Algorithms for massive data, cloud and distributed computing

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

More about the course – 1

- Two modules
  - Cloud Computing and Algorithms for Massive Data (40 hours)
  - Security for Cloud Computing (40 hours)
- Organization and schedule
  - First trimester
    - Security for Cloud Computing, 40 hours (prof. Foresti)
    - Cloud Computing, 20 hours (prof. Ardagna)
  - Second trimester
    - Algorithms for Massive Data, 20 hours (prof. Malchiodi)

More about the course – 2

- Organization of the exam
  - Written test for
    - Security for Cloud Computing, 40 hours (prof. Foresti)
    - Cloud Computing, 20 hours (prof. Ardagna)
  - Project and an oral test for
    - Algorithms for Massive Data, 20 hours (prof. Malchiodi)

For students of the master in Computer Science

- This unit of the course substitutes the course Privacy and Data Protection
- 6 CFU
- Please, consider only the unit Security for Cloud Computing

Teacher

- Sara Foresti:
  - email: sara.foresti@unimi.it
  - homepage: http://www.di.unimi.it/foresti
- Course web page
  - https://homes.di.unimi.it/foresti
  - https://sforestiamdcdc.ariel.ctu.unimi.it

Classes and reference textbook/papers

- Classes
  - Wednesday 12:30 – 16:30
  - Virtual classes through Zoom platform
- Reference textbook and papers
  - Slides and scientific papers will be made available, after each class, on Ariel platform

Exam

- The exam aims at verifying the knowledge and comprehension of the topics discussed during classes

- The exam is a written test, with questions and exercises (possibly followed by a colloquium)

- First call in December 2020

Syllabus (preliminary)

0. Introduction to security and privacy
1. Authentication and access control
2. Macrodata and microdata protection
3. Privacy in data publication
4. Data protection in emerging scenarios
5. Data confidentiality and integrity in the cloud
6. Access confidentiality and integrity in the cloud

Macrodata and Microdata Protection

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

Statistical data dissemination

- Often statistical data (or data for statistical purpose) are released
- Such released data can be used to infer information that was not intended for disclosure
- Disclosure can:
  - occur based on the released data alone
  - result from combination of the released data with publicly available information
  - be possible only through combination of the released data with detailed external (public) data sources
- The disclosure risk from the released data should be very low

Statistical DBMS vs statistical data

Release of data for statistical purpose

- statistical DBMS
  - the DBMS responds only to statistical queries
  - need run time checking to control information (indirectly) released
- statistical data
  - publish statistics
  - control on indirect release performed before publication

Statistical DBMS

- A statistical DBMS is a DBMS that provides access to statistics about groups of individuals
  - should not reveal information about any particular individual
- Confidential information about an individual can be deduced
  - combining the results of different statistics
  - combining the results of statistics with external knowledge (possibly about the database content)

Statistical DBMS – Example (1)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	Female	EE	1980	50k
Cook	Male	EE	1978	70k
Davis	Female	CS	1978	80k
Evans	Male	EE	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	EE	1978	60k
Iles	Male	CS	1979	70k

Query: sum of the incomes of females with major in EE

Statistical DBMS – Example (1)

Name	**Sex**	**Major**	**Class**	**Income**
Allen	Female	CS	1980	68k
Baker	**Female**	**EE**	1980	**50k**
Cook	Male	EE	1978	70k
Davis	Female	CS	1978	80k
Evans	Male	EE	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	EE	1978	60k
Iles	Male	CS	1979	70k

Query: sum of the incomes of females with major in EE

Result: it reveals the income of Baker (only female with EE)

Statistical DBMS – Example (1)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	**Female**	**EE**	1980	**50k**
Cook	Male	EE	1978	70k
Davis	Female	CS	1978	80k
Evans	Male	EE	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	EE	1978	60k
Iles	Male	CS	1979	70k

Query: sum of the incomes of females with major in EE

Result: it reveals the income of Baker (only female with EE)

⟹ The query is sensitive

Statistical DBMS – Example (1)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	**Female**	**EE**	1980	**50k**
Cook	Male	EE	1978	70k
Davis	Female	CS	1978	80k
Evans	Male	EE	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	EE	1978	60k
Iles	Male	CS	1979	70k

Query: sum of the incomes of females with major in EE

Result: it reveals the income of Baker (only female with EE)

$\Longrightarrow$ The query is sensitive

$\Longrightarrow$ Block statistics computed over a too small number of respondents

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	Female	EE	1980	50k
Cook	Male	EE	1978	70k
Davis	Female	CS	1978	80k
Evans	Male	EE	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	EE	1978	60k
Iles	Male	CS	1979	70k

Query 1: sum of the incomes of individuals with major in EE

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	Female	**EE**	1980	**50k**
Cook	Male	**EE**	1978	**70k**
Davis	Female	CS	1978	80k
Evans	Male	**EE**	1981	**60k**
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	**EE**	1978	**60k**
Iles	Male	CS	1979	70k

Query 1: sum of the incomes of individuals with major in EE
Result: it does not reveal the income of any individual (**240k**)

⟹ The query is not sensitive

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	Female	EE	1980	50k
Cook	Male	EE	1978	70k
Davis	Female	CS	1978	80k
Evans	Male	EE	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	Male	EE	1978	60k
Iles	Male	CS	1979	70k

Query 2: sum of the incomes of males with major in EE

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	Female	EE	1980	50k
Cook	**Male**	**EE**	1978	**70k**
Davis	Female	CS	1978	80k
Evans	**Male**	**EE**	1981	**60k**
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	**Male**	**EE**	1978	**60k**
Iles	Male	CS	1979	70k

Query 2: sum of the incomes of males with major in EE
Result: it does not reveal the income of any individual (**190k**)

⟹ The query is not sensitive

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	Female	**EE**	1980	50k
Cook	**Male**	**EE**	1978	70k
Davis	Female	CS	1978	80k
Evans	**Male**	**EE**	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	**Male**	**EE**	1978	60k
Iles	Male	CS	1979	70k

Query 1: sum of the incomes of individuals with major in EE (**240k**) –
Query 2: sum of the incomes of males with major in EE (**190k**)

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	**Female**	**EE**	1980	**50k**
Cook	**Male**	**EE**	1978	70k
Davis	Female	CS	1978	80k
Evans	**Male**	**EE**	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	**Male**	**EE**	1978	60k
Iles	Male	CS	1979	70k

Query 1: sum of the incomes of individuals with major in EE (**240k**) $-$
Query 2: sum of the incomes of males with major in EE (**190k**)
$=$ sum of the incomes of females with major in EE (**50k**)
income of Baker

Statistical DBMS – Example (2)

Name	Sex	Major	Class	Income
Allen	Female	CS	1980	68k
Baker	**Female**	**EE**	1980	**50k**
Cook	**Male**	**EE**	1978	70k
Davis	Female	CS	1978	80k
Evans	**Male**	**EE**	1981	60k
Frank	Male	CS	1978	76k
Good	Male	CS	1981	64k
Hall	**Male**	**EE**	1978	60k
Iles	Male	CS	1979	70k

Query 1: sum of the incomes of individuals with major in EE (**240k**) −
Query 2: sum of the incomes of males with major in EE (**190k**)
$=$ sum of the incomes of females with major in EE (**50k**)
income of Baker

$\Longrightarrow$ The combination of queries is sensitive

Macrodata vs microdata

- In the past data were mainly released in tabular form (macrodata) and through statistical DBMS
- Today many situations require that the specific stored data themselves, called microdata, be released
  - increased flexibility and availability of information for recipients
- Microdata are subject to a greater risk of privacy breaches (linking attacks)

Macrodata

Macrodata tables can be classified into the following two groups (types of tables)

- Count/Frequency. Each cell contains the number (count) or the percentage (frequency) of respondents that have the same value over all attributes in the table
- Magnitude data. Each cell contains an aggregate value of a *quantity of interest* over all attributes in the table

Count table – Example

Two-dimensional table showing the number of employees by department and annual income (in K Euro)

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept_1	2	4	18	20	7	1	52
Dept_2	-	-	7	9	-	-	16
Dept_3	-	6	30	15	4	-	55
Dept_4	-	-	2	-	-	-	2

Magnitude table – Example

Average number of days spent in the hospital by respondents with a given disease

	Hypertension	Obesity	Chest Pain	Short Breath	Tot
M	2	8.5	23.5	3	37
F	3	30.5	0	5	38.5
Tot	5	39	23.5	8	75.5

Microdata table – Example

Records about employees of company Alfa

N	Employee	Company	Education	Salary	Race
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	Pete	Alfa	high	61	white
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	Virginia	Alfa	low	56	black
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	Mike	Alfa	low	48	white
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	Nancy	Alfa	low	37	white

Macrodata Disclosure Protection Techniques: Tables of Counts or Frequencies

Tables of counts or frequencies

- Data collected from most surveys are published in tables of count or frequencies

- The protection techniques include:
    - sampling
    - special rules
    - threshold rules

Sampling

- Conduct (and publish) a sample survey rather than a census
- Estimates are made by multiplying individual responses by a sampling weight before aggregating them
- If weights are not published, weighting helps to make an individual respondent's data less identifiable from published totals
- Estimates must achieve a specified accuracy
    - data that do not meet the accuracy requirements are not published (not considered meaningful)

Special rules

- When macrodata tables are defined on the whole population, disclosure limitation procedures must be applied

- Special rules define restrictions on the level of detail that can be provided in a table

- Special rules differ depending on the agency and the kind of table

Special rules – Example (1)

Social Security Administration (SSA) rules prohibit publishing tables where the value of a cell:

- is equal to a marginal total or
- would allow users to determine
  - an individual's age within a five-year interval
  - earnings within a $1,000 interval
  - benefits within a $50 interval

Special rules – Example (2)

Number of employees by department and annual income (in K Euro)
Special rule: Income within a 5K Euro interval

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept_1	2	4	18	20	7	1	52
Dept_2	-	-	7	9	-	-	16
Dept_3	-	6	30	15	4	-	55
Dept_4	-	-	2	-	-	-	2

Special rules – Example (2)

Number of employees by department and annual income (in K Euro)
Special rule: Income within a 5K Euro interval

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept_1	2	4	18	20	7	1	52
Dept_2	-	-	7	9	-	-	16
Dept_3	-	6	30	15	4	-	55
Dept_4	-	-	**2**	-	-	-	2

Cannot be released

- The value of a cell is equal to the total (Dept_4)

Special rules – Example (2)

Number of employees by department and annual income (in K Euro)
Special rule: Income within a 5K Euro interval

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept_1	2	4	18	20	7	1	52
Dept_2	-	-	**7**	**9**	-	-	16
Dept_3	-	6	30	15	4	-	55
Dept_4	-	-	**2**	-	-	-	2

Cannot be released

- The value of a cell is equal to the total (Dept_4)
- The table allows recipients to determine income within a 5K interval
  - between 23K and 25K for Dept_4
  - between 23K and 27K for Dept_2

Special rules – Example (3)

- To protect confidentiality, the table can be restructured and rows or columns combined (“rolling-up categories”)

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept$_1$	2	4	18	20	7	1	52
Dept$_2$	-	-	**7**	**9**	-	-	16
Dept$_3$	-	6	30	15	4	-	55
Dept$_4$	-	-	**2**	-	-	-	2

Special rules – Example (3)

- To protect confidentiality, the table can be restructured and rows or columns combined (“rolling-up categories”)

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept_1 Dept_2	2	4	25	29	7	1	68
Dept_3 Dept_4	-	6	32	15	4	-	57

- Combining Dept_1 with Dept_2 and Dept_3 with Dept_4 does offer the required protection

Special rules – Example (3)

- To protect confidentiality, the table can be restructured and rows or columns combined (“rolling-up categories”)

	Income						
Dept	**0-21**	**21-23**	**23-25**	**25-27**	**27-29**	**29+**	**Total**
Dept$_1$	2	4	18	20	7	1	52
Dept$_2$ Dept$_4$	-	-	**9**	**9**	-	-	16
Dept$_3$	-	6	30	15	4	-	55

- Combining Dept$_2$ with Dept$_4$ would still reveal that the range of income is from 23K to 26K

U.S. HIPAA

Health Insurance Portability and Accountability Act

“Safe Harbor” rules, include:

- identifying information must be removed
- locations have to be generalized to units that contain at least 20,000 residents
- dates of birth must be rounded up to the year of birth only (or to larger value if the person is older than 90)

Threshold rules

- A cell is sensitive if the number of respondents is less than some specified number (e.g., some agencies consider 5, others 3)
- A sensitive cell cannot be released
- Different techniques can be applied to protect sensitive cells:
    - table restructuring and category combination
    - cell suppression
    - random rounding
    - controlled rounding
    - confidentiality edit

Table with disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	1	3	1	20
Beta	20	10	10	15	55
Gamma	3	10	10	2	25
Delta	12	14	7	2	35
Total	50	35	30	20	135

Table with disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	1	3	1	20
Beta	20	10	10	15	55
Gamma	3	10	10	2	25
Delta	12	14	7	2	35
Total	50	35	30	20	135

A cell with fewer than **5** respondents is defined as sensitive

Table with disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

A cell with fewer than **5** respondents is defined as sensitive

Cell suppression

- One of the mostly used ways of protecting sensitive cells is suppression

- Suppressing sensitive cells (primary suppression) is not sufficient

- At least one additional cell must be suppressed (complementary suppression) for each row or column with a suppressed sensitive cell (primary suppression)
    - the value in the sensitive cell can be calculated from the marginal total

- Even with complementary suppression it is difficult to guarantee adequate protection

Complementary suppressions

- The selection of cells for complementary suppression is complicated

- Linear programming techniques are used to automatically select cells for complementary suppression

- Audit techniques can be applied to evaluate the proposed suppression pattern to see if it provides the required protection

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

A cell with fewer than **5** respondents is defined as sensitive

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	D_1	D_2	D_3	20
Beta	20	10	10	15	55
Gamma	D_4	10	10	D_5	25
Delta	12	14	7	D_6	35
Total	50	35	30	20	135

Suppress sensitive cells

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	D_1	D_2	D_3	20
Beta	20	10	10	15	55
Gamma	D_4	10	10	D_5	25
Delta	12	14	7	D_6	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$35 = D_1 + 10 + 10 + 14$

$\Longrightarrow D_1 = 1$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	$\mathbf{D}_2$	$\mathbf{D}_3$	20
Beta	20	10	10	15	55
Gamma	$\mathbf{D}_4$	10	10	$\mathbf{D}_5$	25
Delta	12	14	7	$\mathbf{D}_6$	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$35 = \mathbf{D}_1 + 10 + 10 + 14$

$\Longrightarrow \mathbf{D}_1 \mathbf{= 1}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	$\mathbf{D}_2$	$\mathbf{D}_3$	20
Beta	20	10	10	15	55
Gamma	$\mathbf{D}_4$	10	10	$\mathbf{D}_5$	25
Delta	12	14	7	$\mathbf{D}_6$	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$30 = \mathbf{D}_2 + 10 + 10 + 7$

$\Longrightarrow \mathbf{D}_2 = \mathbf{3}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**D**$_3$	20
Beta	20	10	10	15	55
Gamma	**D**$_4$	10	10	**D**$_5$	25
Delta	12	14	7	**D**$_6$	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$30 = \mathbf{D}_2 + 10 + 10 + 7$

$\Longrightarrow \mathbf{D}_2 = \mathbf{3}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**D**$_3$	20
Beta	20	10	10	15	55
Gamma	**D**$_4$	10	10	**D**$_5$	25
Delta	12	14	7	**D**$_6$	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

50 = 15 + 20 + **D**$_4$ + 12

$\Longrightarrow$ **D**$_4$ **= 3**

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**D**$_3$	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**D**$_5$	25
Delta	12	14	7	**D**$_6$	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

50 = 15 + 20 + **D**$_4$ + 12

$\Longrightarrow$ **D**$_4$ **= 3**

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	$\mathbf{D}_3$	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	$\mathbf{D}_5$	25
Delta	12	14	7	$\mathbf{D}_6$	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$35 = 12 + 14 + 7 + \mathbf{D}_6$

$\Longrightarrow \mathbf{D}_6 = \mathbf{2}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	$\mathbf{D}_3$	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	$\mathbf{D}_5$	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$35 = 12 + 14 + 7 + \mathbf{D}_6$

$\Longrightarrow \mathbf{D}_6 = 2$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	$\mathbf{D}_3$	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	$\mathbf{D}_5$	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$20 = 15 + 1 + 3 + \mathbf{D}_3$

$\Longrightarrow \mathbf{D}_3 = \mathbf{1}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

Company	Education level Low	Medium	High	Very High	Total
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**D_5**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$20 = 15 + 1 + 3 + \mathbf{D_3}$

$\Longrightarrow \mathbf{D_3 = 1}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	$\mathbf{D}_5$	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$25 = 3 + 10 + 10 + \mathbf{D}_5$

$\Longrightarrow \mathbf{D}_5 = \mathbf{2}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Suppressing sensitive cells is not sufficient

$25 = 3 + 10 + 10 + \mathbf{D}_5$

$\Longrightarrow \mathbf{D}_5 = \mathbf{2}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	D_1	D_2	D_3	20
Beta	20	D_4	D_5	15	55
Gamma	D_6	10	10	D_7	25
Delta	D_8	14	7	D_9	35
Total	50	35	30	20	135

Suppress one additional cell for each row/column with a sensitive cell suppressed

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	$\mathbf{D}_1$	$\mathbf{D}_2$	$\mathbf{D}_3$	20
Beta	20	$\mathbf{D}_4$	$\mathbf{D}_5$	15	55
Gamma	$\mathbf{D}_6$	10	10	$\mathbf{D}_7$	25
Delta	$\mathbf{D}_8$	14	7	$\mathbf{D}_9$	35
Total	50	35	30	20	135

The table appears to offer protection to the sensitive cells but:
$(15 + \mathbf{D}_1 + \mathbf{D}_2 + \mathbf{D}_3) + (20 + \mathbf{D}_4 + \mathbf{D}_5 + 15) - (\mathbf{D}_1 + \mathbf{D}_4 + 10 + 14) - (\mathbf{D}_2 + \mathbf{D}_5 + 10 + 7) = 20 + 55 - 35 - 30$
$\Longrightarrow \mathbf{D}_3 = \mathbf{1}$

Cell suppression: Table without disclosures – Example

Table containing information about employees by company and education level

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	D_1	D_2	D_3	20
Beta	20	10	10	15	55
Gamma	D_4	D_5	10	D_6	25
Delta	D_7	14	D_8	D_9	35
Total	50	35	30	20	135

The table provides adequate protection for the sensitive cells but out of a total of 16 cells, only 7 cells are published, while 9 are suppressed

Rounding

To reduce data loss due to suppression, use rounding of values

- random: random decision on whether cell values will be rounded up or down
    - the sum of the values in a row/column may be different from the published marginal totals (recipients may lose confidence in the data)
- controlled: ensure that the sum of published entries is equal to published marginal totals

Random rounding – Example

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Random rounding – Example

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

	Education level (random rounding)				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	*0	*0	*0	20
Beta	20	10	10	15	55
Gamma	*5	10	10	*0	25
Delta	*15	*15	*10	*0	35
Total	50	35	30	20	135

Controlled rounding – Example

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

Controlled rounding – Example

	Education level				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	**1**	**3**	**1**	20
Beta	20	10	10	15	55
Gamma	**3**	10	10	**2**	25
Delta	12	14	7	**2**	35
Total	50	35	30	20	135

	Education level (controlled rounding)				
Company	**Low**	**Medium**	**High**	**Very High**	**Total**
Alfa	15	*0	*5	*0	20
Beta	20	10	10	15	55
Gamma	*5	10	10	*0	25
Delta	*10	*15	*5	*5	35
Total	50	35	30	20	135

Controlled rounding

- Linear programming methods are used to identify a controlled rounding for a table

- Disadvantages:
  - it requires the use of specialized computer programs
  - controlled rounding solutions may not always exist for complex tables

Confidentiality edit (1)

- Developed by the U.S. Census Bureau to provide protection of tables prepared from the 1990 Census

- Two different approaches:
  - to protect the regular decennial Census data (100% of the population)
  - to protect the long-form of the Census which refers to a sample of the population

- Both approaches apply statistical disclosure limitation techniques to the microdata on which statistics are calculated:
  - statistics are protected by changing input data

Confidentiality edit (2)

- For the 100 percent microdata file, confidentiality edit applies switching

  1. Take a sample of records from the microdata file
  2. Find a match for these records in some other geographic region, matching on a specified set of important attributes
  3. Swap all attributes on the matched records

- For small blocks, the sampling fraction is increased to provide additional protection

- The microdata file can be used directly to prepare tables

Confidentiality edit – Example (1)

Records for the 20 employees of company Alfa

N	Employee	Company	Education	Salary	Race
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	Pete	Alfa	high	61	white
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	Virginia	Alfa	low	56	black
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	Mike	Alfa	low	48	white
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	Nancy	Alfa	low	37	white

Confidentiality edit – Example (2)

1. Take a sample of records from the microdata file (say a 10% sample). Assume that records number 4 and 17 were selected as part of our 10% sample

2. Since we need tables by company and education level, we find a match in some other company on the other variables (race and salary, company totals for these variables remain unchanged)
   - A match for record 4 (Pete) is found in company Beta, the match is with Alonso, who has very high education
   - Record 17 (Mike) is matched with George in company Delta, who has medium education

Confidentiality edit – Example (3)

3. We also assume that part of the randomly selected 10% sample from other companies match records in company Alfa
   - One record from company Delta (June with high education) matches with Virginia (record 12)
   - One record from company Gamma (Heather with low education) matched with Nancy (record 20)
4. After all matches are made, swap attributes on matched records
5. Use the swapped data file directly to produce tables

Confidentiality edit – Example (4)

Records for the 20 employees of company Alfa

N	Employee	Company	Education	Salary	Race
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	Pete	Alfa	high	61	white
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	Virginia	Alfa	low	56	black
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	Mike	Alfa	low	48	white
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	Nancy	Alfa	low	37	white

Confidentiality edit – Example (4)

Take a sample of records from the microdata file (say a 10% sample)

N	Employee	Company	Education	Salary	Race
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	**Pete**	**Alfa**	**high**	**61**	**white**
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	Virginia	Alfa	low	56	black
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	**Mike**	**Alfa**	**low**	**48**	**white**
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	Nancy	Alfa	low	37	white

Confidentiality edit – Example (4)

Since we need tables by company and education level, we find a match in some other company on the other variables

N	Employee	Company	Education	Salary	Race
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	**Alonso**	**Alfa**	**very high**	**61**	**white**
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	Virginia	Alfa	low	56	black
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	**George**	**Alfa**	**medium**	**48**	**white**
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	Nancy	Alfa	low	37	white

Confidentiality edit – Example (4)

Part of the randomly selected 10% sample from other companies match records in company Alfa

N	Employee	Company	Education	Salary	Race
1	John	Alfa	very high	201	black
2	Jim	Alfa	high	103	white
3	Sue	Alfa	high	77	black
4	**Alonso**	**Alfa**	**very high**	**61**	**white**
5	Ramesh	Alfa	medium	72	white
6	Dante	Alfa	low	103	white
7	Virgil	Alfa	low	91	black
8	Wanda	Alfa	low	84	white
9	Stan	Alfa	low	75	white
10	Irmi	Alfa	low	62	black
11	Renee	Alfa	low	58	white
12	**June**	**Alfa**	**high**	**56**	**black**
13	Mary	Alfa	low	54	black
14	Kim	Alfa	low	52	white
15	Tom	Alfa	low	55	black
16	Ken	Alfa	low	48	white
17	**George**	**Alfa**	**medium**	**48**	**white**
18	Joe	Alfa	low	41	black
19	Jeff	Alfa	low	44	black
20	**Heather**	**Alfa**	**low**	**37**	**white**

Confidentiality edit – Example (5)

Company	Education level (original) Low	Medium	High	Very High	Total
Alfa	15	1	3	1	20
Beta	20	10	10	15	55
Gamma	3	10	10	2	25
Delta	12	14	7	2	35
Total	50	35	30	20	135

Confidentiality edit – Example (5)

Education level (original)

Company	Low	Medium	High	Very High	Total
Alfa	15	1	3	1	20
Beta	20	10	10	15	55
Gamma	3	10	10	2	25
Delta	12	14	7	2	35
Total	50	35	30	20	135

Education level (with confidentiality edit)

Company	Low	Medium	High	Very High	Total
Alfa	**13**	**2**	3	**2**	20
Beta	20	10	10	15	55
Gamma	**4**	**9**	10	2	25
Delta	**13**	14	7	**1**	35
Total	50	35	30	20	135

Macrodata Disclosure Protection Techniques: Tables of Magnitude Data

Protection of tables of magnitude data

- Magnitude data are generally nonnegative quantities reported in surveys or censuses
- The distribution of these values is likely to be skewed
- Disclosure limitation techniques focus on preventing precise estimation of the values for outliers
- Sampling is less likely to provide protection
- The units that are most visible because of their size do not receive any protection from sampling

Suppression rules

- Primary suppression rules determine whether a cell could reveal individual respondent information
- Such cells are considered sensitive and cannot be released
- The most common suppression rules are:
  - the p-percent rule
  - the pq rule
  - the (n,k) rule
- These rules are used to identify sensitive cells by verifying whether it is enough difficult for one respondent to estimate the value reported by another respondent too closely

Primary suppression rule: p-percent

- Disclosure of magnitude data occurs if the user can estimate the contribution of a respondent too accurately
- A cell is sensitive, if upper and lower estimates for the respondent's value are closer to the reported value than a pre-specified percentage p
- Formally, a cell is protected if

$$\sum_{i=c+2}^{N} x_i \geq \frac{p}{100} x_1$$

$x_1, x_2, \ldots, x_N$: respondent's value in decreasing order
c: size of a coalition of respondents interested in estimating x_1

- The largest value x_1 is the most exposed

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
    - Alice: 100K
    - Bob: 80K
    - Carol: 30K
    - David: 20K
    - Eve: 10K
    - Frank: 3K
    - . . .
- The most sensitive value is Alice's, because it is easier to estimate
- If Alice's income cannot be estimated accurately, the income of the other citizens is protected

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
    - Alice: 100K
    - Bob: 80K
    - Carol: 30K
    - David: 20K
    - Eve: 10K
    - Frank: 3K
    - . . .

- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income?

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - ...
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income?
  Bob, Carol, David, whose total income is 130K

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income?
  Bob, Carol, David, whose total income is 130K
  can estimate that Alice's income is between **80K** and **120K**

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - ...
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income?

  Bob, Carol, David, whose total income is 130K

  can estimate that Alice's income is between **80K** and **120K**

  $\Longrightarrow$ sensitive for any p$\geq$20

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …

- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David

- Formally the cell is protected if:

$$\sum_{i=c+2}^{N} x_i \geq \frac{p}{100} x_1$$

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - ...

- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David

- Formally the cell is protected if:

$$\sum_{i=3+2}^{N} x_i \geq \frac{p}{100} Alice$$

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - . . .
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David
- Formally the cell is protected if:

$$\sum_{i=5}^{N} x_i \geq \frac{p}{100} 100$$

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
    - Alice: 100K
    - Bob: 80K
    - Carol: 30K
    - David: 20K
    - Eve: 10K
    - Frank: 3K
    - ...
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David
- Formally the cell is protected if:

$$Cell - \sum_{i=1}^{4} x_i \geq p$$

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - ...
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David
- Formally the cell is protected if:

$$Cell - (Alice + Bob + Carol + David) \geq p$$

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - ...
- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David
- Formally the cell is protected if:

$$250 - (100 + 80 + 30 + 20) \geq p$$

Primary suppression rule: p-percent – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - . . .

- Which is the coalition of $c = 3$ respondents that can better estimate **Alice**'s income? Bob, Carol, David

- Formally the cell is protected if:

$$20 \geq p$$

Primary suppression rule: pq (1)

- In the p-percent rule, we assumed that there was no prior knowledge about respondent's values
- Agencies should not make this assumption
- In the pq rule, agencies can specify how much prior knowledge there is by assigning a value q which represents how accurately respondents can estimate another respondent's value before any data are published (p $<$ q $<$ 100)
- Parameter q represents the error in estimation before the cell is published

Primary suppression rule: pq (2)

- Formally, a cell is protected if

$$\frac{q}{100}\sum_{i=c+2}^{N} x_i \geq \frac{p}{100}x_1$$

$x_1, x_2, \ldots, x_N$: respondent's value in decreasing order
c: size of a coalition of respondents interested in estimating x_1

- The pq rule reduces to the p-percent rule when q=100 (i.e., no estimate ability)

Primary suppression rule: pq – Example (1)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assume that the ability of respondents to estimate another respondent's value before data publishing is $q = 80\%$

Primary suppression rule: pq – Example (1)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - . . .
- Assume that the ability of respondents to estimate another respondent's value before data publishing is $q = 80\%$
- Anyone knows that **Alice**'s income is between **20K** and **180K**

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - . . .

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- The coalition of $c = 3$ respondents that can better estimate **Alice**'s income is Bob, Carol, David

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - . . .
- The coalition of $c = 3$ respondents that can better estimate **Alice**'s income is Bob, Carol, David
- The coaction can reduce uncertainty about Alice's income from [20K-180K] to **[80K-120K]**

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$\frac{q}{100} \sum_{i=c+2}^{N} x_i \geq \frac{p}{100} x_1$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$\frac{80}{100}\sum_{i=3+2}^{N} x_i \geq \frac{p}{100} Alice$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$\frac{80}{100}\sum_{i=5}^{N} x_i \geq \frac{p}{100} 100$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$\frac{80}{100}\sum_{i=5}^{N} x_i \geq p$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$\sum_{i=5}^{N} x_i \geq \frac{p}{0.80}$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - ...
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$Cell - \sum_{i=1}^{4} x_i \geq \frac{p}{0.80}$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$Cell - (Alice + Bob + Carol + David) \geq \frac{p}{0.80}$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$250 - (100 + 80 + 30 + 20) \geq \frac{p}{0.80}$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$20 \geq \frac{p}{0.80}$$

Primary suppression rule: pq – Example (2)

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - …
- Assuming $q = 80\%$ and $c = 3$
- Formally the cell is protected if

$$16 \geq p$$

Primary suppression rule: (n,k)

- Regardless of the number of respondents in a cell, if a small number (n or fewer) of these respondents contribute a large percentage (k% or more) of the total cell value, the cell is considered sensitive
- Intuitive rule: if a cell is dominated by one respondent, the published total is an upper estimate for her value
- n selected to be larger than the number of any suspected coalitions
- Many agencies use an (n,k) rule with n = 1 or 2

Primary suppression rule: (n,k) – Example

- Consider the respondents that contribute to the total income in a city, which is equal to 250K, to be (in decreasing order)
  - Alice: 100K
  - Bob: 80K
  - Carol: 30K
  - David: 20K
  - Eve: 10K
  - Frank: 3K
  - . . .
- Assuming n=2 and k=70, the cell is considered sensitive
  The income of Alice and Bob (100K+80K=180K) represents the **72%** of the cell value (250K)

Secondary suppression (1)

- Once sensitive cells have been identified, there are two options:
  - restructure the table and collapse cells until no sensitive cells remain
  - cell suppression: do not publish sensitive cells (primary suppressions) and remove other cells (complementary suppressions)
- An administrative way to avoid cell suppression consists in obtaining written permission from respondents

Secondary suppression (2)

- Other non-sensitive cells must be selected for suppression to assure that the respondent level data in sensitive cells cannot be estimated too accurately
  - a respondent's data cannot be estimated too closely
- Sensitive cells might be leaked due to the fact that:
  - implicitly published unions of suppressed cells may be sensitive according to the sensitivity rule adopted
  - the row and column equations represented by the published table may be solved, and the value for a suppressed cell estimated too accurately

Secondary suppression (3)

- Any complementary suppression is acceptable as long as the sensitive cells are protected
- For small tables the selection of complementary cells can be done manually
- Data analysts know which cells are of greatest interest (and should not be used for complementary suppression)
- Manual selection of complementary cells is acceptable as long as the resulting table provides sufficient protection to sensitive cells
- An automated audit should be applied to ensure this is true

Audit

- If totals are published the sum of the (primary or secondary) suppressed cells can be derived

- Apply the sensitivity rule to these sums to ensure that they are not sensitive

  - Rows and columns can be seen as a large system of linear equations

  - Estimate a lower and upper bound of each suppressed cell using linear programming

  - If bounds are too close to the original value, the cell is sensitive

- Simple for small tables, possibly computationally intractable for large tables

Information loss

- The selection of the complementary cells should result in minimum information loss
- There is no unique definition of information loss
- For instance, we can try to minimize:
  - the sum of the suppressed values (a large number of cells with small values can be suppressed)
  - the total number of suppressed cells

Information in parameter values

While the suppression rules can be published, parameter values should be kept confidential

EXAMPLE: Assume that:

- p-percent rule is used with p=20% and the same value is used for complementary suppression
- a cell x with value 100 has been suppressed along with other suitable complementary cells
- by solving a system of linear equations, the upper bound is 120 and the lower bound is 80: $80 \leq x \leq 120 \Longrightarrow x = 100$

Once the value for one suppressed cell has been uniquely determined, other cell values can easily be derived

Protection of tables of magnitude data – Example

Employees by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	1	2	2	1	6
Male	3	2	0	2	7
Total	4	4	2	3	13

Monthly income by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	1800	5600	4200	2500	14100
Male	4500	5800	0	5500	15800
Total	6300	11400	4200	8000	29900

Protection of tables of magnitude data – Example

Employees by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	1	2	2	1	6
Male	3	2	0	2	7
Total	4	4	2	3	13

Monthly income by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	1800	5600	4200	2500	14100
Male	4500	5800	0	5500	15800
Total	6300	11400	4200	8000	29900

(n,k) rule with n=1, k=90 ⇒ a cell is sensitive if one respondent contributes more than 90%

Protection of tables of magnitude data – Example

Employees by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	**1**	2	2	**1**	6
Male	3	2	0	2	7
Total	4	4	2	3	13

Monthly income by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	**1800**	5600	4200	**2500**	14100
Male	4500	5800	0	5500	15800
Total	6300	11400	4200	8000	29900

(n,k) rule with n=1, k=90 ⇒ a cell is sensitive if one respondent contributes more than 90%

Protection of tables of magnitude data – Example

Employees by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	**1**	2	2	**1**	6
Male	3	2	0	2	7
Total	4	4	2	3	13

Monthly income by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	**D**$_1$	5600	4200	**D**$_2$	14100
Male	4500	5800	0	5500	15800
Total	6300	11400	4200	8000	29900

(n,k) rule with n=1, k=90 ⇒ a cell is sensitive if one respondent contributes more than 90%

Protection of tables of magnitude data – Example

Employees by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	**1**	2	2	**1**	6
Male	3	2	0	2	7
Total	4	4	2	3	13

Monthly income by sex and department

Sex	Dept$_1$	Dept$_2$	Dept$_3$	Dept$_4$	Total
Female	$\mathbf{D}_1$	5600	4200	$\mathbf{D}_2$	14100
Male	$\mathbf{D}_3$	5800	0	$\mathbf{D}_4$	15800
Total	6300	11400	4200	8000	29900

Secondary suppression

Microdata

Microdata (1)

- Many situations require today that the specific stored data themselves (microdata) be released
- The advantage of releasing microdata is an increased flexibility and availability of information for the recipients
- To protect the anonymity of the respondents, data holders often remove or encrypt explicit identifiers such as names, addresses, and phone numbers
- De-identifying data, however, provides no guarantee of anonymity

Microdata (2)

- Released information often contains other quasi-identifying data (e.g., race, birth date, sex, and ZIP code) that can be linked to publicly available information to reidentify respondents
- The data recipients can determine (or restrict uncertainty) to which respondent some pieces of released data refer
- This has created an increasing demand to devote resources for an adequate protection of sensitive data
- The microdata protection techniques follow two main strategies:
  - reduce the information content
  - change the data in such a way that the information content is maintained as much as possible

Disclosure risk – Example

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

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

Microdata disclosure protection techniques

To limit the disclosure risk, the following procedures should be applied:

- including data from a sample of the whole population only
- removal of identifiers
- limiting geographic details
- limiting the number of variables

Limiting geographic details

- Geographic location is a characteristic that:
  - often appears on microdata
  - can be used for re-identifying respondents
- It is therefore important limiting geographic details

EXAMPLE:

- The Census Bureau will not identify any geographic region with less than 100,000 persons in the sampling (250,000 in the '80)
- Microdata contain contextual variables that describe the area in which a respondent resides but do not identify that area (e.g., average temperature of an area)

Classification of microdata protection techniques (1)

These techniques are based on the principle that reidentification can be counteracted by reducing the amount of released information:

- masking the data (e.g., by not releasing or by perturbing their values)
- releasing plausible but made up values instead of the real ones

According to this principle, the microdata protection techniques can be classified into two main categories:

- masking techniques
- synthetic data generation techniques

Classification of microdata protection techniques (2)

They can operate on different data types:

- Continuous. An attribute is said to be continuous if it is numerical and arithmetic operations are defined on it

  EXAMPLE: date of birth, temperature, ...

- Categorical. An attribute is said to be categorical if it can assume a limited and specified set of values and arithmetic operations do not have sense on it

  EXAMPLE: marital status, race, ...

Microdata Disclosure Protection Techniques: Masking Techniques

Masking techniques (1)

- The original data are transformed to produce new data that are valid for statistical analysis and such that they preserve the confidentiality of respondents

- They are classified as:
  - non-perturbative, the original data are not modified, but some data are suppressed and/or some details are removed
  - perturbative, the original data are modified

Masking techniques (2)

Non-perturbative

Technique	Continuous	Categorical
Sampling	yes	yes
Local suppression	yes	yes
Global recoding	yes	yes
Top-coding	yes	yes
Bottom-coding	yes	yes
Generalization	yes	yes

Perturbative

Technique	Continuous	Categorical
Resampling	yes	no
Lossy compression	yes	no
Rounding	yes	no
PRAM	no	yes
MASSC	no	yes
Random noise	yes	yes
Swapping	yes	yes
Rank swapping	yes	yes
Micro-aggregation	yes	yes

Sampling

- The protected microdata table is obtained as a sample of the original microdata table
- The protected microdata table includes only the data of a sample of the whole population
- Since there is uncertainty about whether or not a specific respondent is in the sample, reidentification risk decreases

Sampling – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Asian	64/03/09	M	94138	Married	10	190
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/03/18	M	94141	Married	60	290
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/02	M	94138	Single	22	140
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Sampling – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Asian	64/03/09	M	94138	Married	10	190
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/03/18	M	94141	Married	60	290
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/02	M	94138	Single	22	140
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Compute a sample of 11 tuples out of 14

Sampling – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	12	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Local suppression

- It suppresses the value of an attribute (i.e., it replaces it with a missing value) thus limiting the possibilities of analysis

- This technique blanks out some attribute values (sensitive cells) that are likely to contribute significantly to the disclosure risk of the tuple involved

Local suppression – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Suppress cells that contribute significantly to re-identification

Local suppression – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	**94142**	**Widow**	15	200

Suppress cells that contribute significantly to re-identification

Local suppression – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F			15	200

Global recoding

- The domain of an attribute is partitioned into disjoint intervals, usually of the same width, and each interval is associated with a label

- The protected microdata table is obtained by replacing the values of the attribute with the label associated with the corresponding interval

Global recoding – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Global recoding on **Income**:
[150-199]: low, [200-289]: medium, [290-310] high

Global recoding – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Global recoding on **Income**:

[150-199]: low, [200-289]: medium, [290-310] high

Global recoding – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	med
		Asian	64/09/30	F	94139	Divorced	1	low
		Asian	64/04/18	M	94139	Married	40	med
		Asian	64/04/15	M	94139	Married	17	med
		Black	63/03/13	M	94138	Married	2	low
		Black	63/03/18	M	94138	Married	13	low
		Black	64/09/13	F	94141	Married	15	med
		Black	64/09/07	F	94141	Married	60	high
		White	61/05/14	M	94138	Single	17	low
		White	61/05/08	M	94138	Single	10	high
		White	61/09/15	F	94142	Widow	15	med

Top-coding and bottom-coding

- Top-coding
  - It defines an upper limit, called top-code, for each attribute to be protected. Any value greater than this value is replaced with the top-code
  - It can be applied to categorical attributes that can be linearly ordered as well as to continuous attributes
- Bottom-coding
  - It defines a lower limit, called bottom-code, for each attribute to be protected. Any value lower than this limit is not published and is replaced with the bottom-code
  - It can be applied to categorical attributes that can be linearly ordered as well as to continuous attributes

Top-coding and bottom-coding – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Top-coding on **Holidays** for values higher than 30

Bottom-coding on **Holidays** for values lower than 10

Top-coding and bottom-coding – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Top-coding on **Holidays** for values higher than 30
Bottom-coding on **Holidays** for values lower than 10

Top-coding and bottom-coding – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	<10	170
		Asian	64/04/18	M	94139	Married	>30	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	<10	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	>30	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Top-coding on **Holidays** for values higher than 30
Bottom-coding on **Holidays** for values lower than 10

Generalization

- It consists in representing the values of a given attribute by using more general values

- It is based on the definition of a generalization hierarchy, where the most general value is the root and the leaves correspond to the most specific values

- It replaces values represented by the leaf nodes with one of their ancestors

- Different generalized microdata tables can be built, depending on the number of generalization steps applied

Generalization – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Generalize attribute **DoB** to the granularity of month

Generalization – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Generalize attribute **DoB** to the granularity of month

Generalization – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09	F	94139	Divorced	13	260
		Asian	64/09	F	94139	Divorced	1	170
		Asian	64/04	M	94139	Married	40	200
		Asian	64/04	M	94139	Married	17	280
		Black	63/03	M	94138	Married	2	190
		Black	63/03	M	94138	Married	13	185
		Black	64/09	F	94141	Married	15	200
		Black	64/09	F	94141	Married	60	290
		White	61/05	M	94138	Single	17	170
		White	61/05	M	94138	Single	10	300
		White	61/09	F	94142	Widow	15	200

Random noise

- It perturbs a sensitive attribute by adding or by multiplying it with a random variable with a given distribution

- It is necessary to decide whether or not to publish the distribution(s) used to add noise to the data

- Publishing the distribution(s) might increase disclosure risk of the data

Random noise – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Additive noise over attribute **Holidays** (to preserve average)

Random noise – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Noise	Income
		Asian	64/09/27	F	94139	Divorced	13	+2	260
		Asian	64/09/30	F	94139	Divorced	1	+1	170
		Asian	64/04/18	M	94139	Married	40	-10	200
		Asian	64/04/15	M	94139	Married	17	+3	280
		Black	63/03/13	M	94138	Married	2	+5	190
		Black	63/03/18	M	94138	Married	13	+8	185
		Black	64/09/13	F	94141	Married	15	+4	200
		Black	64/09/07	F	94141	Married	60	-11	290
		White	61/05/14	M	94138	Single	17	-2	170
		White	61/05/08	M	94138	Single	10	-3	300
		White	61/09/15	F	94142	Widow	15	+3	200

Additive noise over attribute **Holidays** (to preserve average)

Random noise – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	15	260
		Asian	64/09/30	F	94139	Divorced	2	170
		Asian	64/04/18	M	94139	Married	30	200
		Asian	64/04/15	M	94139	Married	20	280
		Black	63/03/13	M	94138	Married	7	190
		Black	63/03/18	M	94138	Married	21	185
		Black	64/09/13	F	94141	Married	19	200
		Black	64/09/07	F	94141	Married	49	290
		White	61/05/14	M	94138	Single	15	170
		White	61/05/08	M	94138	Single	7	300
		White	61/09/15	F	94142	Widow	18	200

Additive noise over attribute **Holidays** (to preserve average)

Swapping

- A small percent of records are matched with other records in the same file, perhaps in different geographic regions, on a set of predetermined variables
- The values of all other variables on the file are then swapped between the two records
- This technique reduces the risk of reidentification because it introduces uncertainty about the true value of a respondentent's data

Swapping – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Swap **Holidays** and **Income** for tuples with the same **Sex** and **MarStat**

Swapping – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Identify 3 pairs of tuples with same **Sex** and **MarStat**

Swapping – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	2	190
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	40	200
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	60	290
		Black	64/09/07	F	94141	Married	15	200
		White	61/05/14	M	94138	Single	10	300
		White	61/05/08	M	94138	Single	17	170
		White	61/09/15	F	94142	Widow	15	200

Swap **Holidays** and **Income**

Micro-aggregation (blurring)

- It consists in grouping individual tuples into small aggregates of a fixed dimension k
- The average over each aggregate is published instead of individual values
- Groups are formed by using maximal similarity criteria
- There are different variations of micro-aggregation:
  - the average can substitute the original value only for a tuple in the group or for all of them
  - different attributes can be protected through micro-aggregation using the same or different grouping
  - . . .

Micro-aggregation (blurring) – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Group tuples based on **Sex** and **MarStat**

Micro-aggregation (blurring) – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	260
		Asian	64/09/30	F	94139	Divorced	1	170
		Asian	64/04/18	M	94139	Married	40	200
		Asian	64/04/15	M	94139	Married	17	280
		Black	63/03/13	M	94138	Married	2	190
		Black	63/03/18	M	94138	Married	13	185
		Black	64/09/13	F	94141	Married	15	200
		Black	64/09/07	F	94141	Married	60	290
		White	61/05/14	M	94138	Single	17	170
		White	61/05/08	M	94138	Single	10	300
		White	61/09/15	F	94142	Widow	15	200

Group tuples based on **Sex** and **MarStat**

Micro-aggregation (blurring) – Example

SSN	Name	Race	DoB	Sex	ZIP	MarStat	Holidays	Income
		Asian	64/09/27	F	94139	Divorced	13	215
		Asian	64/09/30	F	94139	Divorced	1	215
		Asian	64/04/18	M	94139	Married	40	213
		Asian	64/04/15	M	94139	Married	17	213
		Black	63/03/13	M	94138	Married	2	213
		Black	63/03/18	M	94138	Married	13	213
		Black	64/09/13	F	94141	Married	15	245
		Black	64/09/07	F	94141	Married	60	245
		White	61/05/14	M	94138	Single	17	235
		White	61/05/08	M	94138	Single	10	235
		White	61/09/15	F	94142	Widow	15	200

Substitute **Income** with the average for each group

Microdata Disclosure Protection Techniques: Synthetic Techniques

Synthetic techniques (1)

- Since the statistical content of the data is not related to the information provided by each respondent, a model well representing the data could in principle replace the data themselves

- An important requirement for the generation of synthetic data is that the synthetic and original data should present the same quality of statistical analysis

- The main advantage of this class of techniques is that the released synthetic data are not referred to any respondent and therefore their release cannot lead to reidentification

Synthetic techniques (2)

Fully Synthetic

Technique	**Continuous**	**Categorical**
Bootstrap	yes	no
Cholesky decomposition	yes	no
Multiple imputation	yes	yes
Maximum entropy	yes	yes
Latin Hypercube Sampling	yes	yes

Partially Synthetic

Technique	**Continuous**	**Categorical**
IPSO	yes	no
Hybrid masking	yes	no
Random response	no	yes
Blank and impute	yes	yes
SMIKe	yes	yes
Multiply imputed partially synthetic dataset	yes	yes

Privacy in Data Publication

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

Statistical DBMS vs statistical data

Release of data for statistical purpose

- statistical DBMS
  - the DBMS responds only to statistical queries
  - need run time checking to control information (indirectly) released
- statistical data
  - publish statistics
  - control on indirect release performed before publication

Macrodata vs microdata

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

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

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

Information disclosure

Disclosure relates to attribution of sensitive information to a respondent (an individual or organization)

There is disclosure when:

- a respondent is identified from the released data (identity disclosure)
- sensitive information about a respondent is revealed through the released data (attribute disclosure)
- the released data make it possible to determine the value of some characteristics of a respondent even if no released record refers to the respondent (inferential disclosure)

Identity disclosure

It occurs if a third party can identify a respondent from the released data

Revealing that an individual is a respondent in a data collection may or may not violate confidentiality requirements

- Macrodata: revealing identity is generally not a problem, unless the identification leads to divulging confidential information (attribute disclosure)

- Microdata: identification is generally regarded as a problem, since microdata records are detailed; identity disclosure usually implies also attribute disclosure

Attribute disclosure

It occurs when confidential information about a respondent is revealed and can be attributed to her

Confidential information may be:

- revealed exactly
- closely estimated

Inferential disclosure

It occurs when information can be inferred with high confidence from statistical properties of the released data

EXAMPLE: the data may show a high correlation between income and purchase price of house. As purchase price of house is typically public information, a third party might use this information to infer the income of a respondent

Inferences are designed to predict aggregate behavior, not individual attributes, and are then often poor predictors of individual data values

- Inference disclosure itself does not always represent a risk
- It may be used together with other information and increase potential for inference

Restricted data and restricted access – 1

- The choice of statistical disclosure limitation methods depends on the nature of the data products whose confidentiality must be protected

- Some microdata include explicit identifiers (e.g., name, address, or Social Security Number)

- Removing such identifiers is a first step in preparing for the release of microdata for which the confidentiality of individual information must be protected

Restricted data and restricted access – 2

Confidentiality can be protected by:

- restricting the amount of information in the released tables (restricted data)
- imposing conditions on access to the data products (restricted access)
- some combination of these two strategies

The anonymity problem

- The amount of privately owned records that describe each citizen's finances, interests, and demographics is increasing every day
- These data are de-identified before release, that is, any explicit identifier (e.g., SSN) is removed
- De-identification is not sufficient
- Most municipalities sell population registers that include the identities of individuals along with basic demographics
- These data can then be used for linking identities with de-identified information $\Longrightarrow$**re-identification**

The anonymity problem – Example

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

The anonymity problem – Example

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

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

The anonymity problem – Example

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

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

Classification of attributes in a microdata table

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

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

Re-identification

A study of the 2000 census data reported that the US population was uniquely identifiable by:

- year of birth, 5-digit ZIP code: 0.2%
- year of birth, county: 0.0%
- year and month of birth, 5-digit ZIP code: 4.2%
- year and month of birth, county: 0.2%
- year, month, and day of birth, 5-digit ZIP code: 63.3%
- year, month, and day of birth, county: 14.8%

Factors contributing to disclosure risk – 1

Possible sources of the disclosure risk of microdata

- Existence of high visibility records. Some records on the file may represent respondents with unique characteristics such as very unusual jobs (e.g., movie star) or very large incomes
- Possibility of matching the microdata with external information. There may be individuals in the population who possess a unique or peculiar combination of the characteristic variables on the microdata
  - if some of those individuals happen to be chosen in the sample of the population, there is a disclosure risk
  - note that the identity of the individuals that have been chosen should be kept secret

Factors contributing to disclosure risk – 2

The possibility of linking or its precision increases with:

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

Factors contributing to decrease the disclosure risk – 1

- A microdata table often contains a subset of the whole population
  - this implies that the information of a specific respondent, which a malicious user may want to know, may not be included in the microdata table
- The information specified in microdata tables released to the public is not always up-to-date (often at least one or two-year old)
  - the values of the attributes of the corresponding respondents may have changed in the meanwhile
  - the age of the external sources of information used for linking may be different from the age of the information contained in the microdata table

Factors contributing to decrease the disclosure risk – 2

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

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

Measures of risk

Measuring the disclosure risk requires considering:

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

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

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

k-anonymity

k-anonymity – 1

- k-anonymity, together with its enforcement via generalization and suppression, has been proposed as an approach to protect respondents' identities while releasing truthful information
- k-anonymity tries to capture the following requirement:
  - the released data should be indistinguishably related to no less than a certain number of respondents
- Quasi-identifier: set of attributes that can be exploited for linking (whose release must be controlled)

k-anonymity – 2

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

Generalization and suppression

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

Domain generalization hierarchy

- A generalization relationship $\leq_D$ defines a mapping between domain D and its generalizations

- Given two domains $D_i, D_j \in$ Dom, $D_i \leq_D D_j$ states that the values in domain D_j are generalizations of values in D_i

- $\leq_D$ implies the existence, for each domain D, of a domain generalization hierarchy DGH$_D$ = (Dom, $\leq_D$):
  - $\forall D_i, D_j, D_z \in$ Dom: $D_i \leq_D D_j, D_i \leq_D D_z \Longrightarrow D_j \leq_D D_z \vee D_z \leq_D D_j$
  - all maximal elements of Dom are singleton

- Given a domain tuple $DT = \langle D_1, \ldots, D_n \rangle$ such that $D_i \in$ Dom, $i = 1, \ldots, n$, the domain generalization hierarchy of DT is DGH$_{DT} =$ DGH$_{D_1} \times \ldots \times$ DGH$_{D_n}$

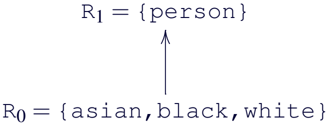
Value generalization hierarchy

- A value generalization relationship $\leq_V$ associates with each value in domain D_i a unique value in domain D_j, direct generalization of D_i

- $\leq_V$ implies the existence, for each domain D, of a value generalization hierarchy VGH$_D$

- VGH$_D$ is a tree
  - the leaves are the values in D
  - the root (i.e., the most general value) is the value in the maximum element in DGH$_D$

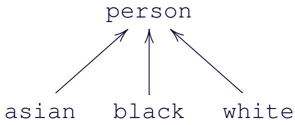
Domain and value generalization hierarchies – Example



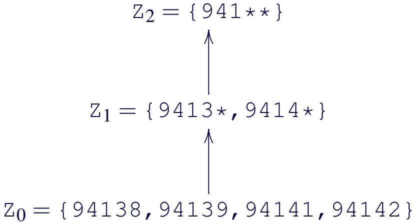
$R_1 = \{\text{person}\}$

$R_0 = \{\text{asian}, \text{black}, \text{white}\}$

DGH_{R_0}



VGH_{R_0}

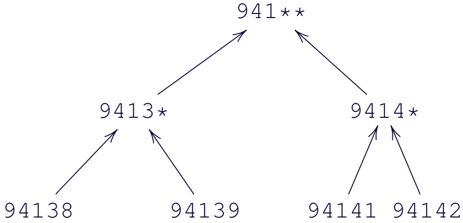


$Z_2 = \{941{\star}{\star}\}$

$Z_1 = \{9413{\star}, 9414{\star}\}$

$Z_0 = \{94138, 94139, 94141, 94142\}$

DGH_{Z_0}



VGH_{Z_0}

Generalized table with suppression

Let T_O and T_G be two tables defined on the same set of attributes. Table T_G is said to be a generalization (with tuple suppression) of table T_O if:

1. the cardinality of T_G is at most that of T_O
2. the domain of each attribute A in T_G is equal to, or a generalization of, the domain of attribute A in T_O
3. it is possible to define a correspondence (an injective function) associating each tuple t_G in T_G with a different tuple t_O in T_O, such that the value of each attribute in t_G is equal to, or a generalization of, the value of the corresponding attribute in t_O (some tuples in T_O might not have corresponding tuples in T_G)

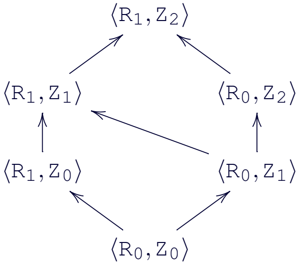
Generalized table with suppression – Example

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

PT

Race:R_1	**ZIP:**z_0
person	94141
person	94139
person	94139
person	94139
person	94139
person	94139
person	94141

GT$_{[1,0]}$



$\langle R_1, Z_2\rangle$
$\langle R_1, Z_1\rangle$
$\langle R_0, Z_2\rangle$
$\langle R_1, Z_0\rangle$
$\langle R_0, Z_1\rangle$
$\langle R_0, Z_0\rangle$

Better to suppress or generalize?

