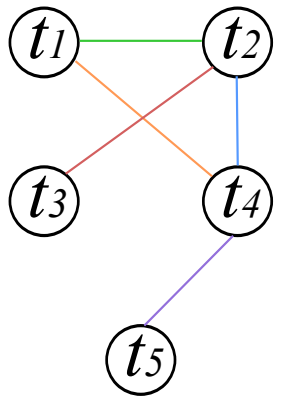
- The conflict graph can also be defined over the whole schema of the outsourced relation, defining a unique partitioning of r
- Each tuple t is associated with a unique salt, used to compute all the index values associated with t
- Conflict graph $G_R(V_R, E_R)$ is a non-directed graph with
  - a vertex in V_R for each tuple in r
  - an edge (t_i, t_j) in E_R if $\exists A \in R$ s.t. $t_i \sim_A t_j$

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

SHOPS

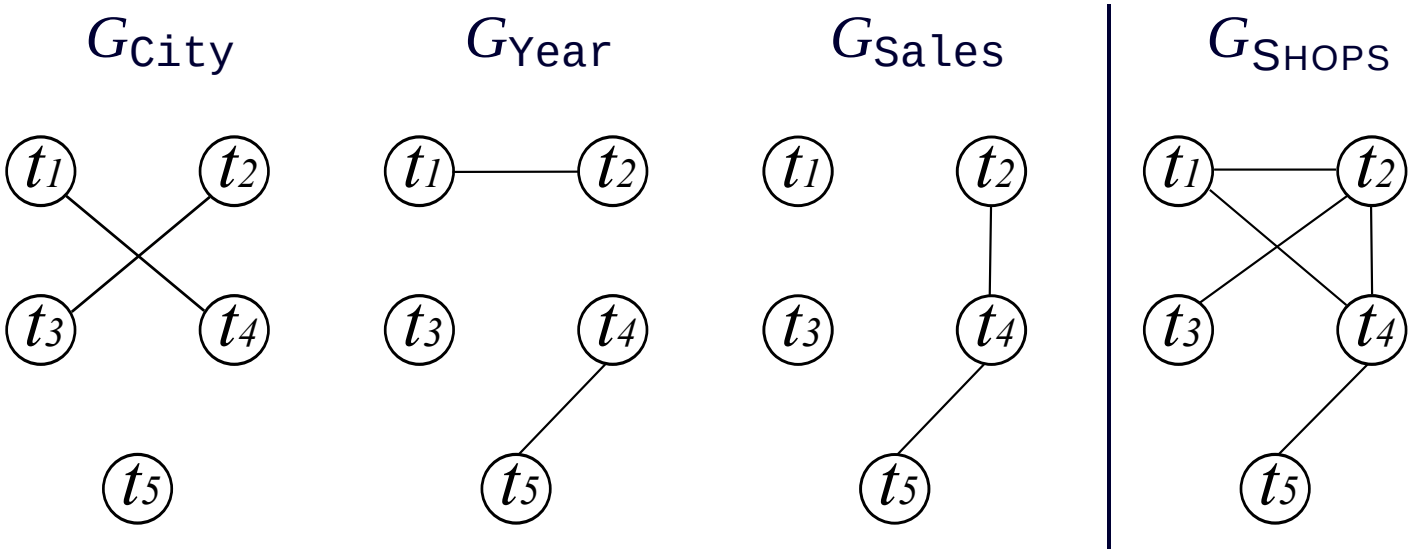
	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