- Suppression is equivalent to generalization to the most (if unique) general value
  $\Longrightarrow$ complete information loss on the cell

- If generalization operates at the level of attribute (column) and suppression at the level of cell (value), generalizing may increase information loss (it hits all the cells in the column)

- Assume a threshold of suppression, if required suppression is:
  - below the threshold $\Longrightarrow$ suppress
  - above the threshold $\Longrightarrow$ generalize

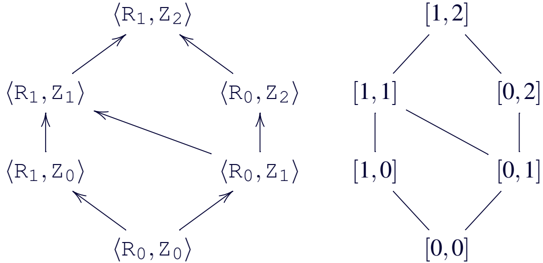
Minimal generalization

- Generalization and suppression cause information loss
  $\Longrightarrow$ do not overdue it

- Minimal solution:
  - suppress and generalize as needed, not more

k-minimal generalization with suppression – 1

- Distance vector. Let $T_i(A_1,\ldots,A_n)$ and $T_j(A_1,\ldots,A_n)$ be two tables such that $T_i \preceq T_j$. The distance vector of T_j from T_i is the vector $DV_{i,j} = [d_1,\ldots,d_n]$, where each d_z, $z = 1,\ldots,n$, is the length of the unique path between dom(A_z,T_i) and dom(A_z,T_j) in the domain generalization hierarchy DGH$_{D_z}$



$\langle R_1, Z_2\rangle$
$\langle R_1, Z_1\rangle$
$\langle R_0, Z_2\rangle$
$\langle R_1, Z_0\rangle$
$\langle R_0, Z_1\rangle$
$\langle R_0, Z_0\rangle$
$[1,2]$
$[1,1]$
$[0,2]$
$[1,0]$
$[0,1]$
$[0,0]$

k-minimal generalization with suppression – 2

Let T_i and T_j be two tables such that $T_i \preceq T_j$, and let MaxSup be the specified threshold of acceptable suppression. T_j is said to be a k-minimal generalization of table T_i iff:

1. T_j satisfies k-anonymity enforcing minimal required suppression, that is, T_j satisfies k-anonymity and $\forall T_z : T_i \preceq T_z, DV_{i,z} = DV_{i,j}$, T_z satisfies k-anonymity $\Longrightarrow |T_j| \geq |T_z|$
2. $|T_i| - |T_j| \leq$ MaxSup
3. $\forall T_z : T_i \preceq T_z$ and T_z satisfies conditions 1 and 2 $\Longrightarrow \neg(DV_{i,z} < DV_{i,j})$

Minimal generalization – Example

MaxSup=0 (no suppression)

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

PT

Minimal generalization – Example

MaxSup=0 (no suppression)

Race:$\mathtt{R}_0$	**ZIP:**$\mathtt{Z}_0$
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Minimal generalization – Example

MaxSup=0 (no suppression)

Race:$\mathtt{R}_0$	**ZIP:**$\mathtt{Z}_0$
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Race:$\mathtt{R}_1$	**ZIP:**$\mathtt{Z}_1$
person	9414*
person	9414*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9414*

$\mathrm{GT}_{[1,1]}$

Minimal generalization – Example

MaxSup=0 (no suppression)

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

PT

Race:R_1	**ZIP:**Z_1
person	9414*
person	9414*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9414*

$\mathrm{GT}_{[1,1]}$

Race:R_0	**ZIP:**Z_2
asian	941**
asian	941**
asian	941**
asian	941**
asian	941**
black	941**
black	941**
white	941**
white	941**

$\mathrm{GT}_{[0,2]}$

Minimal generalization – Example

MaxSup=0 (no suppression)

Race:$\mathtt{R}_0$	**ZIP:**$\mathtt{Z}_0$
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Race:$\mathtt{R}_1$	**ZIP:**$\mathtt{Z}_1$
person	9414*
person	9414*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9414*

$\mathrm{GT}_{[1,1]}$

Race:$\mathtt{R}_1$	**ZIP:**$\mathtt{Z}_2$
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**
person	941**

$\mathrm{GT}_{[1,2]}$

Minimal generalization – Example

MaxSup=0 (no suppression)

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

PT

Race:R_1	**ZIP:**Z_1
person	9414*
person	9414*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9413*
person	9414*

$GT_{[1,1]}$

~~**Race:**R_1~~	~~**ZIP:**Z_2~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~
~~person~~	~~941**~~

$GT_{[1,2]}$

Examples of 2-minimal generalizations

MaxSup=2

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

PT

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

GT$_{[1,0]}$

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

GT$_{[0,1]}$

Computing a preferred generalization

Different preference criteria can be applied in choosing a preferred minimal generalization, among which:

- minimum absolute distance prefers the generalization(s) with the smallest absolute distance, that is, with the smallest total number of generalization steps (regardless of the hierarchies on which they have been taken)
- minimum relative distance prefers the generalization(s) with the smallest relative distance, that is, that minimizes the total number of relative steps (a step is made relative by dividing it over the height of the domain hierarchy to which it refers)
- maximum distribution prefers the generalization(s) with the greatest number of distinct tuples
- minimum suppression prefers the generalization(s) that suppresses less tuples, that is, the one with the greatest cardinality

Classification of k-anonymity techniques – 1

Generalization and suppression can be applied at different levels of granularity

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

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

Classification of k-anonymity techniques – 2

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

2-anonymized tables wrt different models – 1

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

PT

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

AG_TS

2-anonymized tables wrt different models – 2

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

AG_CS

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

AG_≡AG_AS

2-anonymized tables wrt different models – 3

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

CG_ ≡CG_CS

Race	DOB	Sex	ZIP

_TS

2-anonymized tables wrt different models – 4

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

_AS

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

_CS

Algorithms for computing a k-anonymous table

- The problem of finding minimal k-anonymous tables, with attribute generalization and tuple suppression, is computationally hard

- Many efforts in defining algorithms for computing a solution (e.g., exploiting assumptions on the hierarchies or though heuristics)

Incognito algorithm

k-anonymity with respect to a proper subset of QI is a necessary (not sufficient) condition for k-anonymity with respect to QI

- Iteration 1: check k-anonymity for each attribute in QI, discarding generalizations that do not satisfy k-anonymity
- Iteration 2: combine the remaining generalizations in pairs and check k-anonymity for each couple obtained

  ...
- Iteration i: consider all the i-uples of attributes, obtained combining generalizations that satisfied k-anonymity at iteration $i-1$. Discard non k-anonymous solutions

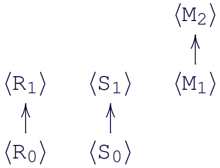
  ...
- Iteration $|QI|$ returns the final result

Incognito adopts a bottom-up approach for the visit of DGHs

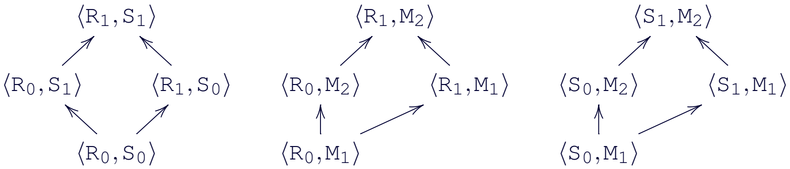
Incognito – Example (1)

Race	Sex	Marital status
asian	F	divorced
asian	F	divorced
asian	F	married
asian	M	married
asian	M	married
black	F	single
black	F	single
white	F	single
white	F	widow

Iteration 1



Iteration 2

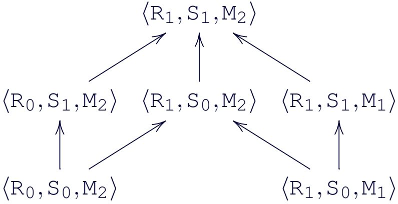


$\langle R_1, S_1 \rangle$
$\langle R_0, S_1 \rangle$
$\langle R_1, S_0 \rangle$
$\langle R_0, S_0 \rangle$
$\langle R_1, M_2 \rangle$
$\langle R_0, M_2 \rangle$
$\langle R_1, M_1 \rangle$
$\langle R_0, M_1 \rangle$
$\langle S_1, M_2 \rangle$
$\langle S_0, M_2 \rangle$
$\langle S_1, M_1 \rangle$
$\langle S_0, M_1 \rangle$

Incognito – Example (2)

Race	Sex	Marital status
asian	F	divorced
asian	F	divorced
asian	F	married
asian	M	married
asian	M	married
black	F	single
black	F	single
white	F	single
white	F	widow

Iteration 3



$\langle R_1, S_1, M_2 \rangle$

$\langle R_0, S_1, M_2 \rangle$ $\langle R_1, S_0, M_2 \rangle$ $\langle R_1, S_1, M_1 \rangle$

$\langle R_0, S_0, M_2 \rangle$ $\langle R_1, S_0, M_1 \rangle$

Mondrian multidimensional algorithm – 1

- Each attribute in QI represents a dimension
- Each tuple in PT represents a point in the space defined by QI
- Tuples with the same QI value are represented by giving a multiplicity value to points
- The multi-dimensional space is partitioned by splitting dimensions such that each area contains at least k occurrences of point values
- All the points in a region are generalized to a unique value
- The corresponding tuples are substituted by the computed generalization

Mondrian multidimensional algorithm – 2

Mondrian algorithm is flexible and can operate

- on a different number of attributes
  - single-dimension
  - multi-dimension
- with different recoding (generalization) strategies
  - global recoding
  - local recoding
- with different partitioning strategies
  - strict (i.e., non-overlapping) partitioning
  - relaxed (i.e., potentially overlapping) partitioning
- using different metrics to determine how to split on each dimension

Mondrian multidimensional algorithm – Example (1)

Private table

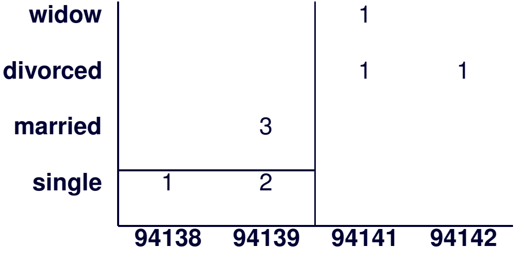
Marital status	**ZIP**
divorced	94142
divorced	94141
married	94139
married	94139
married	94139
single	94138
single	94139
single	94139
widow	94141

	94138	**94139**	**94141**	**94142**
widow			1	
divorced			1	1
married		3		
single	1	2		

Mondrian multidimensional algorithm – Example (2)

3-anonymous table

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



	94138	94139	94141	94142
widow			1	
divorced			1	1
married		3		
single	1	2		

k-anonymity revisited

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

k-anonymity revisited – Example

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

PT

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

2-anonymity

k-anonymity revisited – Example

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

PT

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

2-anonymity

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

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

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

k-anonymity revisited – Example

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

PT

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

2-anonymity

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

2-anonymity (revisited)

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

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

k-anonymity revisited – Example

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

PT

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

2-anonymity

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

2-anonymity (revisited)

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

no 2-anonymity

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

k-anonymity revisited – Example

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

PT

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

2-anonymity

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

2-anonymity (revisited)

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

no 2-anonymity

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

no 2-anonymity

Attribute Disclosure

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

k-anonymity is vulnerable to some attacks

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

Homogeneity of the sensitive attribute values

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

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

Background knowledge

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

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

ℓ-diversity – 1

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

ℓ-diversity – 2

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

  $\Longrightarrow$ the homogeneity attack is not possible anymore
  $\Longrightarrow$ the background knowledge attack becomes more difficult

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

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

BUT

ℓ-diversity leaves space to attacks based on the distribution of values inside q-blocks (skewness and similarity attacks)

Skewness attack

- Skewness attack occurs when the distribution in a q-block is different than the distribution in the original population
- 20% of the population suffers from diabetes; 75% of tuples in a q-block have diabetes

  $\Longrightarrow$ people in the q-block have higher probability of suffering from diabetes

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

Similarity attack

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

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

Group closeness

- A q-block respects t-closeness if the distance between the distribution of the values of the sensitive attribute in the q-block and in the considered population is lower than t

- T respects t-closeness if all its q-blocks respect t-closeness

- t-closeness is monotonic with respect to the generalization hierarchies considered for k-anonymity purposes

- Any algorithm for k-anonymity can be extended to enforce the t-closeness property, which however might be difficult to achieve

External knowledge – 1

- The consideration of the adversary’s background knowledge (or external knowledge) is necessary when reasoning about privacy in data publishing

- External knowledge can be exploited for inferring sensitive information about individuals with high confidence

- Positive inference
    - a respondent has a given value (or a value within a restricted set)

- Negative inference
    - a respondent does not have a given value

- Existing approaches have mostly focused on positive inference

External knowledge – 2

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

External knowledge modeling

- An adversary has knowledge about an individual (target) represented in a released table and knows the individual's QI values
  $\Longrightarrow$ predict the sensitive value of the target
- External knowledge modeled through a logical expression
- Knowledge may be about:
  - the target individual: information that the adversary may know about the target individual
  - others: information about individuals other than the target
  - same-value families: knowledge that a group (or family) of individuals have the same sensitive value
    $\Longrightarrow$ genomic information exposes also information about the relatives and descendants of the genome's owner
- Other types of external knowledge may be identified......

External knowledge – Example (1)

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

Original table

$\Longrightarrow$

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

4-anonymized table

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

External knowledge – Example (2)

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

4-anonymized table

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

External knowledge – Example (2)

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

4-anonymized table

⟹

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

4-anonymized table

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

⟹ Harry belongs to the second group

⟹ Harry has aids with confidence 1/4

External knowledge – Example (3)

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

4-anonymized table

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

External knowledge – Example (3)

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

4-anonymized table

$\Longrightarrow$

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

4-anonymized table

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

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

External knowledge – Example (4)

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

4-anonymized table

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

External knowledge – Example (4)

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

4-anonymized table

⟹

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

4-anonymized table

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

⟹ Harry has aids with confidence 1/2

Multiple releases

- Data may be subject to frequent changes and may need to be published on regular basis
- The multiple release of a microdata table may cause information leakage since a malicious recipient can correlate the released datasets

Multiple independent releases – Example (1)

T_1

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

4-anonymized table at time t_1

T_2

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

4-anonymized table at time t_2

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

Multiple independent releases – Example (1)

T_1

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

4-anonymized table at time t_1

T_2

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

4-anonymized table at time t_2

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

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

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

Multiple independent releases – Example (1)

T_1

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

4-anonymized table at time t_1

T_2

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

4-anonymized table at time t_2

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

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

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

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

Multiple independent releases – Example (2)

T_1

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

4-anonymized table at time t_1

T_2

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

4-anonymized table at time t_2

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

Multiple independent releases – Example (2)

T_1

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

4-anonymized table at time t_1

T_2

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

4-anonymized table at time t_2

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

Multiple independent releases – Example (2)

T_1

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

4-anonymized table at time t_1

T_2

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

4-anonymized table at time t_2

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

$\Longrightarrow$ Frank suffers from short breath
(and it is the only patient in the orange set of time t1 who left)

Multiple releases

Multiple (i.e., longitudinal) releases cannot be independent

$\Longrightarrow$ need to ensure multiple releases are safe with respect to intersection attacks

Extended scenarios

k-anonymity, ℓ-diversity, and t-closeness different variations

- Multiple tuples per respondent
- Release of multiple tables, characterized by (functional) dependencies
- Multiple quasi-identifiers
- Non-predefined quasi-identifiers
- Release of data streams
- Fine-grained privacy preferences

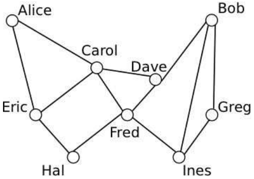
k-anonymity in various applications

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

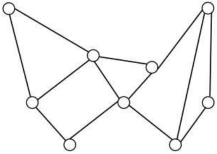
- social networks
- data mining
- location data
- …

k-anonymity in social networks

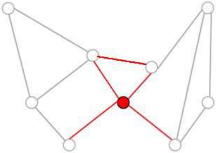
- Neighborhood attack $\Longrightarrow$ given a de-identified graph G' of a social network graph G, exploit knowledge about the neighbors of user u to re-identify the vertex representing u



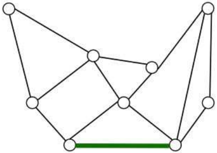
Social network



Anonymized social network



1-neighborhood graph of Fred



2-anonymous social network

k-anonymous data mining

- Privacy preserving data mining techniques depend on the definition of privacy capturing what information is sensitive in the original data and should then be protected

- k-anonymous data mining aims at ensuring that the data mining results do not violate the k-anonymity requirement over the original data

- Threats to k-anonymity can arise from performing mining on a collection of data maintained in a private table PT subject to k-anonymity constraints. E.g.:
    - association rule mining
    - classification mining

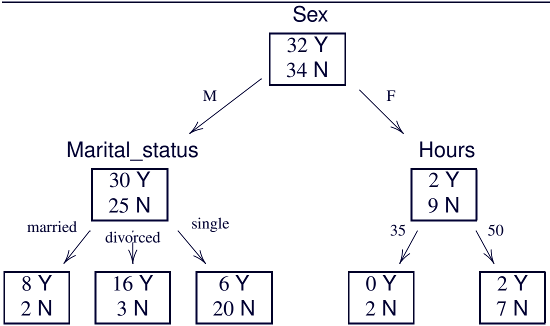
Association rule mining

Marital_status	*Sex*	*Hours*	*#tuples (Hyp. values)*
divorced	M	35	2 (0Y, 2N)
divorced	M	40	17 (16Y, 1N)
divorced	F	35	2 (0Y, 2N)
married	M	35	10 (8Y, 2N)
married	F	50	9 (2Y, 7N)
single	M	40	26 (6Y, 20N)

- {divorced} $\rightarrow$ {M} with support $\frac{19}{66}$ and confidence $\frac{19}{21}$
  If QI includes Marital_status and Sex $\Longrightarrow$
  {divorced} $\rightarrow$ {M}:
  - violates k-anonymity for any $k > 19$
  - violates also k-anonymity for any $k > 2$ since it reflects the existence of 2 divorced and female respondents

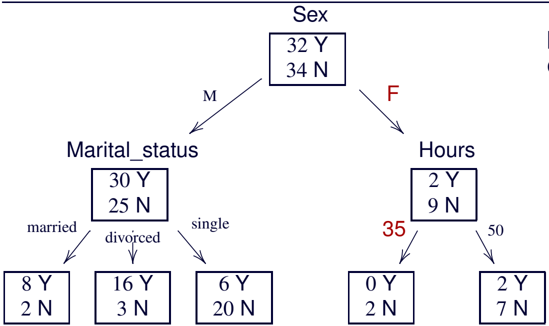
Classification mining – Decision trees

Marital_status	*Sex*	*Hours*	*#tuples (Hyp. values)*
divorced	M	35	2 (0Y, 2N)
divorced	M	40	17 (16Y, 1N)
divorced	F	35	2 (0Y, 2N)
married	M	35	10 (8Y, 2N)
married	F	50	9 (2Y, 7N)
single	M	40	26 (6Y, 20N)



Classification mining – Decision trees

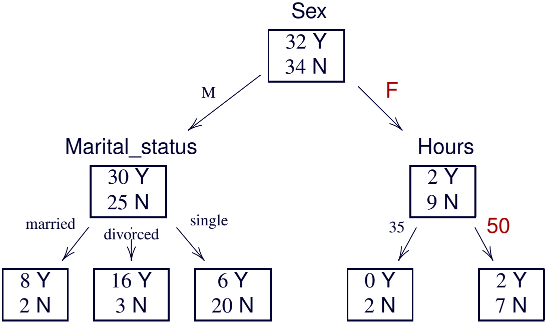
Marital_status	*Sex*	*Hours*	*#tuples (Hyp. values)*
divorced	M	35	2 (0Y, 2N)
divorced	M	40	17 (16Y, 1N)
divorced	F	35	2 (0Y, 2N)
married	M	35	10 (8Y, 2N)
married	F	50	9 (2Y, 7N)
single	M	40	26 (6Y, 20N)



path ⟨F,35⟩ implies the existence of 2 females working 35 hours

Classification mining – Decision trees

Marital_status	*Sex*	*Hours*	*#tuples (Hyp. values)*
divorced	M	35	2 (0Y, 2N)
divorced	M	40	17 (16Y, 1N)
divorced	F	35	2 (0Y, 2N)
married	M	35	10 (8Y, 2N)
married	F	50	9 (2Y, 7N)
single	M	40	26 (6Y, 20N)

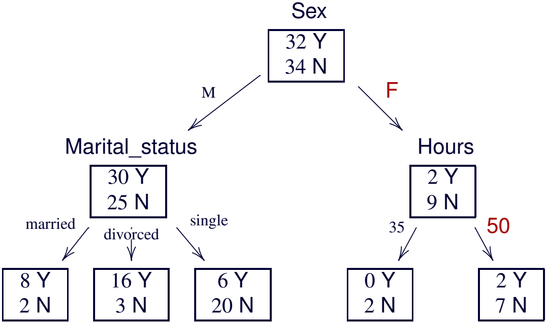


path ⟨F,35⟩ implies the existence of 2 females working 35 hours

paths ⟨F⟩ (#11) and ⟨F,50⟩ (#9) imply the existence of 2 females who do not work 50 hours per week

Classification mining – Decision trees

Marital_status	*Sex*	*Hours*	*#tuples (Hyp. values)*
divorced	M	35	2 (0Y, 2N)
divorced	M	40	17 (16Y, 1N)
divorced	F	35	2 (0Y, 2N)
married	M	35	10 (8Y, 2N)
married	F	50	9 (2Y, 7N)
single	M	40	26 (6Y, 20N)



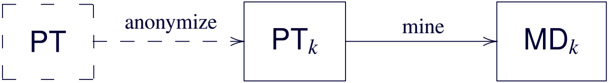
path ⟨F,35⟩ implies the existence of 2 females working 35 hours

paths ⟨F⟩ (#11) and ⟨F,50⟩ (#9) imply the existence of 2 females who do not work 50 hours per week

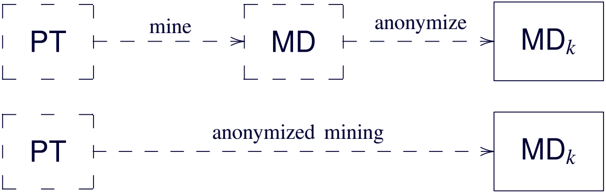
If QI includes Sex and Hours $\Longrightarrow$ k-anonym. is violated for any $k > 2$

Approaches for combining k-anonymity and data mining

Anonymize-and-Mine



Mine-and-Anonymize



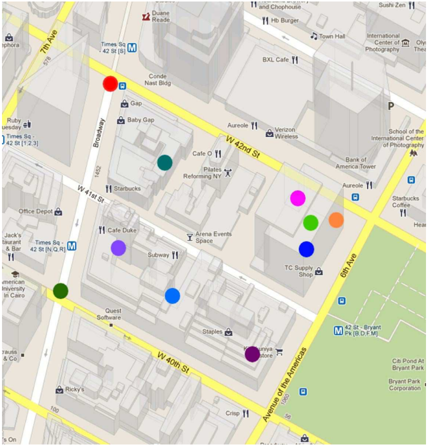
k-anonymity in location-based services

Protect identity of people in locations by considering always locations that contain no less than k individuals:

k-anonymity in location-based services

Protect identity of people in locations by considering always locations that contain no less than k individuals:

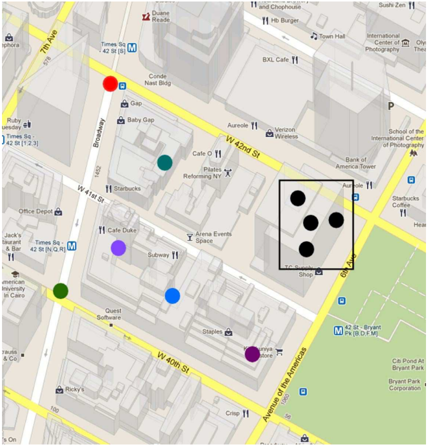
- enlarge the area to include at least other k-1 users (k-anonymity)



k-anonymity in location-based services

Protect identity of people in locations by considering always locations that contain no less than k individuals:

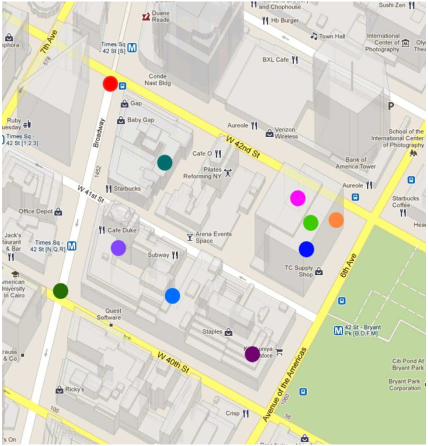
- enlarge the area to include at least other k-1 users (k-anonymity)



Privacy in location-based applications

Protect identity of people in locations by considering always locations that contain no less than k individuals:

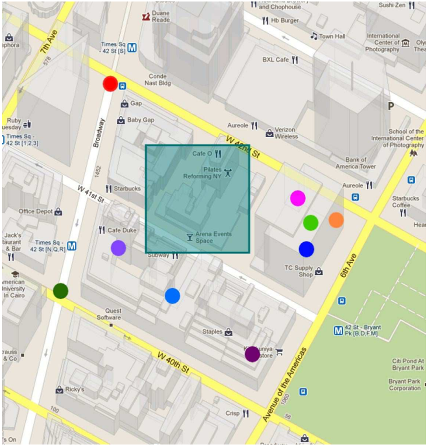
- enlarge the area to include at least other k-1 users (k-anonymity)
- protect the location of users (location privacy)



Privacy in location-based applications

Protect identity of people in locations by considering always locations that contain no less than k individuals:

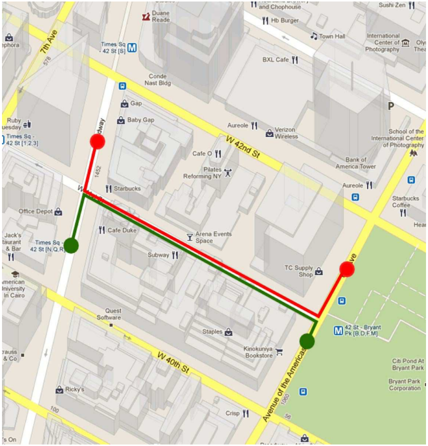
- enlarge the area to include at least other k-1 users (k-anonymity)
- protect the location of users (location privacy)
  $\Longrightarrow$ obfuscate the area so to decrease its precision or confidence



Privacy in location-based applications

Protect identity of people in locations by considering always locations that contain no less than k individuals:

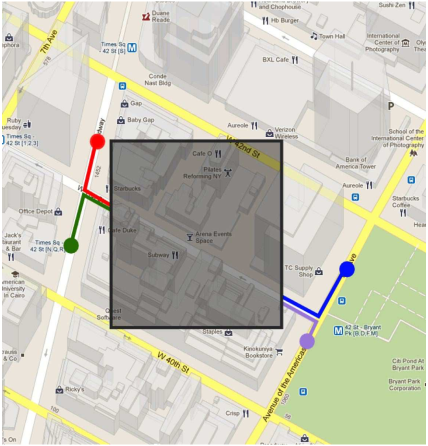
- enlarge the area to include at least other k-1 users (k-anonymity)
- protect the location of users (location privacy)
  $\Longrightarrow$ obfuscate the area so to decrease its precision or confidence
- protect the location path of users (trajectory privacy)



Privacy in location-based applications

Protect identity of people in locations by considering always locations that contain no less than k individuals:

- enlarge the area to include at least other k-1 users (k-anonymity)
- protect the location of users (location privacy)
  ⟹ obfuscate the area so to decrease its precision or confidence
- protect the location path of users (trajectory privacy) [ALS-12]
  ⟹ block tracking by mixing/ modifying trajectories



Re-identification with any information

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

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

- Two examples:
  - AOL
  - Netflix

AOL data release – 1

- In 2006, to embrace the vision of an open research community, AOL (America OnLine) publicly posted to a website 20 million search queries for 650,000 users of AOL's search engine summarizing three months of activity

- AOL suppressed any obviously identifying information such as AOL username and IP address

- AOL replaced these identifiers with unique identification numbers (this made searches by the same user linkable)

AOL data release – 2

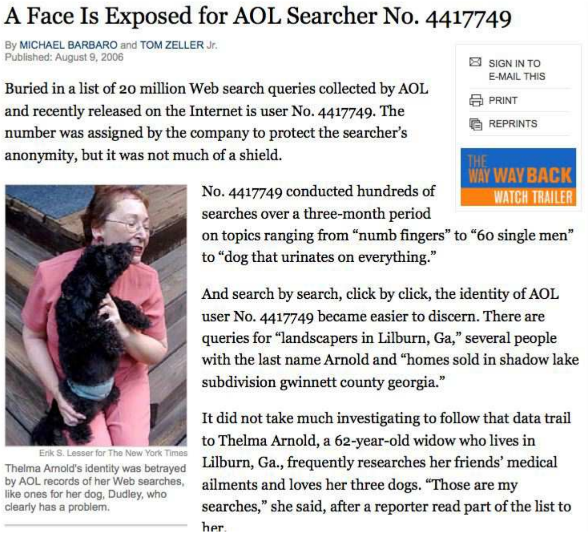
- User 4417749:
  - “numb fingers”, “60 single men”, “dog that urinates on everything”
  - “hand tremors”, “nicotine effects on the body”, “dry mouth”, and “bipolar”
  - “Arnold” (several people with this last name)
  - “landscapers in Lilburn, Ga”, “homes sold in shadow lake subdivision Gwinnett county, Georgia”

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

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

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

AOL data release – 3



A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr.
Published: August 9, 2006

THE WAY WAY BACK
WATCH TRAILER

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.

Erik S. Lesser for The New York Times

Thelma Arnold's identity was betrayed by AOL records of her Web searches, like ones for her dog, Dudley, who clearly has a problem.

No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from "numb fingers" to "60 single men" to "dog that urinates on everything."

And search by search, click by click, the identity of AOL user No. 4417749 became easier to discern. There are queries for "landscapers in Lilburn, Ga," several people with the last name Arnold and "homes sold in shadow lake subdivision gwinnett county georgia."

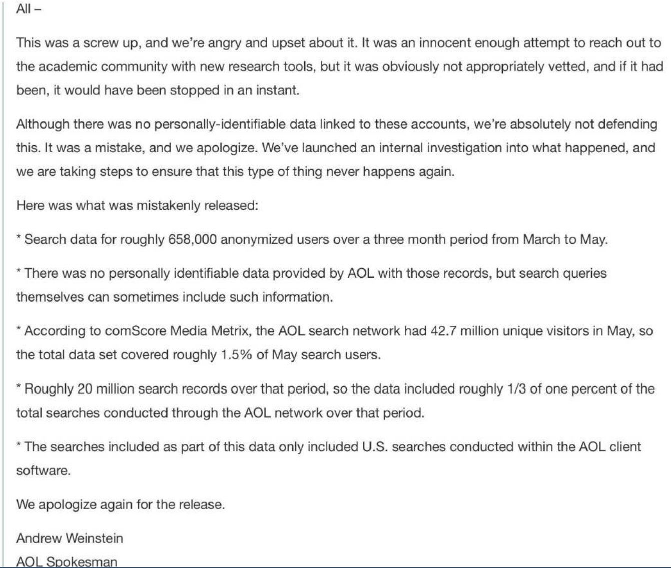
It did not take much investigating to follow that data trail to Thelma Arnold, a 62-year-old widow who lives in Lilburn, Ga., frequently researches her friends' medical ailments and loves her three dogs. "Those are my searches," she said, after a reporter read part of the list to her.

AOL data release – 4

What about user 17556639?

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

AOL data release – 5



All –

This was a screw up, and we're angry and upset about it. It was an innocent enough attempt to reach out to the academic community with new research tools, but it was obviously not appropriately vetted, and if it had been, it would have been stopped in an instant.

Although there was no personally-identifiable data linked to these accounts, we're absolutely not defending this. It was a mistake, and we apologize. We've launched an internal investigation into what happened, and we are taking steps to ensure that this type of thing never happens again.

Here was what was mistakenly released:

* Search data for roughly 658,000 anonymized users over a three month period from March to May.

* There was no personally identifiable data provided by AOL with those records, but search queries themselves can sometimes include such information.

* According to comScore Media Metrix, the AOL search network had 42.7 million unique visitors in May, so the total data set covered roughly 1.5% of May search users.

* Roughly 20 million search records over that period, so the data included roughly 1/3 of one percent of the total searches conducted through the AOL network over that period.

* The searches included as part of this data only included U.S. searches conducted within the AOL client software.

We apologize again for the release.

Andrew Weinstein

AOL Spokesman

Netflix prize data study – 1

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

Netflix prize data study – 2

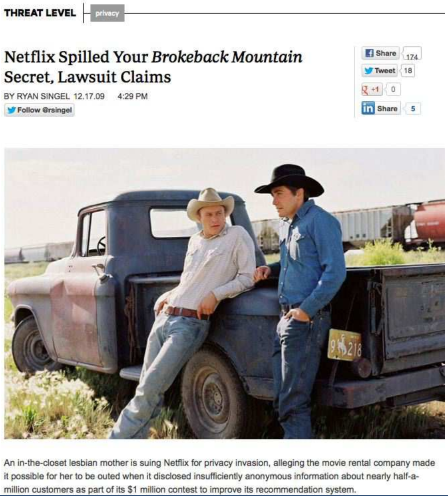
- Only a sample (one tenth) of the database was released
- Some ratings were perturbed (but not much, not to alter statistics)
- Identifying information (e.g., usernames) was removed, but a unique user identifier was assigned to preserve rating-to-rating continuity
- Release was not k-anonymous for any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

Netflix prize data study – 3

- De-identified Netflix data can be re-identified by linking with external sources (e.g., user ratings from IMDb users)
  - Knowing the precise ratings a person has assigned to six obscure (outside the top 500) movies, an adversary is able to uniquely identify that person 84% of the time
  - Knowing approximately when (± 2 weeks) a person has rated six movies (whether or not obscure), an adversary is able to reidentify that person in 99% of the cases
  - Knowing two movies a user has rated, with precise ratings and rating dates (± 3 days), an adversary is able to reidentify 68% of the users

Another example of privacy issue

Movies may reveal your political orientation, religious views, or sexual orientations (Netflix was sued by a lesbian for breaching her privacy)



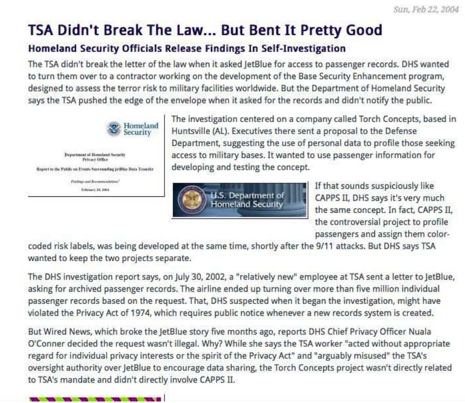
Netflix Spilled Your *Brokeback Mountain* Secret, Lawsuit Claims

BY RYAN SINGEL 12.17.09 4:29 PM

An in-the-closet lesbian mother is suing Netflix for privacy invasion, alleging the movie rental company made it possible for her to be outed when it disclosed insufficiently anonymous information about nearly half-a-million customers as part of its $1 million contest to improve its recommendation system.

JetBlue

- In 2003, JetBlue Airways Corporation gave the travel records of five million customers to Torch Concepts (a private DoD contractor) for an antiterrorism study to track high-risk passengers or suspected terrorists
- Torch Concepts purchased additional customer demographic information (e.g., SSN) about these passengers from Axciom, one of the largest data aggregation companies in the U.S.
- The information from JetBlue and Axciom was then used by Torch Concepts to develop passenger profiles
- Claims of violation of JetBlue Privacy Policy


Homeland Security
Department of Homeland Security
Privacy Office
Report to the Public on Events Surrounding jetBlue Data Transfer
Findings and Recommendations
February 20, 2004
U.S. Department of Homeland Security

Sun, Feb 22, 2004

TSA Didn't Break The Law... But Bent It Pretty Good

Homeland Security Officials Release Findings In Self-Investigation

The TSA didn't break the letter of the law when it asked JetBlue for access to passenger records. DHS wanted to turn them over to a contractor working on the development of the Base Security Enhancement program, designed to assess the terror risk to military facilities worldwide. But the Department of Homeland Security says the TSA pushed the edge of the envelope when it asked for the records and didn't notify the public.

The investigation centered on a company called Torch Concepts, based in Huntsville (AL). Executives there sent a proposal to the Defense Department, suggesting the use of personal data to profile those seeking access to military bases. It wanted to use passenger information for developing and testing the concept.

If that sounds suspiciously like CAPPS II, DHS says it's very much the same concept. In fact, CAPPS II, the controversial project to profile passengers and assign them color-coded risk labels, was being developed at the same time, shortly after the 9/11 attacks. But DHS says TSA wanted to keep the two projects separate.

The DHS investigation report says, on July 30, 2002, a "relatively new" employee at TSA sent a letter to JetBlue, asking for archived passenger records. The airline ended up turning over more than five million individual passenger records based on the request. That, DHS suspected when it began the investigation, might have violated the Privacy Act of 1974, which requires public notice whenever a new records system is created.

But Wired News, which broke the JetBlue story five months ago, reports DHS Chief Privacy Officer Nuala O'Conner decided the request wasn't illegal. Why? While she says the TSA worker "acted without appropriate regard for individual privacy interests or the spirit of the Privacy Act" and "arguably misused" the TSA's oversight authority over JetBlue to encourage data sharing, the Torch Concepts project wasn't directly related to TSA's mandate and didn't directly involve CAPPS II.

Syntactic vs semantic privacy definitions

- Syntactic privacy definitions capture the protection degree enjoyed by data respondents with a numerical value

  E.g., each release of data must be indistinguishably related to no less than a certain number of individuals in the population

- Semantic privacy definitions are based on the satisfaction of a semantic privacy requirement by the mechanism chosen for releasing the data

  E.g., the result of an analysis carried out on a released dataset must be insensitive to the insertion or deletion of a tuple in the dataset

Differential privacy

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

- It defines a property on the data release mechanism

k-anonymity vs differential privacy

Each has its strengths and weaknesses, e.g.,

k-anonymity:

- + nice capturing of real-world requirement
- – not complete protection

Differential privacy:

- + better protection guarantees
- – not easy to understand/enforce, not guaranteeing complete protection either

Still work to be done on both fronts

Some Examples of Other Privacy Issues

Privacy and genomic data

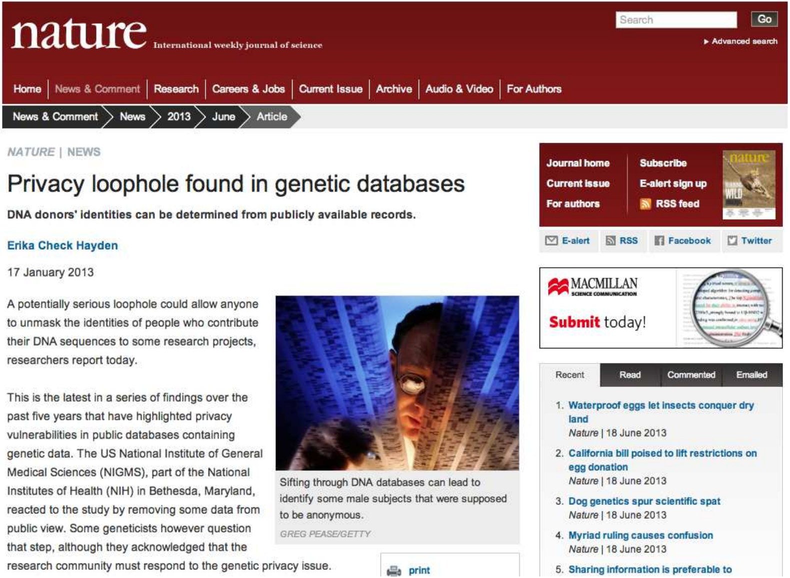
Privacy and genomic data

Genomic information is an opportunity for medicine but there are several privacy issues to be addressed

E.g., human genome:

- identifies its owner
- contains information about ethnic heritage, predisposition to several diseases, and other phenotypic traits
- discloses information about the relatives and descendants of the genome's owner

Individuals’ re-identification – 1



nature International weekly journal of science
Search
Go
▸ Advanced search
Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Authors
News & Comment > News > 2013 > June > Article

NATURE | NEWS

Privacy loophole found in genetic databases

DNA donors' identities can be determined from publicly available records.

Erika Check Hayden

17 January 2013

A potentially serious loophole could allow anyone to unmask the identities of people who contribute their DNA sequences to some research projects, researchers report today.

This is the latest in a series of findings over the past five years that have highlighted privacy vulnerabilities in public databases containing genetic data. The US National Institute of General Medical Sciences (NIGMS), part of the National Institutes of Health (NIH) in Bethesda, Maryland, reacted to the study by removing some data from public view. Some geneticists however question that step, although they acknowledged that the research community must respond to the genetic privacy issue.

Sifting through DNA databases can lead to identify some male subjects that were supposed to be anonymous.

GREG PEASE/GETTY

MACMILLAN SCIENCE COMMUNICATION
Submit today!

1. Waterproof eggs let insects conquer dry land
   Nature | 18 June 2013
2. California bill poised to lift restrictions on egg donation
   Nature | 18 June 2013
3. Dog genetics spur scientific spat
   Nature | 18 June 2013
4. Myriad ruling causes confusion
   Nature | 18 June 2013
5. Sharing information is preferable to

Individuals' re-identification – 2

The 1000 Genomes Project: international project (2008) to establish a catalogue of human genetic variation

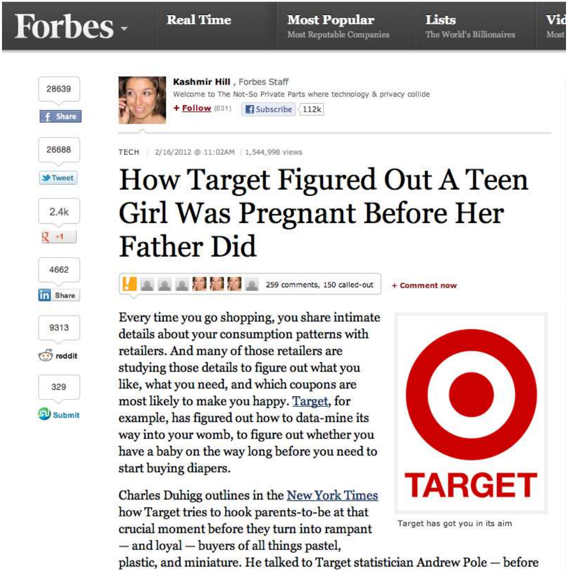
- Five men involved in both the 1000 Genomes Project and a project that studied Mormon families from Utah have been re-identified
    - their identities were determined
    - identities of their male and female relatives were also discovered
- Cross-reference analysis by the Whitehead Institute for Biomedical Research in Cambridge (MA):
    1. extract the haplotypes of short tandem repeats on the donor's Y chromosome (only for males)
    2. enter the haplotypes into genealogical databases to find possible surnames of the donor
    3. enter the surnames into demographic databases

Sensitive inference from data mining

The Target case – 1

- Target is the second-largest discount retailer in the U.S.
- Target assigns every customer a Guest ID number:
  - tied to credit card, name, email address, . . .
  - stores history of bought goods and other (bought) information
  - mining on these data for targeted advertising

The Target case – 2



Kashmir Hill, Forbes Staff

Welcome to The Not-So Private Parts where technology & privacy collide

TECH | 2/16/2012 @ 11:02AM | 1,544,998 views

How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are studying those details to figure out what you like, what you need, and which coupons are most likely to make you happy. Target, for example, has figured out how to data-mine its way into your womb, to figure out whether you have a baby on the way long before you need to start buying diapers.

Target has got you in its aim

Charles Duhigg outlines in the New York Times how Target tries to hook parents-to-be at that crucial moment before they turn into rampant — and loyal — buyers of all things pastel, plastic, and miniature. He talked to Target statistician Andrew Pole — before

The Target case – 3

- Analysts at Target identified $\sim$ 25 products that assign each shopper a pregnancy prediction score
  - e.g., woman, 23 y.o., buying in March cocoa-butter lotion, a purse large enough to double as a diaper bag, zinc and magnesium supplements and a bright blue rug $\Longrightarrow$ 87% due late August
  - due time in a small window to send coupons timed to very specific stages of a pregnancy
- Mining data reveals customers' major life events (e.g., graduating from college or getting a new job or moving to a new town)
  - shopping habits became flexible, predictable, and potential gold mines for retailers
  - between 2002 (starting of similar campaigns) and 2010 Target's revenues grew from $44B to $67B

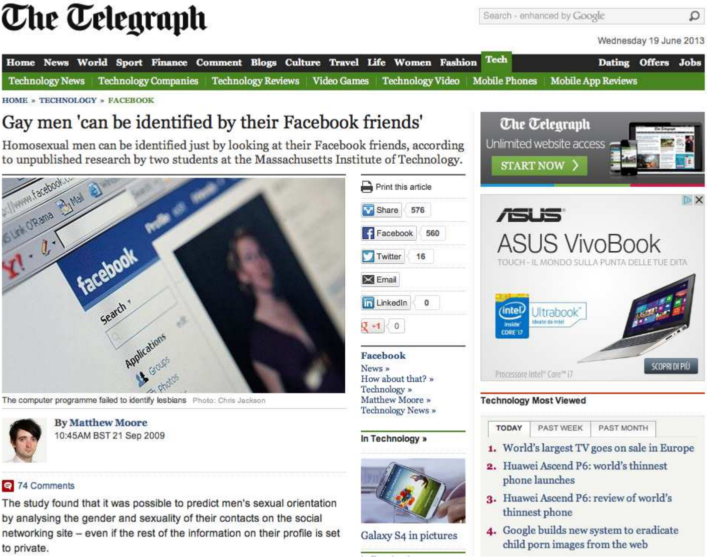
Inferences from social networks – 1

- People tend to connect with others with similar interests / activities / experiences . . .
- What one discloses exposes not only him/her but also others

EXAMPLE: sexual orientation

- a study in 2009 on 1,500 Facebook users showed that homosexual men have more homosexual friends than heterosexual men
- tool to automatically predict the sexual orientation of Facebook users (not indicating it) based on their friends' orientations
- run on 10 men known to be homosexual but not revealing this information on their profiles, the tool correctly inferred it

Inferences from social networks – 2



The Telegraph

Search - enhanced by Google

Wednesday 19 June 2013

Home News World Sport Finance Comment Blogs Culture Travel Life Women Fashion Tech Dating Offers Jobs

Technology News | Technology Companies | Technology Reviews | Video Games | Technology Video | Mobile Phones | Mobile App Reviews

HOME » TECHNOLOGY » FACEBOOK

Gay men 'can be identified by their Facebook friends'

Homosexual men can be identified just by looking at their Facebook friends, according to unpublished research by two students at the Massachusetts Institute of Technology.

The computer programme failed to identify lesbians Photo: Chris Jackson

By Matthew Moore
10:45AM BST 21 Sep 2009

74 Comments

The study found that it was possible to predict men's sexual orientation by analysing the gender and sexuality of their contacts on the social networking site – even if the rest of the information on their profile is set to private.

Print this article

Share 576

Facebook 560

Twitter 16

Email

LinkedIn 0

+1 0

Facebook
News »
How about that? »
Technology »
Matthew Moore »
Technology News »

In Technology »

Galaxy S4 in pictures

The Telegraph
Unlimited website access
START NOW

ASUS
ASUS VivoBook
TOUCH - IL MONDO SULLA PUNTA DELLE TUE DITA
intel inside CORE i7 Ultrabook
SCOPRI DI PIÙ
Processore Intel® Core™ i7

Technology Most Viewed

TODAY	PAST WEEK	PAST MONTH

1. World's largest TV goes on sale in Europe
2. Huawei Ascend P6: world's thinnest phone launches
3. Huawei Ascend P6: review of world's thinnest phone
4. Google builds new system to eradicate child porn images from the web

to be continued . . .

Differential Privacy

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

Our world is guided by data

Source (http://www.agencypja.com/site/assets/files/1826/marketingdata-1

Data are invaluable

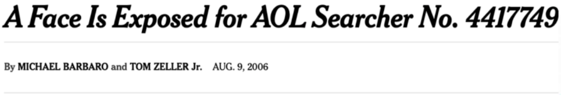
- The big data concept has been adopted by many companies
  ⟹ entered the public vocabolary

- Data are mostly about individuals whose privacy must be ensured

- How can we work on private data?
  - anonymize them and share

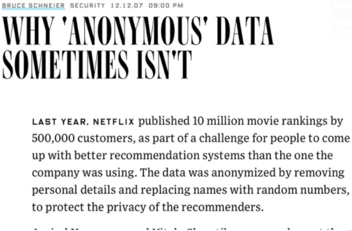
… but …

. . .Anonymity is not enough!



A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr. AUG. 9, 2006



BRUCE SCHNEIER SECURITY 12.12.07 09:00 PM

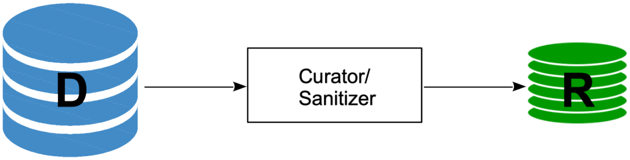
WHY 'ANONYMOUS' DATA SOMETIMES ISN'T

LAST YEAR, NETFLIX published 10 million movie rankings by 500,000 customers, as part of a challenge for people to come up with better recommendation systems than the one the company was using. The data was anonymized by removing personal details and replacing names with random numbers, to protect the privacy of the recommenders.

"Anonymous" Genomes Identified

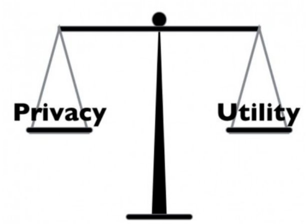
The names and addresses of people participating in the Personal Genome Project can be easily tracked down despite such data being left off their online profiles.

Basic scenario



Database
(set of records, one per individual)

Released data



Classic intuition for privacy

I would feel safe being in a database D if:

- I knew that my data had no impact on the released results
  $\Longrightarrow$ computation over "D without me" = computation over "D"

- I knew that the information learned about an individual by the published results R is no more than the information we can learn about that individual without access to R

Classic intuition for privacy

I would feel safe being in a database D if:

- I know that my data had no impact on the released results $\Longrightarrow$ computation over "D without me" = computation over "D"

If individuals had no impact on the released results…
then the results would have no utility!

- I knew that the information learned about an individual by the published results R is no more than the information we can learn about that individual without access to R

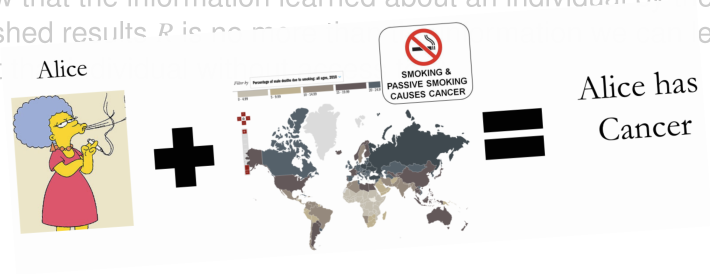
Classic intuition for privacy

I would feel safe being in a database D if:

- I know that my data had no impact on the released results
  $\Longrightarrow$ computation over "D without me" = computation over "D"

If individuals had no impact on the released results…
then the results would have no utility!

- I knew that the information learned about an individual by the published results R is no more than the information we can learn about the individual without


Alice
Filter by Percentage of male deaths due to smoking: all ages, 2010
0 - 4.99
5 - 9.99
10 - 14.99
15 - 19.99
20 - 24.9
SMOKING & PASSIVE SMOKING CAUSES CANCER
Alice has Cancer

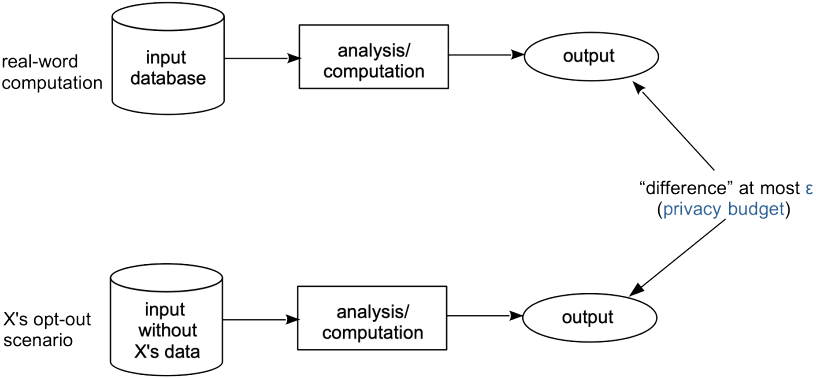
Differential privacy – Intuition (1)

- With or without including Alice in the database, her privacy risk should not change much

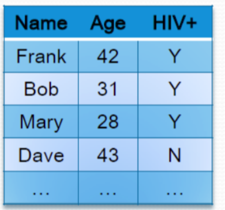
  $\Longrightarrow$ the privacy of an individual is protected whenever the result R does not depend on her specific information

- Inferences about an individual from a differentially private computation are (essentially) limited to what could be inferred from everyone else's data without her own data being included in the computation

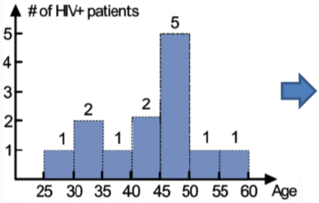
Differential privacy – Intuition (2)

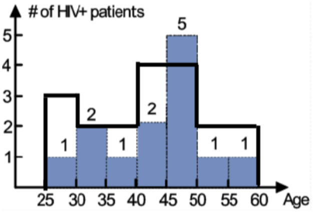


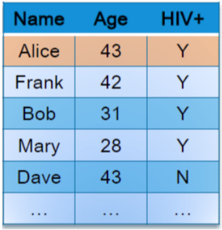
Differential privacy - An example



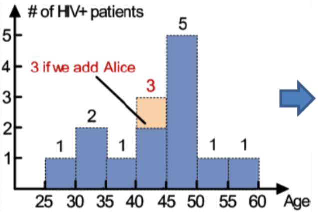
Name	Age	HIV+
Frank	42	Y
Bob	31	Y
Mary	28	Y
Dave	43	N
...	...	...

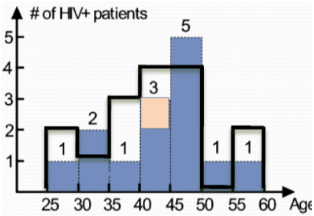






Name	Age	HIV+
Alice	43	Y
Frank	42	Y
Bob	31	Y
Mary	28	Y
Dave	43	N
...	...	...





Original records

Original histogram

Perturbed histogram with differential privacy

Differential privacy and randomness

Differentially private analyses add random noise to the result

- Noise masks the differences between the real-world computation and the opt-out scenario of each individual in the database

- The outcome of a differentially private analysis is not exact but an approximation

- A differentially private analysis may, if performed twice on the same dataset, return different results
    - it is often possible to calculate accuracy bounds for the analysis

Differential privacy – Definition

Let databases D and D' be two neighbors database (e.g., they are the same apart from one of them not having the data of a single individual)

- An algorithm A satisfies ε-differential privacy if for all pairs of neighbor databases D, D', and for all outputs o:

  $$\mathrm{P}[A(D) = o] \leq e^{\varepsilon}\ \mathrm{P}[A(D') = o]$$

  $\Longrightarrow$ an adversary should not be able to use o to distinguish between any D and D'

The privacy budget ε

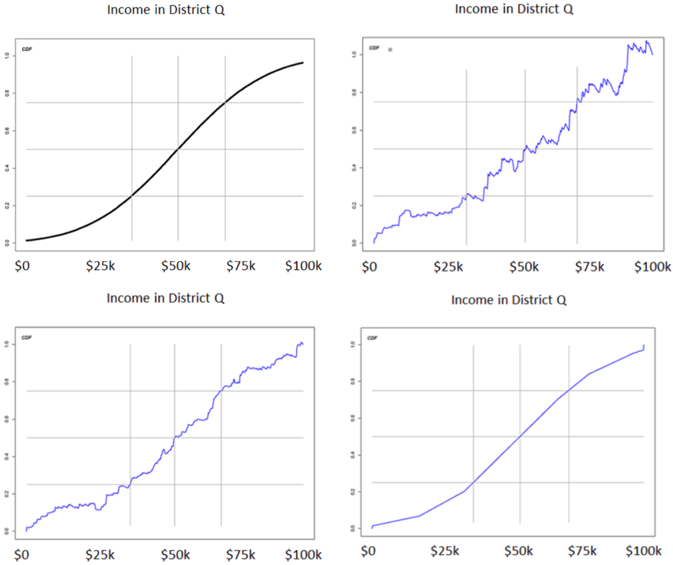
- Determine how much noise is added to the computation
  $\Longrightarrow$ trade-off between privacy and accuracy

- The smaller (larger) the ε the more (less) the noise
  - small $\varepsilon \Longrightarrow$ more privacy, less utility and
  - large $\varepsilon \Longrightarrow$ less privacy, more utility

EXAMPLE

- $\varepsilon = 0 \Longrightarrow$ an analysis could not provide any meaningful output
- $\varepsilon = 0.1 \Longrightarrow$ it provides strong privacy guarantees and useful statistics
- $\varepsilon = 1 \Longrightarrow$ it provides high accuracy but low privacy

Differential privacy and accuracy



$\varepsilon = 0.005; \varepsilon = 0.01; \varepsilon = 0.1$

How to achieve differential privacy

- Need to calibrate the noise to the influence an individual can have on the result

- Global sensitivity: characterizes the scale of the influence of one individual (worst case), and hence how much noise we must add

Global sensitivity – Examples

Database D of patients

- How many patients suffer from diabetes?

Real-world (D)	**Opt-out (D')**
50	49

Global sensitivity – Examples

Database D of patients

- How many patients suffer from diabetes?

Real-world (D)	**Opt-out (D')**
50	**49**

GS(A)=1

Global sensitivity – Examples

Database D of patients

- How many males and females are in the database?

Real-world (D)		**Opt-out (D')**	
M	F	M	F
22	34	21	34

- How many patients suffer from diabetes?

Real-world (D)	**Opt-out (D')**
50	49

Global sensitivity – Examples

Database D of patients

- How many males and females are in the database?

Real-world (D)		**Opt-out (D')**	
M	F	M	F
22	34	**21**	34

- How many patients suffer from diabetes?

Real-world (D)	**Opt-out (D')**
50	**49**

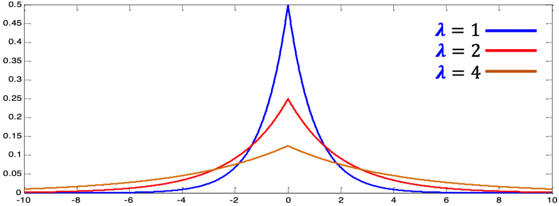
GS(A)=2

Laplace Mechanism with Sensitivity

- Result R is sampled from a Laplace distribution with mean the true result and some scale λ (determined by ε and the global sensitivity of the computation)

$$R = A(D) + Z$$

Z is a random variable drawn from the Laplace distribution

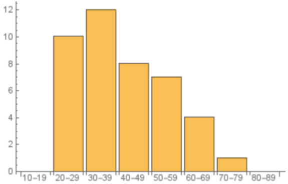


$$\text{Lap}(z, \lambda) = P(z \mid \lambda) = \frac{1}{2\lambda} e^{\frac{-|z|}{\lambda}}, \ \lambda = \frac{GS(A)}{\varepsilon}$$

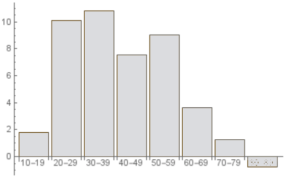
Properties of Differential Privacy

Closure under post-processing

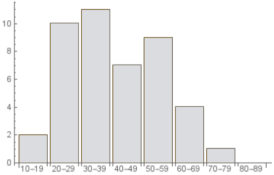
- Differential privacy is resilient to post-processing
  ⟹ the computation of a function over the result of a differentially private computation cannot make it less differentially private



number of users depending on their age ranges ...



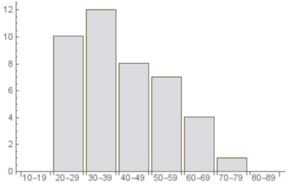
...after the addition of Laplace noise ...



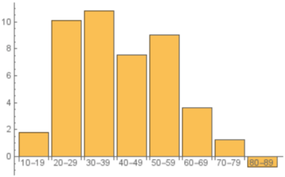
...after rounding all counts and replacing negative numbers with 0

Closure under post-processing

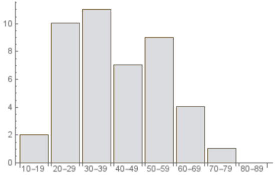
- Differential privacy is resilient to post-processing
  $\Longrightarrow$ the computation of a function over the result of a differentially private computation cannot make it less differentially private



number of users depending on their age ranges ...



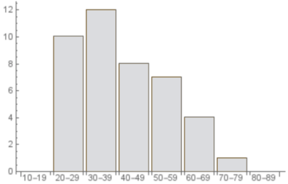
...after the addition of Laplace noise ...



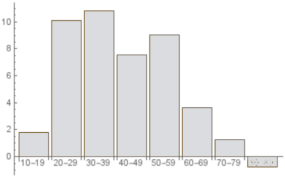
...after rounding all counts and replacing negative numbers with 0

Closure under post-processing

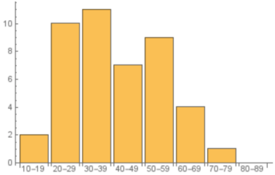
- Differential privacy is resilient to post-processing
  $\Longrightarrow$ the computation of a function over the result of a differentially private computation cannot make it less differentially private



number of users depending on their age ranges ...



...after the addition of Laplace noise ...



...after rounding all counts and replacing negative numbers with 0

Sequential and parallel composition

Differential privacy composes well with itself. But what does it mean?

- Sequencial composition: sequence of m computations over database D with overlapping results

Sequential and parallel composition

Differential privacy composes well with itself. But what does it mean?

- Sequencial composition: sequence of m computations over database D with overlapping results

$$\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_m$$

Sequential and parallel composition

Differential privacy composes well with itself. But what does it mean?

- Sequencial composition: sequence of m computations over database D with overlapping results

$$\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_m$$

- Parallel composition: sequence of m computations over disjoint subsets of a database D

Sequential and parallel composition

Differential privacy composes well with itself. But what does it mean?

- Sequencial composition: sequence of m computations over database D with overlapping results

$$\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_m$$

- Parallel composition: sequence of m computations over disjoint subsets of a database D

$$\max(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m)$$

Sequencial composition – Example

Privacy budget ε

Ask for count of female patients and count of patients suffering from diabetes

Females
34

Diabetes
23

- Cells can be overlapping (e.g., a female who suffers from diabetes)
- Each count must be released in such a way that ε_1 (first count) $+$ ε_2 (second count) be equal to ε

Parallel composition – Example

Privacy budget ε

Ask for count of people broken down by handedness, hair color

	Redhead	**Blond**	**Brunette**
Left-handed	23	35	56
Right-handed	215	360	493

- Each cell is a disjoint set of individuals
- Each cell can be released with ε-differential privacy

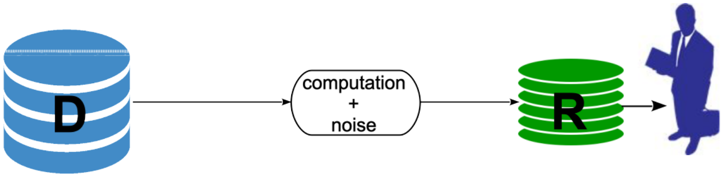
Group privacy

- Differential privacy has been introduced for reasoning about the privacy of a single individual but allows also reasoning about the privacy of groups

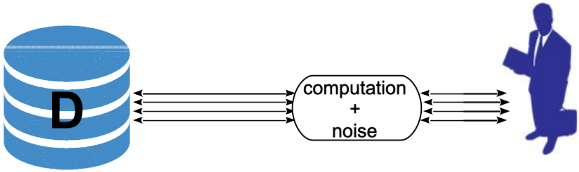
- Privacy guarantees that apply to an individual with ε apply to a group of size n with the privacy parameter becoming $n\varepsilon$

Differential Privacy Models

Non interactive vs interactive

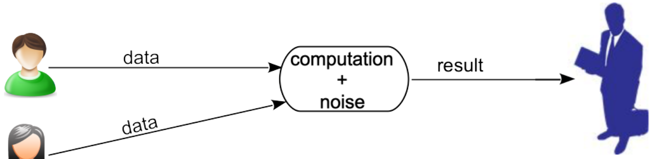


Non-interactive model



Interactive model

Global vs local differential privacy



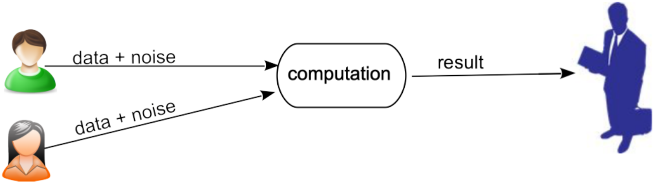
data

data

computation
+
noise

result

Global differential privacy



data + noise

data + noise

computation

result

Local differential privacy

Basic idea behind local differential privacy

- Each user runs a differential private algorithm on her data
- An external party (not necessarly trusted) combines all the (noised) data received from the users to get a final result
- Noise can cancel out or be subtracted
- True answer plus noise; noise is typically larger than in the global case

Local differential privacy – Definition

- A randomized algorithm A satisfies ε-local differential privacy iff for all input x, x' and output o of A:

$$\mathsf{P}[A(x) = o] \leq e^{\varepsilon}\ \mathsf{P}[A(x') = o]$$

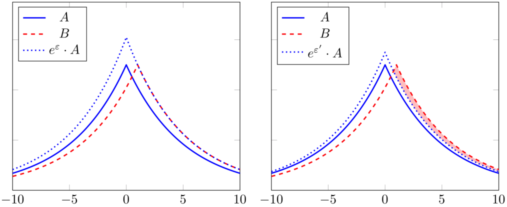
$\Longrightarrow$ any output should no depend on user's secret

Differential Privacy in the Real World

Privacy in practice – 1

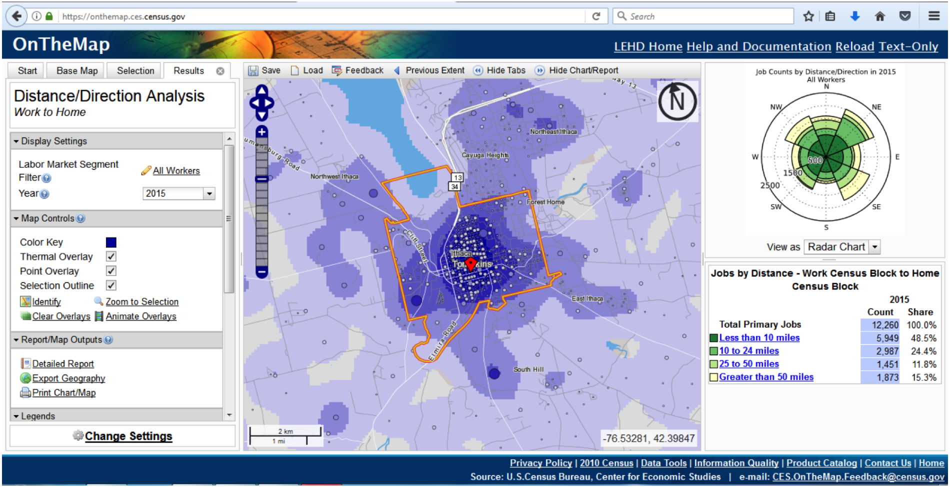
- In 2008 United States Census Bureau deployed OnTheMap, a web-based application that shows where workers are employed and where they live

- Based on a varion of ε-differential privacy, called approximate differential privacy ((ε, δ)-differential privacy):
  - ε is the privacy budget
  - δ is related to the confidence ($1 - \delta$) that the result satisfies ε-differential privacy



Privacy in practice – 2

OnTheMap: $\varepsilon = 8.99$ and $\delta = 0.000001$



Distance/Direction Analysis

Work to Home

Display Settings

Labor Market Segment Filter: All Workers

Year: 2015

Map Controls

Color Key ■

Thermal Overlay ☑

Point Overlay ☑

Selection Outline ☑

Report/Map Outputs

Legends

Jobs by Distance - Work Census Block to Home Census Block

	2015	
	Count	Share
Total Primary Jobs	12,260	100.0%
Less than 10 miles	5,949	48.5%
10 to 24 miles	2,987	24.4%
25 to 50 miles	1,451	11.8%
Greater than 50 miles	1,873	15.3%

Source: U.S.Census Bureau, Center for Economic Studies | e-mail: CES.OnTheMap.Feedback@census.gov

Privacy in practice – 3

- Internal experiments confirmed that confidential microdata from the 2010 Census can be reconstructed quite accurately

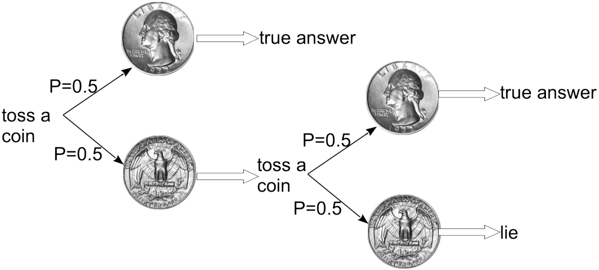
- United States Census Bureau has adopted a new differentially private mechanism for statistical disclosure control in the 2020 Census

- Unclear exactly how they will set ε, a Policy Committee (the Data Stewardship Executive Policy Committee - not technical staff) will decide on the value of ε

Privacy in practice – 4

- Differential privacy based on coin tossing is widely deployed
  - Google Chrome browser to collect browsing statistics (Rappor)
  - Apple iOS and MacOS to collect typing statistics

- All deployments are based on randomized response



  - $P(R = yes \mid Truth = yes) = 3/4 = 1/2 + (1/2 \cdot 1/2)$
  - $P(R = yes \mid Truth = no) = 1/4 = 1/2 \cdot 1/2$

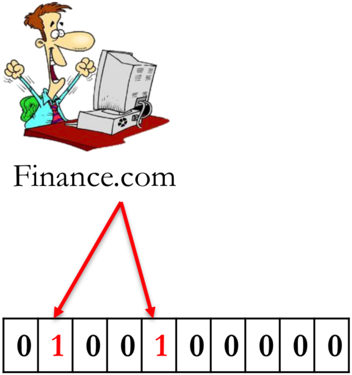
How Rappor works – 1

- Each user has one value v out of a very large set of possibilities (e.g., URL, www.unimi.it)

- Rappor solution is based on:
  - Bloom Filter
  - two levels of randomized response: permanent and instantaneous

How Rappor works – 2

Compression: use h hash functions to hash input string to k-bit vector (Bloom Filter)



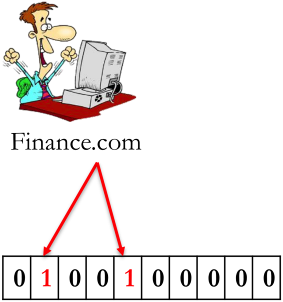
0	1	0	0	1	0	0	0	0	0

Bloom Filter *B*

How Rappor works – 2

Permanent randomized response: from B a B' permanent randomized response is created with (user tunable) probability parameter f

B' is memorized and will be used for all future reports



$$B'_i = \begin{cases} 1, & \text{with probability } \frac{1}{2}f \\ 0, & \text{with probability } \frac{1}{2}f \\ B_i, & \text{with probability } 1-f \end{cases}$$

0	1	0	0	1	0	0	0	0	0

Bloom Filter B

⇨

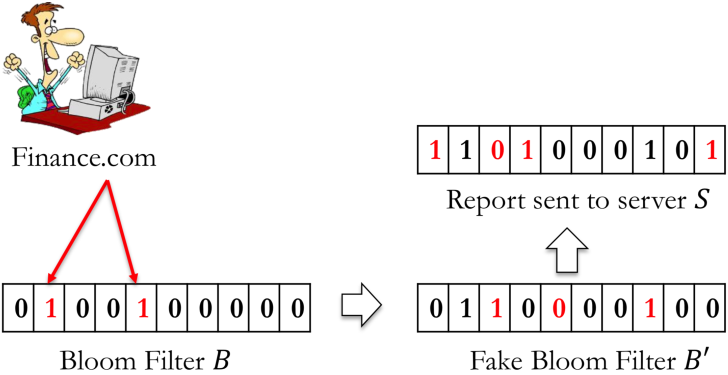
0	1	1	0	0	0	0	1	0	0

Fake Bloom Filter B'

How Rappor works – 2

Instantaneous randomized response: send a report to the server of size k bit generated from B'

- Flip bit value 1 with probability 1-q
- Flip bit value 0 with probability p



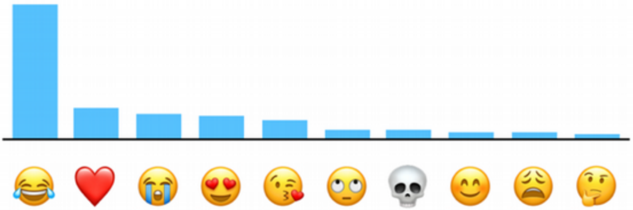
Bloom Filter B: 0 1 0 0 1 0 0 0 0 0

Fake Bloom Filter B': 0 1 1 0 0 0 0 1 0 0

Report sent to server S: 1 1 0 1 0 0 0 1 0 1

Apple at work

- Apple collects data from iOS and OS X users
  - Popular emojis: (heart) (laugh) (smile) (crying) (sadface)
  - “New” words: bruh, hun, bae, tryna, despacito, mayweather
  - Which websites to mute, which to autoplay audio on!



The Count Mean Sketch technique allows Apple to determine the most popular emoji to help design better ways to find and use our favorite emoji. The top emoji for US English speakers contained some surprising favorites.

What is the privacy budget ε?

- Google
  - $\varepsilon =$2 for particular data that are uploaded
  - $\varepsilon =$8-9 is an upper limit over the lifetime of the user
- Apple
  - $\varepsilon =$6 for macOS
  - $\varepsilon =$14 for iOS 10
  - $\varepsilon =$43 for beta version iOS 11 (version unknown)

Frank McSherry (one of the inventor of differential privacy):

> *Say someone has told their phone's health app they have a one-in-a-million medical condition, and their phone uploads that data to the phone's creator on a daily basis, using differential privacy with an epsilon of 14. After one upload obfuscated with an injection of random data, the company's data analysts would be able to figure out with 50 percent certainty whether the person had the condition. After two days of uploads, the analysts would know about that medical condition with virtually 100 percent certainty.*

Problems with Differential Privacy

Sensitivity of computations

- Count, histogram computations: differential privacy works well (presence/absence of a single record can change the result slightly)

- Sum computation: the application of differential privacy can be a problem:

  What is the total income earned by men vs women?
  A single very high income $\Longrightarrow$ lot of noise for this worst-case individual

- How to set ε? What happens when the privacy budget has been exausted?

Authentication and Access Control

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

Security strategies

- Prevention: take measures that prevent your system from being damaged (e.g., lock the door)

- Detection: take measures that detect when, how, and by whom your system has been damaged (e.g., missing items from your house)

- Reaction: take measures so that you can recover your system from damages (e.g., call the police)

Security objectives

- Confidentiality: prevent unauthorized disclosure of information
- Integrity: prevent unauthorized modification of information
- Availability: guarantee that information (or resources) are always available to authorized users

Authentication

Identification and Authentication

- Provide the system with the ability of identifying its users and confirming their identity
  - Identification by the parties to be authenticated (users declare who they are and present proofs of this)
  - Authentication by the system doing the authentication (to be certain of the identity presented)
- Users authentication necessary for
  - access control
  - security logging

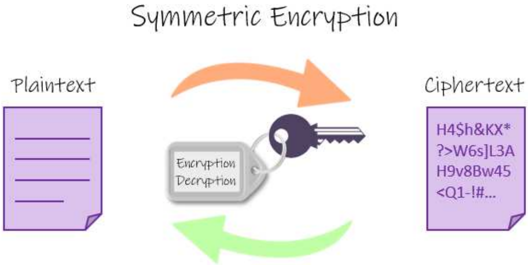
Cryptography

- Cryptography transforms a cleartext into a non intelligible (encrypted text or ciphertext) and viceversa

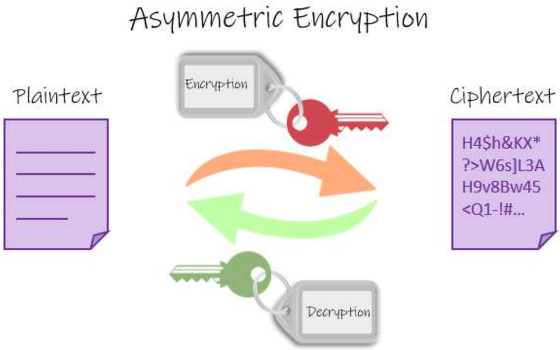
- Cryptography is based on the use of a key to encrypt and decrypt messages

- Classification of encryption algorithms
  - Symmetric encryption
    - the same (private) key is used for encryption and decryption
    - the key is secret and known to both sender and receiver
  - Asymmetric encryption
    - each subject possesses a pair of keys (⟨public,private⟩), one for encryption, the other for decryption
    - the private key is known only to the owner of the key pair
    - the public key is known to everybody

Symmetric encryption



Asymmetric encryption



Authentication

- Establishes the identity of a “party” to another “party”, where a party can be a user or a machine
- Often mutual authentication is needed
  - Authentication of a computer to a user can be needed to prevent attacks (e.g., spoofing, in which a computer masquerades as another one to acquire users passwords)
- Authentication can be considered the primary security service
- Correctness of the access control relies on a correct authentication
- Correctness of intrusion/violation control relies on correct authentication

User to computer authentication

Can be based on:

- something the user knows (e.g., password)
- something the user has (e.g., token)
- something the user is (e.g., biometric trait)

or a combination of the above (multi-factor authentication)

Password-based authentication

- Based on pairs
  - username: the user identifies herself
  - password: the user gives the proof of her identity
- It is the oldest and most widely used authentication method

  + simple

  + cheap

  + easily implementable

  – weakest

Vulnerabilities of passwords – 1

- Often passwords can be
  - easily guessed (guessing)
  - read by people observing the legitimate users typing it in (snooping)
  - observed by third parties when passing over the network (sniffed)
  - acquired by third parties impersonating the login interface (spoofing)
- Anybody that acquires the password of a user can impersonate the user (masquerading) in getting access to the system

Vulnerabilities of passwords – 2

One of the primary causes of password vulnerability is due to the users that do not choose or manage them properly.

ASSWORD
abc123
WEAK!
BUT 'ABC123' ARE THE ONLY LETTERS AND NUMBERS I CAN REMEMBER.

Causes of password vulnerability

- The first step to limit password vulnerability is good password management
- Often passwords are vulnerable because users do not put enough care
  - do not change passwords for a long time
  - share passwords with colleagues and friends
  - choose “weak” passwords because they are easy to remember (e.g., name or date of births of relatives or pets)
  - use the same password on different services
  - write password on a piece of paper not to forget it

Authentication based on possession

- Based on possession by users of tokens (small in size)
- Each token has a cryptographic key (stored in the token) that can be used to prove the identity of the token to a computer
- Tokens are safer than passwords: by keeping control on the tokens, users maintain control on their identity

Vulnerabilities of tokens

- Token-based authentication proves only the identity of the token, not the identity of the user
  - tokens can be lost, stolen, forged
  - everybody who acquires a token can impersonate the user
- Often token-based authentication is combined with authentication based on knowledge (two-factor authentication)
  - to masquerade as a user, third parties need both to have the token and to know the password

Authentication based on user characteristics – 1

- Based on biometric characteristics of the user
  - physical characteristics: fingerprints, face recognition, . . .
  - behavioral characteristics: typing cadence, signature, . . .
- Requires an initial enrollment phase that
  - performs several measures on the characteristic
  - defines a profile (template)

Authentication based on user characteristics – 2

- Authentication compares the characteristic measured for the user with the stored template

- Authentication succeeds if they correspond, provided a tolerance interval (to be properly tuned))

- Impose a maximum number of failed attempts

- Important to have a backup authentication factor

Access Control

Access control

- It evaluates access requests to the resources by the authenticated users and, based on some access rules, it determines whether they must be granted or denied
  - It may be limited to control only *direct* access
  - It may be enriched with inference, information flow, and *non-interference* controls

Access control vs other services

Correctness of access control rests on

- Proper user identification/authentication ⇒ No one should be able to acquire the privileges of someone else

- Correctness of the authorizations against which access is evaluated (which must be protected from improper modifications)

Access control and authentication

- Authentication also necessary for accountability and establishing responsibility

- Each principal (logged subject) should correspond to a single user $\Longrightarrow$ no shared accounts

- In open systems it should rely on authenticity of the information, in contrast to authenticity of the identity (authentication) $\Longrightarrow$ credential-based access control

Policies, models, and mechanisms

In studying access control, it is useful to separate

- Policy: it defines (high-level) guidelines and rules describing the accesses to be authorized by the system (e.g., closed vs open policies)
    - often the term policy is abused and used to refer to actual authorizations (e.g., Employees can read bulletin-board)
- Model: it formally defines the access control specification and enforcement
- Mechanism: it implements the policies via low level (software and hardware) functions

Separation between policies and mechanisms

The separation between policies and mechanisms allows us to:

- Discuss access requirements independent of their implementation
- Compare different access control policies as well as different mechanisms that enforce the same policy
- Design mechanisms able to enforce multiple policies

Access control mechanisms – 1

Based on the definition of a reference monitor that must be

- tamper-proof: cannot be altered
- non-bypassable: mediates all accesses to the system and its resources
- security kernel confined in a limited part of the system (scattering security functions all over the system implies all the code must be verified)
- small enough to be susceptible of rigorous verification methods

Access control mechanisms – 2

The implementation of a correct mechanism is far from being trivial and is complicated by need to cope with

- storage channels (residue problem) Storage elements such as memory pages and disk sectors must be cleared before being released to a new subject, to prevent data scavenging

- covert channels Channels that are not intended for information transfer (e.g., program's effect on the system load) that can be exploited to infer information

Assurance How well does the mechanism do?

Security policies

Security policies can be distinguished in

- Access control policies: define who can (or cannot) access the resources.
    - Discretionary (DAC)
    - Mandatory (MAC)
    - Role-based (RBAC)
    - Credential-based
    - Attribute-based (ABAC)
- Administrative policies: define who can specify authorizations/rules governing access control

Discretionary (DAC) policies: Basic approaches

Discretionary policies

Enforce access control on the basis of

- the identity of the requestors (or on properties they have)

- and explicit access rules that establish who can or cannot execute which actions on which resources

They are called discretionary as users can be given the ability of passing on their rights to other users (granting and revocation of rights regulated by an administrative policy)

Access Matrix model

- It provides a framework for describing protection systems
- Often reported as HRU model (from later formalization by Harrison, Ruzzo, and Ullmann)
- Called access matrix since the authorization state (or protection system) is represented as a matrix
- Abstract representation of protection system found in real systems (many subsequent systems may be classified as access matrix-based)

Access Matrix model – protection state

State of the system defined by a triple (S,O,A) where

- S set of subjects (who can exercise privileges)
- O set of objects (on which privileges can be exercised) subjects may be considered as objects, in which case $S \subseteq O$
- A access matrix, where
  - rows correspond to subjects
  - columns correspond to objects
  - $A[s,o]$ reports the privileges of s on o

Changes of states via commands calling primitive operations: **enter** r into $A[s,o]$, **delete** r from $A[s,o]$, **create subject** s', **destroy subject** s', **create object** o', **destroy object** o'

Access Matrix – Example

	File 1	**File 2**	**File 3**	**Program 1**
Ann	own read write	read write		execute
Bob	read		read write	
Carl		read		execute read

Access Matrix – implementation

Matrix is generally large and sparse
Storing the matrix $\Longrightarrow$ waste of memory space

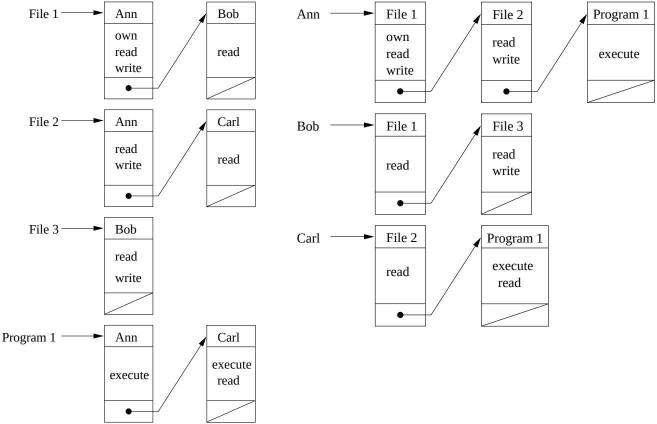
Alternative approaches

- Authorization table Store table of non-null triples (s,o,a)
  Generally used in DBMS
- Access control lists (ACLs) Store by column
- Capability lists (tickets) Store by row

Authorization Tables

User	Access mode	Object
Ann	own	File 1
Ann	read	File 1
Ann	write	File 1
Ann	read	File 2
Ann	write	File 2
Ann	execute	Program 1
Bob	read	File 1
Bob	read	File 2
Bob	write	File 2
Carl	read	File 2
Carl	execute	Program 1
Carl	read	Program 1

Access control lists vs. Capability Lists



File 1 → Ann: own, read, write → Bob: read

File 2 → Ann: read, write → Carl: read

File 3 → Bob: read, write

Program 1 → Ann: execute → Carl: execute, read

Ann → File 1: own, read, write → File 2: read, write → Program 1: execute

Bob → File 1: read → File 3: read, write

Carl → File 2: read → Program 1: execute, read

ACL vs Capabilities

- ACLs require authentication of subjects
- Capabilities do not require authentication of subjects, but require *unforgeability* and control of propagation of capabilities
- ACLs provide superior for access control and revocation on a per-object basis
- Capabilities provide superior for access control and revocation on a per-subject basis
- The per-object basis usually wins out so most systems are based on ACLs
- Some systems use abbreviated form of ACL (e.g., Unix 9 bits)

DAC weaknesses

Discretionary access controls constraint only direct access

No control on what happens to information once released
$\Longrightarrow$ DAC is vulnerable from Trojan horses exploting access privileges of calling subject

Trojan Horse: Rogue software. It contains a hidden code that performs (unlegitimate) functions not known to the caller.

Viruses and logic bombs can be transmitted in the form of Trojan Horse

The Trojan Horse problem

Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

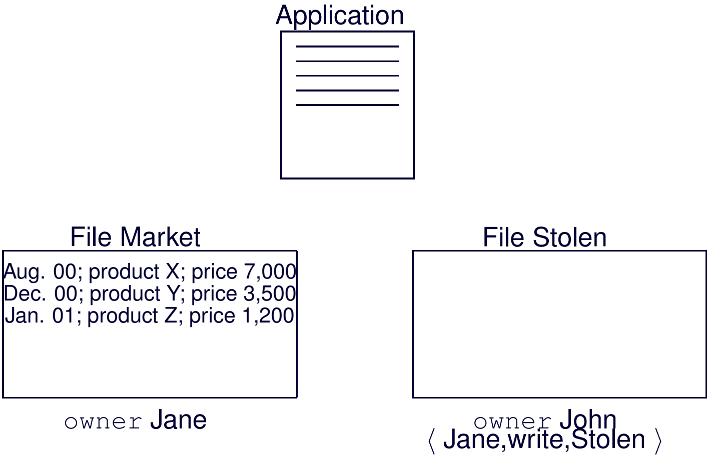
The Trojan Horse problem

File Market

Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

`owner` Jane

The Trojan Horse problem



Application

File Market

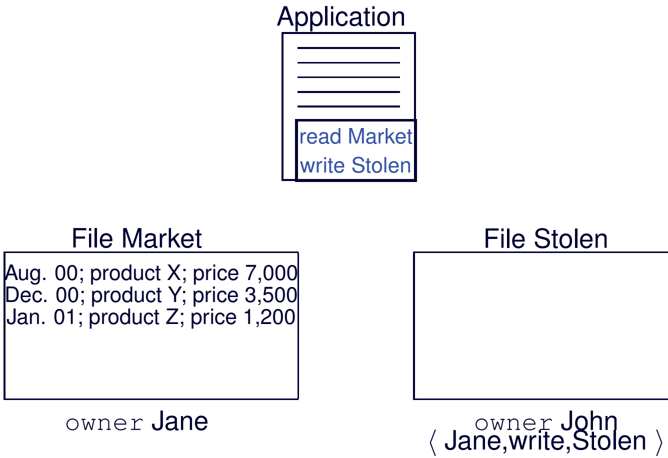
Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

`owner` Jane

File Stolen

`owner` John
⟨ Jane,write,Stolen ⟩

The Trojan Horse problem



Application

read Market
write Stolen

File Market

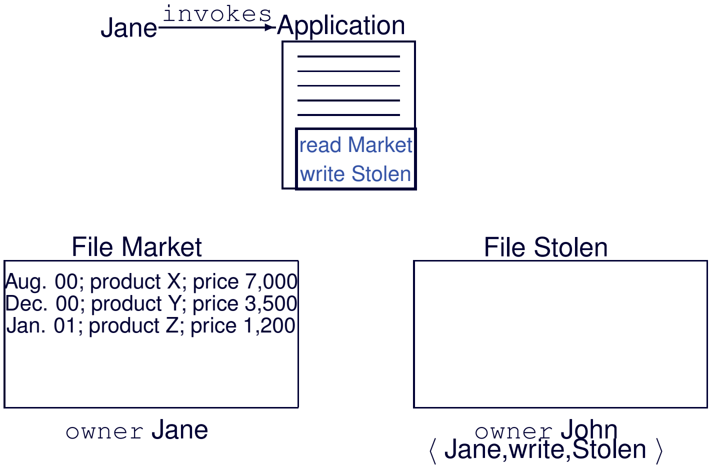
Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

owner Jane

File Stolen

owner John
⟨ Jane,write,Stolen ⟩

The Trojan Horse problem



Jane invokes Application

read Market
write Stolen

File Market

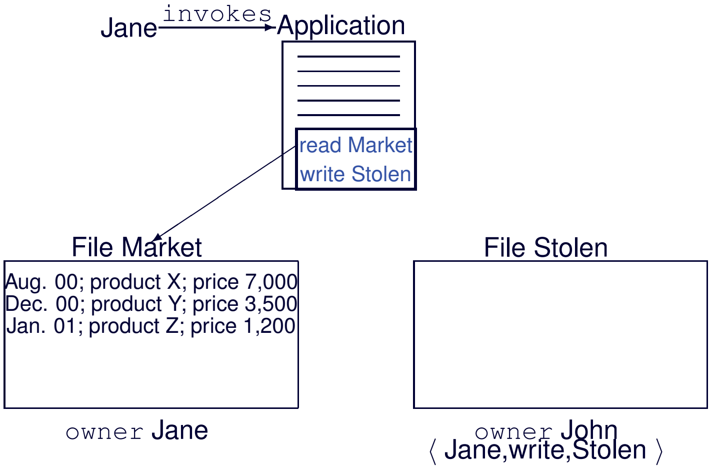
Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

owner Jane

File Stolen

owner John
⟨ Jane,write,Stolen ⟩

The Trojan Horse problem



Jane invokes Application

read Market
write Stolen

File Market

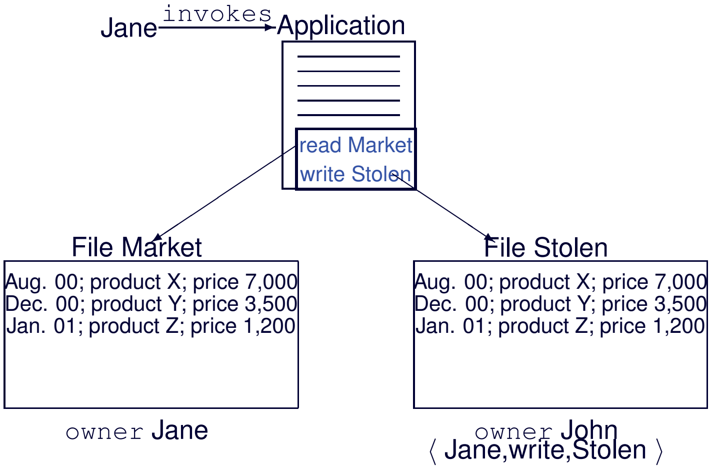
Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

owner Jane

File Stolen

owner John
⟨ Jane,write,Stolen ⟩

The Trojan Horse problem



Jane invokes Application

read Market

write Stolen

File Market

Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

owner Jane

File Stolen

Aug. 00; product X; price 7,000
Dec. 00; product Y; price 3,500
Jan. 01; product Z; price 1,200

owner John

⟨ Jane,write,Stolen ⟩

Mandatory (MAC) policies

Mandatory policies

Mandatory access control: Impose restrictions on information flow which cannot be bypassed by Trojan Horses
Makes a distinction between users and subjects operating on their behalf

- User Human being
- Subject Process in the system (program in execution)
  It operates on behalf of a user

While users may be trusted not to behave improperly, the programs they execute are not

Mandatory policies

Most common form of mandatory policy is multilevel security policy

- Based on classification of subjects and objects

- Two classes of policies
  - Secrecy-based (e.g., Bell La Padula model)
  - Integrity-based (e.g., Biba model)

Security classification

Security class usually formed by two components

- Security level element of a hierarchical set of elements. E.g., TopSecret(TS), Secret(S), Confidential(C), Unclassified(U)

  $$TS > S > C > U$$

  Crucial (C), Very Important (VI), Important (I)

  $$C > VI > I$$

- Categories set of a non-hierarchical set of elements (e.g., Administrative, Financial). It may partition different area of competence within the system. It allows enforcement of “need-to-know” restrictions.

The combination of the two introduces a partial order on security classes, called dominates

$$(L_1, C_1) \succeq (L_2, C_2) \iff L_1 \geq L_2 \wedge C_1 \supseteq C_2$$

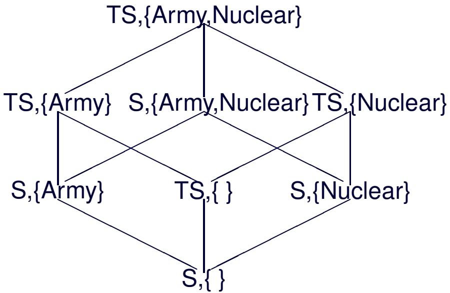
Classification lattice

Security classes together with $\succeq$ introduce a lattice $(SC, \succeq)$

- Reflexivity of $\succeq$ $\forall x \in SC : x \succeq x$
- Transitivity of $\succeq$ $\forall x, y, z \in SC : x \succeq y, y \succeq z \Longrightarrow x \succeq z$
- Antisymmetry of $\succeq$ $\forall x, y \in SC : x \succeq y, y \succeq x \Longrightarrow x = y$
- Least upper bound $\forall x, y \in SC : \exists\ !z \in SC$
  - $z \succeq x$ and $z \succeq y$
  - $\forall t \in SC : t \succeq x$ and $t \succeq y \Longrightarrow t \succeq z$.
- Greatest lower bound $\forall x, y \in SC : \exists\ !z \in SC$
  - $x \succeq z$ and $y \succeq z$
  - $\forall t \in SC : x \succeq t$ and $y \succeq t \Longrightarrow z \succeq t$.

Classification lattice – example

Levels: Top Secret (TS), Secret (S)
Categories: Army, Nuclear



- lub(⟨TS,{Nuclear}⟩,⟨S,{Army,Nuclear}⟩) = ⟨TS,{Army,Nuclear}⟩
- glb(⟨TS,{Nuclear}⟩,⟨S,{Army,Nuclear}⟩) = ⟨S,{Nuclear}⟩

Semantics of security classifications

Each user is assigned a security class (clearance).
A user can connect to the system at any class dominated by his clearance.
Subjects activated in a session take on the security class with which the user has connected.

Secrecy classes

- assigned to users reflect user's trustworthiness not to disclose sensitive information to individuals who do not hold appropriate clearance
- assigned to objects reflect the sensitivity of information contained in the objects and the potential damage that could result from their improper leakage

Categories define the area of competence of users and data

Bell La Padula

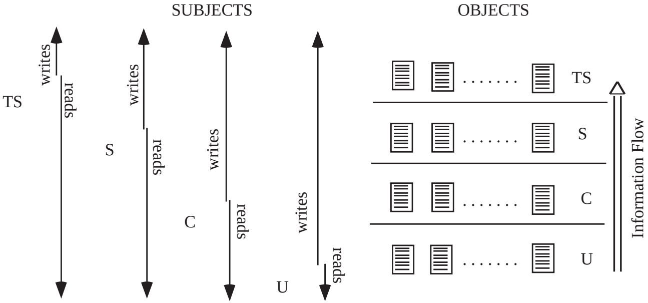
Defines mandatory policy for secrecy.
Different versions of the model have been proposed (with small differences or related to specific application environments); but the basic principles remain the same.
Goal: prevent information flow to lower or incomparable security classes

- simple property A subject s can read object o only if $\lambda(s) \succeq \lambda(o)$
- *-property A subject s can write object o only if $\lambda(o) \succeq \lambda(s)$

$\Longrightarrow$ NO READ UP
NO WRITE DOWN

Easy to see that Trojan Horses leaking information through *legitimate* channels are blocked

Information flow for secrecy



Exceptions to axioms

Real-word requirements may need mandatory restrictions to be bypassed

- Data association: A set of values seen together is to be classified higher than the value singularly taken (e.g., *name* and *salary*)

- Aggregation: An aggregate may have higher classification than its individual items. (e.g., the location of a *single* military ship is unclassified but the location of *all* the ships of a fleet is secret)

- Sanitization and Downgrading: Data may need to be downgraded after some time (embargo). A process may produce data less sensitive than those it has read
  $\Longrightarrow$ Trusted process
  A trusted subject is allowed to bypass (in a controlled way) some restrictions imposed by the mandatory policy

Coexistence of DAC and MAC

DAC and MAC not mutually exclusive

- E.g., BLP enforces DAC as well
  DAC property $b \subseteq \{(s,o,a) \;\text{ s.t. }\; a \in M[s,o]\}$

If both DAC and MAC are applied only accesses which satisfy both are permitted

DAC provides discretionality within the boundaries of MAC

Limitation of mandatory policies

- Secrecy mandatory policy controls only overt channels of information (flow through legitimate channels)
  Remain vulnerable to covert channels

- Covert channels are channels not intended for communicating information but that can, however, be exploited to leak information

- Every resource or observable of the system shared by processes of different levels can be exploited to create a covert channel

Covert and timing channels – examples

- Low level subject asks to write a high level file. The system returns that the file does not exist (if the system creates the file the user may not be aware when necessary)
- Low level subject requires a resource (e.g., CPU or lock) that is busy by a high level subject. Can be exploited by high level subjects to leak information to subjects at lower levels
- A high level process can lock shared resources and modify the response times of process at lower levels (timing channel). With timing channel the response returned to a low level process is the same, it is the time to return it that changes

Locking and concurrency mechanisms must be redefined for multilevel systems
(Careful to not introduce denial-of-service)

Covert channel analysis

Covert channel analysis usually done in the implementation phase (to assure that a system's implementation of the model primitive is not too weak)

Interface models attempt to rule out such channels in the modeling phase

- Non interference: the activity of high level processes must not have any effect on processes at lower or incomparable levels

Integrity mandatory policy

Secrecy mandatory policies control only improper leakage of information

Do not safeguard integrity $\Longrightarrow$ information can be tampered

Dual policy can be applied for integrity, based on assignment of (integrity) classifications

Integrity classes

- assigned to users reflect users' trustworthiness not to improperly modify information
- assigned to objects reflect the degree of trust in information contained in the objects and the potential damage that could result from its improper modification/deletion

Categories define the area of competence of users and data

Biba model for integrity

Defines mandatory policy for integrity
Goal: prevent information to flow to higher or uncomparable security classes

- Strict integrity policy
  Based on principles dual to those of BLP
  - simple property A subject s can read object o only if $\lambda(o) \succeq \lambda(s)$
  - *-property
    Drawback: it does not safeguard integrity but simply signals its compromise

Limitations of Biba policies

Biba's model for the protection of integrity has shortcomings

- flow restrictions may result too restrictive

- it enforces integrity only by preventing information flows from lower to higher access classifications $\Longrightarrow$ it captures only a very small part of the integrity problem

Integrity

Integrity is a more complex concept: ensuring that no resource has been modified in an unauthorized or improper way and that data stored in the system correctly reflect the real word they are intended to represent
$\Longrightarrow$ need to prevent flaws and errors
Any data management system has functionalities for ensuring integrity

- concurrency control and recovery techniques: to ensure that no concurrent access can lead to data loss or inconsistencies
- recovery techniques: to recover the state of the system in case of errors or violations
- integrity constraints: that enforce limitation on the values that can be given to data

Role-Based (RBAC) policies

Role-based access control model – 1

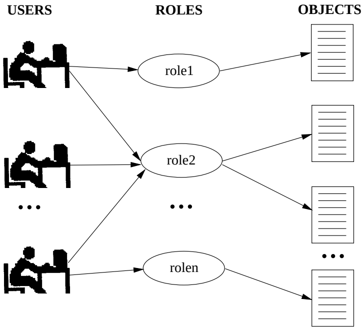
Role named set of privileges related to execution of a particular activity
Access of users to objects mediated by roles

- Roles are granted authorizations to access objects
- Users granted authorizations to activate roles
- By activating a role r a user can execute all access granted to r
- The privileges associated with a role are not valid when the role is not active

Note difference between

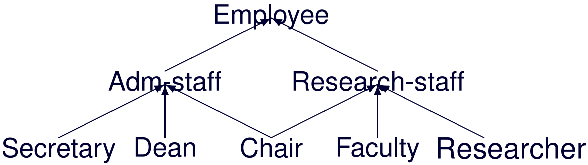
- *group*: set of users
- *role*: set of privileges

Role-based access control model – 2



Role-based access control model – 3

Role hierarchy defines specialization relationships



Hierarchical relationship $\Longrightarrow$ authorization propagation

- If a role r is granted authorization to execute (action, object) $\Longrightarrow$ all roles generalization of r can execute (action, object)
- If u is granted authorization to activate role $r \Longrightarrow u$ can activate all generalizations of r

RBAC – Advantages

- Easy management easy to specify authorizations (e.g., it is sufficient to assign or remove a role for a user to enable the user to execute a whole set of tasks)
- Role hierarchy can be exploited to support implication. Makes authorization management easier
- Restrictions Further restrictions can be associated with roles, such as cardinality or mutual exclusions
- Least privilege It allows associating with each subject the least set of privileges the subject needs to execute its work $\Longrightarrow$ Limits abuses and damages due to violations and errors
- Separation of duty Roles allow the enforcement of separation of duty (split privileges among different subjects)

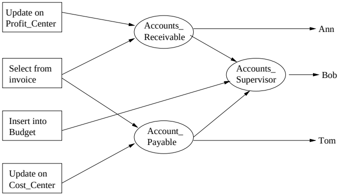
Role-based models

Work on role-based models has been addressing also:

- relationships beyond hierarchical (e.g., secretary can operate on behalf of his manager)
- hierarchy-based propagation not always wanted (some privileges may not propagate to subroles)
- enriched administrative policies (authority confinement)
- relationships with user identifiers (needed for individual relationships – e.g., “my secretary”)
- additional constraints (e.g., dynamic separation of duty; completion of an activity requires participation of at least n individuals)

Roles in SQL

In SQL privileges can be grouped in roles that can be assigned to users or to other roles (nested)



By activating a role, a user is enabled for all the privileges in a subset rooted at that role

- roles can be granted to users with grant option
  $\Longrightarrow$ the user can grant it to others

Administrative policies

Administrative policies

Define who can grant and revoke access authorizations

- Centralized: a privileges authority (system security officer) is in charge of authorization specification
- Ownership The creator of an object is its owner and as such can administer access authorization on the object
  Ownership not always clear in:
  - hierarchical data models (e.g., object-oriented)
  - RBAC framework

Authority to specify authorizations can be delegated.
Delegation often associated with ownership: the owner of an object delegates administrative privileges to others.
Decentralized administration introduces flexibility, but complicates the scenario.