G_{SHOPS}



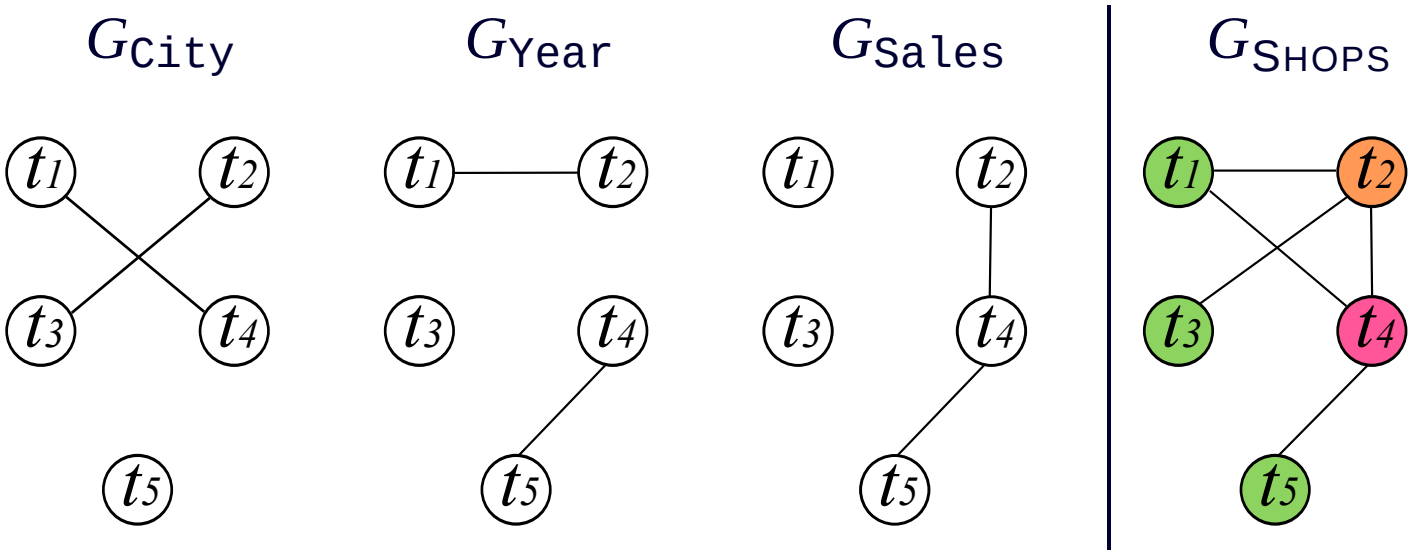
Relation level approach (2)

- Conflict graph $G_R(V_R,E_R)$ can be obtained by composing the conflict graphs $G_A(V_A,E_A)$ of attributes in R
  - a coloring for G_R is a coloring for G_A, with $A \in R$, but not viceversa
  - a minimum coloring for G_R may not be minimum for G_A, with $A \in R$



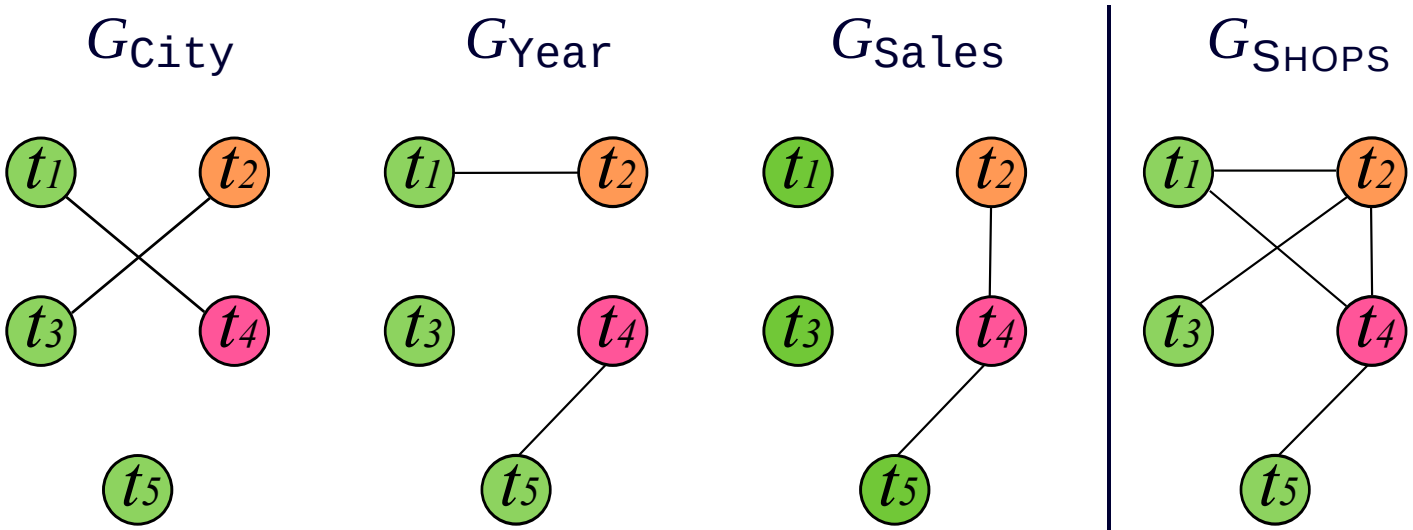
Relation level approach (2)

- Conflict graph $G_R(V_R,E_R)$ can be obtained by composing the conflict graphs $G_A(V_A,E_A)$ of attributes in R
  - a coloring for G_R is a coloring for G_A, with $A \in R$, but not viceversa
  - a minimum coloring for G_R may not be minimum for G_A, with $A \in R$



Relation level approach (2)

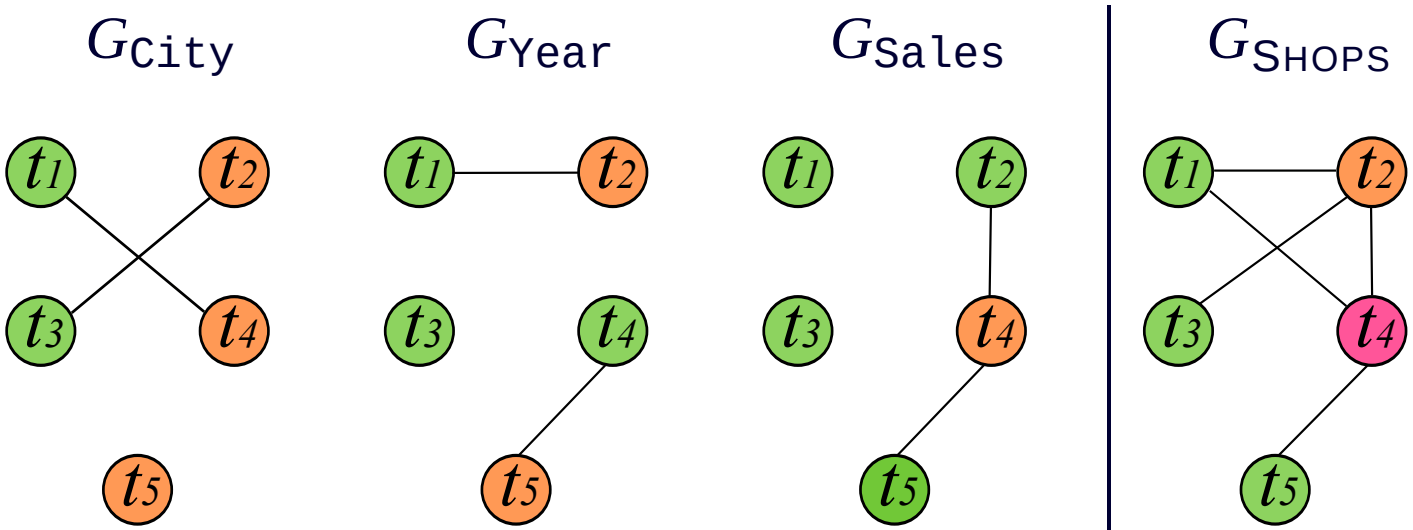
- Conflict graph $G_R(V_R,E_R)$ can be obtained by composing the conflict graphs $G_A(V_A,E_A)$ of attributes in R
  - a coloring for G_R is a coloring for G_A, with $A \in R$, but not viceversa
  - a minimum coloring for G_R may not be minimum for G_A, with $A \in R$



Safe but not minimum coloring

Relation level approach (2)

- Conflict graph $G_R(V_R,E_R)$ can be obtained by composing the conflict graphs $G_A(V_A,E_A)$ of attributes in R
  - a coloring for G_R is a coloring for G_A, with $A \in R$, but not viceversa
  - a minimum coloring for G_R may not be minimum for G_A, with $A \in R$



Safe and minimum coloring

Query evaluation

- Each user u knows
  - index function ι_u
  - the maximum number of salts $n_{\mathsf{A},u}$ used to define the index for attribute A
  - the pseudo-random function used to generate salts
- Condition A=v in a query submitted by user u is translated as $\mathtt{I}_\mathsf{A}$ IN V, with
  - $\mathtt{I}_\mathsf{A}$: index over A
  - V=$\{\iota_u(v, s_1), \ldots, \iota_u(v, s_{n_{\mathsf{A},u}})\}$: values obtained applying ι_u to v combined with each of the $n_{\mathsf{A},u}$ salts