Separation of duty

Separation of duty principle: no user (or restricted set of users) should have enough privileges to be able to abuse the system.

- static who specifies the authorizations must make sure not to give “too much privileges” to a single user

- dynamic the control on limiting privileges is enforced at runtime: a user cannot use “too many” privileges but he can choose which one to use. The system will consequently deny other accesses $\Longrightarrow$ more flexible

Separation of duty – Example

Operations: order-goods, send-order, record-invoice, pay

Four employees. Protection requirements:
at least two people must be involved in the process

- static: the administrator assigns tasks to users so that none can execute all the four operations
- dynamic: each user can execute any operation, but cannot complete the process and execute all four

Expanding authorizations

DAC – Expanding authorizations

Traditionally supported:

- user groups Users collected in groups and authorizations specified for groups
- conditional Validity of authorizations dependent on satisfaction of some conditions
  - *system-dependent* evaluate satisfaction of system predicates
    - location
    - time
  - *content-dependent* dependent on value of data (DBMS)
  - *history dependent* dependent on history of requests

Relatively easy to implement in simple systems
Introduce complications in richer models

Expanding authorizations – 1

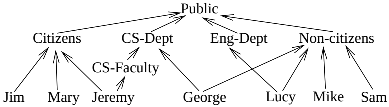
Specifications for single entities (users, files, ...) too heavy

- support abstractions (grouping of them). Usually hierarchical relationships: users/groups; objects/classes; files/directories; ..... Authorizations may propagate along the hierarchies

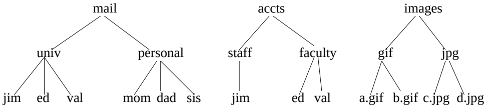
Hierarchical data systems

Support of hierarchies can be applied to all dimensions of authorizations.

Subjects (e.g., users vs groups)



Objects (e.g., files vs directories, objects vs classes)



Actions action grouping (e.g., write modes)
subsumption (e.g., write $\succeq$ read)

Expanding authorizations – 2

Usefulness of abstractions limited if exceptions are not possible. E.g., all Employees but Sam can read a file

- support negative authorizations
  (Employees, read, file, +)   (Sam, read, file, -)

Presence of permissions and denials can bring inconsistencies

- how should the system deal with them?

Permissions and denials

Easy way to support exceptions via negative authorizations.
Negative authorizations first introduced by themselves as:

- open policy: whatever is not explicitely denied can be executed; as opposed to
- closed policy: only accesses explicitely authorized can be executed

Recent hybrid policies support both, but

- what if for an access we have both + and -? (inconsistency)
- what if for an access we have neither + nor -? (incompleteness)

Incompleteness may be solved by either

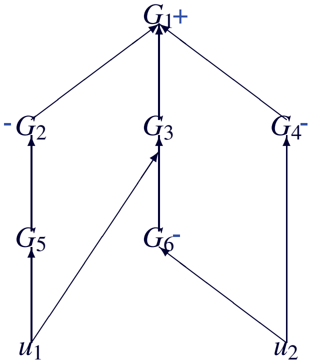
- assuming completeness: for every access either a negation or a permission must exist $\Longrightarrow$ too heavy
- assuming either closed or open as a basis default decision

Permissions and denials – 2

Possible conflict resolution policies

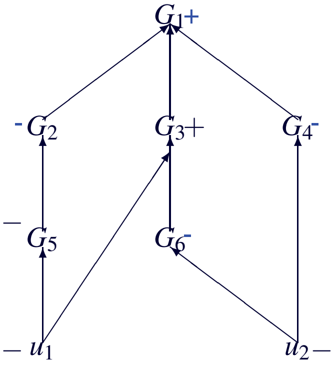
- denials-take-precedence negative authorization wins (fail safe principle)
- most-specific-takes-precedence the authorization that is “more specific” wins
- most-specific-along-a-path-takes-precedence the authorization that is “more specific” wins only on the paths passing through it ⇒ authorizations propagate until overridden by more specific authorizations
- Other.....

Example of conflict resolution

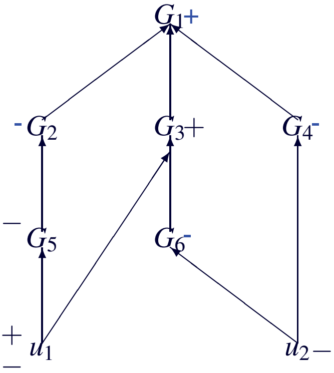


explicit authorizations

Examples of conflict resolution



most specific



most specific along a path

Most specific takes precedence

Most specific intuitive and natural ...... but

- what is more specific if multiple hierarchies?
  `(Employees, read, file1, +)`
  `(Sam, read, directory1, -)`

- in some cases not wanted.
  E.g., authorizations that do not allow exceptions
  - `(Employees, read, bulletin-board, +)`
    I do not want anybody to be able to forbid
  - `(Employees, read, budget, +)`
    `(Temporary_employees, read, budget, -)`
    I do not want my restriction on temporary employees to be bypassed

Other conflict resolution policies

- Explicit priority authorizations have associated explicit priorities
  - difficult to manage
- Positional strength of authorizations depend on order in authorization list
  - gives responsibility of explicitely resolving conflicts to security administrator
  - controlled administration difficult to enforce
- Grantor-dependent strength of authorizations depend on who granted them
  - need to be coupled with others to support exceptions among authorizations stated by a single administrator
- Time-dependent strength of authorizations depend on time they have been granted (e.g., more recent wins)
  - limited applicability

Conflict resolution policies

Different conflict resolution policies are not in mutual exclusion. E.g., I can first apply “most specific” and then “denials-take-precedence” on the remaining conflicts

There is no policy better than the others:

- Different policies correspond to different choices that we can apply for solving conflicts.

Trying to support all the different semantics that negation can have (strong negation, exception,....) can lead to models not manageable.
$\Longrightarrow$ Often negative authorizations are not used.

However, they can be useful.
$\Longrightarrow$ Systems that support negative authorizations usually adopt one specific conflict resolution policy.

Recent directions in access control

Access control in the global infrastructure ......

- need to interact with remote parties and access remote resources

- accesses as (action,object) limiting. E.g., service

- relationships with authentication may change
  - in some cases authentication not even wanted (anonymous transactions)
  - in an open system like Internet new users (not known at the server) can present requests
    - group and role administration may not be centralized
    - the system protecting resources may not know its users in advance

⟹ access control based on the use of digital certificates (credentials)

A more general approach supporting certificates

Allow users to present digital certificates, signed by some authority trusted for making a statement, and can

- bind a public key to an identity (identity)
- bind a public key or identity to some properties (e.g., membership in groups)
- bind a public key or identity to the ability of enjoying some privileges (authorization)

The server can use certificates to enforce access control.
Certificate management relates to the context of:

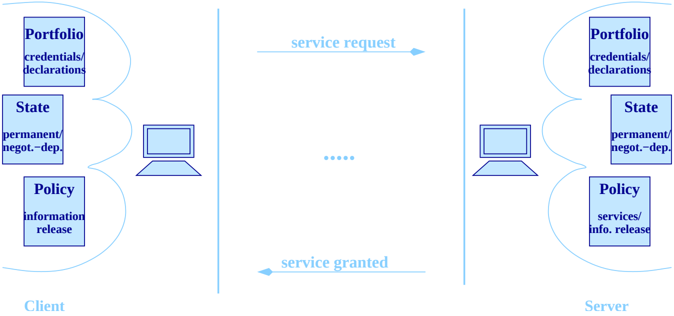
- Certification Authorities
- Public Key Infrastructure
- Trust Management

Recent Access Control Models

- Attribute-based access control (ABAC)
  Authorizations defined on attributes/properties of the requester

- Credential-based access control
  Attributes proved by presenting certificates

Client-Server Interplay



Privacy and Data Protection in Emerging Scenarios

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

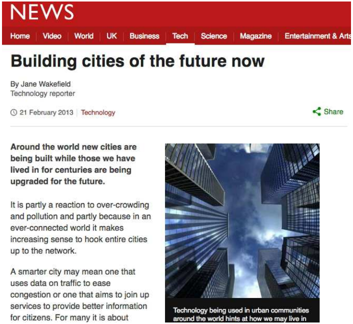
Privacy in Data Outsourcing

ICT ecosystem

- Advancements in the ICT have changed our society
- Infrastructures and services are more powerful, efficient, and complex

- ICT is the enabling factor for a smart society …

Smart home, smart grid, ...



NEWS

Home | Video | World | UK | Business | Tech | Science | Magazine | Entertainment & Arts

Building cities of the future now

By Jane Wakefield
Technology reporter

21 February 2013 | Technology   Share

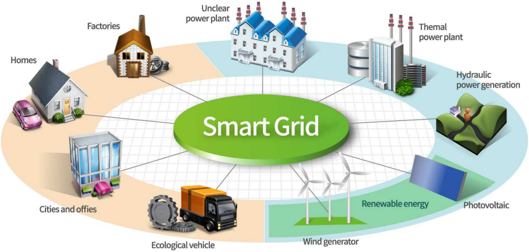
Around the world new cities are being built while those we have lived in for centuries are being upgraded for the future.

It is partly a reaction to over-crowding and pollution and partly because in an ever-connected world it makes increasing sense to hook entire cities up to the network.

A smarter city may mean one that uses data on traffic to ease congestion or one that aims to join up services to provide better information for citizens. For many it is about

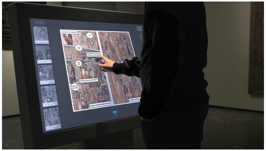
Technology being used in urban communities around the world hints at how we may live in

COMMUNICATION
AUTO CURTAIN
HOME ENTERTAINMENT
MOODS
HOME AUTOMATION
GARAGE
AUTO SPRINKLERS
AUTO LIGHTING
ON
OFF
HOME BACK UP SYSTEM
SECURITY
eCAM



. . . Everything is getting smart

Smart car


Museum and exhibitions

Health Care

Augmented reality

Smart e-commerce

Intelligent shops

Smart entertainment systems

Smart governance

Smart transportation

Smart society

Smart Parking
Public Safety
Smart Buildings
Smart Health
Waste Management
Open Data
Education
Traffic Management
Smart Environment
Electromagnetic Emissions
Internet of Things
Smart Energy
Gas & Water Leak Detection
Smart Street Lights
Air Pollution
Water Quality
Smart Home
Intelligent Shopping
Electric Vehicle Charging

Smart services and security – Advantages

+ Better protection mechanisms

+ Business continuity and disaster recovery

+ Prevention and response

. . . but . . .

Smart services and security – Disadvantages

- More complexity . . .

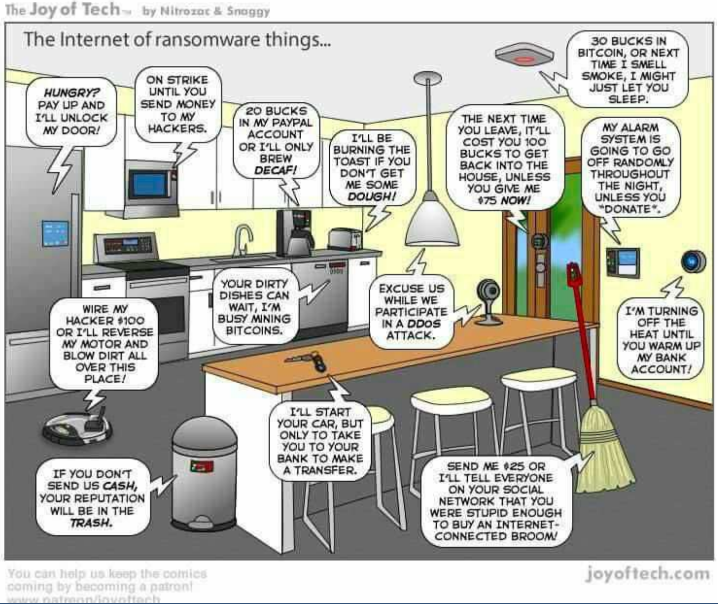
  . . . weakest link becomes a point of attack

  - system hacking
  - improper information leakage
  - data and process tampering

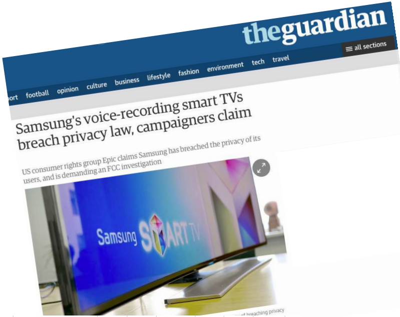
- Explosion of damages and violations

- Loss of control over data and processes

Maybe too smart? – 1

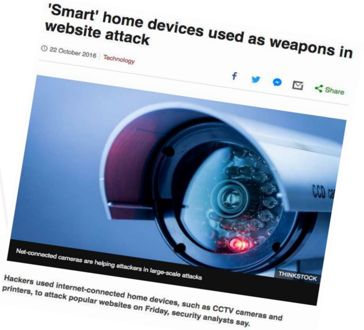

The Joy of Tech by Nitrozac & Snaggy
The Internet of ransomware things...
HUNGRY? PAY UP AND I'LL UNLOCK MY DOOR!
ON STRIKE UNTIL YOU SEND MONEY TO MY HACKERS.
20 BUCKS IN MY PAYPAL ACCOUNT OR I'LL ONLY BREW DECAF!
I'LL BE BURNING THE TOAST IF YOU DON'T GET ME SOME DOUGH!
THE NEXT TIME YOU LEAVE, IT'LL COST YOU 100 BUCKS TO GET BACK INTO THE HOUSE, UNLESS YOU GIVE ME $75 NOW!
30 BUCKS IN BITCOIN, OR NEXT TIME I SMELL SMOKE, I MIGHT JUST LET YOU SLEEP.
MY ALARM SYSTEM IS GOING TO GO OFF RANDOMLY THROUGHOUT THE NIGHT, UNLESS YOU "DONATE".
YOUR DIRTY DISHES CAN WAIT, I'M BUSY MINING BITCOINS.
EXCUSE US WHILE WE PARTICIPATE IN A DDoS ATTACK.
WIRE MY HACKER $100 OR I'LL REVERSE MY MOTOR AND BLOW DIRT ALL OVER THIS PLACE!
I'M TURNING OFF THE HEAT UNTIL YOU WARM UP MY BANK ACCOUNT!
I'LL START YOUR CAR, BUT ONLY TO TAKE YOU TO YOUR BANK TO MAKE A TRANSFER.
IF YOU DON'T SEND US CASH, YOUR REPUTATION WILL BE IN THE TRASH.
SEND ME $25 OR I'LL TELL EVERYONE ON YOUR SOCIAL NETWORK THAT YOU WERE STUPID ENOUGH TO BUY AN INTERNET-CONNECTED BROOM!
You can help us keep the comics coming by becoming a patron! www.patreon/joyoftech
joyoftech.com

Maybe too smart? – 2


theguardian
all sections
ort football opinion culture business lifestyle fashion environment tech travel

Samsung's voice-recording smart TVs breach privacy law, campaigners claim

US consumer rights group Epic claims Samsung has breached the privacy of its users, and is demanding an FCC investigation

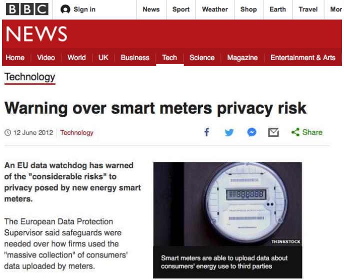

'Smart' home devices used as weapons in website attack

22 October 2016 | Technology

Net-connected cameras are helping attackers in large-scale attacks

THINKSTOCK

Hackers used internet-connected home devices, such as CCTV cameras and printers, to attack popular websites on Friday, security analysts say.


BBC Sign in News Sport Weather Shop Earth Travel Mor
NEWS
Home Video World UK Business Tech Science Magazine Entertainment & Arts
Share

Technology

Warning over smart meters privacy risk

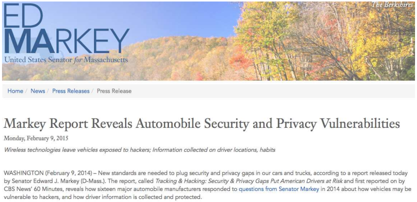
12 June 2012 | Technology

An EU data watchdog has warned of the "considerable risks" to privacy posed by new energy smart meters.

The European Data Protection Supervisor said safeguards were needed over how firms used the "massive collection" of consumers' data uploaded by meters.

THINKSTOCK

Smart meters are able to upload data about consumers' energy use to third parties


ED MARKEY
United States Senator for Massachusetts
The Berkshires
Home / News / Press Releases / Press Release

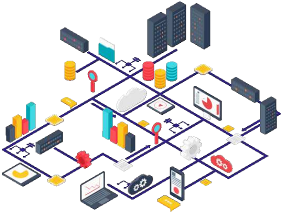
Markey Report Reveals Automobile Security and Privacy Vulnerabilities

Monday, February 9, 2015

Wireless technologies leave vehicles exposed to hackers; Information collected on driver locations, habits

WASHINGTON (February 9, 2014) – New standards are needed to plug security and privacy gaps in our cars and trucks, according to a report released today by Senator Edward J. Markey (D-Mass.). The report, called *Tracking & Hacking: Security & Privacy Gaps Put American Drivers at Risk* and first reported on by CBS News' 60 Minutes, reveals how sixteen major automobile manufacturers responded to questions from Senator Markey in 2014 about how vehicles may be vulnerable to hackers, and how driver information is collected and protected.

Security . . . a complex problem



Protection of infrastructure

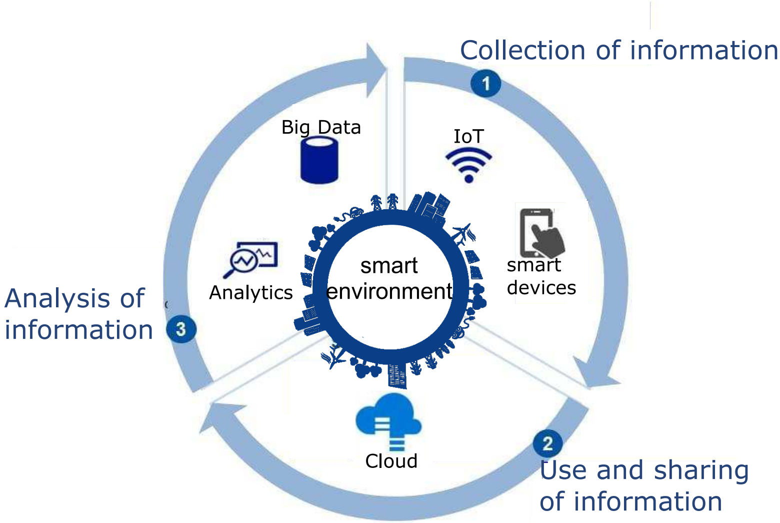
Protection of communication

Protection against malware and attacks

Protection of devices

Protection of data

The role of data in a smart environment



⟹ better governance and intelligent systems

The most valuable resource - Data

INQUIRER

The new oil: data is the world's most valuable resource

'Data is the new oil': Your personal information is now the world's most valuable commodity

Huge amounts of data are controlled by just 5 global mega-corporations t
bigger than most governments

By Ramona Pringle, CBC News Posted: Aug 25, 2017 5:00 AM ET | Last Updated: Aug 25, 2017 11:28 AN

Big Data and Analytics Play an Important Role in the Energy Industry

Real-Time DAILY

AROUND THE NET

Data Is Now The World's Most Valuable Resource

The Economist, Monday, May 8, 2017 6:22 AM

Data is now the world's most valuable resource according to *The Economist*, which reports on antitrust concerns about Alphabet (Google's parent company), Amazon, Apple, Facebook, and Microsoft, all of which have tons of data. The

Fuel of the future

Data is giving rise to a new economy

How is it shaping up?

Why is data protection so important?

BLOG 06 February 2017

digitally needs to be properly protected. From financial
ct information for your staff, data usage in the UK is protected by

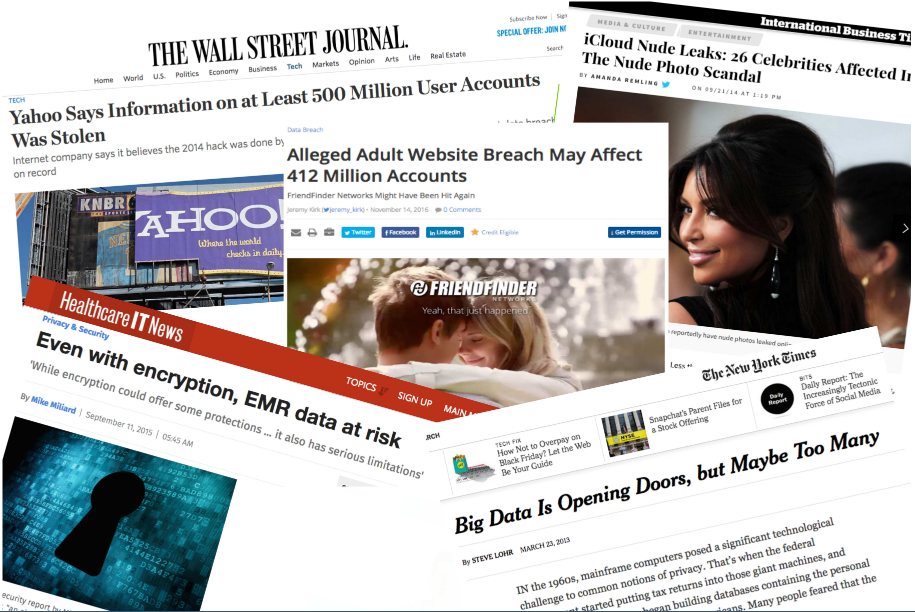
a legal necessity, but crucial to protecting and maintaining your

bus

PARTNER CONTENT ARVIND SINGH

IS BIG DATA THE NEW BLACK GOLD?

Impact on data protection and privacy – 1



THE WALL STREET JOURNAL.

Subscribe Now | Sign

SPECIAL OFFER: JOIN N

Search

Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life Real Estate

TECH

Yahoo Says Information on at Least 500 Million User Accounts Was Stolen

Internet company says it believes the 2014 hack was done by

on record

Data Breach

Alleged Adult Website Breach May Affect 412 Million Accounts

FriendFinder Networks Might Have Been Hit Again

Jeremy Kirk (jeremy_kirk) • November 14, 2016 0 Comments

International Business Ti

MEDIA & CULTURE ENTERTAINMENT

iCloud Nude Leaks: 26 Celebrities Affected In The Nude Photo Scandal

BY AMANDA REMLING ON 09/21/14 AT 1:19 PM

reportedly have nude photos leaked onli

Less th

Healthcare IT News

Privacy & Security

Even with encryption, EMR data at risk

'While encryption could offer some protections ... it also has serious limitations'

By Mike Miliard | September 11, 2015 | 05:45 AM

ecurity report by

"an al

The New York Times

RCH

TECH FIX How Not to Overpay on Black Friday? Let the Web Be Your Guide

Snapchat's Parent Files for a Stock Offering

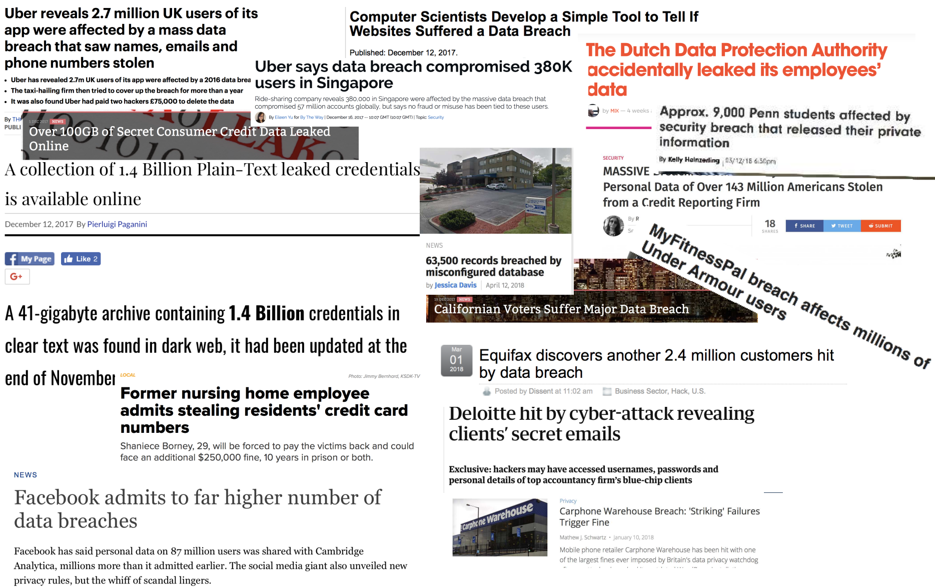
BITS Daily Report: The Increasingly Tectonic Force of Social Media

Big Data Is Opening Doors, but Maybe Too Many

By STEVE LOHR MARCH 23, 2013

IN the 1960s, mainframe computers posed a significant technological challenge to common notions of privacy. That's when the federal nt started putting tax returns into those giant machines, and began building databases containing the personal icans. Many people feared that the

Impact on data protection and privacy – 2



Uber reveals 2.7 million UK users of its app were affected by a mass data breach that saw names, emails and phone numbers stolen

- Uber has revealed 2.7m UK users of its app were affected by a 2016 data brea
- The taxi-hailing firm then tried to cover up the breach for more than a year
- It was also found Uber had paid two hackers £75,000 to delete the data

By TH
PUBLI

Computer Scientists Develop a Simple Tool to Tell If Websites Suffered a Data Breach

Published: December 12, 2017.

Uber says data breach compromised 380K users in Singapore

Ride-sharing company reveals 380,000 in Singapore were affected by the massive data breach that compromised 57 million accounts globally, but says no fraud or misuse has been tied to these users.

By Eileen Yu for By The Way | December 16, 2017 -- 10:07 GMT (10:07 GMT) | Topic: Security

1 DEC 2017 NEWS
Over 100GB of Secret Consumer Credit Data Leaked Online

The Dutch Data Protection Authority accidentally leaked its employees' data

by MIX — 4 weeks

Approx. 9,000 Penn students affected by security breach that released their private information

By Kelly Heinzerling | 03/12/18 6:50pm

SECURITY
MASSIVE

Personal Data of Over 143 Million Americans Stolen from a Credit Reporting Firm

By R
Sr

18 SHARES
SHARE TWEET SUBMIT

A collection of 1.4 Billion Plain-Text leaked credentials is available online

December 12, 2017 By Pierluigi Paganini

My Page Like 2
G+

A 41-gigabyte archive containing **1.4 Billion** credentials in clear text was found in dark web, it had been updated at the end of Novembe

NEWS

63,500 records breached by misconfigured database

by Jessica Davis | April 12, 2018

19 DEC 2017 NEWS
Californian Voters Suffer Major Data Breach

MyFitnessPal breach affects millions of Under Armour users

LOCAL

Photo: Jimmy Bernhard, KSDK-TV

Former nursing home employee admits stealing residents' credit card numbers

Shaniece Borney, 29, will be forced to pay the victims back and could face an additional $250,000 fine, 10 years in prison or both.

Mar 01 2018
Equifax discovers another 2.4 million customers hit by data breach

Posted by Dissent at 11:02 am   Business Sector, Hack, U.S.

Deloitte hit by cyber-attack revealing clients' secret emails

Exclusive: hackers may have accessed usernames, passwords and personal details of top accountancy firm's blue-chip clients

NEWS

Facebook admits to far higher number of data breaches

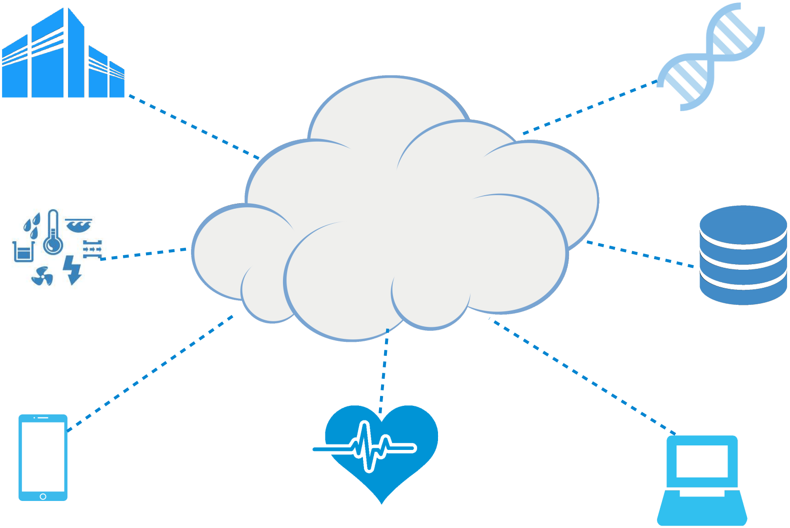
Facebook has said personal data on 87 million users was shared with Cambridge Analytica, millions more than it admitted earlier. The social media giant also unveiled new privacy rules, but the whiff of scandal lingers.

Privacy
Carphone Warehouse Breach: 'Striking' Failures Trigger Fine

Mathew J. Schwartz • January 10, 2018

Mobile phone retailer Carphone Warehouse has been hit with one of the largest fines ever imposed by Britain's data privacy watchdog

Huge amount of data stored at external providers

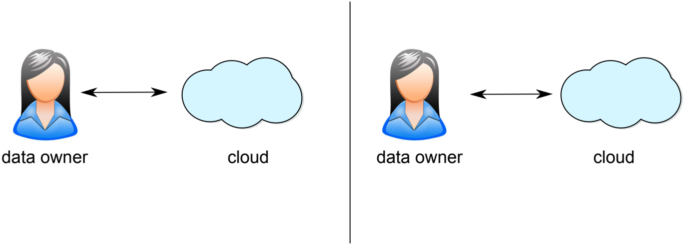


Cloud computing

- The Cloud allows users and organizations to rely on external providers for storing, processing, and accessing their data
  - + high configurability and economy of scale
  - + data and services are always available
  - + scalable infrastructure for applications
- Users lose control over their own data
  - – new security and privacy problems
- Need solutions to protect data and to securely process them in the cloud

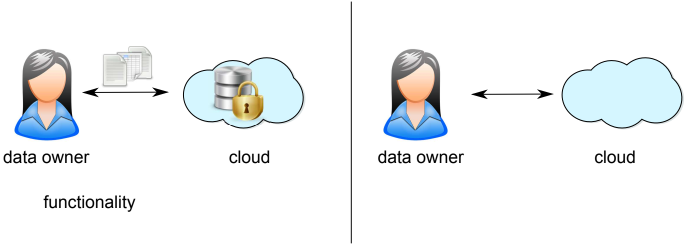
Cloud computing: Today

Cloud Service Providers (CSPs) apply security measures in the services they offer but these measures protect only the perimeter and storage against outsiders



Cloud computing: Today

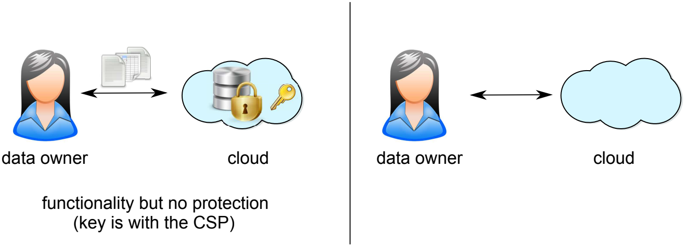
Cloud Service Providers (CSPs) apply security measures in the services they offer but these measures protect only the perimeter and storage against outsiders



- functionality

Cloud computing: Today

Cloud Service Providers (CSPs) apply security measures in the services they offer but these measures protect only the perimeter and storage against outsiders

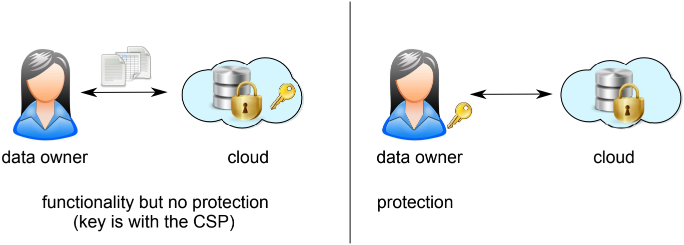


functionality but no protection
(key is with the CSP)

- functionality implies full trust in the CSP that has full access to the data (e.g., Google Cloud Storage, iCloud)

Cloud computing: Today

Cloud Service Providers (CSPs) apply security measures in the services they offer but these measures protect only the perimeter and storage against outsiders



data owner

cloud

functionality but no protection
(key is with the CSP)

data owner

cloud

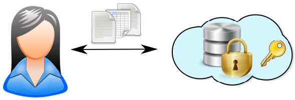
protection

- functionality implies full trust in the CSP that has full access to the data (e.g., Google Cloud Storage, iCloud)

- protection

Cloud computing: Today

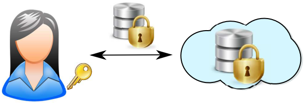
Cloud Service Providers (CSPs) apply security measures in the services they offer but these measures protect only the perimeter and storage against outsiders



data owner

cloud

functionality but no protection
(key is with the CSP)



data owner

cloud

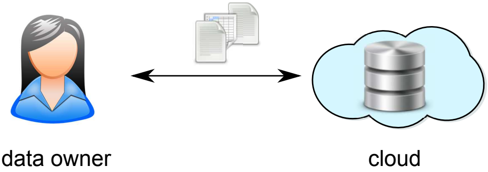
protection but limited functionality
(you cannot access data as you like)

- functionality implies full trust in the CSP that has full access to the data (e.g., Google Cloud Storage, iCloud)

- protection but limited functionality since the CSP cannot access data (e.g., Boxcryptor, SpiderOak)

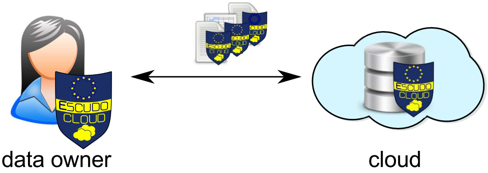
Cloud computing: ESCUDO-CLOUD's vision

Solutions that provide protection guarantees giving the data owners both: full control over their data and cloud functionality over them



Cloud computing: ESCUDO-CLOUD's vision

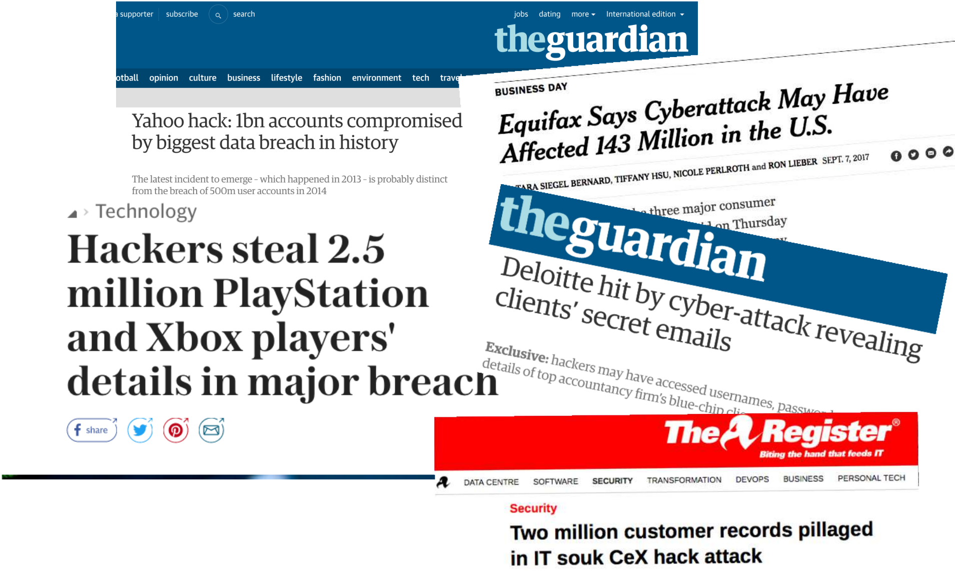
Solutions that provide protection guarantees giving the data owners both: full control over their data and cloud functionality over them



- client-side trust boundary: only the behavior of the client should be considered trusted

  ⟹ techniques and implementations supporting direct processing of encrypted data in the cloud

Data protection – Base level



Yahoo hack: 1bn accounts compromised by biggest data breach in history

The latest incident to emerge - which happened in 2013 - is probably distinct from the breach of 500m user accounts in 2014

Hackers steal 2.5 million PlayStation and Xbox players' details in major breach

Equifax Says Cyberattack May Have Affected 143 Million in the U.S.

TARA SIEGEL BERNARD, TIFFANY HSU, NICOLE PERLROTH and RON LIEBER SEPT. 7, 2017

three major consumer on Thursday

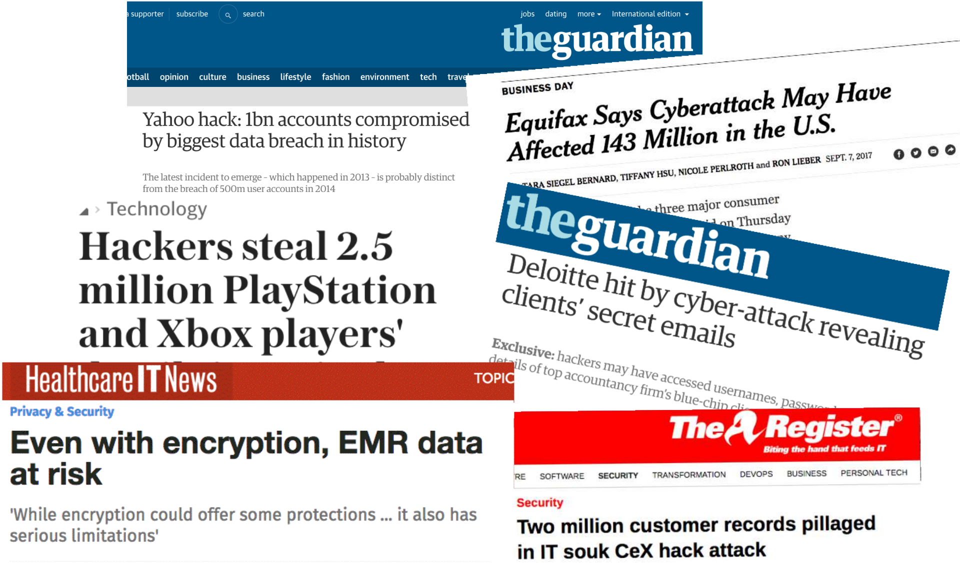
Deloitte hit by cyber-attack revealing clients' secret emails

Exclusive: hackers may have accessed usernames, passwo details of top accountancy firm's blue-chip cli

Security

Two million customer records pillaged in IT souk CeX hack attack

Data protection – Base level



a supporter subscribe search jobs dating more International edition

theguardian

otball opinion culture business lifestyle fashion environment tech trave

Yahoo hack: 1bn accounts compromised by biggest data breach in history

The latest incident to emerge - which happened in 2013 - is probably distinct from the breach of 500m user accounts in 2014

Technology

Hackers steal 2.5 million PlayStation and Xbox players'

BUSINESS DAY

Equifax Says Cyberattack May Have Affected 143 Million in the U.S.

TARA SIEGEL BERNARD, TIFFANY HSU, NICOLE PERLROTH and RON LIEBER SEPT. 7, 2017

three major consumer

on Thursday

theguardian

Deloitte hit by cyber-attack revealing clients' secret emails

Exclusive: hackers may have accessed usernames, passwo details of top accountancy firm's blue-chip cli

Healthcare IT News TOPIC

Privacy & Security

Even with encryption, EMR data at risk

'While encryption could offer some protections ... it also has serious limitations'

The Register® Biting the hand that feeds IT

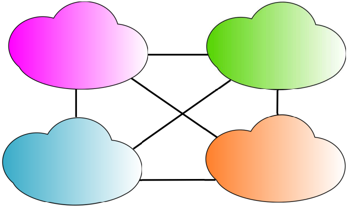
RE SOFTWARE SECURITY TRANSFORMATION DEVOPS BUSINESS PERSONAL TECH

Security

Two million customer records pillaged in IT souk CeX hack attack

Data protection – Regulation

Access/use control



Controlled sharing

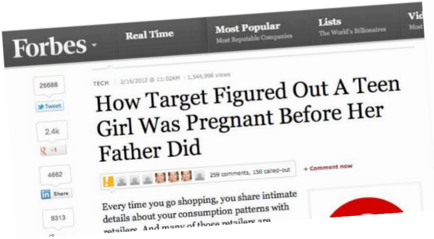
General Data Protection Regulation

Privacy Shield Framework

Governance and regulation

Data protection – Confidentiality (1)

- Minimize release/exposition
  - correlation among different data sources
  - indirect exposure of sensitive information
  - de-identification ≠ anonymization



Forbes · Real Time · Most Popular · Most Reputable Companies · Lists · The World's Billionaires · Vid · Most

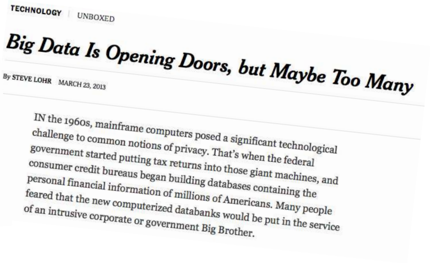
TECH | 2/16/2012 @ 11:02AM | 1,544,998 views

26688 · Tweet · 2.4k · +1 · 4662 · Share · 9313

How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

259 comments, 150 called-out · + Comment now

Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are



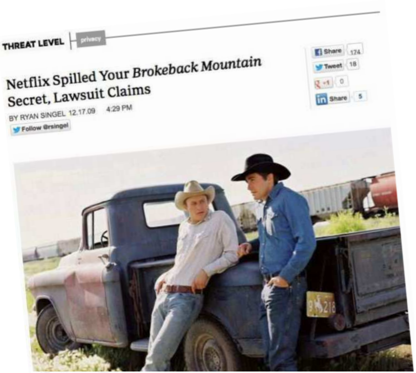
TECHNOLOGY | UNBOXED

Big Data Is Opening Doors, but Maybe Too Many

By STEVE LOHR MARCH 23, 2013

IN the 1960s, mainframe computers posed a significant technological challenge to common notions of privacy. That's when the federal government started putting tax returns into those giant machines, and consumer credit bureaus began building databases containing the personal financial information of millions of Americans. Many people feared that the new computerized databanks would be put in the service of an intrusive corporate or government Big Brother.

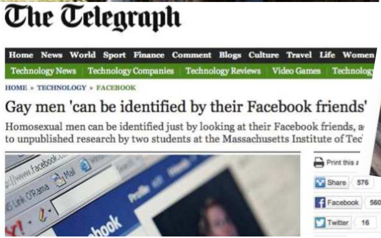
Data protection – Confidentiality (2)



Netflix Spilled Your *Brokeback Mountain* Secret, Lawsuit Claims

BY RYAN SINGEL 12.17.09 4:29 PM

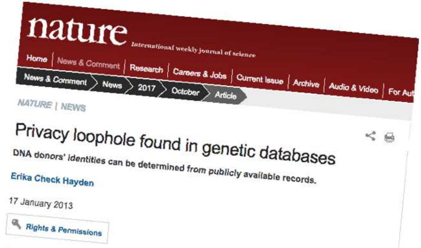
Follow @rsingel



The Telegraph
Home News World Sport Finance Comment Blogs Culture Travel Life Women
Technology News | Technology Companies | Technology Reviews | Video Games | Technolog
HOME » TECHNOLOGY » FACEBOOK
Print this a
Share 576
Facebook 560
Twitter 16

Gay men 'can be identified by their Facebook friends'

Homosexual men can be identified just by looking at their Facebook friends, a
to unpublished research by two students at the Massachusetts Institute of Tec



nature
International weekly journal of science
Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Aut
News & Comment > News > 2017 > October > Article
NATURE | NEWS

Privacy loophole found in genetic databases

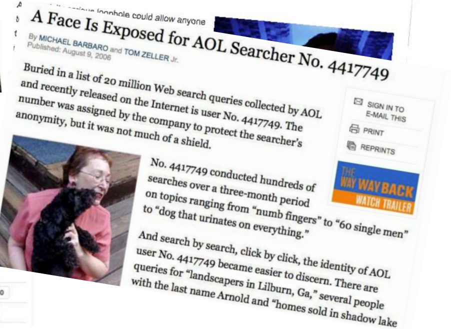
DNA donors' identities can be determined from publicly available records.

Erika Check Hayden

17 January 2013

Rights & Permissions

A … loophole could allow anyone



A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr.
Published: August 9, 2006

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.

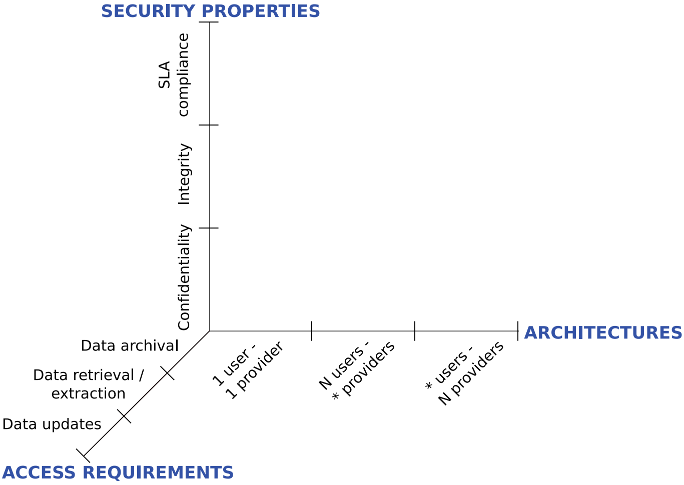
No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from "numb fingers" to "60 single men" to "dog that urinates on everything."

And search by search, click by click, the identity of AOL user No. 4417749 became easier to discern. There are queries for "landscapers in Lilburn, Ga," several people with the last name Arnold and "homes sold in shadow lake

Characterization of Data Protection Challenges

Scientific and technical challenges

Three dimensions characterize the problems and challenges



Security properties

Confidentiality

- data externally stored
- users identities
- actions that users perform on the data

Integrity

- data externally stored
- computation and query results

SLA compliance

- assurance and certification

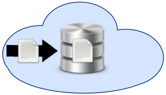
Access requirements

Data archival

- upload/download
- protection of data in storage

Data retrieval/extraction

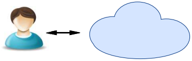
- support for fine-grained data retrieval and queries
- protection of computations and query results



Data update

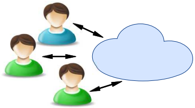
- support for access retrieval and enforcement of updates
- protection of the actions and of their effects on the data

Architectures



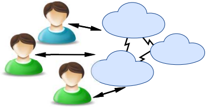
1 user - 1 provider

- protection of data at rest
- fine-grained retrieval
- query privacy/integrity



n users - * providers

- authorizations and access control
- multiple writers



*** users - n providers**

- controlled data sharing and computation

Combinations of the dimensions

- Every combination of the different instances of the dimensions identifies new problems and challenges

- The security properties to be guaranteed can depend on the access requirements and on the trust assumption on the providers involved in storage and/or processing of data

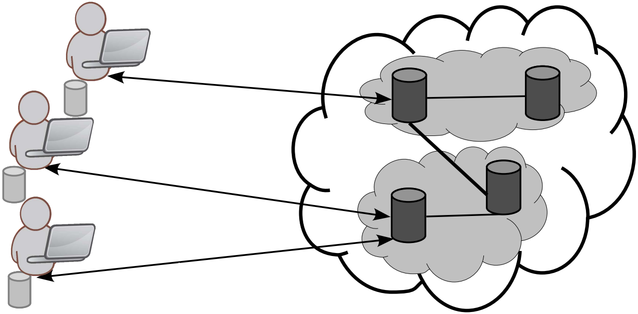
- Providers can be:
  - curious
  - lazy
  - malicious

Some Challenges in Data Protection

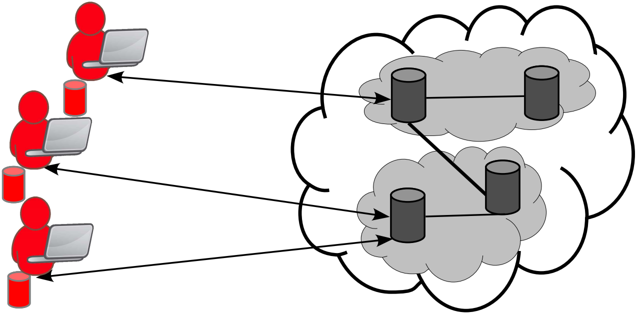
Issues to be addressed

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

Security and privacy problems

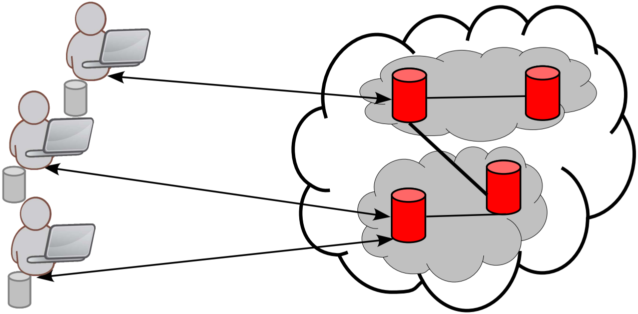


Security and privacy problems



Privacy of users

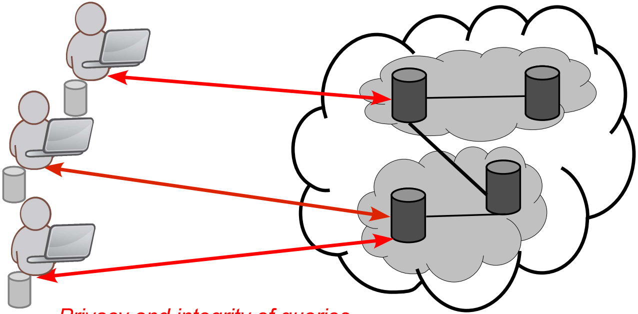
Security and privacy problems



Privacy of users

Privacy and integrity of data storage

Security and privacy problems



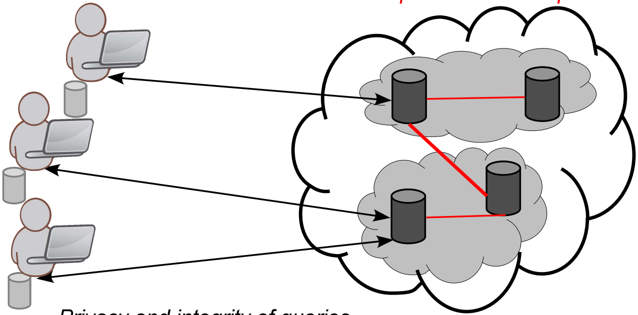
Privacy and integrity of queries and computations

Privacy of users

Privacy and integrity of data storage

Security and privacy problems

Secure and private data computations



Privacy and integrity of queries
and computations

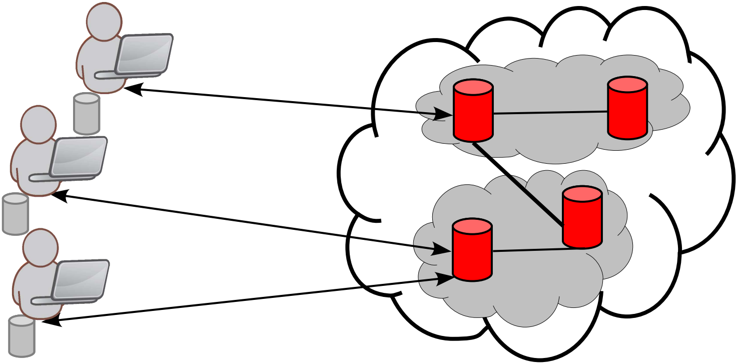
Privacy of users

Privacy and integrity of data storage

Privacy and Data Protection in Emerging Scenarios

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

Privacy and integrity of data storage



Privacy of users

Privacy and integrity of data storage

Contributions and advancements

The research community has been very active and produced several contributions and advancements. E.g.,:

- Solutions for protecting confidentiality of stored data [ABGGKMSTX-05, CDJJPS-09b, CDFJPS-10, HIML-02]
- Indexes supporting different types of queries [CDDJPS-05, HIML-02, WL-06]
- Inference exposure evaluation [CDDJPS-05]
- Data integrity [S-05, XWYM-07, WYPY-08]
- Selective access to outsourced data [DFJPS-10b]
- …

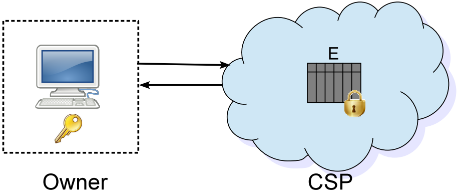
Protecting data confidentiality

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

Encryption

Encryption

- The server can be honest-but-curious and should not have access to the resource content
- Data confidentiality is provided by wrapping a layer of encryption around sensitive data [HIML-02]
  - for performance reasons, encryption is typically applied at the tuple level



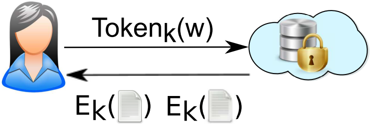
Fine-grained access to data in the cloud

- For confidentiality reasons, CSPs storing data cannot decrypt them for data processing/access

- Need mechanisms to support access to the outsourced data
  - effective and efficient
  - should not open the door to inferences

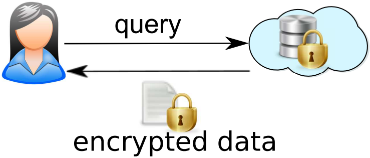
Fine-grained access: Approaches – 1

Keyword-based searches directly on the encrypted data: supported by specific cryptographic techniques (e.g., [CWLRL-11])



Fine-grained access: Approaches – 2

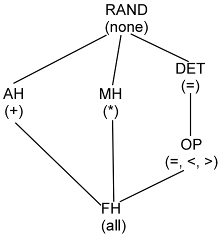
Homomorphic encryption: supports the execution of operations directly on the encrypted data (e.g., [BV11,G-09,GSW13])



Fine-grained access: Approaches – 3

- Encryption schemas: each column can be encrypted with a different encryption schema, depending on the conditions to be evaluated on it (e.g., Google encrypted BigQuery)

- Onion encryption (CryptDB): different onion layers each of which supports the execution of a specific SQL operation (e.g., HanaDB SEEED framework) [PRZB-11]



RAND
(none)

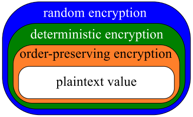
AH
(+)

MH
(*)

DET
(=)

OP
(=, <, >)

FH
(all)

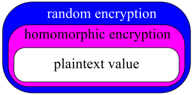


random encryption

deterministic encryption

order-preserving encryption

plaintext value



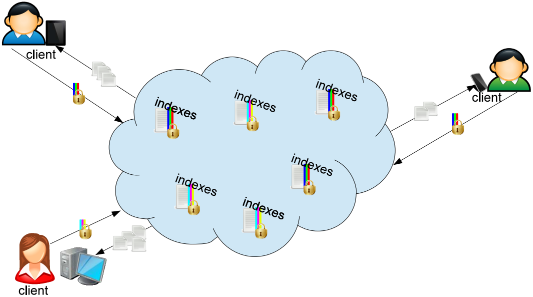
random encryption

homomorphic encryption

plaintext value

Fine-grained access: Approaches – 4

Indexes: metadata attached to the data and used for fine-grained information retrieval and query execution (e.g., [CDDJPS-05, HIML-02, WL-06])



can also be complementary to encryption (even with encryption users want to have the ability to perform searches based on metadata)

Encryption and indexes – Example

Indexes associated with attributes are used by the server to select data to be returned in response to a query

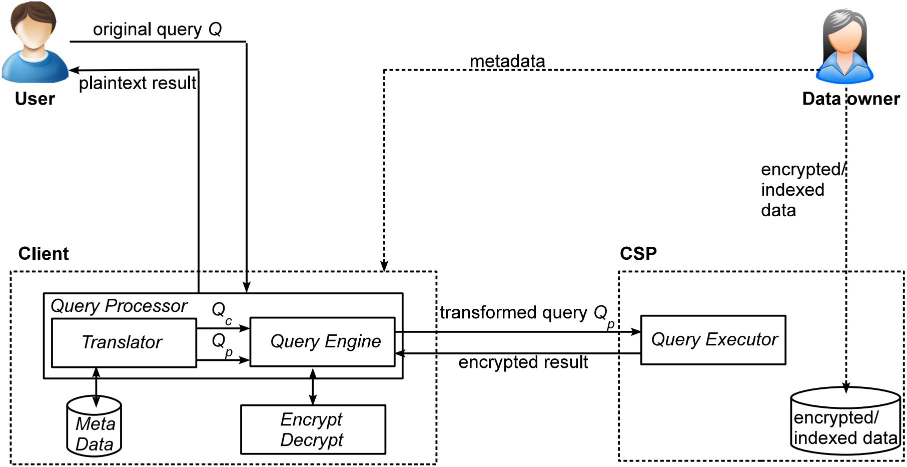
Accounts

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

Accounts$_1^k$

Counter	**Etuple**	$\mathbf{I}_A$	$\mathbf{I}_C$	$\mathbf{I}_B$
1	x4Z3tfX2ShOSM	π	α	μ
2	mNHg1oC010p8w	ϖ	α	κ
3	WsIaCvfyF1Dxw	ξ	β	η
4	JpO8eLTVgwV1E	ρ	γ	κ
5	qctG6XnFNDTQc	ς	δ	θ
6	4QbqCeq3hxZHklU	ι	ε	κ

Query evaluation process



Indexes for queries: Direct (1:1)

Actual value or coding

+ simple and precise for equality queries
– preserves plaintext value distinguishability (inference attacks)

Indexes for queries: Direct (1:1)

Actual value or coding

- + simple and precise for equality queries
- – preserves plaintext value distinguishability (inference attacks)

Patients

SSN	**Name**	**Illness**	**Doctor**
123…89	Alice	Asthma	Angel
234…91	Bob	Asthma	Angel
345…12	Carol	Asthma	Bell
456…23	David	Bronchitis	Clark
567…34	Eva	Gastritis	Dan
232…11	Eva	Stroke	Ellis

Patients^k

Tid	**Etuple**	I_S	I_N	I_I	I_D
1	x4Z3tfX2ShOSM	π	κ	α	δ
2	mNHg1oC010p8w	ϖ	ω	α	δ
3	WsIaCvfyF1Dxw	ξ	λ	α	ν
4	JpO8eLTVgwV1E	ρ	υ	β	γ
5	qctG6XnFNDTQc	ι	μ	α	σ
6	kotG8XnFNDTaW	χ	o	β	ψ

Indexes for queries: Direct (1:1)

Actual value or coding

+ simple and precise for equality queries

– preserves plaintext value distinguishability (inference attacks)

Patients

SSN	**Name**	**Illness**	**Doctor**
123…89	Alice	Asthma	Angel
234…91	Bob	Asthma	Angel
345…12	Carol	Asthma	Bell
456…23	David	Bronchitis	Clark
567…34	Eva	Gastritis	Dan
232…11	Eva	Stroke	Ellis

Patients^k

Tid	**Etuple**	$\mathbf{I_S}$	$\mathbf{I_N}$	$\mathbf{I_I}$	$\mathbf{I_D}$
1	x4Z3tfX2ShOSM	π	κ	α	δ
2	mNHg1oC010p8w	ϖ	ω	α	δ
3	WsIaCvfyF1Dxw	ξ	λ	α	ν
4	JpO8eLTVgwV1E	ρ	υ	β	γ
5	qctG6XnFNDTQc	ι	μ	α	σ
6	kotG8XnFNDTaW	χ	o	β	ψ

Indexes for queries: Bucket (n:1)

Partition-based or hash-based

- + supports for equality queries
- + collisions remove plaintext distinguishability
- – result may contain spurious tuples (postprocessing query)
- – still vulnerable to inference attacks

Indexes for queries: Bucket (n:1)

Partition-based or hash-based

- + supports for equality queries
- + collisions remove plaintext distinguishability
- – result may contain spurious tuples (postprocessing query)
- – still vulnerable to inference attacks

Patients

SSN	Name	Illness	Doctor
123…89	Alice	Asthma	Angel
234…91	Bob	Asthma	Angel
345…12	Carol	Asthma	Bell
456…23	David	Bronchitis	Clark
567…34	Eva	Gastritis	Dan
232…11	Eva	Stroke	Ellis

Patients^k

Tid	Etuple	I_S	I_N	I_I	I_D
1	x4Z3tfX2ShOSM	π	κ	α	δ
2	mNHg1oC010p8w	ϖ	ω	α	δ
3	WsIaCvfyF1Dxw	ξ	λ	α	ν
4	JpO8eLTVgwV1E	ρ	υ	β	γ
5	qctG6XnFNDTQc	ι	μ	α	σ
6	kotG8XnFNDTaW	χ	o	β	ψ

Indexes for queries: Bucket (n:1)

Partition-based or hash-based

- + supports for equality queries
- + collisions remove plaintext distinguishability
- – result may contain spurious tuples (postprocessing query)
- – still vulnerable to inference attacks

Patients

SSN	Name	Illness	Doctor
123…89	Alice	Asthma	Angel
234…91	Bob	Asthma	Angel
345…12	Carol	Asthma	Bell
456…23	David	Bronchitis	Clark
567…34	Eva	Gastritis	Dan
232…11	Eva	Stroke	Ellis

Patients^k

Tid	Etuple	I_S	I_N	I_I	I_D
1	x4Z3tfX2ShOSM	π	κ	α	δ
2	mNHg1oC010p8w	ϖ	ω	α	δ
3	WsIaCvfyF1Dxw	ξ	λ	α	ν
4	JpO8eLTVgwV1E	ρ	υ	β	γ
5	qctG6XnFNDTQc	ι	μ	α	σ
6	kotG8XnFNDTaW	χ	o	β	ψ

Indexes for queries: Flattened (1:n)

Flat indexes

- + decreases exposure to inference attacks
- – remains vulnerabile to dynamic observations

Indexes for queries: Flattened (1:n)

Flat indexes

+ decreases exposure to inference attacks
– remains vulnerabile to dynamic observations

Patients

SSN	Name	Illness	Doctor
123…89	Alice	Asthma	Angel
234…91	Bob	Asthma	Angel
345…12	Carol	Asthma	Bell
456…23	David	Bronchitis	Clark
567…34	Eva	Gastritis	Dan
232…11	Eva	Stroke	Ellis

Patients^k

Tid	Etuple	I_S	I_N	I_I	I_D
1	x4Z3tfX2ShOSM	π	κ	α	δ
2	mNHg1oC010p8w	ϖ	ω	α	δ
3	WsIaCvfyF1Dxw	ξ	λ	α	ν
4	JpO8eLTVgwV1E	ρ	υ	β	γ
5	qctG6XnFNDTQc	ι	μ	α	σ
6	kotG8XnFNDTaW	χ	o	β	ψ

Indexes for queries: Flattened (1:n)

Flat indexes

- + decreases exposure to inference attacks
- – remains vulnerabile to dynamic observations

Patients

SSN	Name	Illness	Doctor
123…89	Alice	Asthma	Angel
234…91	Bob	Asthma	Angel
345…12	Carol	Asthma	Bell
456…23	David	Bronchitis	Clark
567…34	Eva	Gastritis	Dan
232…11	Eva	Stroke	Ellis

Patients^{k}

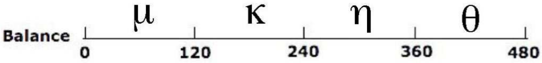
Tid	Etuple	I_S	I_N	I_I	I_D
1	x4Z3tfX2ShOSM	π	κ	α	δ
2	mNHg1oC010p8w	ϖ	ω	α	δ
3	WslaCvfyF1Dxw	ξ	λ	α	ν
4	JpO8eLTVgwV1E	ρ	υ	β	γ
5	qctG6XnFNDTQc	ι	μ	α	σ
6	kotG8XnFNDTaW	χ	o	β	ψ

Partition-based index [HIML-02]

- Consider an arbitrary plaintext attribute A_i in relational schema R, with domain D_i
- D_i is partitioned in a number of non-overlapping subsets of values, called partitions, containing contiguous values
- Given a plaintext tuple t in r, the value of attribute A_i for t belongs to a partition
  - function $\mathit{ident}_{R.A_i}(p_j)$ assigns to each partition p_j of attribute A_i in R an identifier
- The corresponding index value is the unique value associated with the partition to which the plaintext value $t[A_i]$ belongs
  - $\mathit{Map}_{R.A_i}(v) = \mathit{ident}_{R.A_i}(p_j)$, where p_j is the partition containing v
- $\mathit{Map}_{R.A_i}$ can be order-preserving or random

Partition-based index – Example

Random mapping



- $Map_{Balance}(100) = \mu$
- $Map_{Balance}(200) = \kappa$
- $Map_{Balance}(300) = \eta$
- $Map_{Balance}(400) = \theta$

Query conditions supported by the partition-based index

- Support queries where conditions are boolean formulas over terms of the form
  - *Attribute op Value*
  - *Attribute op Attribute*
- Allowed operations for *op* include $\{=, <, >, \leq, \geq\}$

Mapping conditions Map_{cond} – 1

- $A_i = v$. The mapping is defined as:

  $$Map_{cond}(A_i = v) \Longrightarrow I_i = Map_{A_i}(v)$$

  Example

  $Map_{cond}(\text{Balance} = 100) \Longrightarrow I_B = Map_{Balance}(100) = \mu$

- $A_i < v$. The mapping depends on whether or not the mapping function Map_{A_i} is order-preserving or random
  - order-preserving: $Map_{cond}(A_i < v) \Longrightarrow I_i \leq Map_{A_i}(v)$
  - random: check if attribute I_i lies in any of the partitions that may contain a value v' where $v' < v$: $Map_{cond}(A_i < v) \Longrightarrow I_i \in Map^{<}_{A_i}(v)$

  Example

  $Map_{cond}(\text{Balance} < 200) \Longrightarrow I_B \in \{\mu, \kappa\}$

- $A_i > v$. Symmetric with respect to $A_i < v$

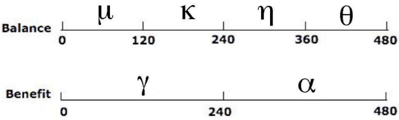
Mapping conditions Map_{cond} – 2

- $A_i = A_j$. The translation is performed by considering all possible pairs of partitions of A_i and A_j that overlap. Formally:

$$Map_{cond}(A_i = A_j) \Longrightarrow \bigvee_{\varphi}(I_i = ident_{A_i}(p_k) \wedge I_j = ident_{A_j}(p_l))$$

where φ is $p_k \in$partition(A_i), $p_l \in$partition(A_j), $p_k \cap p_l \neq \emptyset$

Example

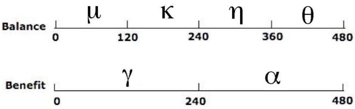


$$\begin{aligned} Map_{cond}(\text{Balance=Benefit}) \Longrightarrow \quad & (\text{Balance=}\mu \wedge \text{Benefit=}\gamma) \\ \vee\ & (\text{Balance=}\kappa \wedge \text{Benefit=}\gamma) \\ \vee\ & (\text{Balance=}\eta \wedge \text{Benefit=}\alpha) \\ \vee\ & (\text{Balance=}\theta \wedge \text{Benefit=}\alpha) \end{aligned}$$

Mapping conditions Map_{cond} – 3

- $A_i < A_j$. The mapping depends on whether or not the mapping functions Map_{A_i} and Map_{A_j} are order-preserving or random
  - Map_{A_i} and Map_{A_j} are both random: the translation considers all pairs of partitions of A_i and A_j that could satisfy the condition.
$$Map_{cond}(A_i < A_j) \Longrightarrow \bigvee_{\varphi}(I_i = ident_{A_i}(p_k) \wedge I_j = ident_{A_j}(p_l))$$
    where φ is $p_k \in$partition(A_i), $p_l \in$partition(A_j), $p_l.high \geq p_k.low$

Example



$$\begin{aligned} Map_{cond}(\text{Balance}<\text{Benefit}) \Longrightarrow \quad & (\text{Balance}=\mu \wedge \text{Benefit}=\gamma) \\ \vee\ & (\text{Balance}=\mu \wedge \text{Benefit}=\alpha) \\ \vee\ & (\text{Balance}=\kappa \wedge \text{Benefit}=\gamma) \\ \vee\ & (\text{Balance}=\kappa \wedge \text{Benefit}=\alpha) \\ \vee\ & (\text{Balance}=\eta \wedge \text{Benefit}=\alpha) \\ \vee\ & (\text{Balance}=\theta \wedge \text{Benefit}=\alpha) \end{aligned}$$

Query execution

- Each query Q on the plaintext DB is translated into:
  - a query Q_s to be executed at the server
  - a query Q_c to be executed at client on the result
- Query Q_s is defined by exploiting the definition of *Map*$_{cond}(C)$
- Query Q_c is executed on the decrypted result of Q_s to filter out spurious tuples
- The translation should be performed in such a way that the server is responsible for the majority of the work

Query execution – Simple example

Accounts

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

Accounts$_2^k$

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

Original query on Accounts	**Translation over Accounts$_2^k$**

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

Query execution – Simple example

Accounts

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

Accounts_2^k

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

Original query on Accounts	Translation over Accounts_2^k

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

Query execution – Simple example

Accounts

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

Accounts$_2^k$

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

Original query on Accounts	**Translation over Accounts$_2^k$**

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

Query execution – Simple example

Accounts

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

Accounts$_2^k$

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

Original query on Accounts	Translation over Accounts$_2^k$

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

Query execution – Simple example

Accounts

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

Accounts$_2^k$

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

Original query on Accounts	**Translation over Accounts$_2^k$**

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

Hash-based index [CDDJPS-05]

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

An example of encrypted relation with hashing

Accounts

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

Accounts$_2^k$

<u>Enc_tuple</u>	$\mathbf{I_A}$	$\mathbf{I_C}$	$\mathbf{I_B}$
x4Z3tfX2ShOSM	π	α	μ
mNHg1oC010p8w	ϖ	α	κ
WsIaCvfyF1Dxw	ξ	δ	θ
JpO8eLTVgwV1E	ρ	α	κ
qctG6XnFNDTQc	ς	β	κ
4QbqC3hxZHklU	ι	β	κ

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

Query conditions supported by the hash-based index

- Support queries where conditions are boolean formulas over terms of the form
  - *Attribute = Value*
  - *Attribute1 = Attribute2*, if *Attribute1* and *Attribute2* are indexed with the same hash function
- It does not support range queries (a solution similar to the one adopted for partition-based methods is not viable)
  - colliding values in general are not contiguous in the plaintext domain
- Query translation works like in the partition-based method

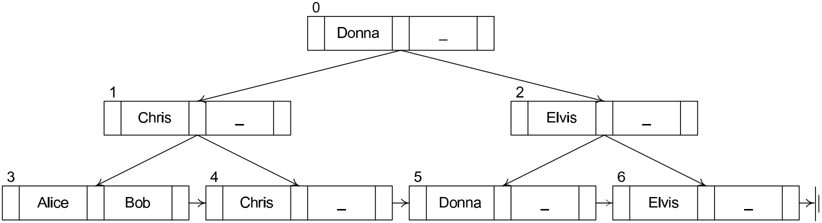
Interval-based queries [CDDJPS-05]

- Order-preserving indexing techniques (e.g., [AKSX-04]): support interval-based queries but expose to inference
  - comparing the ordered sequences of plaintext and indexes would lead to reconstruct the correspondence
- Non order-preserving techniques: data are not exposed to inference but interval-based queries are not supported
- DBMSs support interval-based queries using B+-trees, but the B+-tree defined by the server on indexes is of no use

  Possible solution:

  - Calculate the nodes in the B+-tree at the client and encrypt each node as a whole at the server
  - B+-tree traversal must be performed at the trusted front-end

B+-tree example – 1



B+-tree Table

ID	Node
0	(1,Donna,2,_,_)
1	(3,Chris,4,_,_)
2	(5,Elvis,6,_,_)
3	(Alice,Bob,4)
4	(Chris,_,5)
5	(Donna,_,6)
6	(Elvis,_,_)

Encrypted B+-tree Table

ID	Enc_Node
0	/WKu5y8laqK82(
1	AXYaqohgyVObU
2	IUf7R.PK5h5fU
3	uOtdm/HDXNSqU
4	GLDWRnBGIvYBA
5	a9yl36PA3LeLk
6	H6GwdJpXiU8MY

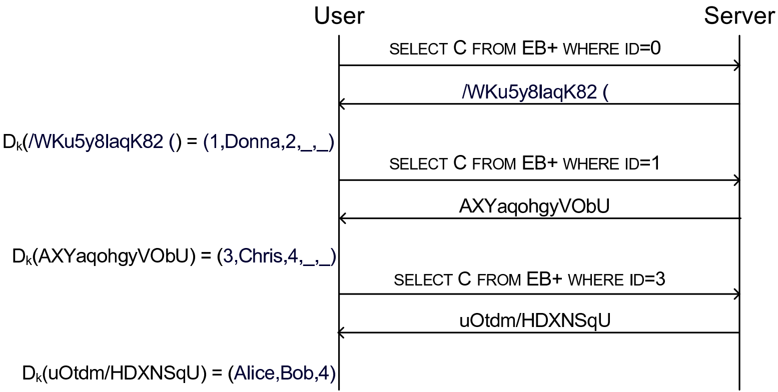
©Security, Privacy, and Data Protection Laboratory (SPDP Lab)

B+-tree example – 2

Query on the plaintext relation

SELECT * FROM Accounts WHERE Customer = ‘Bob’

Interaction for query evaluation



User

Server

SELECT C FROM EB+ WHERE ID=0

/WKu5y8IaqK82 (

D_k(/WKu5y8IaqK82 () = (1,Donna,2,_,_)

SELECT C FROM EB+ WHERE ID=1

AXYaqohgyVObU

D_k(AXYaqohgyVObU) = (3,Chris,4,_,_)

SELECT C FROM EB+ WHERE ID=3

uOtdm/HDXNSqU

D_k(uOtdm/HDXNSqU) = (Alice,Bob,4)

Searchable encryption

Order preserving encryption

- Order Preserving Encryption Schema (OPES) takes as input a target distribution of index values and applies an order preserving transformation [AKS-04] so that the resulting index values follow the target distribution
  - + comparison can be directly applied on the encrypted data
  - + query evaluation does not produce spurious tuples
  - – vulnerable with respect to inference attacks
- Order Preserving Encryption with Splitting and Scaling (OPESS) schema creates index values so that their frequency distribution is flat [WL-06]

Fully homomorphic encryption [G-09, GKPVZ-13]

Fully homomorphic encryption schema:

- allows performing specific computation on encrypted data
- decryption of the computation result, yields the result of operations performed on the plaintext data

Recent advancement: a functional-encryption schema that fits together several existing schemes (homomorphic encryption, garbled circuit, attribute-based encryption) [GKPVZ-13]

- still too computationally intensive for practical DBMS applications

Inference exposure

A. Ceselli, E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, "Modeling and Assessing Inference Exposure in Encrypted Databases," in *ACM TISSEC*, vol. 8, no. 1, February 2005.

Inference exposure

There are two conflicting requirements in indexing data:

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

It is important to measure quantitatively the level of exposure due to the publication of indexes:

ε = Exposure Coefficient

Scenarios

The computation of the exposure coefficient ε depends on two factors:

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

Possible inferences

Freq+DB^k

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

DB+DB^k

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

Exposure coefficient computation [CDDJPS-05]

	Direct Encryption	**Hashing**
Freq+DB^k	Quotient Table	Multiple subset sum problem
DB+DB^k	RCV graph	RCV line graph

Freq+DB^k – Example

Knowledge

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

Accounts$_1^k$

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

Inference

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

Direct encryption – Freq+DB^k

- Correspondence between an index and a plaintext value can be determined based on the number of occurrences of the index/value
  - Basic protection: values with the same number of occurrences are indistinguishable to the attacker
- Assessment of index exposure based on equivalence relation where index/plaintext values with same number of occurrences belong to the same class
  - Exposure of values in equivalence class C is $1/\mid C\mid$

Freq+DB^k – Example of exposure computation

A.1 = $\{\pi, \varpi, \xi, \rho, \varsigma, \iota\}$ = {Acc1,...,Acc6}

C.1 = $\{\beta, \gamma, \delta, \varepsilon\}$ = {Bob,Chris,Donna,Elvis}

C.2 = $\{\alpha\}$ = {Alice}

B.1 = $\{\mu, \eta, \theta\}$ = {100,300,400}

B.3 = $\{\kappa\}$ = {200}

INDEX_VALUES

$\mathbf{I_A}$	$\mathbf{I_C}$	$\mathbf{I_B}$
π	α	μ
ϖ	α	κ
ξ	β	η
ρ	γ	κ
ς	δ	θ
ι	ε	κ

QUOTIENT

qt$_A$	**qt**$_C$	**qt**$_B$
A.1	C.2	B.1
A.1	C.2	B.3
A.1	C.1	B.1
A.1	C.1	B.3
A.1	C.1	B.1
A.1	C.1	B.3

INVERSE CARDINALITY

ic$_A$	**ic**$_C$	**ic**$_B$
$1/6$	1	$1/3$
$1/6$	1	1
$1/6$	$1/4$	$1/3$
$1/6$	$1/4$	1
$1/6$	$1/4$	$1/3$
$1/6$	$1/4$	1

$$\mathscr{E} = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{k} \mathrm{IC}_{i,j} = 1/18$$

Direct encryption – DB+DB^k

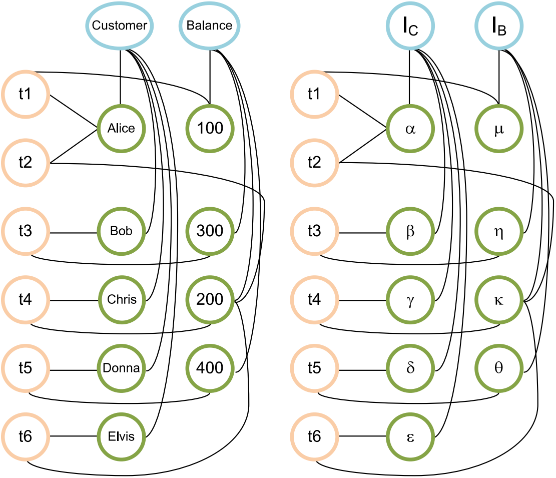
- 3-colored undirected Row-Column-Value graph:
  - one vertex of color “column” for every attribute
  - one vertex of color “row” for every tuple
  - one vertex for every distinct value in a column
  - an arc connects every value to the column and row(s) in which it appears
- RCV on plaintext values is identical to the one on indexes
- Inference exposure can be measured by evaluating the automorphisms of the graph
- Not sufficient to count the number of automorphisms:
  - if there are K automorphisms and in k of them the label assigned to v_i is the same, there is a probability of k/K of identifying the value

DB+DB^k – Example (1)

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

$\mathbf{I_C}$	$\mathbf{I_B}$
α	μ
α	κ
β	η
γ	κ
δ	θ
ε	κ

DB+DB^k – Example (2)



Inference

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

Computing the exposure coefficient

- The set of automorphisms constitutes a group described by the coarsest equitable partition of the vertices:
    - each subset appearing in the partition contains vertices that can be substituted one for the other in an automorphism

- Nauty algorithm: iteratively derives the partition

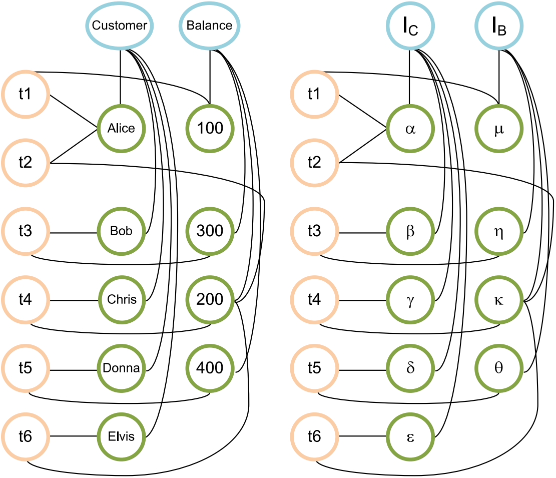
- Probability of identifying a vertex in partition C: $1/\mid C\mid$

Exposure with equitable partition of n elements over a total number of m: n/m

Example

- β indistinguishable from δ
- η indistinguishable from θ
- γ indistinguishable from ε

Computing the exposure coefficient – Example



Inference

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

Equitable partition: $\{(\alpha),(\beta,\delta),(\gamma,\varepsilon),(\mu),(\eta,\theta),(\kappa)\}$

$\mathscr{E} = 6/9 = 2/3$

Hashing exposure – Freq+DB^k

- The hash function is characterized by a collision factor, denoting the number of attribute values that on average collide on the same index value
- There are different possible mappings of plaintext values in index values, w.r.t. the constraints imposed by frequencies
- Need to enumerate the different mappings by using an adaptation of Pisinger's algorithm for the subset sum problem
- Compute the exposure coefficient for each mapping

Hashing exposure – DB+DB^k

- The RCV-graph built on plaintext and encrypted data are not identical

- Different vertexes of the plaintext RCV-graph may collapse to the same encrypted RCV-graph vertex

- The number of edges connecting row vertexes to value vertexes in the plaintext and encrypted RCV-graph is the same

- The problem becomes finding a correct matching between the edges of the plaintext RCV-graph and the edges of the encrypted RCV-graph

Bloom Filter

Bloom filter [B-70]

A Bloom filter is at the basis of the construction of some indexing techniques. It is an efficient method to encode set membership

- Set of n elements (n is large)
- Vector of l bits (l is small)
- h independent hash functions $H_i : \{0,1\}^* \rightarrow [1,l]$

Insert element x:

- Sets to 1 the bit values at index positions $H_1(x), H_2(x), \ldots, H_h(x)$

Search element x:

- Compute $H_1(x), H_2(x), \ldots, H_h(x)$ and check whether those values are set in the bit vector

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

1	2	3	4	5	6	7	8	9	10

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

	1			1				1	
1	2	3	4	5	6	7	8	9	10

- Insert sun: H_1(sun)=2; H_2(sun)=5; H_3(sun)=9

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

1	1			1		1		1	
1	2	3	4	5	6	7	8	9	10

- Insert sun: H_1(sun)=2; H_2(sun)=5; H_3(sun)=9
- Insert frog: H_1(frog)=1; H_2(frog)=5; H_3(frog)=7

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

1	**1**			**1**		1		1	
1	2	3	4	5	6	7	8	9	10

- Insert sun: H_1(sun)=2; H_2(sun)=5; H_3(sun)=9

- Insert frog: H_1(frog)=1; H_2(frog)=5; H_3(frog)=7

- Search dog: H_1(dog)=2; H_2(dog)=5; H_3(dog)=10

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

1	**1**			**1**		1		1	
1	2	3	4	5	6	7	8	9	10

- Insert sun: H_1(sun)=2; H_2(sun)=5; H_3(sun)=9
- Insert frog: H_1(frog)=1; H_2(frog)=5; H_3(frog)=7
- Search dog: H_1(dog)=2; H_2(dog)=5; H_3(dog)=10
  $\Longrightarrow$ No

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

1	1			**1**		1		**1**	
1	2	3	4	5	6	7	8	9	10

- Insert sun: H_1(sun)=2; H_2(sun)=5; H_3(sun)=9
- Insert frog: H_1(frog)=1; H_2(frog)=5; H_3(frog)=7
- Search dog: H_1(dog)=2; H_2(dog)=5; H_3(dog)=10 $\Longrightarrow$ No
- Search car: H_1(car)=1; H_2(car)=5; H_3(car)=9

Bloom filter [B-70] – Example

Let $l = 10$ and $h = 3$

1	1			**1**		1		**1**	
1	2	3	4	5	6	7	8	9	10

- Insert sun: H_1(sun)=2; H_2(sun)=5; H_3(sun)=9
- Insert frog: H_1(frog)=1; H_2(frog)=5; H_3(frog)=7
- Search dog: H_1(dog)=2; H_2(dog)=5; H_3(dog)=10
  $\Longrightarrow$ No
- Search car: H_1(car)=1; H_2(car)=5; H_3(car)=9
  $\Longrightarrow$ Maybe Yes; false positive!

Bloom filter – Properties

- Generalization of hashing (Bloom filter with one hash function is equivalent to ordinary hashing)
  - + space efficient (roughly ten bit for every element in the dictionary with 1% error)
  - – elements cannot be removed
- Yield a constant false positive probability
  - – theoretically considered not acceptable
  - + acceptable in practical applications as fine price to pay for space efficiency

Data Integrity

Integrity of outsourced data

Two aspects:

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

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

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

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

Integrity in storage

- Data integrity in storage relies on digital signatures

- Signatures are usually computed at tuple level
  - table and attribute level signatures can be verified only after downloading the whole table/column
  - cell level signature causes a high verification overhead

- The verification cost grows linearly with the number of tuples in the query result

  $\Longrightarrow$ the signature of a set of tuples can be combined to generate the aggregated signature [MNT-06]

Selective Encryption and Over-Encryption

S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Encryption Policies for Regulating Access to Outsourced Data,” in *ACM TODS,* vol. 35, no. 2, April 2010.

Selective information sharing

- Different users might need to enjoy different views on the outsourced data

- Enforcement of the access control policy requires the data owner to mediate access requests

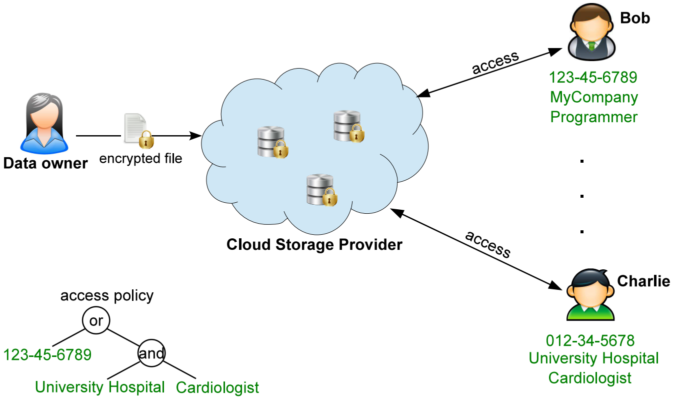
  $\Longrightarrow$ impractical (if not inapplicable)

- Authorization enforcement may not be delegated to the provider

  $\Longrightarrow$ data owner should remain in control

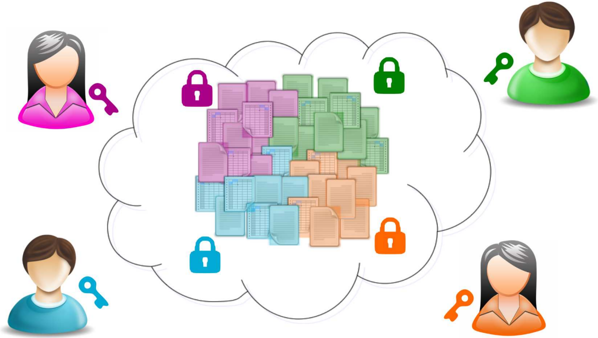
Selective information sharing: Approaches – 1

- Attribute-based encryption (ABE): allow derivation of a key only by users who hold certain attributes (based on asymmetric cryptography)

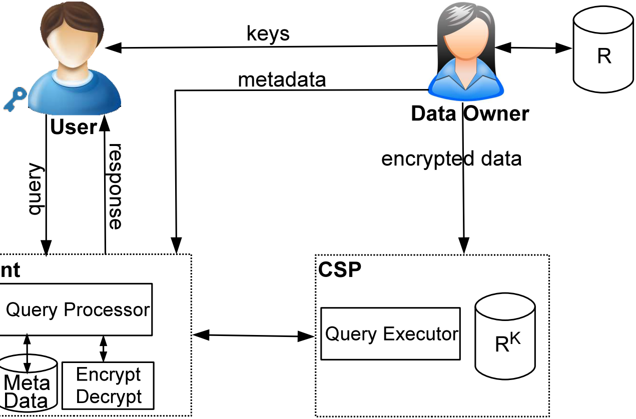


Selective information sharing: Approaches – 2

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



Selective encryption – Scenario



Selective encryption [DFJPS-10b]

Basic idea/desiderata:

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

Authorization policy

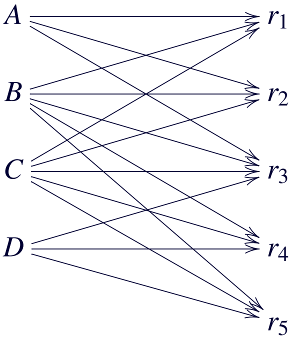
- The data owner defines a discretionary access control (authorization) policy to regulate read access to the resources

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

- Basic idea:
  - different ACLs implies different encryption keys

Authorization policy – Example

$$\begin{array}{c|ccccc|}
 & r_1 & r_2 & r_3 & r_4 & r_5 \\
\hline
A & 1 & 1 & 1 & 0 & 0 \\
B & 1 & 1 & 1 & 1 & 1 \\
C & 1 & 1 & 1 & 1 & 1 \\
D & 0 & 0 & 1 & 1 & 1 \\
\hline
\end{array}$$



Encryption policy

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

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

Key derivation methods

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

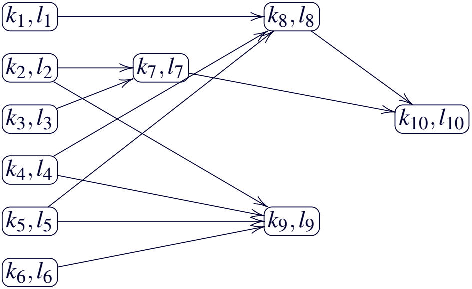
Token-based key derivation methods [AFB-05]

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

Key and token graph

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

Example



Key assignment and encryption schema

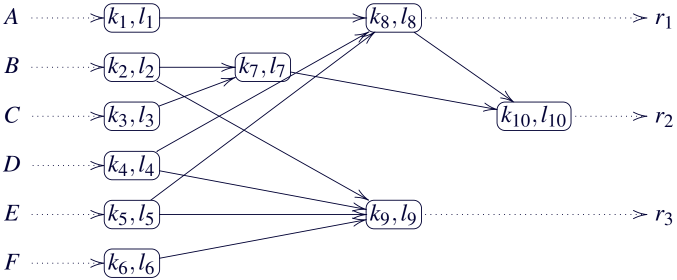
Translation of the authorization policy into an encryption policy:

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

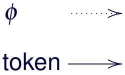
Formal definition of encryption policy

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

Encryption policy graph – Example



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



Policy transformation

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

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

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

Translating $\mathscr{A}$ into $\mathscr{E}$ – 1

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

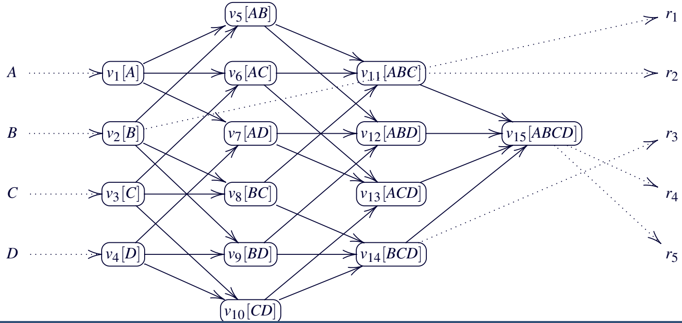
$\Longrightarrow$ producing and managing a token for each single permission can be unfeasible in practice

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

Translating $\mathscr{A}$ into $\mathscr{E}$ – 2

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

$\Longrightarrow$ inefficient key derivation



Minimum encryption policy

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

Construction of the key and token graph

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

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

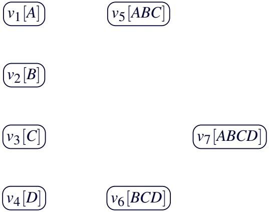
2. For each vertex v corresponding to a non-singleton *acl*, find a cover without redundancies (covering)
   - for each user u in v.*acl*, find an ancestor v' of v with $u \in v'$.*acl*

3. Factorize common ancestors (factorization)

Key and token graph – Example

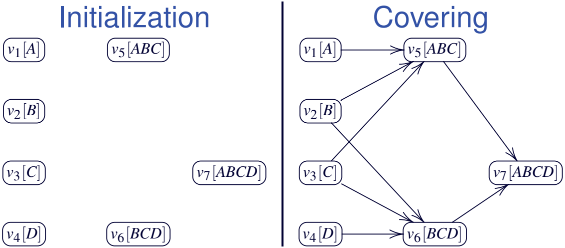
$$
\begin{array}{c|ccccc|}
 & r_1 & r_2 & r_3 & r_4 & r_5 \\
\hline
A & 0 & 1 & 0 & 1 & 1 \\
B & 1 & 1 & 1 & 1 & 1 \\
C & 0 & 1 & 1 & 1 & 1 \\
D & 0 & 0 & 1 & 1 & 1 \\
\hline
\end{array}
$$

Initialization



Key and token graph – Example

$$
\begin{array}{c|ccccc|}
 & r_1 & r_2 & r_3 & r_4 & r_5 \\
\hline
A & 0 & 1 & 0 & 1 & 1 \\
B & 1 & 1 & 1 & 1 & 1 \\
C & 0 & 1 & 1 & 1 & 1 \\
D & 0 & 0 & 1 & 1 & 1 \\
\hline
\end{array}
$$

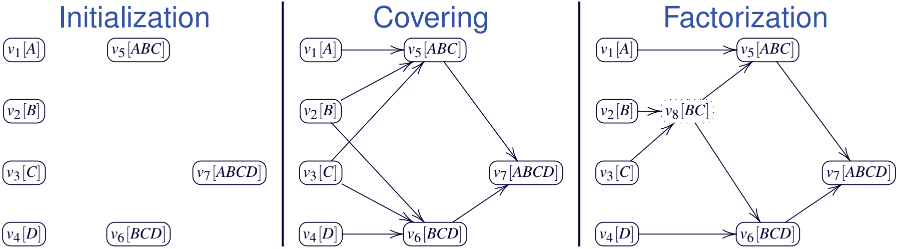


Initialization

Covering

Key and token graph – Example

$$\begin{array}{c|ccccc|}
 & r_1 & r_2 & r_3 & r_4 & r_5 \\
\hline
A & 0 & 1 & 0 & 1 & 1 \\
B & 1 & 1 & 1 & 1 & 1 \\
C & 0 & 1 & 1 & 1 & 1 \\
D & 0 & 0 & 1 & 1 & 1 \\
\hline
\end{array}$$

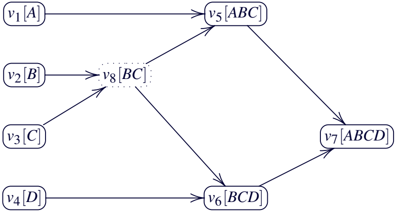


Initialization

Covering

Factorization

Key assignment and encryption schema ϕ and catalog



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

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

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

Policy updates

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

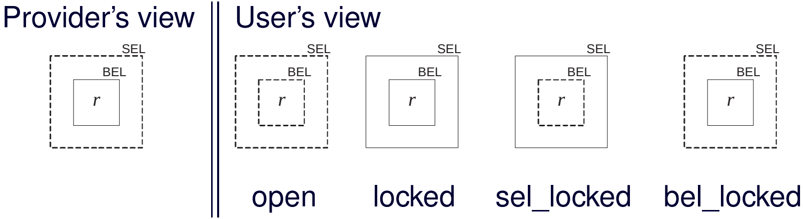
  $\Longrightarrow$ inefficient

- Possible solution: over-encryption

Over-encryption – 1

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

Over-encryption – 2



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

Over-encryption – 3

- Revoke

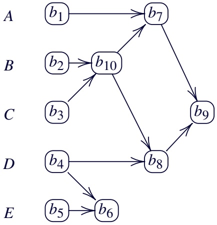
  to protect resources for which the revokee has the BEL key

- Grant

  if a BEL key protects multiple resources and access is to be granted only to a subset of them, there is the need to protect at SEL level the resources on which access is not being granted

An example of revoke operation

BEL	SEL



r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4,r_5	$b_6.l_a$

$\boxed{s_1[A]}$

$\boxed{s_2[B]}$

$\boxed{s_3[C]}$

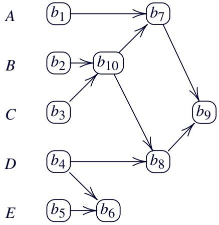
$\boxed{s_4[D]}$

$\boxed{s_5[E]}$

r	$\phi_S(r)$
r_1,r_2,r_3,r_4,r_5	NULL

An example of revoke operation

BEL	SEL
revoke(B, r_3)	



r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4, r_5	$b_6.l_a$

$s_1[A]$

$s_2[B]$

$s_3[C]$

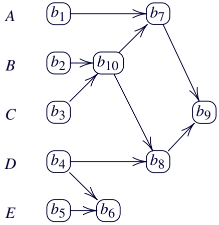
$s_4[D]$

$s_5[E]$

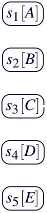
r	$\phi_S(r)$
r_1, r_2, r_3, r_4, r_5	NULL

An example of revoke operation

BEL	SEL
revoke(B, r_3)	**over_encrypt**(CD, r_3)



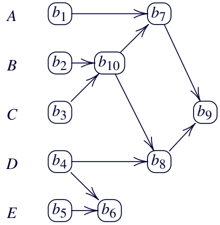
r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4, r_5	$b_6.l_a$



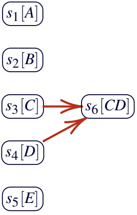
r	$\phi_S(r)$
r_1, r_2, r_3, r_4, r_5	NULL

An example of revoke operation

BEL	SEL
revoke(B, r_3)	**over_encrypt**(CD, r_3)



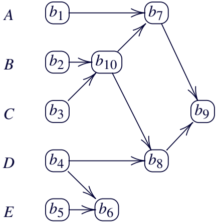
r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4, r_5	$b_6.l_a$



r	$\phi_s(r)$
$r_1, r_2,$ ~~r_3~~$, r_4, r_5$	NULL
r_3	$s_6.l$

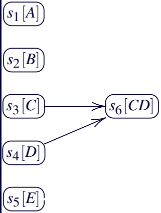
An example of grant operation

BEL



r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4,r_5	$b_6.l_a$

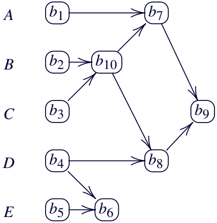
SEL



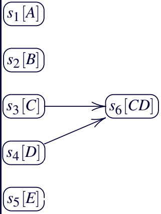
r	$\phi_S(r)$
r_1,r_2,r_4,r_5	NULL
r_3	$s_6.l$

An example of grant operation

BEL	SEL
grant(C, r_4)	



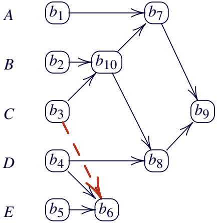
r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4,r_5	$b_6.l_a$



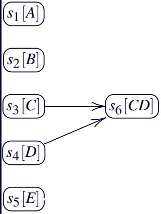
r	$\phi_S(r)$
r_1,r_2,r_4,r_5	NULL
r_3	$s_6.l$

An example of grant operation

BEL	SEL
grant(C, r_4)	



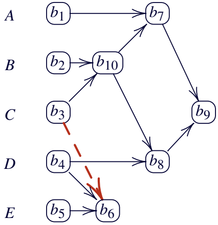
r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4, r_5	$b_6.l_a$



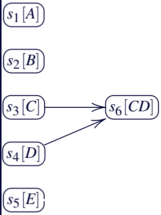
r	$\phi_S(r)$
r_1, r_2, r_4, r_5	NULL
r_3	$s_6.l$

An example of grant operation

BEL	SEL
grant(C, r_4)	**over_encrypt**(DE, r_5)



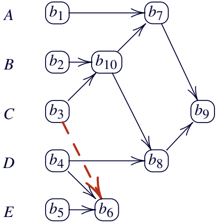
r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4, r_5	$b_6.l_a$



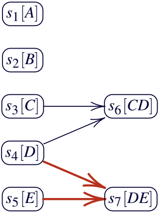
r	$\phi_S(r)$
r_1, r_2, r_4, r_5	NULL
r_3	$s_6.l$

An example of grant operation

BEL	SEL
grant(C,r_4)	**over_encrypt**(DE,r_5)



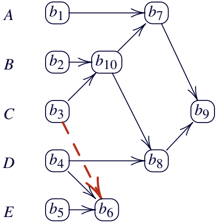
r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4,r_5	$b_6.l_a$



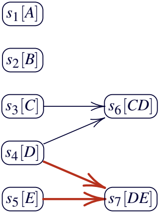
r	$\phi_S(r)$
$r_1,r_2,r_4,$~~r_5~~	NULL
r_3	$s_6.l$
r_5	$s_7.l$

An example of grant operation

BEL	SEL
grant(C,r_4)	**over_encrypt**(DE,r_5) **over_encrypt**(ALL,r_4)



r	$\phi_b(r)$
r_1	$b_7.l_a$
r_2	$b_9.l_a$
r_3	$b_8.l_a$
r_4,r_5	$b_6.l_a$



r	$\phi_S(r)$
r_1,r_2,r_4,~~r_5~~	NULL
r_3	$s_6.l$
r_5	$s_7.l$

Mix&Slice for Policy Revocation

E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, P. Samarati, “Mix&Slice: Efficient Access Revocation in the Cloud,” in *Proc. of the 23rd ACM Conference on Computer and Communications Security (CCS 2016)*, Vienna, Austria, October 2016.