Query evaluation: Example

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1	001	NY	2010	600
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4	004	NY	2011	700
t_5	005	Oslo	2011	700

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	$\iota_A(\text{NY},s_A)$	$\iota_A(2010,s_A)$	$\iota_A(600,s_A)$
2	β	$\iota_A(\text{Rome},s'_A)\iota_B(\text{Rome},s_B)$	$\iota_A(2010,s'_A)\iota_B(2010,s_B)$	$\iota_A(700,s_A)\iota_B(700,s_B)$
3	γ	$\iota_B(\text{Rome},s'_B)$	$\iota_B(2011,s'_B)$	$\iota_B(600,s_B)$
4	δ	$\iota_A(\text{NY},s'_A)\iota_C(\text{NY},s_C)$	$\iota_A(2011,s_A)\iota_C(2011,s_C)$	$\iota_A(700,s'_A)\iota_C(700,s_C)$
5	ε	$\iota_C(\text{Oslo},s_C)$	$\iota_C(2011,s'_C)$	$\iota_C(700,s'_C)$

Query evaluation: Example

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1				
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	$\iota_A(\text{NY},s_A)$	$\iota_A(2010,s_A)$	$\iota_A(600,s_A)$
2	β	$\iota_A(\text{Rome},s'_A)\iota_B(\text{Rome},s_B)$	$\iota_A(2010,s'_A)\iota_B(2010,s_B)$	$\iota_A(700,s_A)\iota_B(700,s_B)$
3	γ	$\iota_B(\text{Rome},s'_B)$	$\iota_B(2011,s'_B)$	$\iota_B(600,s_B)$
4	δ	$\iota_A(\text{NY},s'_A)\iota_C(\text{NY},s_C)$	$\iota_A(2011,s_A)\iota_C(2011,s_C)$	$\iota_A(700,s'_A)\iota_C(700,s_C)$
5	ε	$\iota_C(\text{Oslo},s_C)$	$\iota_C(2011,s'_C)$	$\iota_C(700,s'_C)$

Query by B, who has 2 salts for `Year`

```
SELECT City, Sales
FROM   SHOPS
WHERE  Year=2010
```

$\Longrightarrow$

Query evaluation: Example

SHOPS

	acl
t_1	A
t_2	A,B
t_3	B
t_4	A,C
t_5	C

	Id	City	Year	Sales
t_1				
t_2	002	Rome	2010	700
t_3	003	Rome	2011	600
t_4				
t_5				

SHOPS^e

tid	etuple	$\mathtt{I}_c$	$\mathtt{I}_y$	$\mathtt{I}_s$
1	α	$\iota_A(\text{NY},s_A)$	$\iota_A(2010,s_A)$	$\iota_A(600,s_A)$
2	β	ι_A**(Rome,**s'_A**)**ι_B**(Rome,**s_B**)**	ι_A**(2010,**s'_A**)**ι_B**(2010,**s_B**)**	ι_A**(700,**s_A**)**ι_B**(700,**s_B**)**
3	γ	$\iota_B(\text{Rome},s'_B)$	$\iota_B(2011,s'_B)$	$\iota_B(600,s_B)$
4	δ	$\iota_A(\text{NY},s'_A)\iota_C(\text{NY},s_C)$	$\iota_A(2011,s_A)\iota_C(2011,s_C)$	$\iota_A(700,s'_A)\iota_C(700,s_C)$
5	ε	$\iota_C(\text{Oslo},s_C)$	$\iota_C(2011,s'_C)$	$\iota_C(700,s'_C)$

Query by B, who has 2 salts for `Year` translates to

```
SELECT City, Sales
FROM   SHOPS
WHERE  Year=2010
```