Mix&Slice

- Over-encryption requires support by the server (i.e., the server implements more than simple get/put methods)

- Alternative solution to enforce revoke operations: Mix&Slice

- Use different rounds of encryption to provide complete mixing of the resource

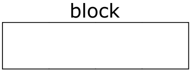
  ⟹ unavailability of a small portion of the encrypted resource prevents its (even partial) reconstruction

- Slice the resource into fragments and, every time a user is revoked access to the resource, re-encrypt a randomly chosen fragment

  ⟹ lack of a fragment prevents resource decryption

Resource organization

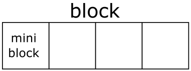
- Block: sequence of bits input to a block cipher
  AES uses block of 128 bits



Resource organization

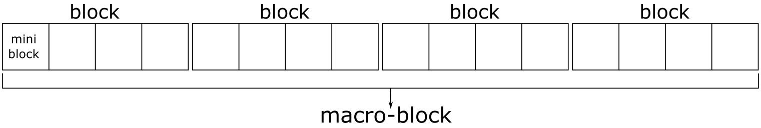
- Block: sequence of bits input to a block cipher
  AES uses block of 128 bits

- Mini-block: sequence of bits in a block
  it is our atomic unit of protection
  mini-blocks of 32 bits imply a cost of
  2^{32} for brute-force attacks



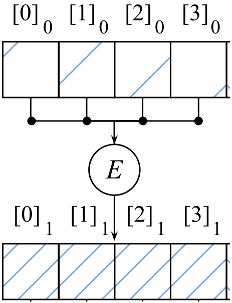
Resource organization

- Block: sequence of bits input to a block cipher
  AES uses block of 128 bits
- Mini-block: sequence of bits in a block
  it is our atomic unit of protection
  mini-blocks of 32 bits imply a cost of
  2^{32} for brute-force attacks
- Macro-block: sequence of blocks
  mixing operates at the level of macro-block
  a macro-block of 1KB includes 8 blocks



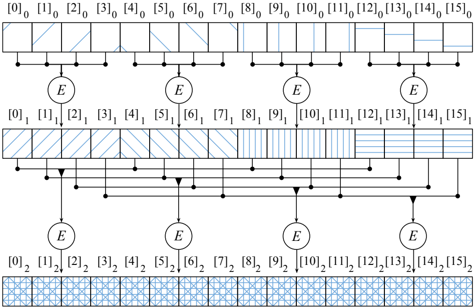
Mixing – 1

- When encryption is applied to a block, all the mini-blocks are mixed
  - + absence of a mini-block in a block from the result prevents reconstruction of the block
  - – does not prevent the reconstruction of other blocks in the resource



Mixing – 2

- Extend mixing to a macro-block
  - iteratively apply block encryption
  - at iteration i, each block has a mini-block for each encrypted block obtained at iteration $i-1$ (at distance 2^i)
  - x rounds mix 4^x mini-blocks

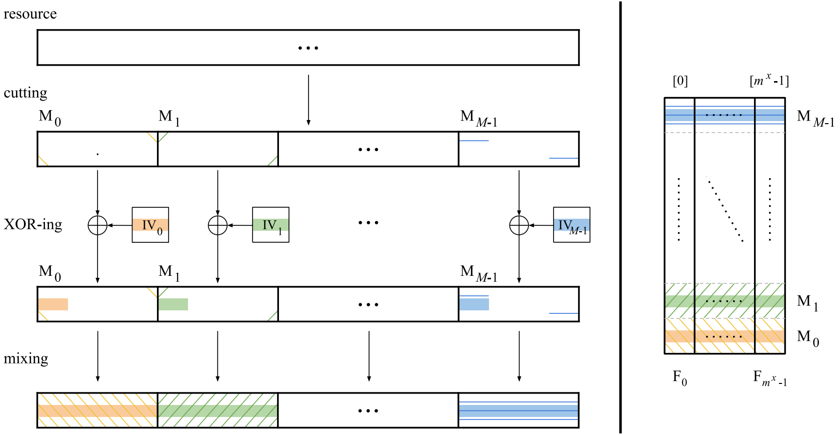


Slicing – 1

- To be mixed, large resources require large macro-blocks
  - many rounds of encryption
  - considerable computation and data transfer overhead
- Large resources are split in different macro-blocks for encryption
- Absence of a mini-block for each macro-block prevents the (even partial) reconstruction of the resource

Slicing – 2

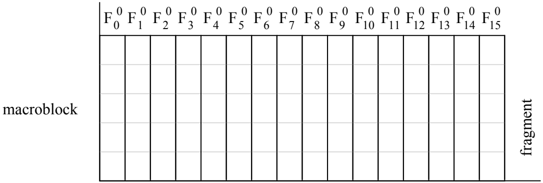
- Slice resources in fragments having a mini-block for each macro-block (the ones in the same position)
  - absence of a fragment prevents reconstruction of the resource



Revoke

To revoke user u access to a resource r

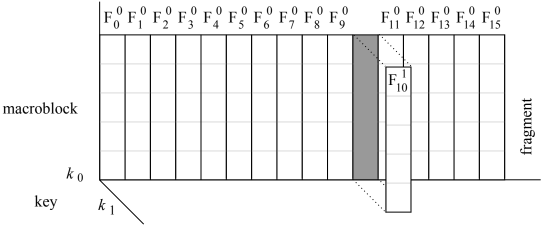
1. randomly select a fragment F_i of r and download it
2. decrypt F_i
3. generate a new key k_l that u does not know and cannot derive (generated with key regression and seed encrypted with new ACL)
4. re-encrypt F_i with the new key k_l
5. upload the encrypted fragment


F_0^0 F_1^0 F_2^0 F_3^0 F_4^0 F_5^0 F_6^0 F_7^0 F_8^0 F_9^0 F_{10}^0 F_{11}^0 F_{12}^0 F_{13}^0 F_{14}^0 F_{15}^0
macroblock
fragment

Revoke

To revoke user u access to a resource r

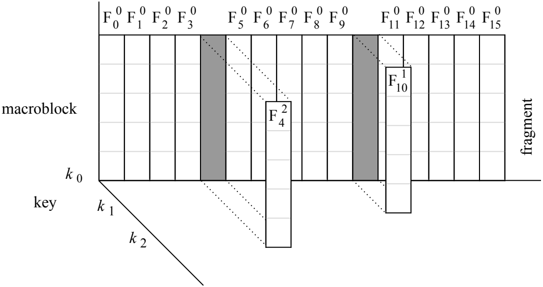
1. randomly select a fragment F_i of r and download it
2. decrypt F_i
3. generate a new key k_l that u does not know and cannot derive (generated with key regression and seed encrypted with new ACL)
4. re-encrypt F_i with the new key k_l
5. upload the encrypted fragment


F_0^0 F_1^0 F_2^0 F_3^0 F_4^0 F_5^0 F_6^0 F_7^0 F_8^0 F_9^0 F_{11}^0 F_{12}^0 F_{13}^0 F_{14}^0 F_{15}^0
F_{10}^1
macroblock
fragment
k_0
key
k_1

Revoke

To revoke user u access to a resource r

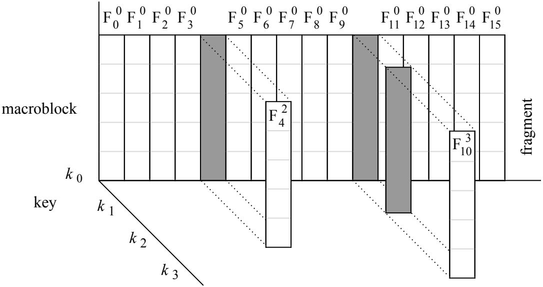
1. randomly select a fragment F_i of r and download it
2. decrypt F_i
3. generate a new key k_l that u does not know and cannot derive (generated with key regression and seed encrypted with new ACL)
4. re-encrypt F_i with the new key k_l
5. upload the encrypted fragment


F_0^0 F_1^0 F_2^0 F_3^0 F_5^0 F_6^0 F_7^0 F_8^0 F_9^0 F_{11}^0 F_{12}^0 F_{13}^0 F_{14}^0 F_{15}^0
F_{10}^1
F_4^2
macroblock
fragment
k_0
key
k_1
k_2

Revoke

To revoke user u access to a resource r

1. randomly select a fragment F_i of r and download it
2. decrypt F_i
3. generate a new key k_l that u does not know and cannot derive (generated with key regression and seed encrypted with new ACL)
4. re-encrypt F_i with the new key k_l
5. upload the encrypted fragment



F_0^0 F_1^0 F_2^0 F_3^0 F_5^0 F_6^0 F_7^0 F_8^0 F_9^0 F_{11}^0 F_{12}^0 F_{13}^0 F_{14}^0 F_{15}^0
macroblock
F_4^2
F_{10}^3
fragment
k_0
key
k_1
k_2
k_3

Effectiveness of the approach

- A revoked user does not know the encryption key of at least one fragment
  - a brute force attack is needed to reconstruct the fragment (and the resource)
  - 2^{msize} attempts, with msize the number of bits in a mini-block
- A user can locally store f_{loc} of the f fragments of a resource
  - probability to be able to reconstruct the resource after f_{miss} fragments have been re-encrypted: $P = (f_{\text{loc}}/f)^{f_{\text{miss}}}$
    - proportional to the number of locally stored fragments
    - decreases exponentially with the number of policy updates

Other issues

- Support for write privileges for data collections with multiple owners

- Selective encryption for supporting subscription-based authorization policies [DFJL-12]
  - users are authorized to access all and only the resources published during their subscribed periods
  - user authorizations remain valid also after the expiration of their subscriptions

    ⟹ need to take into account both the subscriptions of the users and the time when resources have been published

Fragmentation and Encryption

Fragmentation and encryption

- Encryption makes query evaluation and application execution more expensive or not always possible

- Often what is sensitive is the association between values of different attributes, rather than the values themselves
    - ○ e.g., association between employee's names and salaries

  ⟹protect associations by breaking them, rather than encrypting

- Recent solutions for enforcing privacy requirements couple:
    - ○ encryption
    - ○ data fragmentation

Confidentiality constraints

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

- Sensitive attributes: the values of some attributes are considered sensitive and should not be visible
  $\Longrightarrow$ singleton constraints

- Sensitive associations: the associations among values of given attributes are sensitive and should not be visible
  $\Longrightarrow$ non-singleton constraints

Confidentiality constraints – Example

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

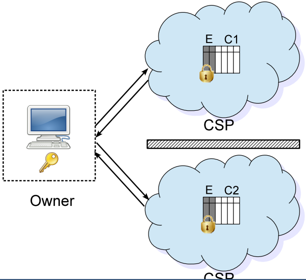
- {Telephone}, {Email}
  - attributes Telephone and Email are sensitive (cannot be stored in the clear)
- {Name,Salary}, {Name,Position}, {Name,DoB}
  - attributes Salary, Position, and DoB are private of an individual and cannot be stored in the clear in association with the name
- {DoB,Gender,Zip,Salary}, {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

Outline

- Data fragmentation
  - Non-communicating pair of servers [ABGGKMSTX-05]
  - Multiple non-linkable fragments [CDFJPS-07,CDFJPS-10]
  - Departing from encryption: Keep a few [CDFJPS-09b]
  - Fragmentation and inferences [DFJLPS-14]
- Publishing obfuscated associations
  - Anonymizing bipartite graph [CSYZ-08]
  - Fragments and loose associations [DFJPS-10]

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) [ABGGKMSTX-05]
  - Sensitive associations are protected by distributing the attributes among the two servers
  - Encryption is applied only when explicitly demanded by the confidentiality constraints or when storing an attribute in any of the two servers 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 $c \not\subseteq (R_2 - E)$

Non-communicating pair of servers – Example

PATIENTS

	SSN	**Name**	**YoB**	**Job**	**Disease**
t_1	123456789	Alice	1980	Clerk	Asthma
t_2	234567891	Bob	1980	Doctor	Asthma
t_3	345678912	Carol	1970	Nurse	Asthma
t_4	456789123	David	1970	Lawyer	Bronchitis
t_5	567891234	Eva	1970	Doctor	Bronchitis
t_6	678912345	Frank	1960	Doctor	Gastritis
t_7	789123456	Gary	1960	Teacher	Gastritis
t_8	891234567	Hilary	1960	Nurse	Diabetes

c_0 = {SSN}

c_1 = {Name, Disease}

c_2 = {Name, Job}

c_3 = {Job, Disease}

F_1

tid	**Name**	**YoB**	**SSN**^{k}	**Disease**^{k}
1	Alice	1980	jdkis	hyaf4k
2	Bob	1980	u9hs9	j97;qx
3	Carol	1970	j9und	9jp‘md
4	David	1970	p0vp8	p;nd92
5	Eva	1970	8nn[	0-mw-n
6	Frank	1960	j9jMK	wqp9[i
7	Gary	1960	87l’D	L0MB2G
8	Hilary	1960	8pm}n	@h8hwu

F_2

tid	**Job**	**SSN**^{k}	**Disease**^{k}
1	Clerk	uwq8hd	jsd7ql
2	Doctor	j-0.dl;	0],nid
3	Nurse	8ojqdkf	j-0/?n
4	Lawyer	j0i12nd	5lkdpq
5	Doctor	mj[9;’s	j0982e
6	Doctor	aQ14l[	jnd%d
7	Teacher	8qsdQW	OP[’
8	Nurse	0890UD	UP0D@

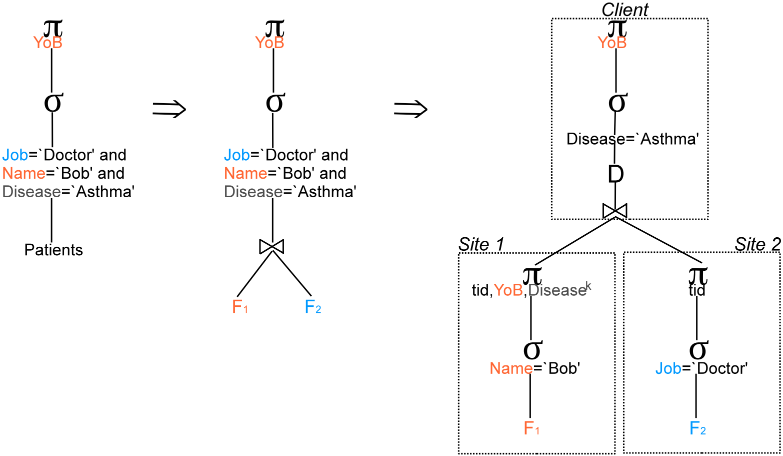
Query execution

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

- F_1: (tid,Name,YoB,SSN^k,Disease^k)
- F_2: (tid,Job,SSN^k,Disease^k)



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

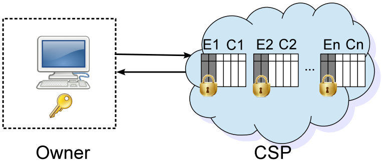
$$\sum_{i,j:i \in (R_1 - E), j \in (R_2 - E)} M_{i,j} + \sum_{i \in E} M_{i,i}$$

Multiple non-linkable fragments – 1

Coupling fragmentation and encryption is interesting and provides advantages, but assumption of two non-communicating 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

⟹ allow for more than two non-linkable fragments [CDFJPS-10]



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

Multiple non-linkable 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 non-linkable 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 non-linkable fragments – Example

PATIENTS

	SSN	**Name**	**YoB**	**Job**	**Disease**
t_1	123456789	Alice	1980	Clerk	Asthma
t_2	234567891	Bob	1980	Doctor	Asthma
t_3	345678912	Carol	1970	Nurse	Asthma
t_4	456789123	David	1970	Lawyer	Bronchitis
t_5	567891234	Eva	1970	Doctor	Bronchitis
t_6	678912345	Frank	1960	Doctor	Gastritis
t_7	789123456	Gary	1960	Teacher	Gastritis
t_8	891234567	Hilary	1960	Nurse	Diabetes

$c_0 = \{\text{SSN}\}$

$c_1 = \{\text{Name}, \text{Disease}\}$

$c_2 = \{\text{Name}, \text{Job}\}$

$c_3 = \{\text{Job}, \text{Disease}\}$

F_1

salt	**enc**	**Name**	**YoB**
s_{11}	Bd6!l3	Alice	1980
s_{12}	Oij3X.	Bob	1980
s_{13}	9kEf6?	Carol	1970
s_{14}	ker5/2	David	1970
s_{15}	C:mE91	Eva	1970
s_{16}	4lDwqz	Frank	1960
s_{17}	me3,op	Gary	1960
s_{18}	zWf4g>	Hilary	1960

F_2

salt	**enc**	**Job**
s_{21}	8de6TO	Clerk
s_{22}	X’mlE3	Doctor
s_{23}	wq.vy0	Nurse
s_{24}	nh=l3a	Lawyer
s_{25}	hh%kj)	Doctor
s_{26}	;vf5eS	Doctor
s_{27}	e4+YUp	Teacher
s_{28}	pgt6eC	Nurse

F_3

salt	**enc**	**Disease**
s_{31}	ew3)V!	Asthma
s_{32}	LkEd69	Asthma
s_{33}	w8vd66	Asthma
s_{34}	1"qPdd	Bronchitis
s_{35}	(mn2eW	Bronchitis
s_{36}	wD}x1X	Gastritis
s_{37}	0opAuEl	Gastritis
s_{38}	Sw@Fez	Diabetes

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**

Q :=SELECT SSN, Name
FROM PATIENTS
WHERE (Disease=‘Gastritis’ OR
Disease=‘Asthma’) AND
Job=‘Doctor’
	Q^3 :=SELECT salt, enc
	FROM F_3
	WHERE (Disease=‘Gastritis’ OR
	Disease=‘Asthma’)
	
	Q' := SELECT SSN, Name
	FROM *Decrypt*(Q^3, *Key*)
	WHERE Job=‘Doctor’

Optimization criteria

- Goal: find a fragmentation that makes query execution efficient
- The fragmentation process can then take into consideration different optimization criteria:
  - number of fragments [CDFJPS-07]
  - affinity among attributes [CDFJPS-10]
  - query workload [CDFJPS-09a]
- All criteria obey maximal visibility
  - only attributes that appear in singleton constraints (sensitive attributes) are encrypted
  - all attributes that are not sensitive appear in the clear in one fragment

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

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		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

<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		-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

<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		-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

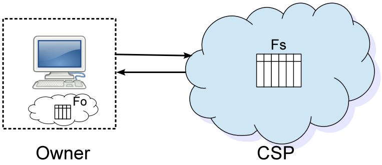
Fragmentation

Keep a few

Basic idea (hybrid scenarios):

- 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 [CDFJPS-09b, CDFJPS-11]



- $F_o \cup F_s = R$

Keep a few – 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

Keep a few – Example

PATIENTS

	SSN	Name	YoB	Job	Disease
t_1	123456789	Alice	1980	Clerk	Asthma
t_2	234567891	Bob	1980	Doctor	Asthma
t_3	345678912	Carol	1970	Nurse	Asthma
t_4	456789123	David	1970	Lawyer	Bronchitis
t_5	567891234	Eva	1970	Doctor	Bronchitis
t_6	678912345	Frank	1960	Doctor	Gastritis
t_7	789123456	Gary	1960	Teacher	Gastritis
t_8	891234567	Hilary	1960	Nurse	Diabetes

c_0 = {SSN}
c_1 = {Name, Disease}
c_2 = {Name, Job}
c_3 = {Job, Disease}

F_o

tid	SSN	Job	Disease
1	123456789	Clerk	Asthma
2	234567891	Doctor	Asthma
3	345678912	Nurse	Asthma
4	456789123	Lawyer	Bronchitis
5	567891234	Doctor	Bronchitis
6	678912345	Doctor	Gastritis
7	789123456	Teacher	Gastritis
8	891234567	Nurse	Diabetes

F_s

tid	Name	YoB
1	Alice	1980
2	Bob	1980
3	Carol	1970
4	David	1970
5	Eva	1970
6	Frank	1960
7	Gary	1960
8	Hilary	1960

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,Job,Disease}, F_s={Name,YoB}

- $q =$ SELECT SSN, YoB
  FROM Patients
  WHERE (Disease=“Bronchitis”)
  AND (YoB=“1970”)
  AND (Name=Job)

- The conditions in the WHERE clause are split as follows
  - $C_o = \{$Disease = “Bronchitis”$\}$
  - $C_s = \{$YoB = “1970”$\}$
  - $C_{so} = \{$Name = Job$\}$

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, YoB
FROM Patients
WHERE (Disease = “Bronchitis”)
AND (YoB = “1970”)
AND (Name = Job)

C_o={Disease = “Bronchitis”}
C_s={YoB = “1970”}
C_{so}={Name $=$ Job}

q_s = SELECT tid,Name,YoB
FROM F_s
WHERE YoB = “1970”

q_{so} = SELECT SSN, YoB
FROM F_o JOIN r_s
ON F_o.tid=r_s.tid
WHERE (Disease = “Bronchitis”) AND (Name = Job)

Client-server strategy – Example

q = SELECT SSN, YoB
FROM Patients
WHERE (Disease = “Bronchitis”)
AND (YoB = “1970”)
AND (Name = Job)

C_o={Disease = “Bronchitis”}
C_s={YoB = “1970”}
C_{so}={Name = Job}

q_o = SELECT tid
FROM F_o
WHERE Disease = “Bronchitis”

q_s = SELECT tid,Name,YoB
FROM F_s JOIN r_o ON F_s.tid=r_o.tid
WHERE YoB = “1970”

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

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

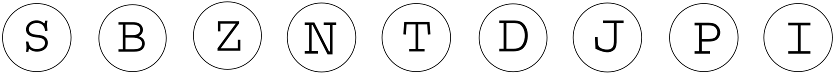
Different metrics could be applied splitting the attributes between F_o and F_s, such as minimizing:

- 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

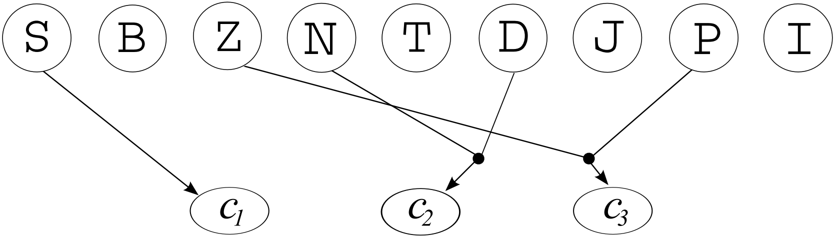
Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)



Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)

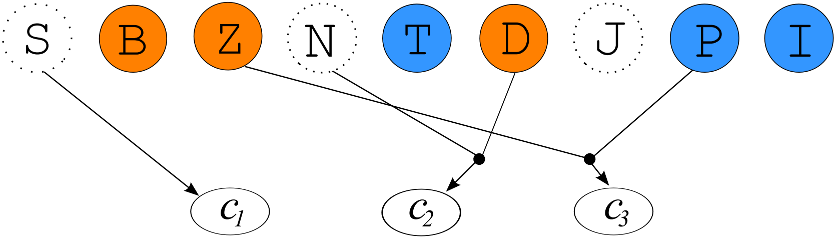


Constraints

c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)

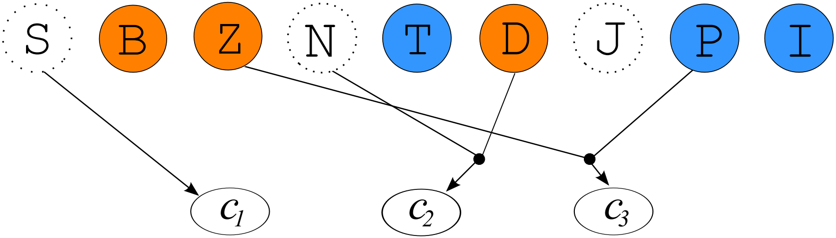


Constraints

c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)



Constraints

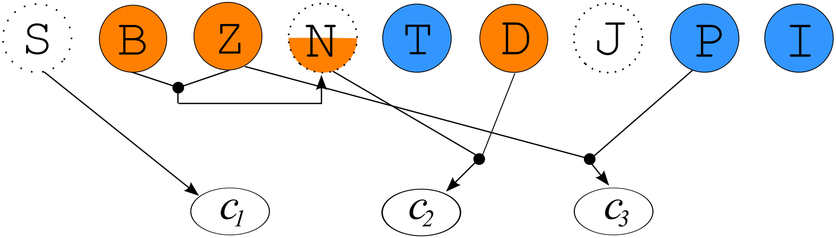
c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Dependencies

d_1 = {Birth, ZIP}⇝ Name
d_2 = {Treatment}⇝ Disease
d_3 = {Disease}⇝ Job
d_4 = {Insurance, Premium}⇝ Job

Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)



Constraints

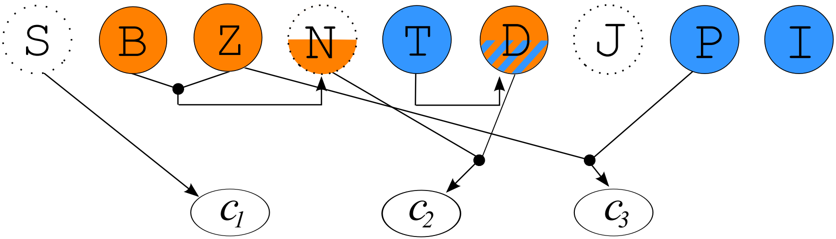
c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Dependencies

d_1 = {Birth, ZIP} $\rightsquigarrow$ Name
d_2 = {Treatment} $\rightsquigarrow$ Disease
d_3 = {Disease} $\rightsquigarrow$ Job
d_4 = {Insurance, Premium} $\rightsquigarrow$ Job

Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)



Constraints

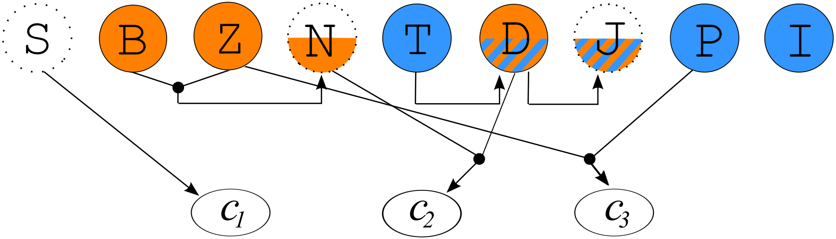
c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Dependencies

d_1 = {Birth, ZIP}$\rightsquigarrow$ Name
d_2 = {Treatment}$\rightsquigarrow$ Disease
d_3 = {Disease}$\rightsquigarrow$ Job
d_4 = {Insurance, Premium}$\rightsquigarrow$ Job

Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)



Constraints

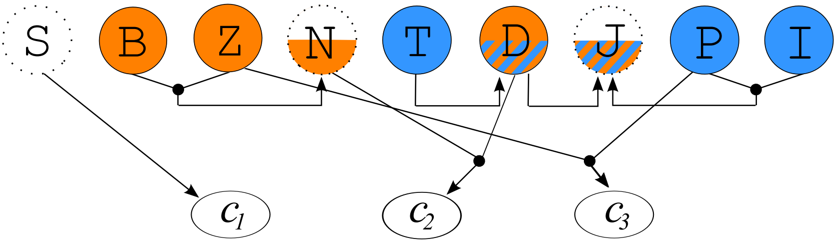
c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Dependencies

d_1 = {Birth, ZIP}$\rightsquigarrow$ Name
d_2 = {Treatment}$\rightsquigarrow$ Disease
d_3 = {Disease}$\rightsquigarrow$ Job
d_4 = {Insurance, Premium}$\rightsquigarrow$ Job

Fragmentation and inference – Example

R(SSN, Birth, ZIP, Name, Treatment, Disease, Job, Premium, Insurance)



Constraints

c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

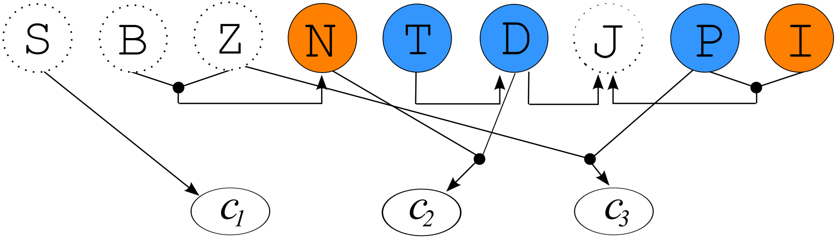
Dependencies

d_1 = {Birth, ZIP}$\rightsquigarrow$ Name
d_2 = {Treatment}$\rightsquigarrow$ Disease
d_3 = {Disease}$\rightsquigarrow$ Job
d_4 = {Insurance, Premium}$\rightsquigarrow$ Job

Fragmenting with data dependencies

Take into account data dependencies in fragmentation

- Fragments should not contain sensitive attributes/associations neither directly nor indirectly



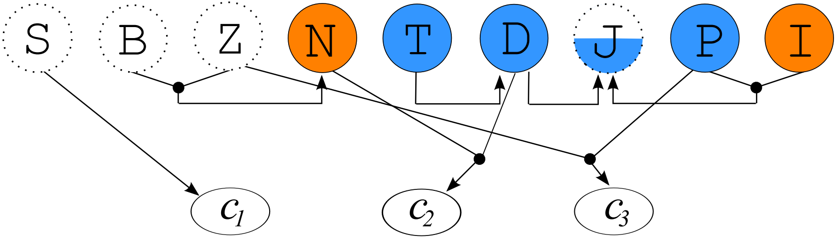
Constraints c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Dependencies d_1 = {Birth, ZIP}$\rightsquigarrow$ Name
d_2 = {Treatment}$\rightsquigarrow$ Disease
d_3 = {Disease}$\rightsquigarrow$ Job
d_4 = {Insurance, Premium}$\rightsquigarrow$ Job

Fragmenting with data dependencies

Take into account data dependencies in fragmentation

- Fragments should not contain sensitive attributes/associations neither directly nor indirectly



Constraints

c_1 = {SSN}
c_2 = {Name, Disease}
c_3 = {ZIP, Premium}

Dependencies

d_1 = {Birth, ZIP}$\leadsto$ Name
d_2 = {Treatment}$\leadsto$ Disease
d_3 = {Disease}$\leadsto$ Job
d_4 = {Insurance, Premium}$\leadsto$ Job

Combining Indexes, Selective Encryption, and Fragmentation

Exposure of confidential information

- Indexes, fragmentation, and selective encryption are all solutions providing the required security and privacy guarantees but...

- ...What happens when such solutions are combined?

Exposure of confidential information

- Indexes, fragmentation, and selective encryption are all solutions providing the required security and privacy guarantees but...

- ...What happens when such solutions are combined?

$\Longrightarrow$ They may open the door to inferences by users

Exposure of confidential information

- Indexes, fragmentation, and selective encryption are all solutions providing the required security and privacy guarantees but...

- ...What happens when such solutions are combined?

$\Longrightarrow$ They may open the door to inferences by users

  - Indexes and selective encryption

  - Indexes and fragmentation

Access Control and Indexes

S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Private Data Indexes for Selective Access to Outsourced Data,” in *Proc. of the 10th Workshop on Privacy in the Electronic Society (WPES 2011)*, Chicago, IL, USA, October 2011.

Access control and indexes

- Selective encryption for access control combined with indexes for query execution
  - + provide effectiveness and efficiency in query execution
  - + provide different data views to different users
  - – can open the door to inferences by users

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 – Example

- With direct indexes, plaintext values are always represented by the same index value and viceversa

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

Exposure risk – Example

- With direct indexes, 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 – Example

- With direct indexes, 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)

Intuitive approach: User-based index – 1

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

Intuitive approach: User-based index – 1

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

	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

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 – 1

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

	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

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 – 1

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

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 – 1

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

	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	▨	▨	**2010**	▨
t_2	002	Rome	**2010**	**700**
t_3	003	Rome	2011	600
t_4	▨	▨	▨	**700**
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
if $t_i[\mathtt{A}]$=$t_j[\mathtt{A}]$ and *acl*(t_i), *acl*(t_j) are different but overlapping

Intuitive approach: User-based index – 2

- For tuples t_i and t_j such that t_i[A]=t_j[A] and their acls are different but overlapping
  - the index values for t_i[A] and t_j[A] of all users in *acl*(t_i)∩*acl*(t_j) must be different
  - use a random salt to differentiate index values

	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)	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)

Variations/open issues

- Protect against the server observing multiple queries
- Protect against collusion between users and server
- Use of indexes associated with clusters of tuples in contrast to individual tuples

Indexes and Fragmentation

S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “On Information Leakage by Indexes over Data Fragments,” in *Proc. of PrivDB*, Brisbane, Australia, April 2013.

Information exposure

+ Provides effectiveness and efficiency in query execution
    ◦ enables the partial server-side evaluation of selection conditions over encrypted attributes

– Indexes combined with fragmentation can cause information leakage of confidential (encrypted or fragmented) information
    ◦ exposure to leakage varies depending on the kind of indexes

Kinds of knowledge

A curious observer can exploit

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t^e_{11}	Adams	VA	α
s_{12}	t^e_{12}	Brown	MN	α
s_{13}	t^e_{13}	Cooper	CA	α
s_{14}	t^e_{14}	Davis	VA	β
s_{15}	t^e_{15}	Eden	NY	β
s_{16}	t^e_{16}	Falk	CA	γ
s_{17}	t^e_{17}	Green	NY	δ
s_{18}	t^e_{18}	Hack	NY	δ

F_2^e

salt	enc	Disease
s_{21}	t^e_{21}	Flu
s_{22}	t^e_{22}	Flu
s_{23}	t^e_{23}	Flu
s_{24}	t^e_{24}	Diabetes
s_{25}	t^e_{25}	Diabetes
s_{26}	t^e_{26}	Gastritis
s_{27}	t^e_{27}	Arthritis
s_{28}	t^e_{28}	Arthritis

Kinds of knowledge

A curious observer can exploit

- vertical knowledge due to values appearing in the clear in one fragment and indexed in other fragments

F_1^e

salt	enc	Name	State	i_d
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

Kinds of knowledge

A curious observer can exploit

- vertical knowledge due to values appearing in the clear in one fragment and indexed in other fragments
- horizontal knowledge due to external knowledge of the presence of specific tuples in the table

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Direct index

F_1^e

<u>salt</u>	enc	Name	State	$\mathrm{i_d}$
s_{11}	t^e_{11}	Adams	VA	α
s_{12}	t^e_{12}	Brown	MN	α
s_{13}	t^e_{13}	Cooper	CA	α
s_{14}	t^e_{14}	Davis	VA	β
s_{15}	t^e_{15}	Eden	NY	β
s_{16}	t^e_{16}	Falk	CA	γ
s_{17}	t^e_{17}	Green	NY	δ
s_{18}	t^e_{18}	Hack	NY	δ

vertical knowledge

<u>salt</u>	enc	Disease
s_{21}	t^e_{21}	Flu
s_{22}	t^e_{22}	Flu
s_{23}	t^e_{23}	Flu
s_{24}	t^e_{24}	Diabetes
s_{25}	t^e_{25}	Diabetes
s_{26}	t^e_{26}	Gastritis
s_{27}	t^e_{27}	Arthritis
s_{28}	t^e_{28}	Arthritis

horizontal

Name	Disease
Adams	Flu

Direct index

F_1^e

salt	enc	Name	State	i_d
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

Direct index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

- $\iota(\text{Flu}) = \alpha$
- $\iota(\text{Gastritis}) = \gamma$

Direct index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

- ι(Flu) = $\alpha \Longrightarrow$ Adams, Brown, Cooper have Flu
- ι(Gastritis) = $\gamma \Longrightarrow$ Falk has Gastritis
- the other patients have Diabetes or Arthritis with $p = 50\%$

Direct index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

Direct index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

- $\iota(\text{Flu}) = \alpha$

Direct index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	α
s_{12}	t_{12}^e	Brown	MN	α
s_{13}	t_{13}^e	Cooper	CA	α
s_{14}	t_{14}^e	Davis	VA	β
s_{15}	t_{15}^e	Eden	NY	β
s_{16}	t_{16}^e	Falk	CA	γ
s_{17}	t_{17}^e	Green	NY	δ
s_{18}	t_{18}^e	Hack	NY	δ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

- ι(Flu) = $\alpha \implies$ also Brown and Cooper have Flu

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

Bucket index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

- $\iota(\text{Flu}) = \zeta$

Bucket index

F_1^e

salt	enc	Name	State	i_d
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

- $\iota(\text{Flu}) = \zeta \implies \iota(\text{Gastritis}) = \zeta$

Bucket index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

- ι(Flu) = ι(Gastritis) = ζ $\Longrightarrow$ Adams, Brown, Cooper, and Falk have Flu with $p = 75\%$, Gastritis with $p = 25\%$

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

Bucket index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

- $\iota(\text{Flu}) = \zeta$

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

- $\iota(\text{Flu}) = \zeta \Longrightarrow$ no inference

Bucket index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical and Horizontal knowledge

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t^e_{11}	Adams	VA	ζ
s_{12}	t^e_{12}	Brown	MN	ζ
s_{13}	t^e_{13}	Cooper	CA	ζ
s_{14}	t^e_{14}	Davis	VA	η
s_{15}	t^e_{15}	Eden	NY	η
s_{16}	t^e_{16}	Falk	CA	ζ
s_{17}	t^e_{17}	Green	NY	θ
s_{18}	t^e_{18}	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t^e_{21}	Flu
s_{22}	t^e_{22}	Flu
s_{23}	t^e_{23}	Flu
s_{24}	t^e_{24}	Diabetes
s_{25}	t^e_{25}	Diabetes
s_{26}	t^e_{26}	Gastritis
s_{27}	t^e_{27}	Arthritis
s_{28}	t^e_{28}	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical and Horizontal knowledge

- ι(Flu) = ι(Gastritis) = ζ

Bucket index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical and Horizontal knowledge

- ι(Flu) = ι(Gastritis) = $\zeta \Longrightarrow$ Brown, Cooper, and Falk have Flu with $p = 66\%$, Gastritis with $p = 33\%$

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t^e_{11}	Adams	VA	ζ
s_{12}	t^e_{12}	Brown	MN	ζ
s_{13}	t^e_{13}	Cooper	CA	ζ
s_{14}	t^e_{14}	Davis	VA	η
s_{15}	t^e_{15}	Eden	NY	η
s_{16}	t^e_{16}	Falk	CA	ζ
s_{17}	t^e_{17}	Green	NY	θ
s_{18}	t^e_{18}	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t^e_{21}	Flu
s_{22}	t^e_{22}	Flu
s_{23}	t^e_{23}	Flu
s_{24}	t^e_{24}	Diabetes
s_{25}	t^e_{25}	Diabetes
s_{26}	t^e_{26}	Gastritis
s_{27}	t^e_{27}	Arthritis
s_{28}	t^e_{28}	Arthritis

horizontal

Name	Disease
Davis	Diabetes

Vertical and Horizontal knowledge

Bucket index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Davis	Diabetes

Vertical and Horizontal knowledge

- $\iota(\text{Diabetes}) = \eta$

Bucket index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ζ
s_{12}	t_{12}^e	Brown	MN	ζ
s_{13}	t_{13}^e	Cooper	CA	ζ
s_{14}	t_{14}^e	Davis	VA	η
s_{15}	t_{15}^e	Eden	NY	η
s_{16}	t_{16}^e	Falk	CA	ζ
s_{17}	t_{17}^e	Green	NY	θ
s_{18}	t_{18}^e	Hack	NY	θ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Davis	Diabetes

Vertical and Horizontal knowledge

- ι(Diabetes) = $\eta \Longrightarrow$ Eden has Diabetes

Flattened index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	κ
s_{12}	t_{12}^e	Brown	MN	λ
s_{13}	t_{13}^e	Cooper	CA	μ
s_{14}	t_{14}^e	Davis	VA	ν
s_{15}	t_{15}^e	Eden	NY	ξ
s_{16}	t_{16}^e	Falk	CA	π
s_{17}	t_{17}^e	Green	NY	ρ
s_{18}	t_{18}^e	Hack	NY	σ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Flattened index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	κ
s_{12}	t_{12}^e	Brown	MN	λ
s_{13}	t_{13}^e	Cooper	CA	μ
s_{14}	t_{14}^e	Davis	VA	ν
s_{15}	t_{15}^e	Eden	NY	ξ
s_{16}	t_{16}^e	Falk	CA	π
s_{17}	t_{17}^e	Green	NY	ρ
s_{18}	t_{18}^e	Hack	NY	σ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

Flattened index

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t^e_{11}	Adams	VA	κ
s_{12}	t^e_{12}	Brown	MN	λ
s_{13}	t^e_{13}	Cooper	CA	μ
s_{14}	t^e_{14}	Davis	VA	ν
s_{15}	t^e_{15}	Eden	NY	ξ
s_{16}	t^e_{16}	Falk	CA	π
s_{17}	t^e_{17}	Green	NY	ρ
s_{18}	t^e_{18}	Hack	NY	σ

vertical knowledge

salt	enc	Disease
s_{21}	t^e_{21}	Flu
s_{22}	t^e_{22}	Flu
s_{23}	t^e_{23}	Flu
s_{24}	t^e_{24}	Diabetes
s_{25}	t^e_{25}	Diabetes
s_{26}	t^e_{26}	Gastritis
s_{27}	t^e_{27}	Arthritis
s_{28}	t^e_{28}	Arthritis

horizontal

Name	Disease
Adams	Flu

Vertical knowledge

- each correspondence between plaintext and index values is equally likely

Flattened index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	κ
s_{12}	t_{12}^e	Brown	MN	λ
s_{13}	t_{13}^e	Cooper	CA	μ
s_{14}	t_{14}^e	Davis	VA	ν
s_{15}	t_{15}^e	Eden	NY	ξ
s_{16}	t_{16}^e	Falk	CA	π
s_{17}	t_{17}^e	Green	NY	ρ
s_{18}	t_{18}^e	Hack	NY	σ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

Flattened index

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	κ
s_{12}	t_{12}^e	Brown	MN	λ
s_{13}	t_{13}^e	Cooper	CA	μ
s_{14}	t_{14}^e	Davis	VA	ν
s_{15}	t_{15}^e	Eden	NY	ξ
s_{16}	t_{16}^e	Falk	CA	π
s_{17}	t_{17}^e	Green	NY	ρ
s_{18}	t_{18}^e	Hack	NY	σ

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

Horizontal knowledge

- $\iota(\text{Flu}) = \kappa$

Flattened index

F_1^e

salt	**enc**	**Name**	**State**	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	κ
s_{12}	t_{12}^e	Brown	MN	λ
s_{13}	t_{13}^e	Cooper	CA	μ
s_{14}	t_{14}^e	Davis	VA	ν
s_{15}	t_{15}^e	Eden	NY	ξ
s_{16}	t_{16}^e	Falk	CA	π
s_{17}	t_{17}^e	Green	NY	ρ
s_{18}	t_{18}^e	Hack	NY	σ

vertical knowledge

salt	**enc**	**Disease**
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	**Disease**
Adams	Flu

Horizontal knowledge

- ι(Flu) = $\kappa \implies$ no inference

Intuitive approach: flattening and collisions

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ϕ
s_{12}	t_{12}^e	Brown	MN	ϕ
s_{13}	t_{13}^e	Cooper	CA	ψ
s_{14}	t_{14}^e	Davis	VA	χ
s_{15}	t_{15}^e	Eden	NY	χ
s_{16}	t_{16}^e	Falk	CA	ψ
s_{17}	t_{17}^e	Green	NY	ω
s_{18}	t_{18}^e	Hack	NY	ω

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

+ blocks inference exposure

Intuitive approach: flattening and collisions

F_1^e

salt	enc	Name	State	$\mathrm{i_d}$
s_{11}	t_{11}^e	Adams	VA	ϕ
s_{12}	t_{12}^e	Brown	MN	ϕ
s_{13}	t_{13}^e	Cooper	CA	ψ
s_{14}	t_{14}^e	Davis	VA	χ
s_{15}	t_{15}^e	Eden	NY	χ
s_{16}	t_{16}^e	Falk	CA	ψ
s_{17}	t_{17}^e	Green	NY	ω
s_{18}	t_{18}^e	Hack	NY	ω

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

+ blocks inference exposure

– exposed to inferences exploiting dynamic observations

Intuitive approach: flattening and collisions

F_1^e

salt	enc	Name	State	$\mathbf{i_d}$
s_{11}	t_{11}^e	Adams	VA	ϕ
s_{12}	t_{12}^e	Brown	MN	ϕ
s_{13}	t_{13}^e	Cooper	CA	ψ
s_{14}	t_{14}^e	Davis	VA	χ
s_{15}	t_{15}^e	Eden	NY	χ
s_{16}	t_{16}^e	Falk	CA	ψ
s_{17}	t_{17}^e	Green	NY	ω
s_{18}	t_{18}^e	Hack	NY	ω

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

+ blocks inference exposure

– exposed to inferences exploiting dynamic observations

Disease=‘Flu’ translates to i_d IN $\{\phi,\psi\} \Longrightarrow \iota(\text{Flu})=\{\phi,\psi\}$

Intuitive approach: flattening and collisions

F_1^e

salt	enc	Name	State	i_d
s_{11}	t_{11}^e	Adams	VA	ϕ
s_{12}	t_{12}^e	Brown	MN	ϕ
s_{13}	t_{13}^e	Cooper	CA	ψ
s_{14}	t_{14}^e	Davis	VA	χ
s_{15}	t_{15}^e	Eden	NY	χ
s_{16}	t_{16}^e	Falk	CA	ψ
s_{17}	t_{17}^e	Green	NY	ω
s_{18}	t_{18}^e	Hack	NY	ω

vertical knowledge

salt	enc	Disease
s_{21}	t_{21}^e	Flu
s_{22}	t_{22}^e	Flu
s_{23}	t_{23}^e	Flu
s_{24}	t_{24}^e	Diabetes
s_{25}	t_{25}^e	Diabetes
s_{26}	t_{26}^e	Gastritis
s_{27}	t_{27}^e	Arthritis
s_{28}	t_{28}^e	Arthritis

horizontal

Name	Disease
Adams	Flu

+ blocks inference exposure

– exposed to inferences exploiting dynamic observations

Disease=‘Flu’ translates to i_d IN $\{\phi,\psi\} \Longrightarrow \iota(Flu)=\{\phi,\psi\}$

$\iota($Flu$)=\{\phi,\psi\} \Longrightarrow$ Brown, Cooper, Frank have Flu with $p = 66\%$

Still several open issues

- Protection against observation of accesses to fragments
- Protection against the release of multiple indexes
  - multiple indexes in the same fragment
  - indexes on the same attribute in multiple fragments
  - two attributes appear one in plaintext and the other indexed in one fragment and reversed in another fragment
- Protection against different types of observer's knowledge
- Development of flattened index functions that generate collisions
- Definition of metrics for assessing exposures due to indexes
- . . .

References – 1

- [ABGGKMSTX-05] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, Y. Xu, “Two Can Keep a Secret: A Distributed Architecture for Secure Database Services,” in *Proc. of CIDR*, Asilomar, CA, USA, January 2005.
- [AFB-05] M. Atallah, K. Frikken, M. Blanton, “Dynamic and Efficient Key Management for Access Hierarchies,” in *Proc. of CCS*, Alexandria, VA, USA, November 2005.
- [AKSX-04] R. Agrawal, J. Kierman, R. Srikant, Y. Xu, “Order Preserving Encryption for Numeric Data,” in *Proc. of ACM SIGMOD*, Paris, France, 2004.
- [AT-83] S. Akl, P. Taylor, “Cryptographic Solution to a Problem of Access Control in a Hierarchy,” *ACM TOCS*, vol. 1, no. 3, August 1983.
- [B-70] B.H. Bloom, “Trade-offs in Hash Coding with Allowable Error,” in *Communication of the ACM*, vol. 13, no. 7, July 1970.
- [BV11] Z. Brakerski, V. Vaikuntanathan, “Efficient Fully Homomorphic Encryption from (standard) LWE,” in Proc. of FOCS, Palm Springs, CA, USA, October 2011.

References – 2

- [CDDJPS-05] A. Ceselli, E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Paraboschi, P. Samarati, “Modeling and Assessing Inference Exposure in Encrypted Databases,” in *ACM TISSEC*, vol. 8, no. 1, February 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 ESORICS*, Dresden, Germany, September 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 ICDCS*, Montreal, Quebec, Canada, June 2009.
- [CDFJPS-09b] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Keep a Few: Outsourcing Data while Maintaining Confidentiality,” in *Proc. of ESORICS*, Saint Malo, France, September 2009.
- [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 TISSEC*, vol. 13, no. 3, July 2010.
- [CMW-06] J. Crampton, K. Martin, P. Wild, “On Key Assignment for Hierarchical Access Control,” in *Proc. of CSFW*, Venice, Italy, July 2006.

References – 3

- [CSYZ-08] G. Cormode, D. Srivastava, T. YU, Q. Zhang, “Anonymizing Bipartite Graph Data Using Safe Groupings,” in *Proc. of VLDB*, Auckland, New Zealand, August 2008.
- [DFJL-12] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, “Enforcing Subscription-based Authorization Policies in Cloud Scenarios,” in *Proc. of DBSec*, Paris, France, July 2012.
- [DFJLPS-14] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, P. Samarati, “Fragmentation in Presence of Data Dependencies,” in *IEEE TDSC*, 2014 (to appear).
- [DFJPPS-10] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, G. Pelosi, P. Samarati, “Encryption-based Policy Enforcement for Cloud Storage,” in *Proc. of SPCC 2010,* Genova, Italy, June 2010.
- [DFJPS-07] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Over-encryption: Management of Access Control Evolution on Outsourced Data,” in *Proc. of VLDB*, Vienna, Austria, 2007.
- [DFJPS-10a] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Fragments and Loose Associations: Respecting Privacy in Data Publishing,” in *Proc. of the VLDB Endowment*, vol. 3, no. 1, September 2010.

References – 4

- [DFJPS-10b] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Encryption Policies for Regulating Access to Outsourced Data,” in *ACM TODS*, vol. 35, no. 2, April 2010.
- [DFJPS-11] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Private Data Indexes for Selective Access to Outsourced Data,” in *Proc. of WPES*, Chicago, IL, USA, October 2011.
- [DFJPS-12] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Support for Write Privileges on Outsourced Data,” in *Proc. of SEC*, Heraklion, Crete, Greece, June 2012.
- [DFJPS-13] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “On Information Leakage by Indexes over Data Fragments,” in *Proc. of PrivDB*, Brisbane, Australia, April 2013.
- [DFM-04] A. De Santis, A.L. Ferrara, B. Masucci, “Cryptographic Key Assignment Schemes for any Access Control Policy,” in *Inf. Process. Lett.*, vol. 92, no. 4, 2004.
- [G-09] C. Gentry, “Fully Homomorphic Encryption using Ideal Lattices,” in *Proc. of STOC*, Bethesda, MA, USA, 2009.

References – 5

- [G-80] E. Gudes, “The Design of a Cryptography based Secure File System,” in *IEEE TSE*, vol. 6, no. 5, September 1980.
- [GKPVZ-13] S. Goldwasser, Y.T. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich, “Reusable Garbled Circuits and Succinct Functional Encryption,” in *Proc. of STOC*, Palo Alto, CA, USA, June 2013.
- [GSW13] C. Gentry, A. Sahai, B. Waters, “Homomorphic Encryption from Learning with Errors: Conceptually-simpler, Asymptotically-faster, Attribute-based,” in Proc. of CRYPTO, Santa Barbara, CA, USA, August 2013.
- [HIML-02] H. Hacigümüş, B. Iyer, S. Mehrotra, C. Li, “Executing SQL over Encrypted Data in the Database-Service-Provider Model,” in *Proc. of the ACM SIGMOD*, Madison, Wisconsin, USA, June 2002.
- [HL-90] L. Harn, H. Lin, “A Cryptographic Key Generation Scheme for Multilevel Data Security,” *Computers and Security*, vol. 9, no. 6, October 1990.
- [HY-03] M. Hwang, W. Yang, “Controlling Access in Large Partially Ordered Hierarchies using Cryptographic Keys,” *The Journal of Systems and Software*, vol. 67, no. 2, August 2003.

References – 6

- [LWL-89] H. Liaw, S. Wang, C. Lei, “On the Design of a Single-Key-Lock Mechanism Based on Newton’s Interpolating Polynomial,” in *IEEE TSE*, vol. 15, no. 9, September 1989.
- [M-85] S. MacKinnon et al., “An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy,” in *IEEE TC,* vol. 34, no. 9, September 1985.
- [MNT-06] E. Mykletun, M. Narasimha, G. Tsudik, “Authentication and Integrity in Outsourced Databases,” in *ACM TS*, vol. 2, no. 2, May 2006.
- [S-87] R. Sandhu, “On Some Cryptographic Solutions for Access Control in a Tree Hierarchy,” in *Proc. of the 1987 Fall Joint Computer Conference on Exploring Technology: Today and Tomorrow*, Dallas, TX, USA, October 1987.
- [S-88] R. Sandhu, “Cryptographic Implementation of a Tree Hierarchy for Access Control,” in *Information Processing Letters*, vol. 27, no. 2, 1988.
- [SC-02] V. Shen, T. Chen, “A Novel Key Management Scheme Based on Discrete Logarithms and Polynomial Interpolations,” in *Computers and Security*, vol. 21, no. 2, March 2002.

References – 7

- [WL-06] H. Wang, Laks V. S. Lakshmanan, “Efficient Secure Query Evaluation over Encrypted XML Databases,” in *Proc. of VLDB*, Seoul, Korea, September 2006.

Privacy and Data Protection in Emerging Scenarios

Security, Privacy, and Data Protection Laboratory
Dipartimento di Informatica
Università degli Studi di Milano

Access and pattern confidentiality

Guaranteeing privacy of outsourced data entails protecting the confidentiality of the data (content confidentiality) as well as of the accesses to them

- Access confidentiality: confidentiality of the fact that an access aims at a specific data

- Pattern confidentiality: confidentiality of the fact that two accesses aim at the same data

Approaches for protecting data accesses

- Private Information Retrieval (PIR) proposals (e.g., [CKGS-98, SC-07])
- Oblivious traversal of tree-structured data/indexes [LC-04]
- Pyramid-shaped database layout of Oblivious RAM [WSC-08, WS-12]]
- Path ORAM protocol, working on a tree structure [SVSFRYD-13]
- Ring ORAM, variation of Path ORAM with better performance and same protection guarantees [RFKSSvD-15]
- Shuffle index based on the definition of a B+-tree structure with dynamic allocation of data [DFPPS-11a, DFPPS-11b, DFPPS-13]

Path ORAM

Server side

- Tree structure with L levels ($L = \lceil log_2(N) - 1 \rceil$, with N the number of blocks)
- Each node in the tree is a bucket that contains up to Z real blocks (padded with dummy blocks)
- Any leaf node x defines a unique path $P(x)$ from x to the root

Client side

- The client locally stores a small number of blocks in a stash
- The client stores a position map: x = position$[a]$ means that a block identified by a is currently mapped to the x-th leaf node
  $\Longrightarrow$ block a (if it exists) resides in some bucket in path $P(x)$ or in the stash
- The position map changes every time blocks are accessed and remapped

Path ORAM – Main invariant

At any time:

- each block is mapped to a uniformly random leaf bucket in the tree

- unstashed blocks are always placed in some bucket along the path to the mapped leaf

Path ORAM reads and writes

1. Remap block: Let x be the old position of a. Randomly remap the position of a to a new random position (a new leaf node)

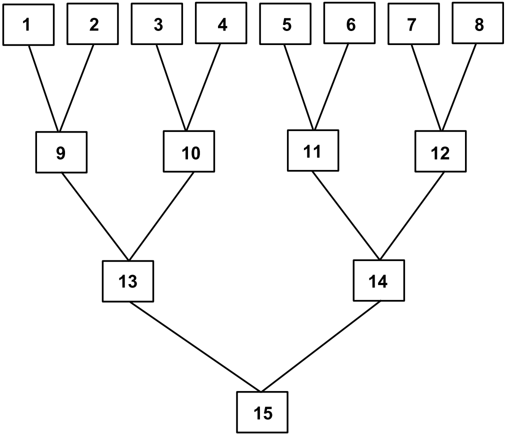
2. Read path: read nodes in $P(x)$ containing a.
   If the access is a write, update the data stored for block a

3. Write path: write the nodes in $P(x)$ back possibly including some additional blocks from the stash if they can be placed into the path (i.e., the main invariant is satisfied)

Path ORAM – Example

Client

Server



stash

a	b				

position[a] = 4

position[b] = 5

position[c] = 7

...

Path ORAM – Example

Client

Read access to block: c

stash

a	b				

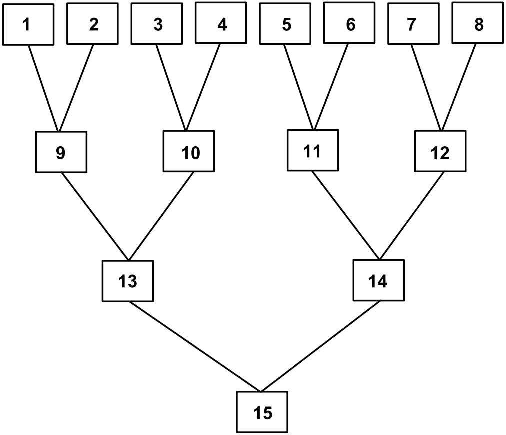
position[a] = 4

position[b] = 5

position[c] = 7

...

Server



Path ORAM – Example

Client

Read access to block: c

1. *x* := position[c] = 7

   position[c] := `Random`(1,...,8) = 8

stash

a	b				

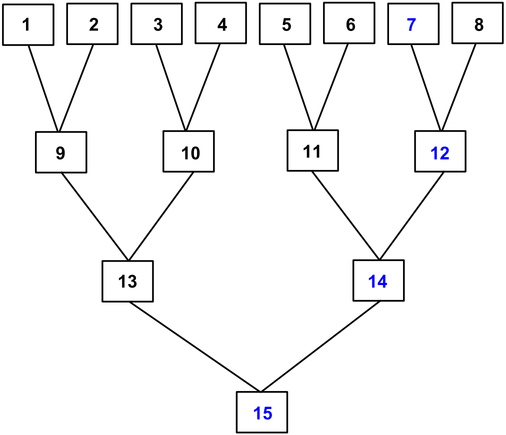
position[a] = 4

position[b] = 5

position[c] = 8

...

Server



Path ORAM – Example

Client

Read access to block: c

1. x := position[c] = 7

   position[c] := `Random`(1,...,8) = 8

2. Read path $P(7)$

stash

a	b	c	■	■	■

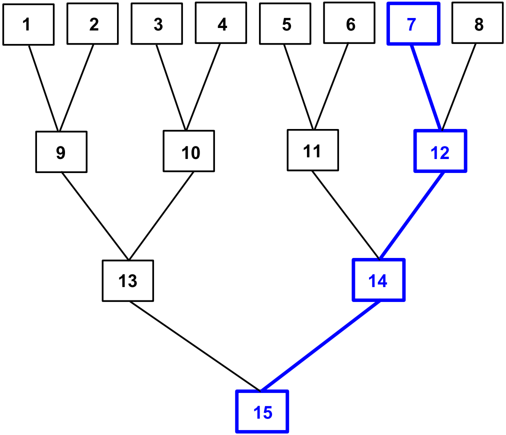
position[a] = 4

position[b] = 5

position[c] = 8

...

Server



Path ORAM – Example

Client

Read access to block: c

1. x := position[c] = 7

   position[c] := `Random`(1,...,8) = 8

2. Read path P(7)

3. Write back nodes in P(7); move nodes whose path intersects P(7) to the highest intersecting node
   (a not written back - the highest
    intersecting node is full
    b written in node 14
    c written in node 12)

stash

a					

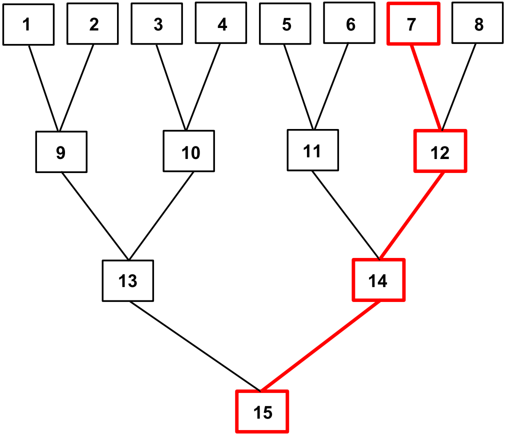
position[a] = 4

position[b] = 5

position[c] = 8

...

Server



Ring ORAM

- Variation of Path ORAM that reduces the online access bandwidth to $O(1)$ and the overall bandwidth to $\sim 2-2.5\log(N)$
- Same server-side structure as Path ORAM but each node has
  - S additional dummy blocks
  - a small map of the offsets of its blocks
  - a counter of accesses
- Protocol
  - Remap (step 1) is the same as Path ORAM
  - Read path (step 2) is revised to download only one block per bucket
  - Write path (step 3) is factorized among multiple access operations (eviction phase)

Path ORAM and Ring ORAM: Pros and cons

Path ORAM and Ring ORAM provide access and pattern confidentiality

- + same protection guarantees as ORAM (no inferences)
- + much more efficient than ORAM $\Longrightarrow$ more applicable in practice
- + limited access time
- – range queries are not supported
- – accesses by multiple clients are not supported
- – vulnerable to failures of the client
- – $\sim 2 - 2.5 \log(N)$ overall bandwidth overhead w.r.t. non protected accesses

Shuffle Index

S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, P. Samarati, “Efficient and Private Access to Outsourced Data,” in *Proc. of ICDCS,* Minneapolis, MN, USA, June 2011.

Shuffle index data structure

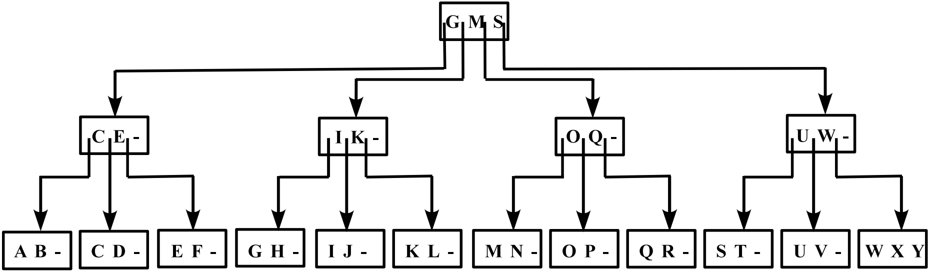
- Data are indexed over a candidate key K and organized as an unchained $B+$-tree with fan out F

- Data are stored in the leaves in association with their index values

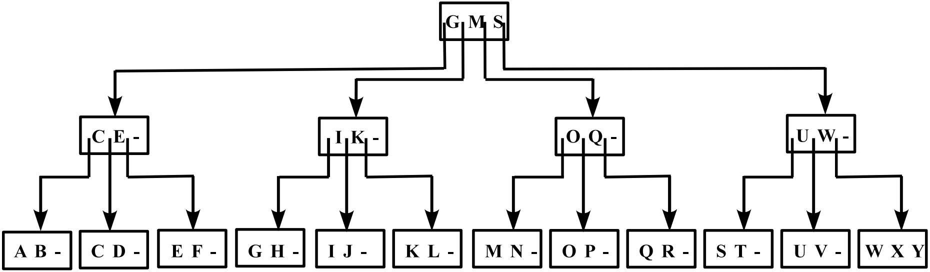
- Accesses to the data (searches) are based on the value of the index

- Node structure:
  - $q \geq \lceil F/2 \rceil$ children with $q-1$ values $v_1 \leq \ldots \leq v_{q-1}$
  - i-th child is the root of a subtree containing the values v with: $v < v_1$; $v_{i-1} \leq v < v_i$, $i = 2, \ldots, q-2$; $v \geq v_{q-1}$

Abstract representation of shuffle index – Example

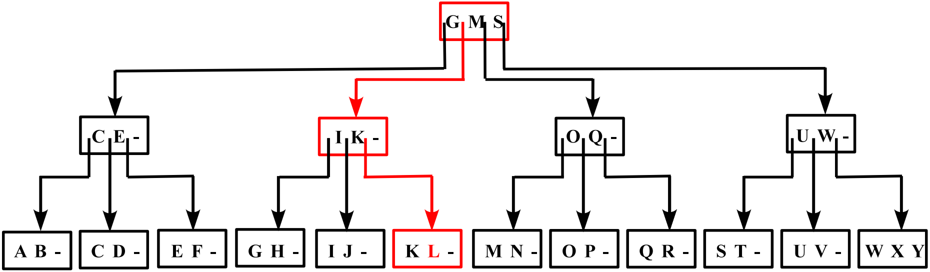


Abstract representation of shuffle index – Example



Search: L

Abstract representation of shuffle index – Example



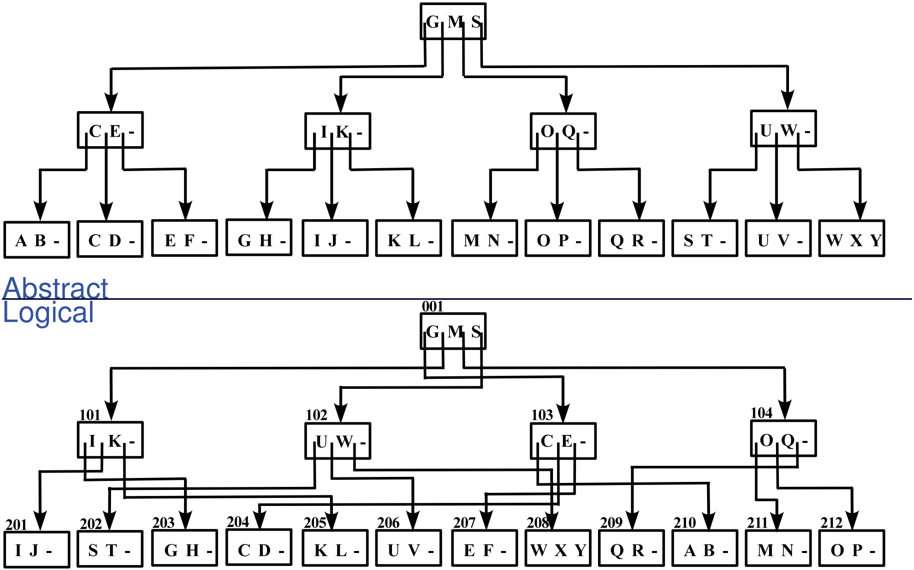
Search: L

Logical representation of shuffle index

- Pointers between nodes of the abstract data structure correspond, at logical level, to node identifiers

- Set of pairs $\langle$id, n$\rangle$, with id the node identifier and n the node content
  - the order between identifiers does not necessarily correspond to the order in which nodes appear in the abstract representation

Abstract and logical shuffle index – Example

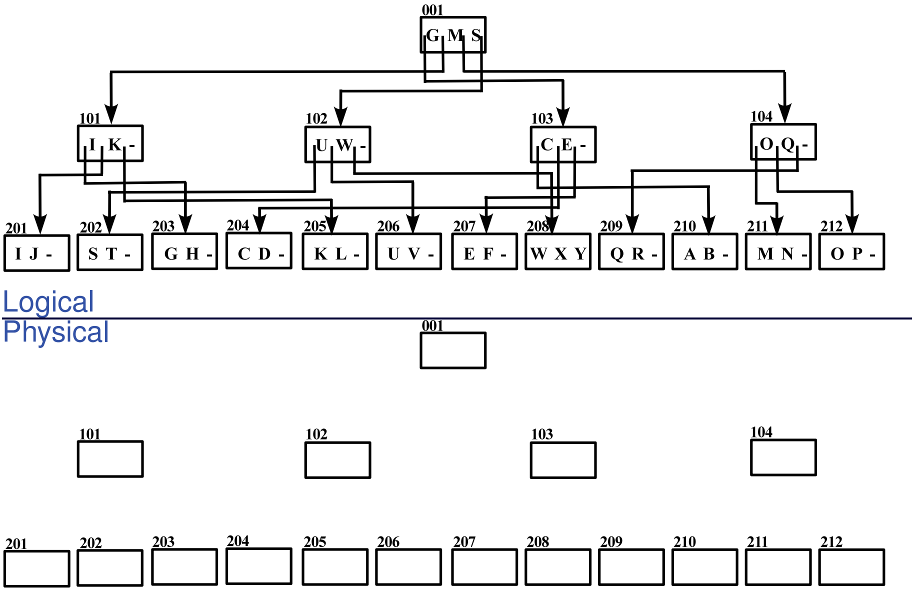


Physical representation of shuffle index

- Each node $\langle$id, n$\rangle$ of the logical shuffle index is stored on the server in encrypted form (content confidentiality)

- A node $\langle$id, n$\rangle$ corresponds to a block $\langle$id, b$\rangle$, with b$=\mathscr{C}||\mathscr{T}$, $\mathscr{C}=E_k(s||\text{n})$, $\mathscr{T}=MAC_k(\text{id}||\mathscr{C})$, s a value chosen at random during each encryption

Logical and physical shuffle index – Example

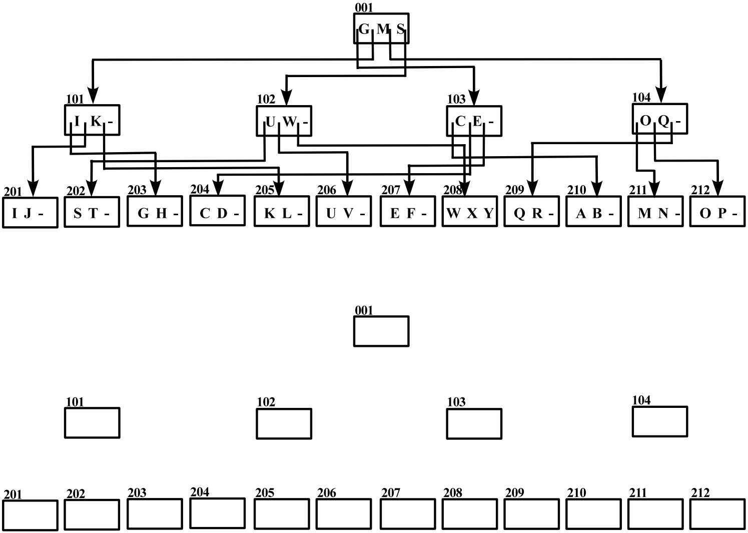


Data accesses

- Access to the data requires an iterative process between the client and the server
- The client performs an iteration for each level of the shuffle index starting from the root
- At each iteration, the client:
  - decrypts the retrieved block
  - determines the block to be retrieved from the server at the next level
- The process ends when a leaf block is retrieved

Data accesses – Example

Search: F

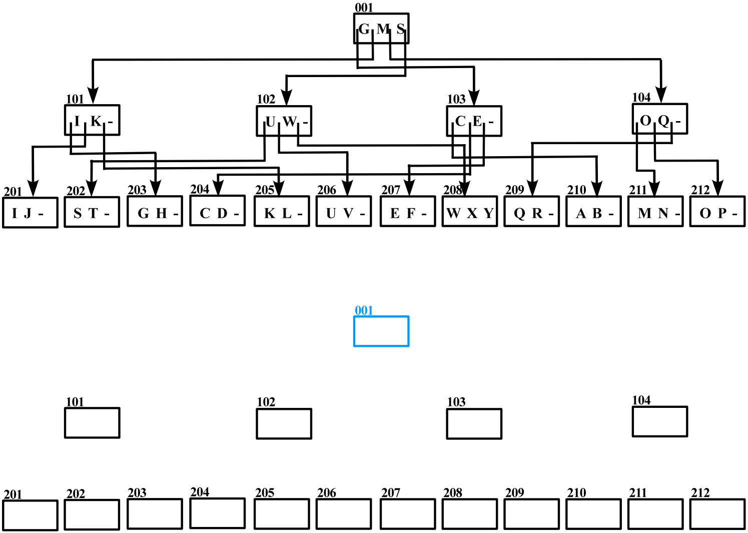


Data accesses – Example

Search: F

level: 0

download: 001



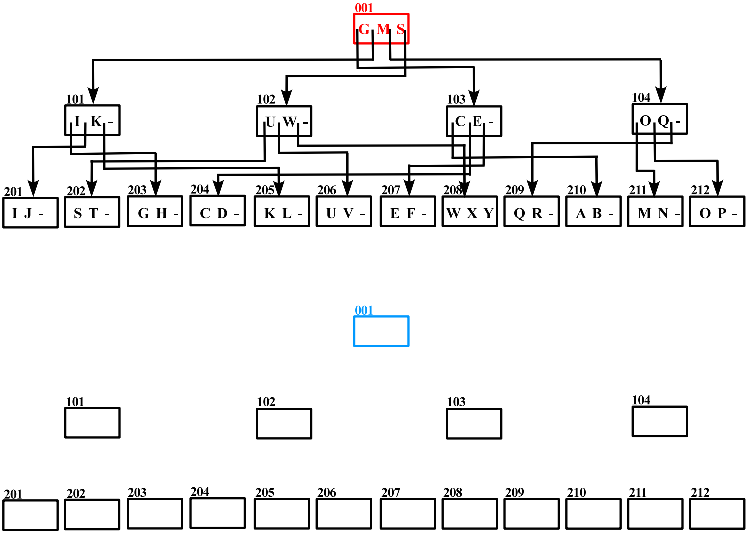
Data accesses – Example

Search: F

level: 0

download: 001

decrypt: 001



Data accesses – Example

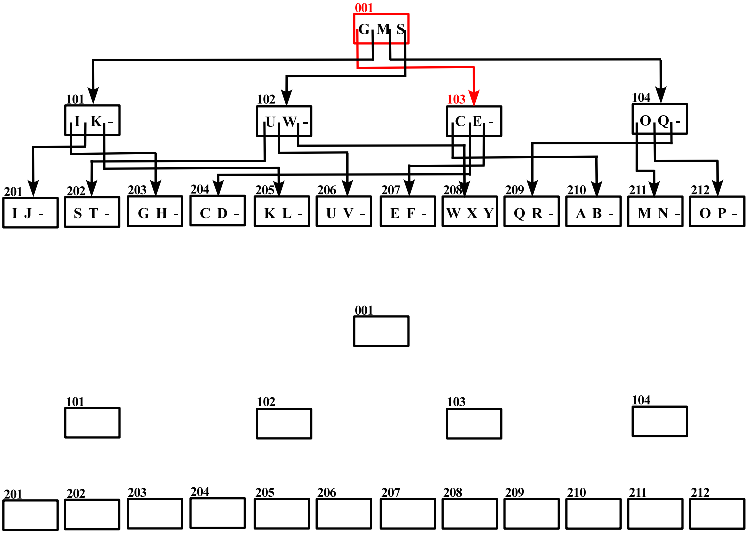
Search: F

level: 0

download: 001

decrypt: 001

level: 1



Data accesses – Example

Search: F

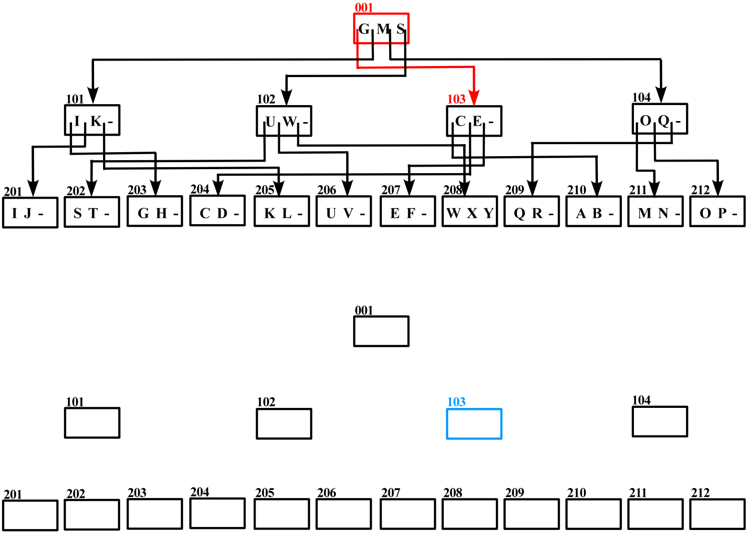
level: 0

download: 001

decrypt: 001

level: 1

download: 103



Data accesses – Example

Search: F

level: 0

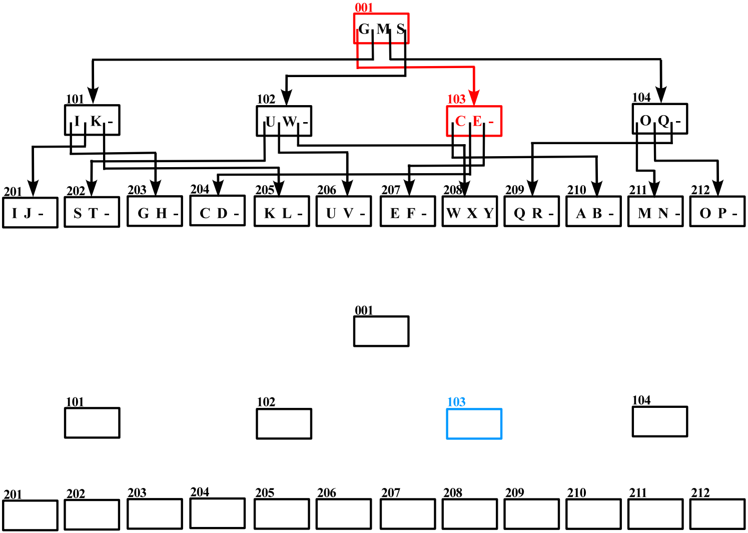
download: 001

decrypt: 001

level: 1

download: 103

decrypt: 103



Data accesses – Example

Search: F

level: 0

download: 001

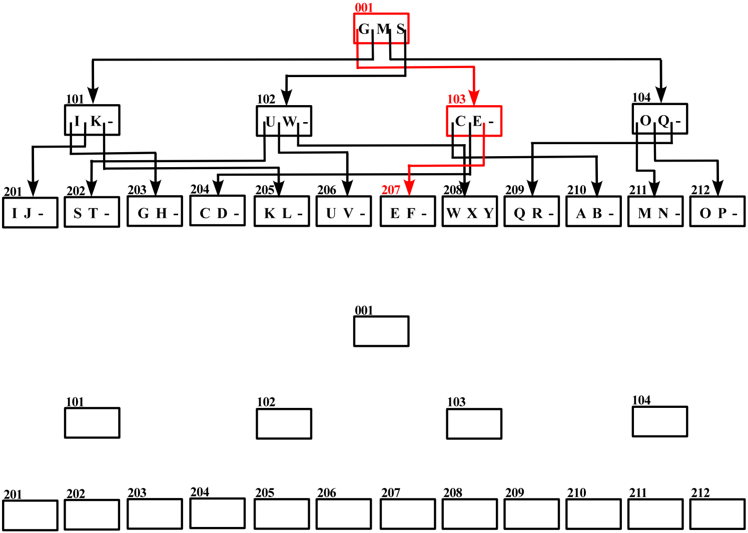
decrypt: 001

level: 1

download: 103

decrypt: 103

level: 2



Data accesses – Example

Search: F

level: 0

download: 001

decrypt: 001

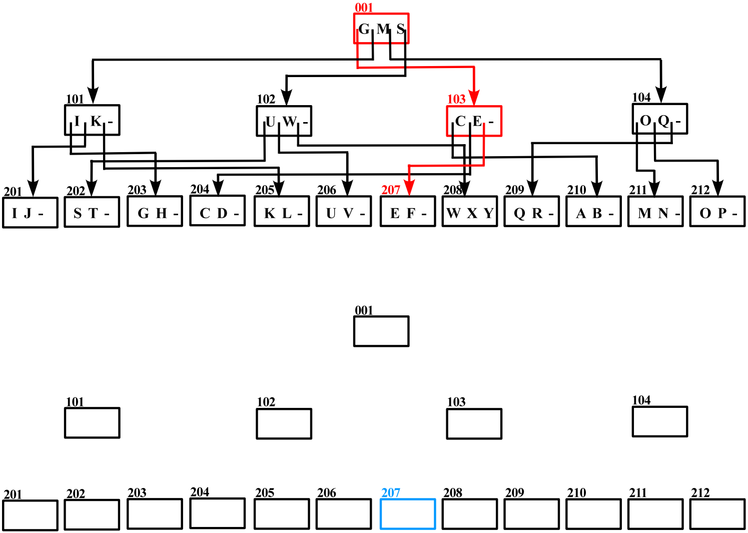
level: 1

download: 103

decrypt: 103

level: 2

download: 207



Data accesses – Example

Search: F

level: 0

download: 001

decrypt: 001

level: 1

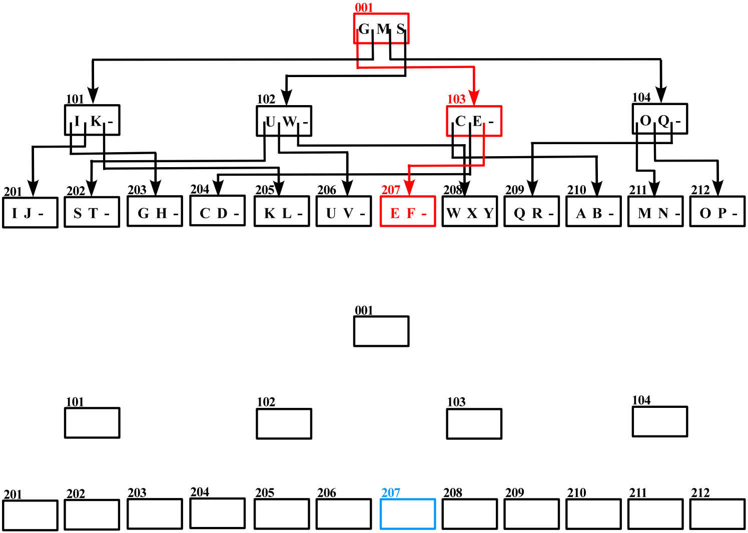
download: 103

decrypt: 103

level: 2

download: 207

decrypt: 207



Knowledge of the observer (server)

- The server receives a set of blocks to store
- The server receives requests to access the blocks that translate into observations
  - an observation o_i corresponds to a sequence of blocks $\{b_{i1}, \ldots, b_{ih}\}$
- The server knows or can easily infer:
  - the number m of blocks and their identifiers
  - the height h of the shuffle index
  - the level associated with each block (after the observation of a long history of accesses)

Problem statement

Given a sequence of observations $\{o_1, \ldots, o_z\}$ the server should not be able to infer:

- the data stored in the shuffle index (content confidentiality)
- the data to which access requests are aimed, that is, $\forall i = 1, \ldots, z$, the server should not infer that o_i aims at a specific node (access confidentiality)
- o_i aims at accessing the same node as o_j, $\forall i,j = 1, \ldots, z, i \neq j$ (pattern confidentiality)

Is encryption enough?

+ It protects:
  - content confidentiality of data at rest
  - access confidentiality of individual requests

– Access and pattern confidentiality is not provided
  - accesses to the same blocks imply accesses to the same data

⟹ frequency-based attacks allow the server to reconstruct the correspondence between plaintext values and blocks

Rationale of the approach

- Destroy the correspondence between the frequencies with which blocks are accessed and the frequencies of accesses to different values

- Combine three strategies:
  - cover searches
    - provide confusion in individual accesses
  - cached searches
    - allow protection of accesses to the same values
  - shuffling
    - dynamically changes node allocation to blocks at every access, so destroying the fixed node-block correspondence

Cover searches

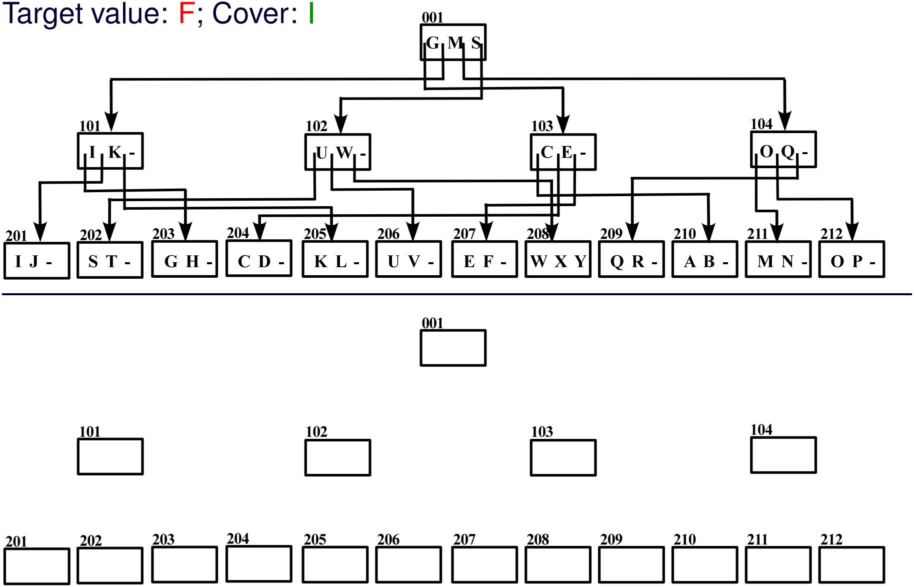
- Introduce confusion on the target of an access by hiding it within a group of other requests that act as covers

- The number of covers (num_cover) is a protection parameter

- Cover searches must:
  - provide block diversity (i.e., on a path disjoint from the target searched, apart from the root)
  - be indistinguishable from actual searches (i.e., enjoy a believable frequency of access)

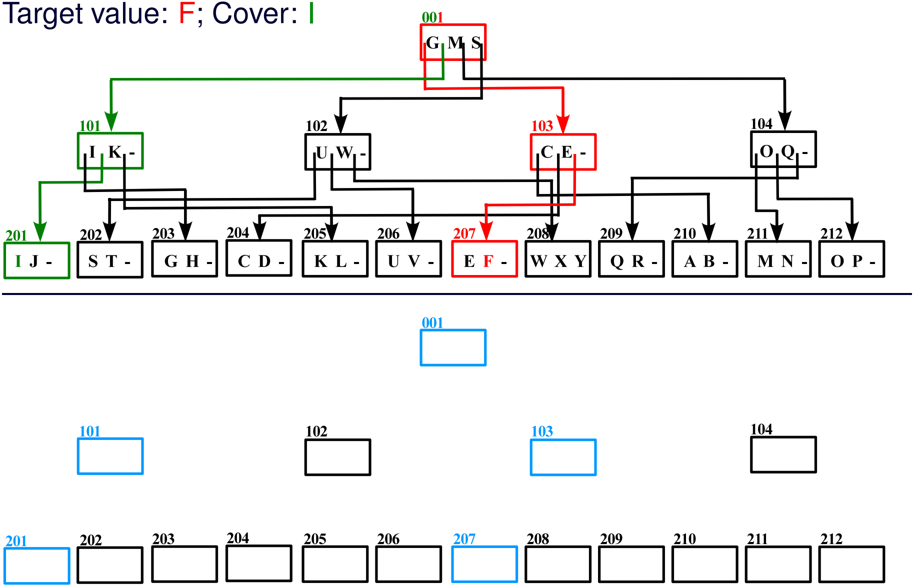
Cover searches – Example (1)

Target value: F; Cover: I



Cover searches – Example (1)

Target value: F; Cover: I



Cover searches – Protection offered

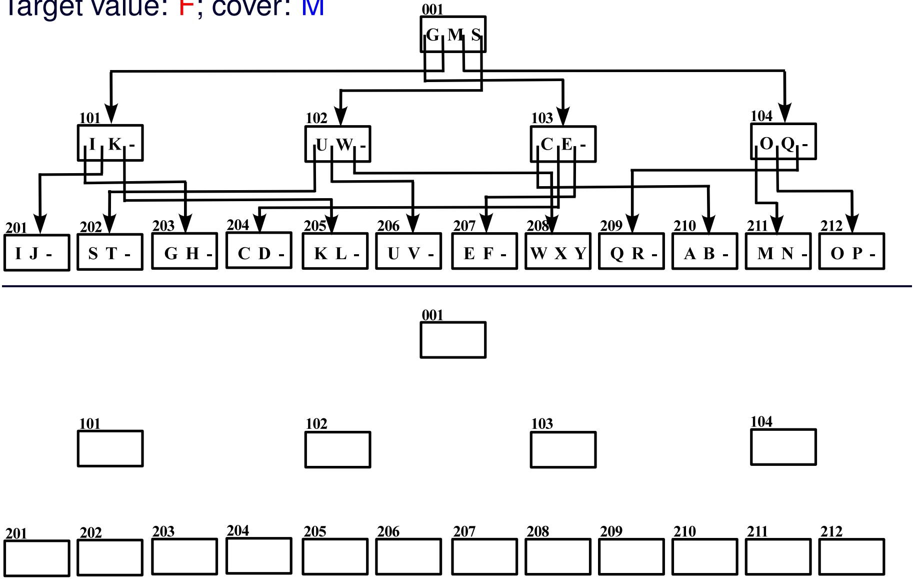
- + Leaf blocks have the same probability of containing the actual target
    - e.g., blocks 201 and 207 can be both the target block

- + The parent-child relationship between accessed blocks is confused
    - e.g., block 201 could be child of either 101 or 103

- – Parent-child relationship can be disclosed by intersection attacks

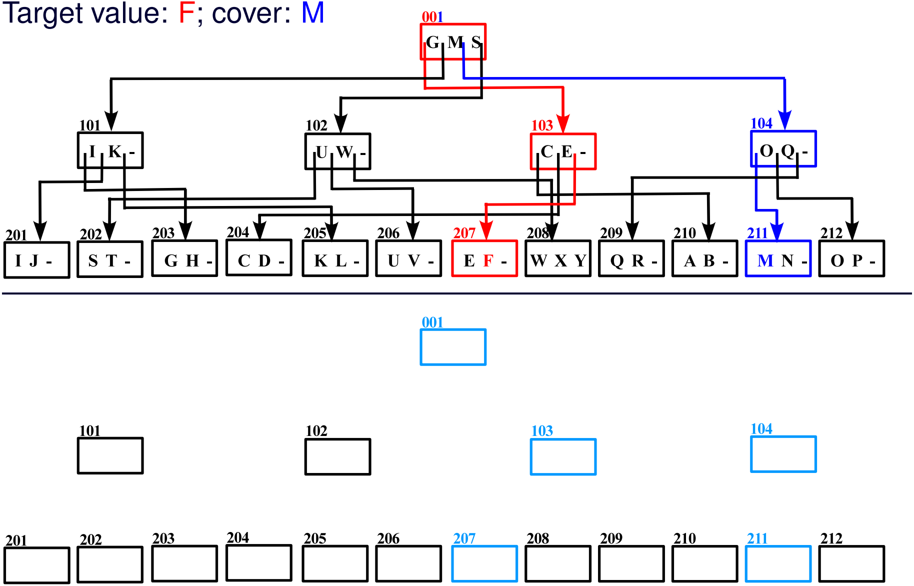
Cover searches – Example (2)

Target value: F; cover: M

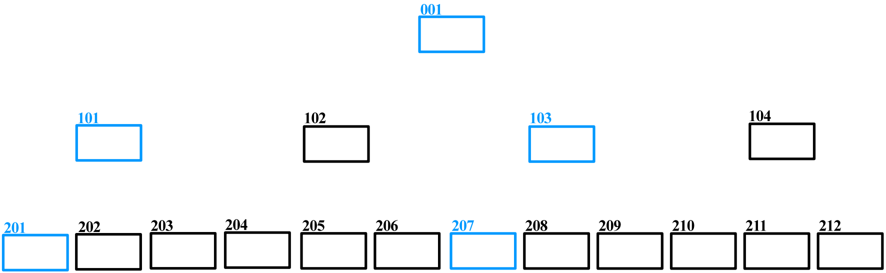


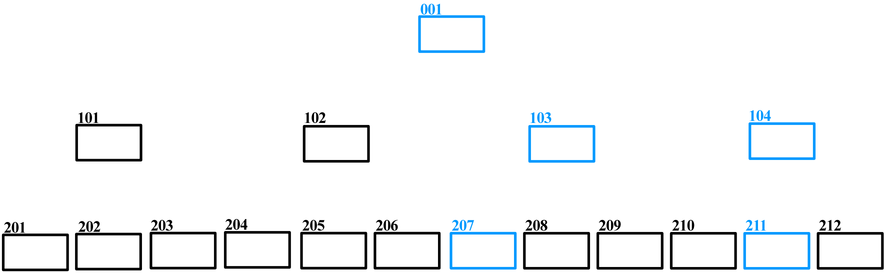
Cover searches – Example (2)

Target value: F; cover: M

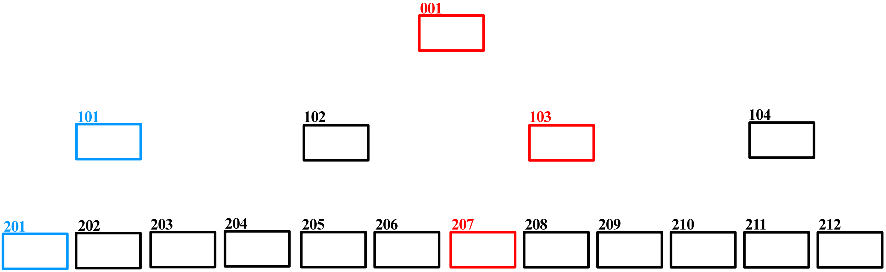


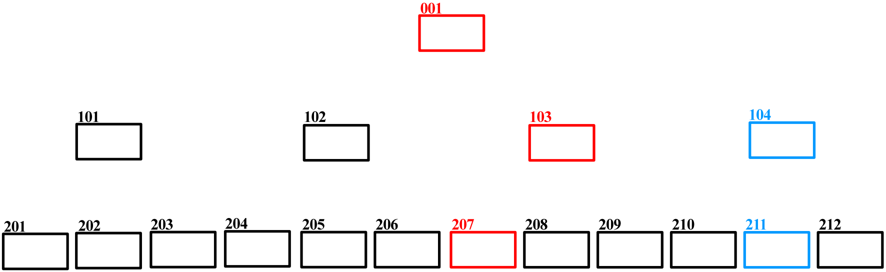
Cover searches – Intersection attack



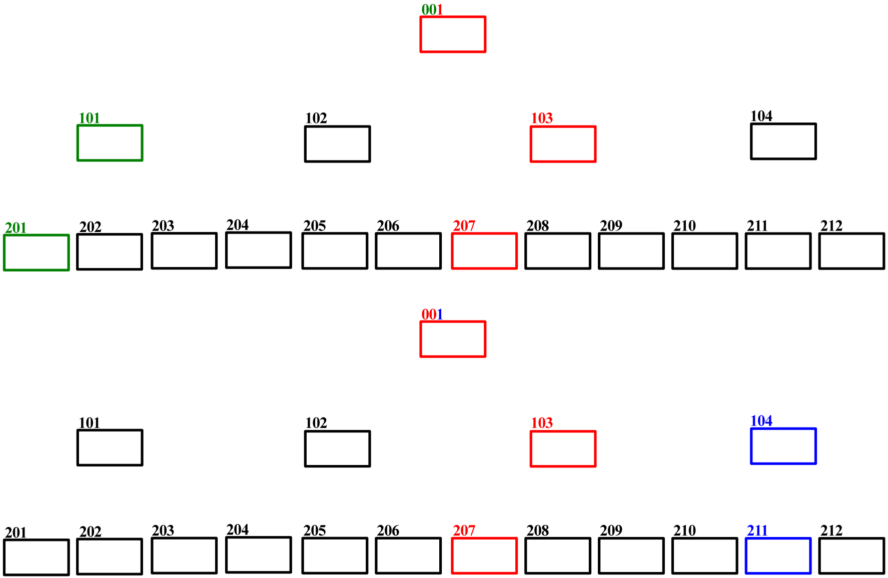


Cover searches – Intersection attack





Cover searches – Intersection attack



Cached searches

- The client maintains a local cache of nodes in the path to the target for counteracting intersection attacks
  - initialized with num_cache disjoint paths and is managed according to the LRU policy
  - if a node is in cache, its parent also is (path continuity property)
  - refreshed at every access
  - recently searched nodes will be found in the cache
  - if a target node is in cache, only cover searches will be performed
    - provides fake observations for the server
    - allows (with shuffling) refreshing the cache

Cached searches – Example (1)

l	$Cache_l$
0	001 $[{}_{103}\mathrm{G}_{101}\mathrm{M}_{104}\mathrm{S}_{102}]$
1	102 $[{}_{202}\mathrm{U}_{206}\mathrm{W}_{208}\text{- -}]$
2	203 [GH-]

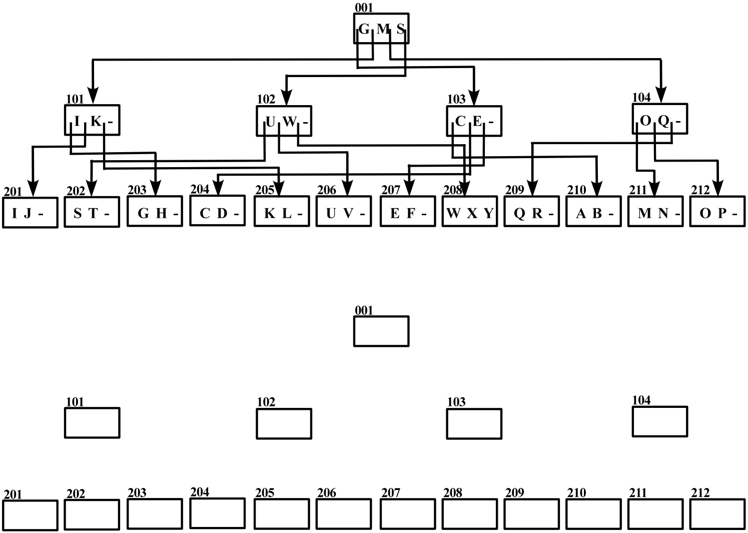
num_cover=1

num_cache=1

first search:

target= F

cover= I



Cached searches – Example (1)

l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	102 $[_{202}U_{206}W_{208}- -]$
2	203 [GH-]

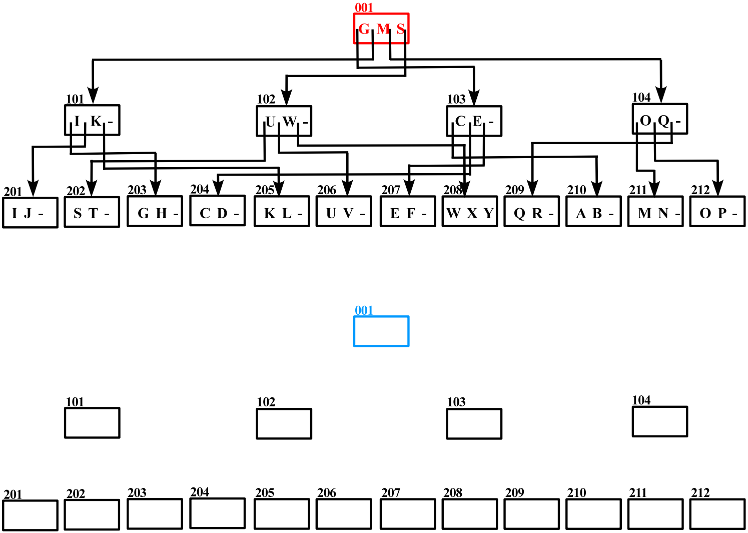
num_cover=1

num_cache=1

first search:

target= F

cover= I



Cached searches – Example (1)

l	$Cache_l$
0	001 [$_{103}$G$_{101}$M$_{104}$S$_{102}$]
1	102 [$_{202}$U$_{206}$W$_{208}$- -]
2	203 [GH-]

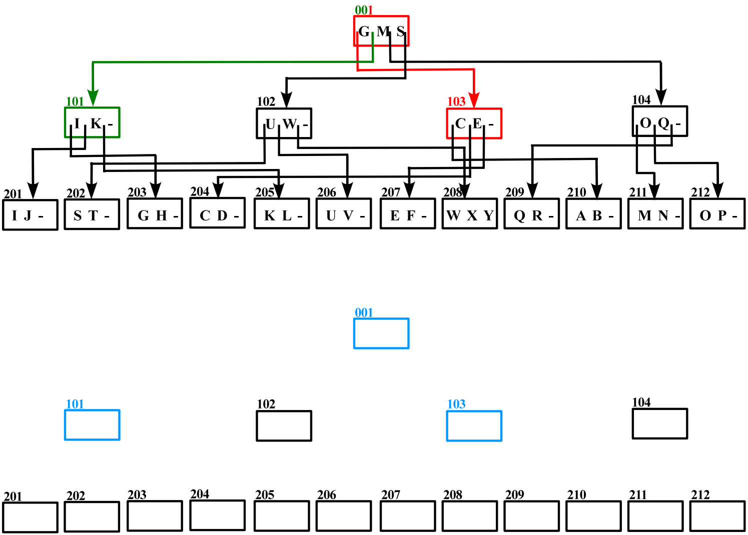
num_cover=1

num_cache=1

first search:

target= F

cover= I



Cached searches – Example (1)

l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	103 $[_{210}C_{204}E_{207}$ - -$]$
2	203 [GH-]

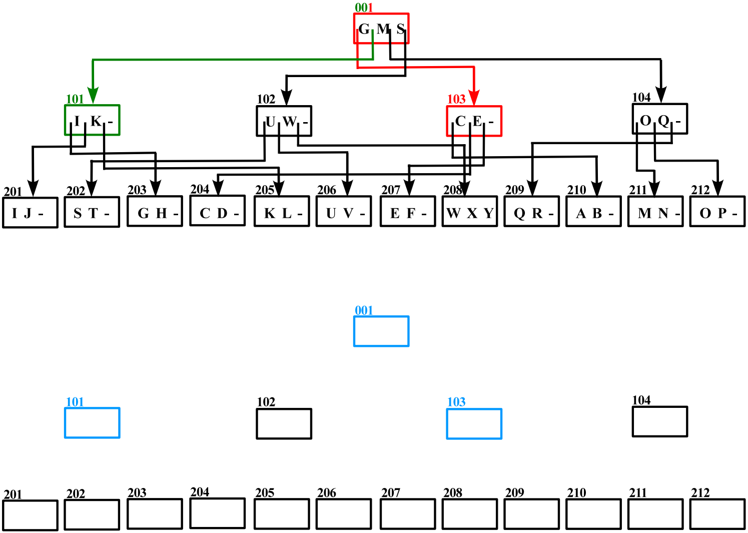
num_cover=1

num_cache=1

first search:

target= F

cover= I



Cached searches – Example (1)

l	*Cache$_l$*
0	001 [$_{103}$G$_{101}$M$_{104}$S$_{102}$]
1	103 [$_{210}$C$_{204}$E$_{207}$- -]
2	203 [GH-]

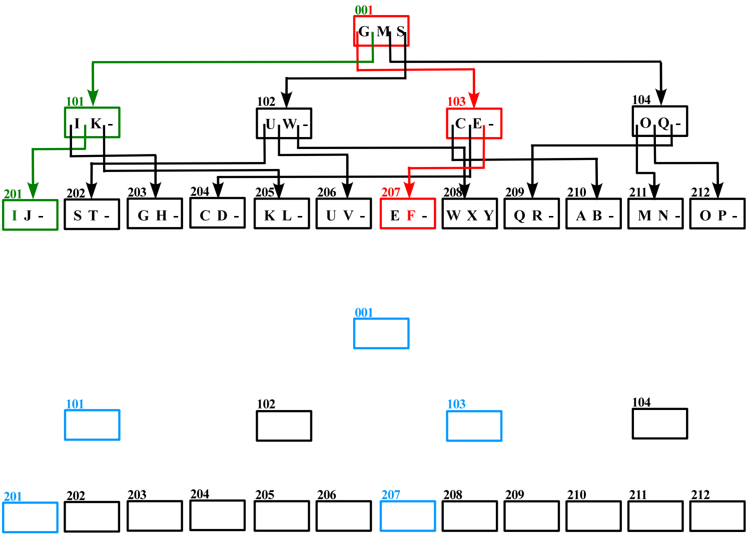
num_cover=1

num_cache=1

first search:

target= F

cover= I



Cached searches – Example (1)

l	$Cache_l$
0	001 [$_{103}$G$_{101}$M$_{104}$S$_{102}$]
1	103 [$_{210}$C$_{204}$E$_{207}$- -]
2	207 [EF-]

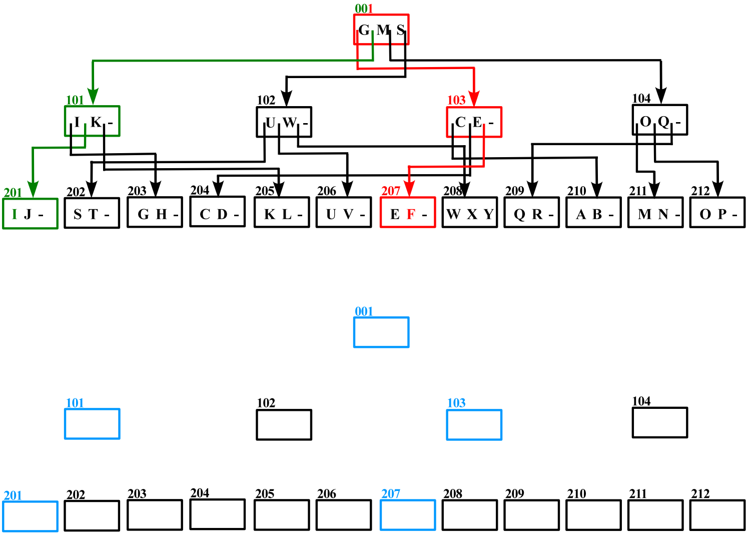
num_cover=1

num_cache=1

first search:

target= F

cover= I



Cached searches – Example (2)

l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	103 $[_{210}C_{204}E_{207}- -]$
2	207 [EF-]