$\Longrightarrow$

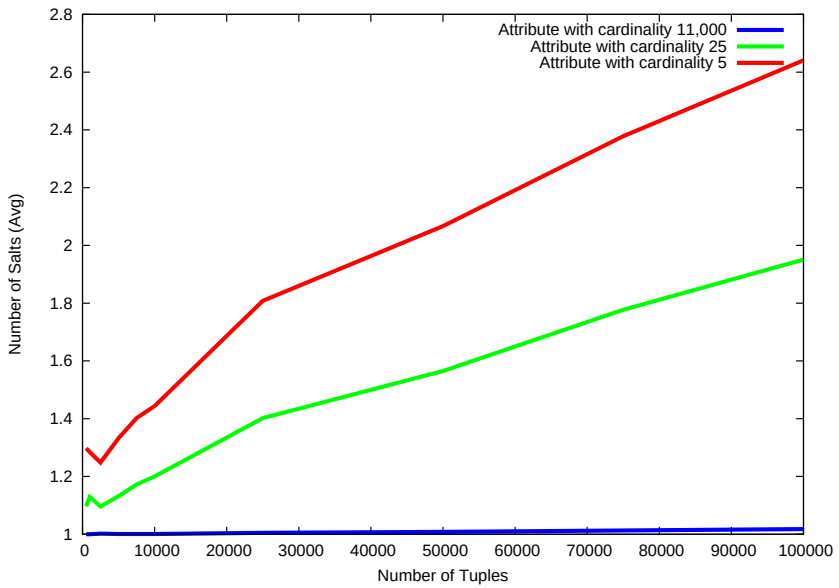
```
SELECT etuple
FROM   SHOPS^e
WHERE  I_y IN {ι_B(2010,s_B), ι_B(2010,s'_B)}
```

Experimental results (1)

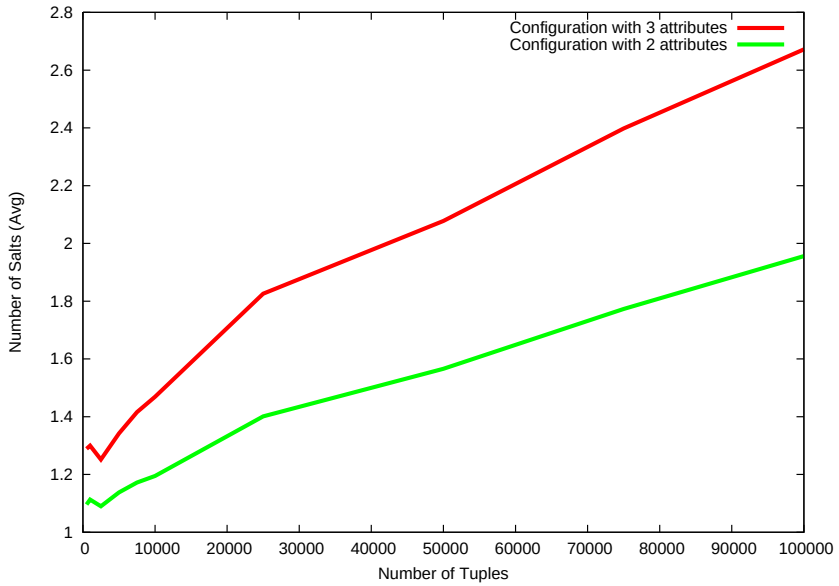
- Relational table built starting from the TPC-H benchmark
  - three attributes with 5, 25, and 11,000 distinct values
  - from 500 to 100,000 tuples
- Access control policy obtained extracting the authorship information from the DBLP repository
  - each paper is represented by a tuple in the table
  - each author can access all and only her papers
- Attribute level and relation level approaches compared w.r.t.
  - the number of clusters composing a safe partitioning (i.e., upper bound of the number of salts required)
  - the average number of salts per user (i.e., user overhead in query translation)

Experimental results (2)

Attribute level



Relation level



- Attribute level salt, three attributes:
  - cardinality 5
  - cardinality 25
  - cardinality 11000

- Relation level salt, two relations:
  - three attributes, with cardinality 5, 25, 11000
  - two attributes, with cardinality 25, 11000

Experimental results (3)

Specifying salts at the attribute level (in contrast to relation)

+ permits to reduce the overhead of queries with condition on the most selective attributes (the difference for non-selective attributes is minimal)

– requires storing a different value for the number of salts for every attribute (in contrast to a value for the whole relation), for every user

$\Longrightarrow$ If queries over selective attributes are more frequent: the attribute level approach is preferred; otherwise, the relation level approach is preferred for its simplicity and limited storage overhead

Conclusions

- We have proposed an index for accessing encrypted data that are made accessible selectively
  - it is safe from inference
  - it enjoys limited overhead in query evaluation
  - the approach is also applicable to indexing techniques other than direct index
- Future works
  - protection against the server observing multiple queries
  - protection against collusion between users and server