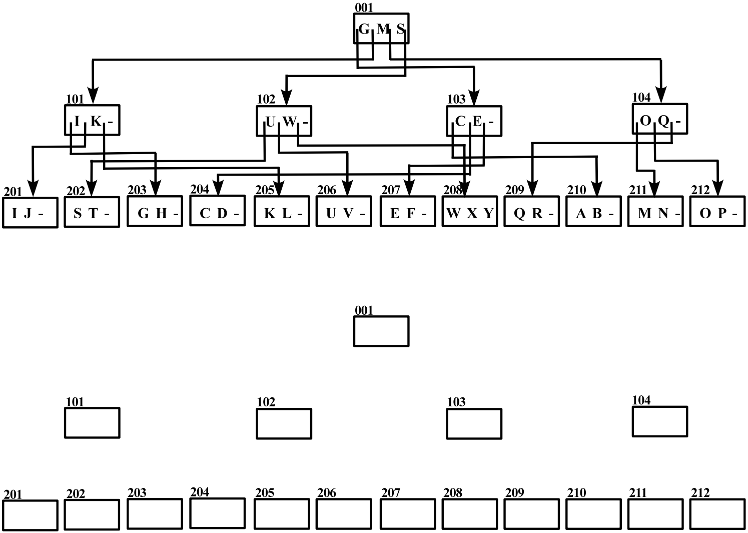
num_cover=1

num_cache=1

second search:

target= F

covers= M,W



Cached searches – Example (2)

l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	103 $[_{210}C_{204}E_{207}- \ -]$
2	207 [EF-]

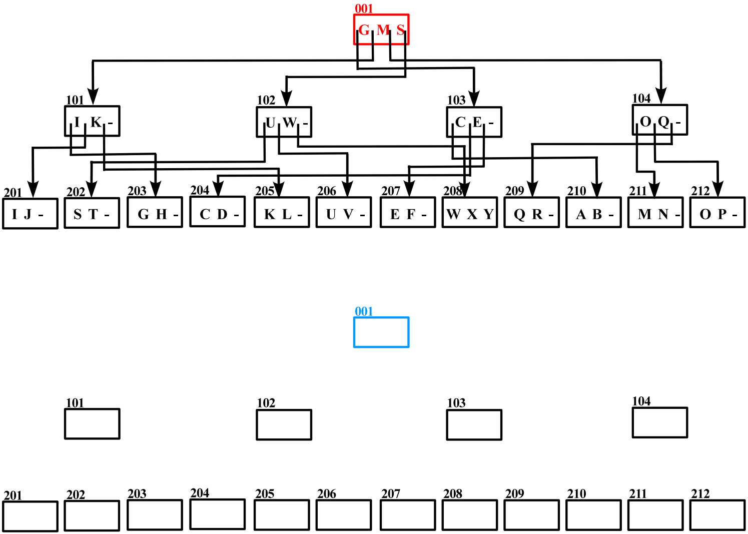
num_cover=1

num_cache=1

second search:

target= F

covers= M,W



Cached searches – Example (2)

l	$Cache_l$
0	$001\ [{}_{103}G_{101}M_{104}S_{102}]$
1	$103\ [{}_{210}C_{204}E_{207}-\ -]$
2	$207\ [EF-]$

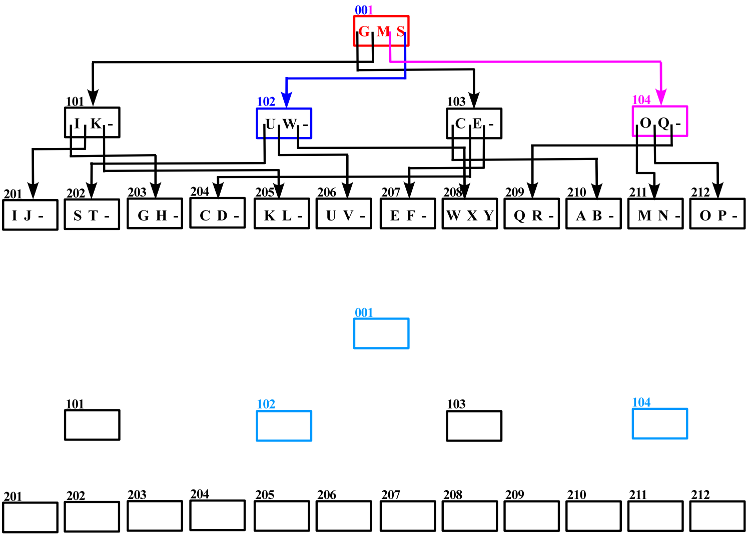
num_cover=1

num_cache=1

second search:

target= F

covers= M,W



Cached searches – Example (2)

l	$Cache_l$
0	001 [$_{103}$G$_{101}$M$_{104}$S$_{102}$]
1	103 [$_{210}$C$_{204}$E$_{207}$- -]
2	207 [EF-]

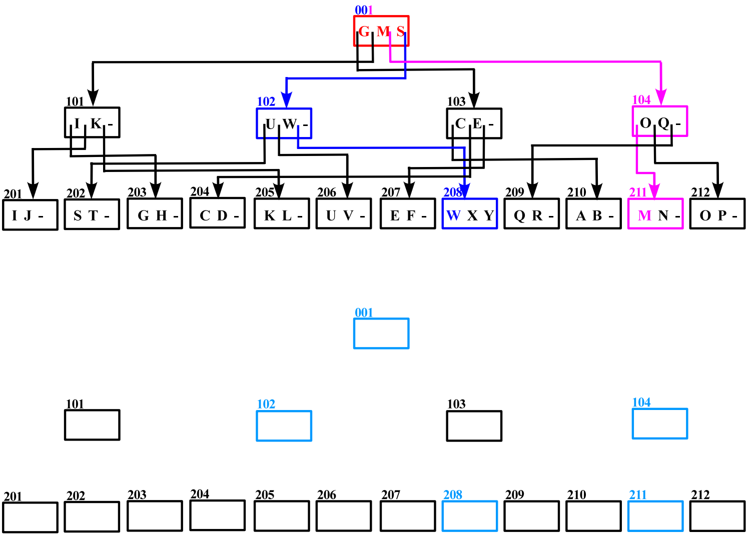
num_cover=1

num_cache=1

second search:

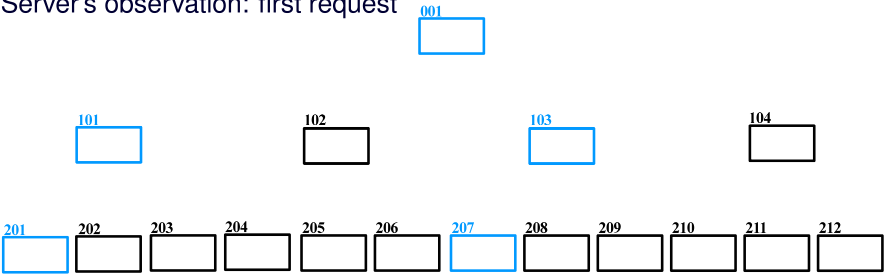
target= F

covers= M,W

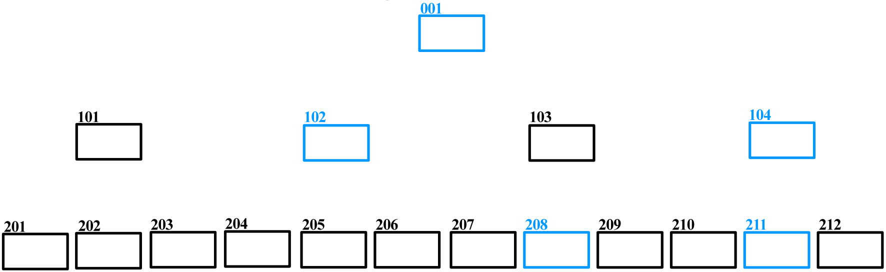


Cached searches – No intersection attack

Server's observation: first request



Server's observation: second request



Cached searches – Protection offered

+ Caching helps in counteracting short term intersection attacks

  ◦ e.g., the observations of the server on the two previous requests would be {(001); (101,103); (201,207)} and {(001); (102,104); (208,211)}

    ⟹ the server would not be able to determine whether the two requests aim at the same target

– Caching does not prevent intersection attacks on observations that go beyond the size of the cache

– A long history of observations will allow the server to reconstruct the topology (parent-child relationship) of the shuffle index

Shuffling

- Shuffling breaks the one-to-one correspondence blocks-nodes by exchanging the content among nodes (and therefore blocks)

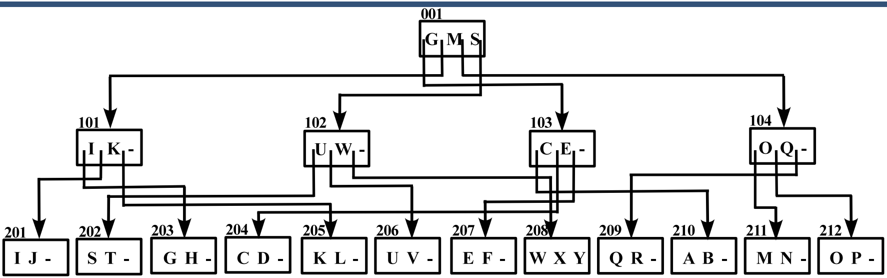
- Shuffling requires node decryption and re-encryption
    - encrypted text corresponding to a given node changes at each access (different node identifier and salt)

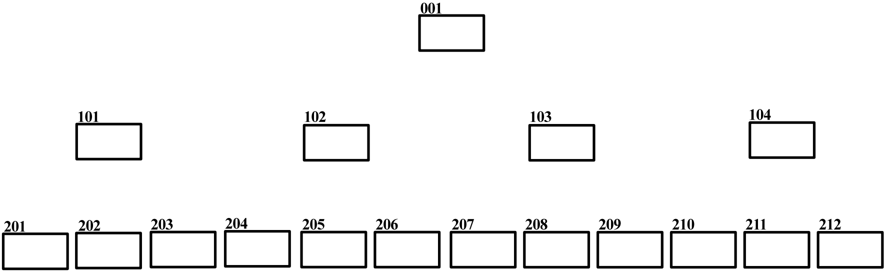
- The contents of all blocks read in the execution of an access and the nodes in cache are exchanged

- The shuffled blocks are rewritten back on the server

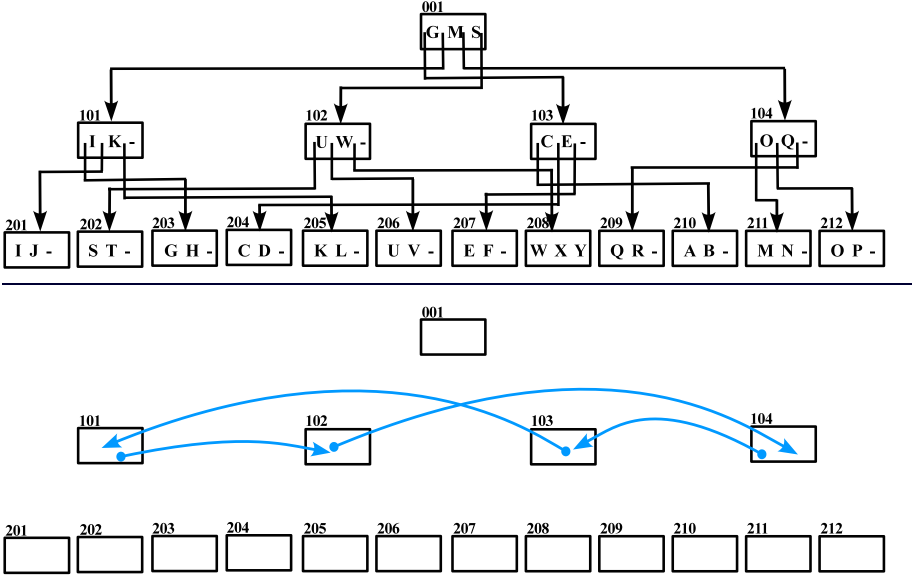
$\Longrightarrow$ node shuffling at a given level requires to update the parents of the nodes

Shuffling – Example

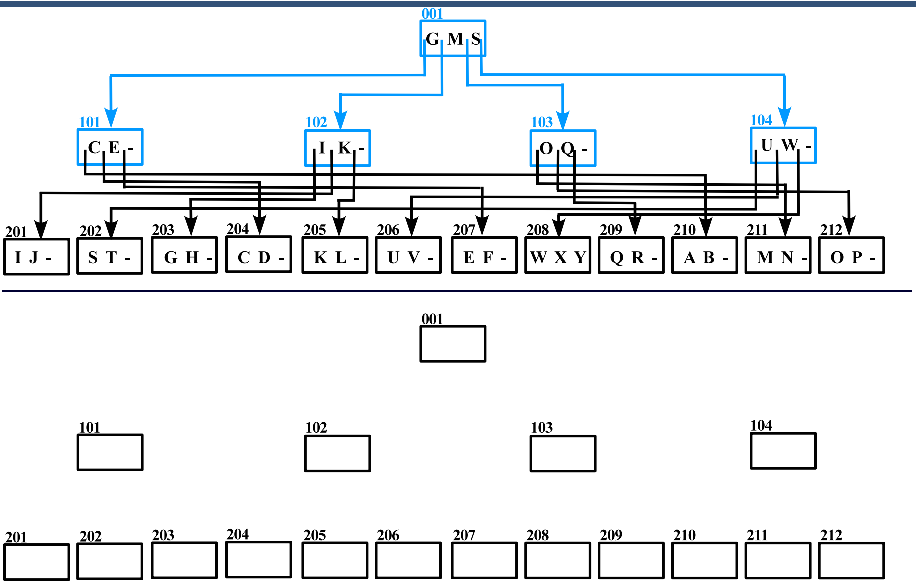




Shuffling – Example



Shuffling – Example



Access execution and shuffle index management

Let v be the target value. Determine num_cover+1 cover values and for each level l of the shuffle index:

- determine the identifiers (ToRead_ids) of the blocks in the path to v and cover values
- if the node in the path to v does not belong to $Cache_l$ (cache miss), only num_cover cover searches are performed
- send to the server a request for the blocks with identifier in ToRead_ids and decrypt their content (set Read of nodes)
- shuffle nodes in Read and in $Cache_l$ according to a permutation π
- update the pointers of the parents of the shuffled nodes
- update $Cache_l$ by inserting the most recently accessed node in the path to v (only if a cache miss occurred)

Access execution – Example

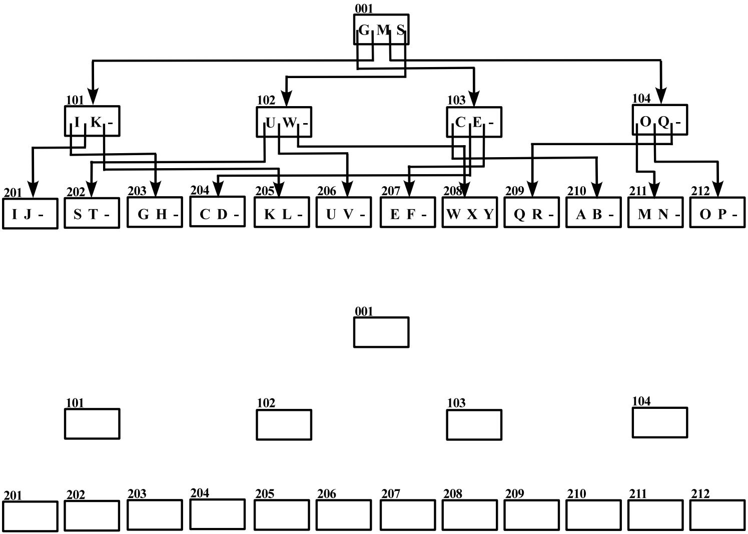
l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	101 $[_{203}I_{201}K_{205}- -]$
	103 $[_{210}C_{204}E_{207}- -]$
2	203 [GH-]
	210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

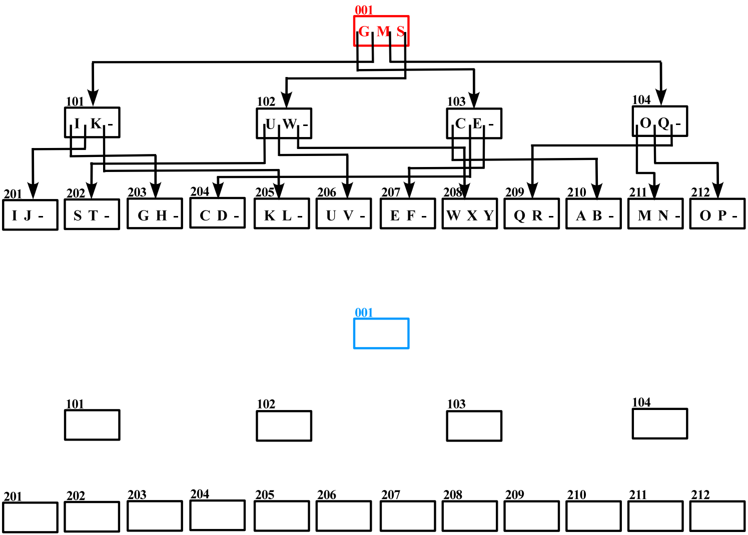
l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	101 $[_{203}I_{201}K_{205}- -]$
	103 $[_{210}C_{204}E_{207}- -]$
2	203 [GH-]
	210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

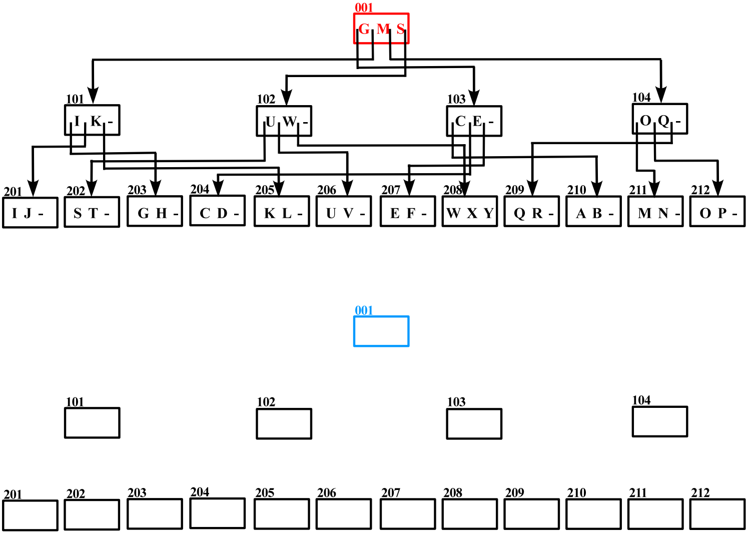
l	$Cache_l$
0	001 $[_{103}G_{101}M_{104}S_{102}]$
1	101 $[_{203}I_{201}K_{205}- -]$
	103 $[_{210}C_{204}E_{207}- -]$
2	203 [GH-]
	210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

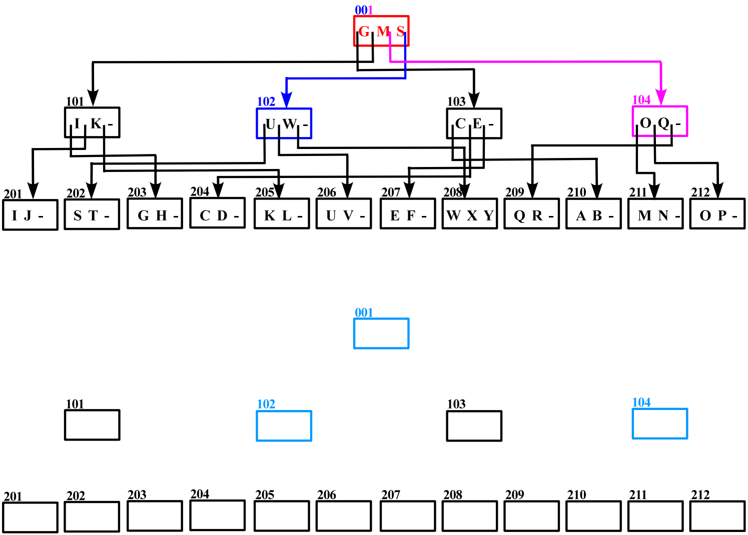
l	$Cache_l$
0	001 $[_{103}\mathrm{G}_{101}\mathrm{M}_{104}\mathrm{S}_{102}]$
1	101 $[_{203}\mathrm{I}_{201}\mathrm{K}_{205}\text{- -}]$
	103 $[_{210}\mathrm{C}_{204}\mathrm{E}_{207}\text{- -}]$
2	203 [GH-]
	210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

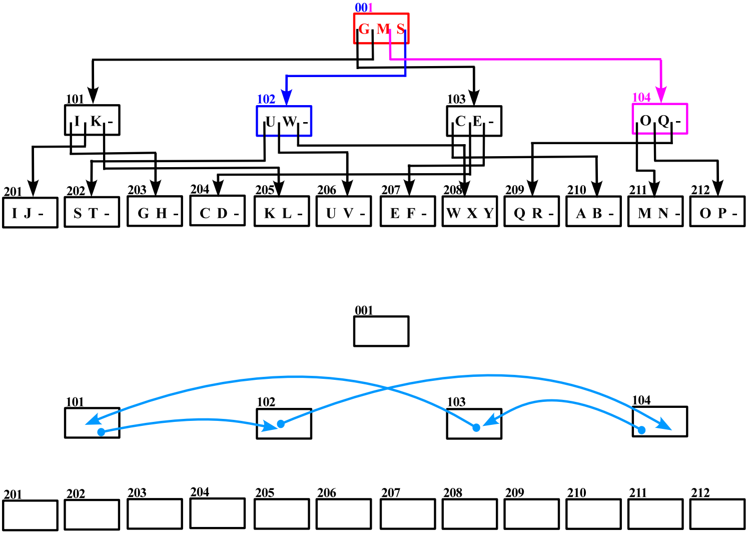
l	$Cache_l$
0	001 $[_{103}\mathrm{G}_{101}\mathrm{M}_{104}\mathrm{S}_{102}]$
1	101 $[_{203}\mathrm{I}_{201}\mathrm{K}_{205}- -]$ 103 $[_{210}\mathrm{C}_{204}\mathrm{E}_{207}- -]$
2	203 [GH-] 210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

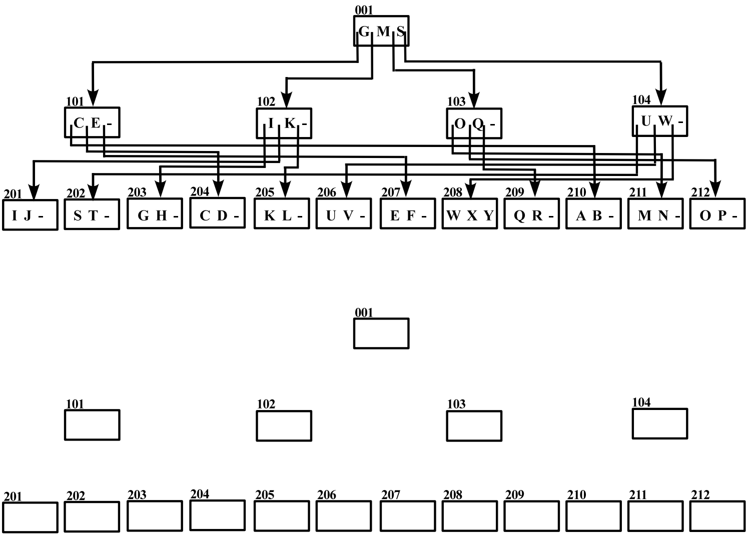
l	$Cache_l$
0	001 $[_{101}G_{102}M_{103}S_{104}]$
1	102 $[_{203}I_{201}K_{205}\text{- -}]$
	101 $[_{210}C_{204}E_{207}\text{- -}]$
2	203 [GH-]
	210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

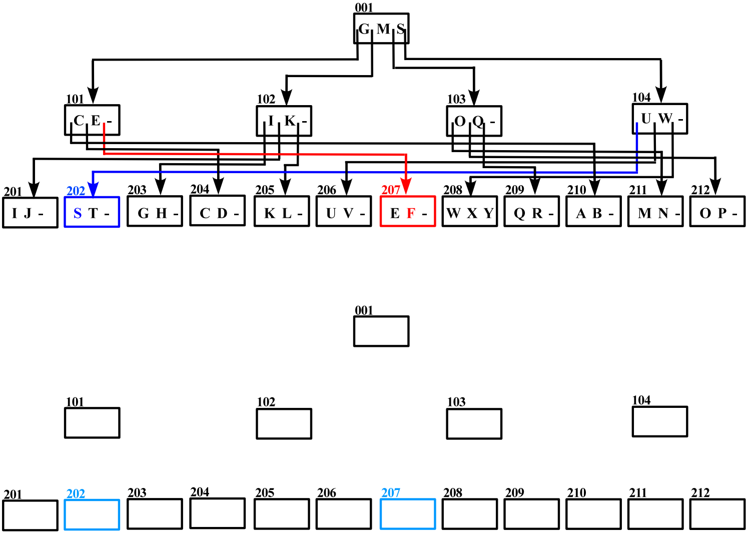
l	$Cache_l$
0	001 $[_{101}\text{G}_{102}\text{M}_{103}\text{S}_{104}]$
1	102 $[_{203}\text{I}_{201}\text{K}_{205}\text{- -}]$ 101 $[_{210}\text{C}_{204}\text{E}_{207}\text{- -}]$
2	203 [GH-] 210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

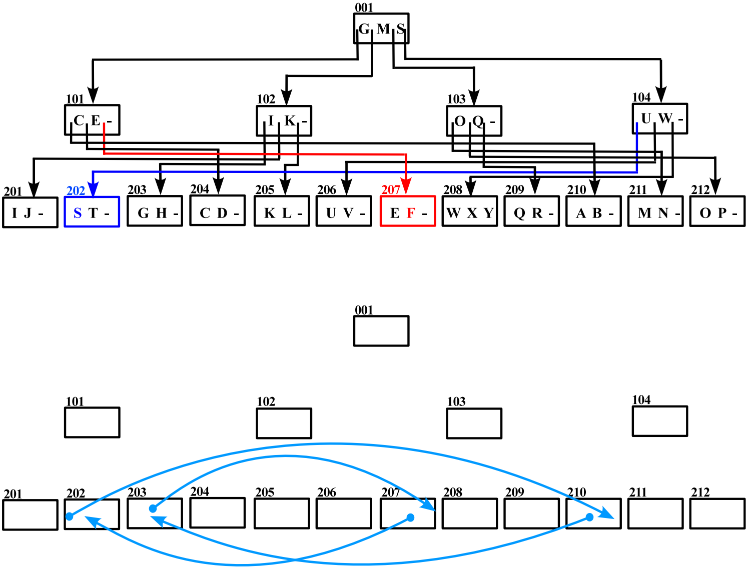
l	$Cache_l$
0	001 $[_{101}G_{102}M_{103}S_{104}]$
1	102 $[_{203}I_{201}K_{205}- -]$
	101 $[_{210}C_{204}E_{207}- -]$
2	203 [GH-]
	210 [AB-]

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

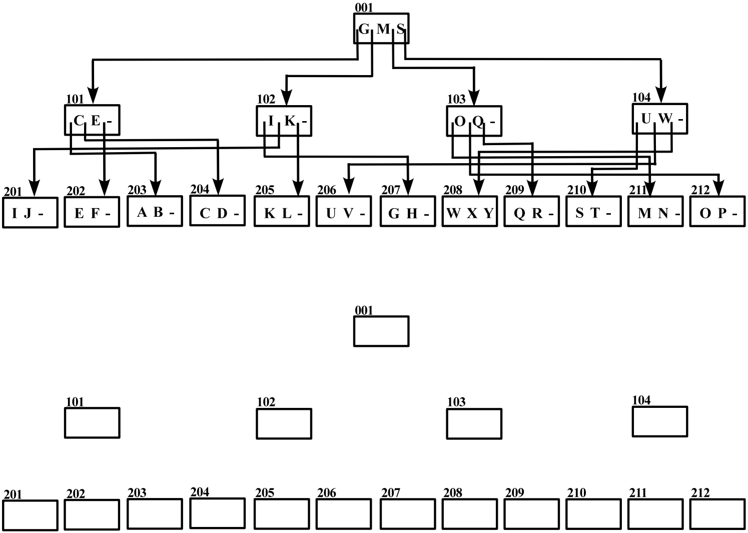
l	$Cache_l$
0	$001\ [_{101}G_{102}M_{103}S_{104}]$
1	$102\ [_{207}I_{201}K_{205}-\ -]$
	$101\ [_{203}C_{204}E_{202}-\ -]$
2	$207\ [GH-]$
	$202\ [EF-]$

num_cover=1

num_cache=2

target= F

covers= S,M



Access execution – Example

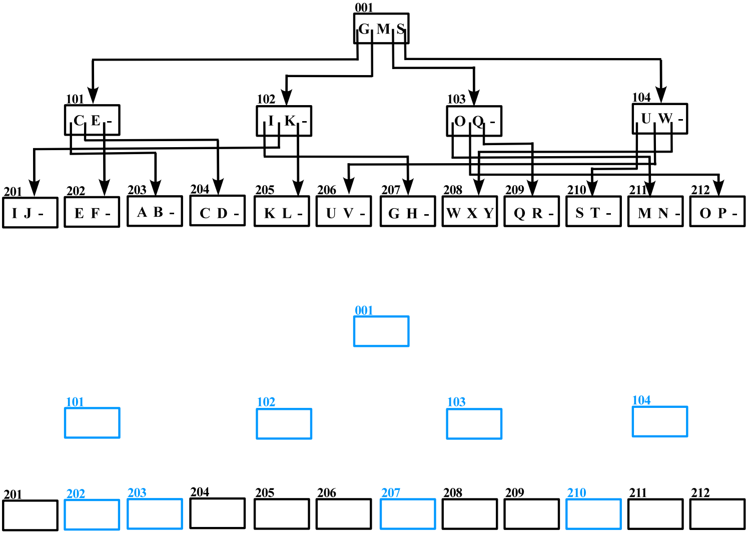
l	$Cache_l$
0	$001\,[{}_{101}G{}_{102}M{}_{103}S{}_{104}]$
1	$102\,[{}_{207}I{}_{201}K{}_{205}\text{- -}]$
	$101\,[{}_{203}C{}_{204}E{}_{202}\text{- -}]$
2	$207\,[GH\text{-}]$
	$202\,[EF\text{-}]$

num_cover=1

num_cache=2

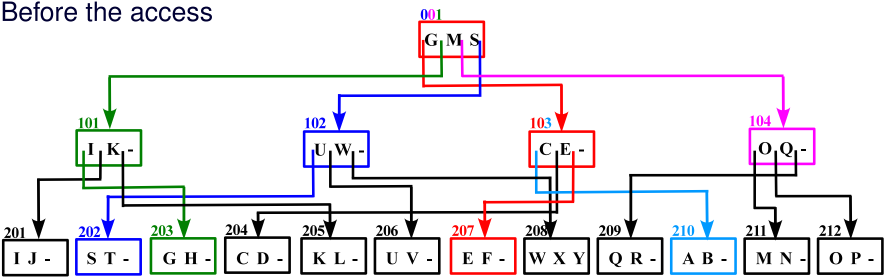
target= F

covers= S,M

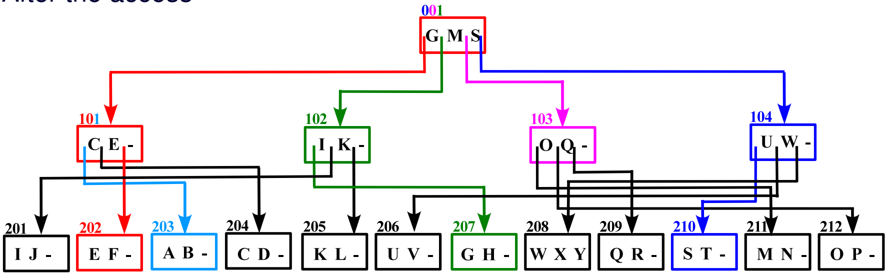


Access execution – Impact on the logical index

Before the access



After the access



Protection analysis

- Degradation due to shuffling: shuffling degrades any information the server may possess on the correspondence between nodes and blocks
- Access confidentiality: every time an access is performed any information on the specific access has to be divided among num_cover + 1 nodes and shuffling destroys the correspondence nodes-blocks
- Pattern confidentiality: accesses separated by a significant number of steps cannot be recognized (shuffling):
  - protection by covers and cache (short term)
  - protection by covers and shuffling (long term)

Protection vs performance

- Protection comes at a cost:
  - – one read access implies num_cover + num_cache + 1 writes back to the server
  - + no solution providing support for access and pattern confidentiality offers comparable performance
  - + even in a WAN configuration the shuffle index enjoys better performance with respect to approaches providing comparable protection

Extensions to the shuffle index

The shuffle index can be extended to efficiently:

- support concurrent accesses (delta versions) [DFPPS-11b]
- operate on multiple servers for storing and accessing data (shadows) [DFPPS-13]

Integrity in Query Computation

Integrity of storage and query computation – 1

- Data owner and users need mechanisms that provide integrity for query results:
  - correctness: computed on genuine data
  - completeness: computed on the whole data collection
  - freshness: computed on the most recent version of the data
- Two approaches:
  - deterministic: uses authenticated data structures (e.g., signature chains, Merkle hash trees, skip lists) or encryption-based solutions (e.g., verifiable homomorphic encryption schema [LDPW-14])
  - probabilistic: exploits insertion of fake tuples in query results, replication of tuples in query results, pre-computed tokens (e.g., [DFJPS-13b,DFJPS-14,DFJLPS-14b,XWYM-07])

Integrity of storage and query computation – 2

- Other approaches consider the verification of the integrity of query results of complex queries (joins):
  - fake tuples [XWYM-07]
    - spurious tuples
    - network overhead
  - Merkle hash tree or its variations [LHKR-06 YPPK-09]
    - support only joins on which the Merkle hash tree has been constructed

Merkle hash tree

- Binary tree where:
  - each leaf contains the hash of one tuple
  - each internal node contains the result of the hash of the concatenation of its children
- The hash function used to build the tree is collision-resistant
- The root is signed by the data owner and communicated to authorized users
- Tuples in the leaves are ordered according to the value of the attribute A on which the tree is defined
- The tree is created by the data owner and stored at the server

Merkle hash tree – Example

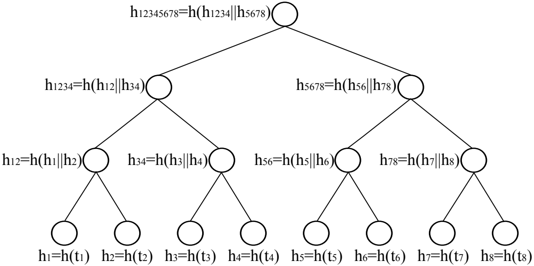
Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes

Merkle hash tree – Example

Patients

	<u>SSN</u>	Name	Disease
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes



Merkle hash tree over attribute SSN

Merkle hash tree verification

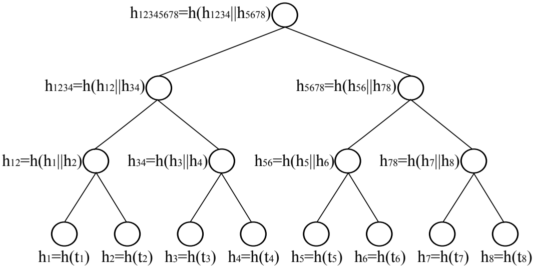
- The Merkle hash tree defined over A supports the verification of equality and range queries over A
- The server returns, together with the query result, a verification object (hash of other tuples allowing to derive the hash of the root)
- The client uses the verification object and query result to recompute the root of the tree
- The query result is correct and complete iff the computed root is the same as the one she knows
  - if a tuple is not correct or is missing from the query result, the recomputed root value is not the same as the one known to the client

Merkle hash tree verification – Example

```
SELECT *
FROM Patients
WHERE SSN = ‘345-67-8912’
```

Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes

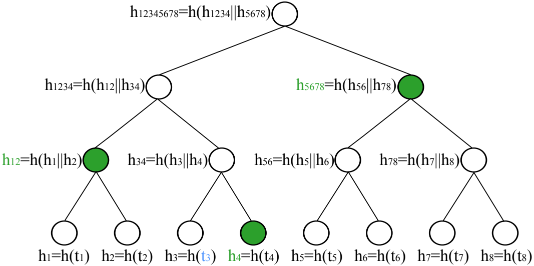


Merkle hash tree verification – Example

SELECT *
FROM Patients
WHERE SSN = ‘345-67-8912’

Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes



Result: t_3

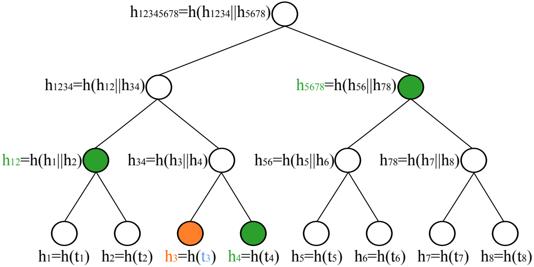
Verification Object: h_4, h_{12}, h_{5678}

Merkle hash tree verification – Example

SELECT *
FROM Patients
WHERE SSN = '345-67-8912'

Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes



Result: t_3

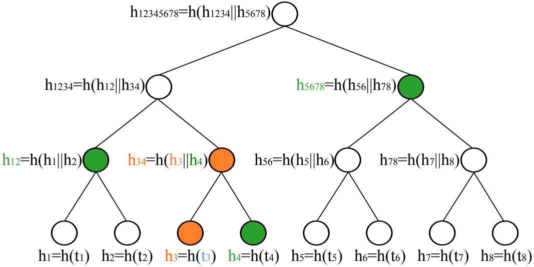
Verification Object: h_4, h_{12}, h_{5678}
h_3 = h(t_3)

Merkle hash tree verification – Example

SELECT *
FROM Patients
WHERE SSN = ‘345-67-8912’

Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes



Result: t_3

Verification Object: h_4, h_{12}, h_{5678}

$\mathrm{h}_3 = \mathrm{h}(t_3)$

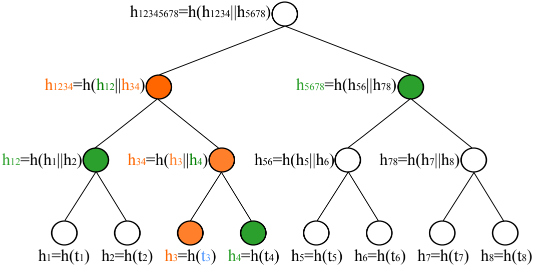
$\mathrm{h}_{34} = \mathrm{h}(\mathrm{h}_3 || \mathrm{h}_4)$

Merkle hash tree verification – Example

SELECT *
FROM Patients
WHERE SSN = ‘345-67-8912’

Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes



Result: t_3

Verification Object: h_4, h_{12}, h_{5678}

$h_3 = h(t_3)$

$h_{34} = h(h_3||h_4)$

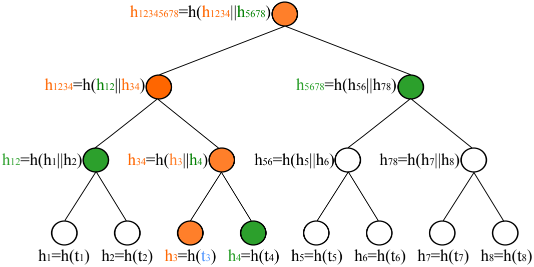
$h_{1234} = h(h_{12}||h_{34})$

Merkle hash tree verification – Example

SELECT *
FROM Patients
WHERE SSN = ‘345-67-8912’

Patients

	SSN	**Name**	**Disease**
t_1	123-45-6789	Alice	Asthma
t_2	234-56-7891	Bob	Asthma
t_3	345-67-8912	Carol	Asthma
t_4	456-78-9123	David	Bronchitis
t_5	567-89-1234	Eva	Bronchitis
t_6	678-91-2345	Frank	Gastritis
t_7	789-12-3456	Gary	Gastritis
t_8	891-23-4567	Hilary	Diabetes



Result: t_3

Verification Object: h_4, h_{12}, h_{5678}

$h_3 = h(t_3)$

$h_{34} = h(h_3||h_4)$

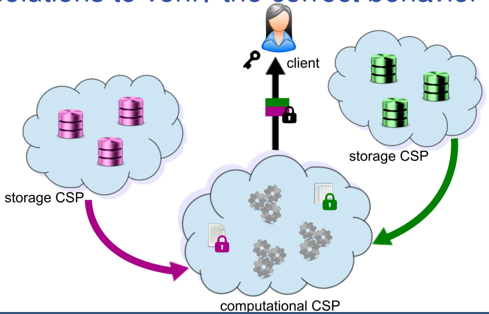
$h_{1234} = h(h_{12}||h_{34})$

$h_{12345678} = h(h_{1234}||h_{5678})$

Computation with multiple providers

- Different CSPs are available on the market, offering a variety of services (e.g., storage, computation) at different prices
- Users can select the CSP that better matches their security, economic, and functional requirements
- Multiple CSPs can help enhancing security but

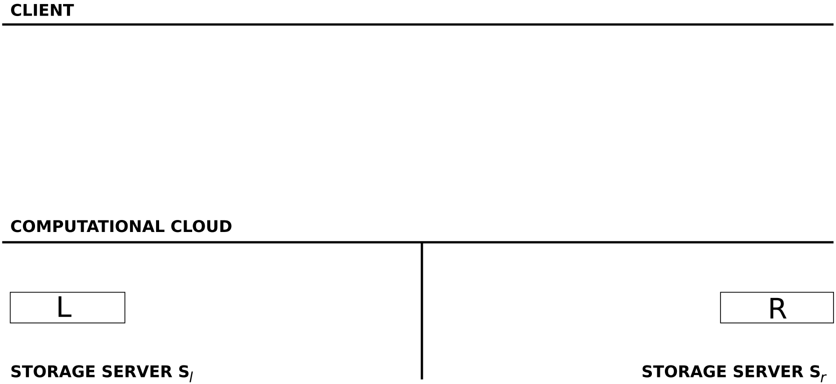
  $\Longrightarrow$ need solutions to verify the correct behavior of these CSPs



Probabilistic approach for join queries

- A client, with the cooperation of the storage servers, can assess the integrity of joins performed by a computational cloud
- Protection techniques [DFJPS-13b,DFJPS-14]:
  - encryption makes data unintelligible
  - markers, fake tuples not recognizable as such by the computational cloud (and not colliding with real tuples)
  - twins, replication of existing tuples
  - salts/buckets, replications with salts (at side 1) and dummy tuples (at side many) to flatten occurrences of matches in 1:n joins
- A marker missing or a twin appearing solo $\Longrightarrow$ integrity violation
- Probabilistic guarantee depending on the amount of control (markers and twins) inserted

Probabilistic approach for join queries – Example



CLIENT

COMPUTATIONAL CLOUD

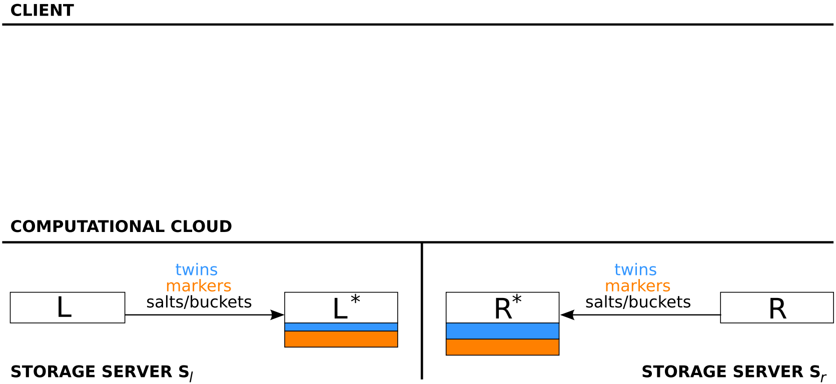
L

R

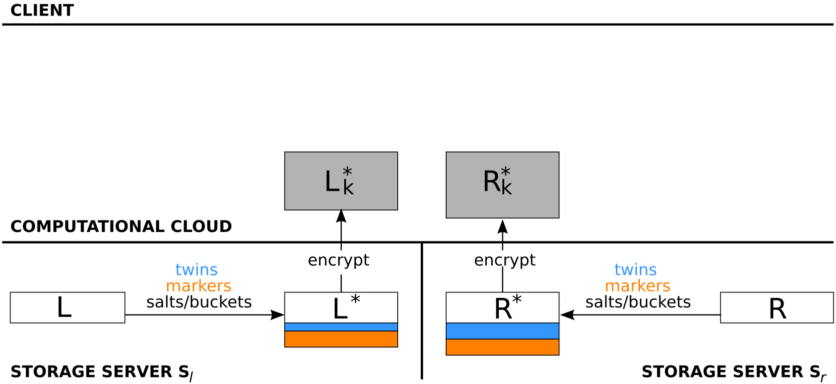
STORAGE SERVER S_l

STORAGE SERVER S_r

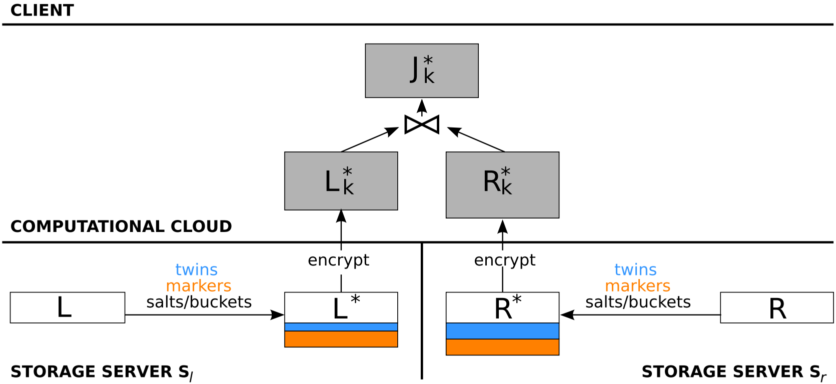
Probabilistic approach for join queries – Example



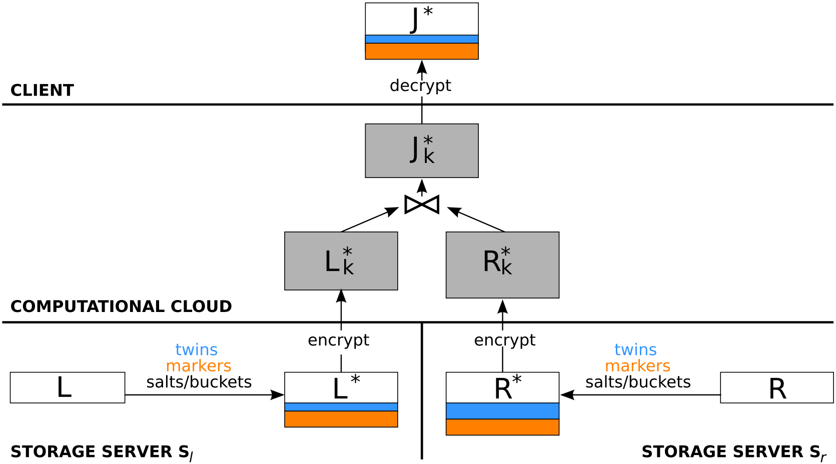
Probabilistic approach for join queries – Example



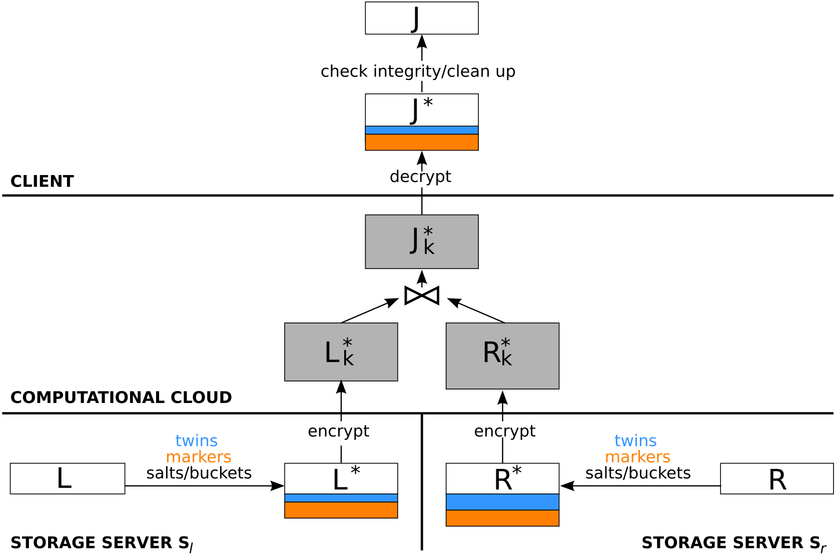
Probabilistic approach for join queries – Example



Probabilistic approach for join queries – Example



Probabilistic approach for join queries – Example



On-the-fly encryption

- Server S encrypts $B(I, \mathit{Att})$, obtaining $B_k(I_k, B.\mathit{Tuple}_k)$
    - For each t in B, there is τ in B_k: $\tau[I_k]{=}E_k(t[I])$ and $\tau[B.\mathit{Tuple}_k]{=}E_k(t)$
    - E is a symmetric encryption function with key k
    - k is defined by the client and changes at every query
- Encryption provides data confidentiality

R_l

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe

R_r

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer

J

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3

On-the-fly encryption

- Server S encrypts $B(I, Att)$, obtaining $B_k(I_k, B.Tuple_k)$
  - For each t in B, there is τ in B_k: $\tau[I_k]{=}E_k(t[I])$ and $\tau[B.Tuple_k]{=}E_k(t)$
  - E is a symmetric encryption function with key k
  - k is defined by the client and changes at every query
- Encryption provides data confidentiality

R_{lk}

$\mathbf{I_k}$	$\mathbf{L.Tuple_k}$
α	λ_1
β	λ_2
γ	λ_3

R_{rk}

$\mathbf{I_k}$	$\mathbf{R.Tuple_k}$
α	ρ_1
α	ρ_2
β	ρ_3
ε	ρ_4
ε	ρ_5
ε	ρ_6

J_k

$\mathbf{L.I_k}$	$\mathbf{L.Attr_k}$	$\mathbf{R.I_k}$	$\mathbf{R.Attr_k}$
α	λ_1	α	ρ_1
α	λ_1	α	ρ_2
β	λ_2	β	ρ_3

Markers

- Artificial tuples injected into R_l by S_l and R_r by S_r
  - not recognizable by the computational server
  - do not generate spurious tuples
  - inserted in a concerted manner to guarantee that they belong to the join result
- The absence of markers signals incompleteness of the join result

R_l

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe

R_r

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer

J

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3

Markers

- Artificial tuples injected into R_l by S_l and R_r by S_r
  - not recognizable by the computational server
  - do not generate spurious tuples
  - inserted in a concerted manner to guarantee that they belong to the join result
- The absence of markers signals incompleteness of the join result

$R_l{}^*$

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe
m_1	x	*marker*$_1$

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer
m_2	x	*marker*$_2$

J^*

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3
m_1	x	*marker*$_1$	x	*marker*$_2$	m_2

Twins

- Duplicates of tuples that satisfy condition C_{twin} that
  - is defined on the join attribute I
  - tunes the percentage p_t of twins
  - is defined by the client and communicated to S_l and S_r

- Twin pairs are not recognizable by the computational server

- A twin appearing solo signals incompleteness of the join result

R_l

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe

R_r

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer

J

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3

Twins

- Duplicates of tuples that satisfy condition C_{twin} that
  - is defined on the join attribute I
  - tunes the percentage p_t of twins
  - is defined by the client and communicated to S_l and S_r

- Twin pairs are not recognizable by the computational server

- A twin appearing solo signals incompleteness of the join result

$R_l{}^*$

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe
$\bar{l_2}$	$\bar{\text{b}}$	Beth

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer
$\bar{r_3}$	$\bar{\text{b}}$	ulcer

J^*

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3
$\bar{l_2}$	$\bar{\text{b}}$	Beth	$\bar{\text{b}}$	ulcer	$\bar{r_3}$

Salts and buckets

- Destroy recognizable frequencies of combinations in one-to-many joins
- Operate on original tuples, markers, and twins and can be adopted in alternative or in combination
- Salts
  - map different occurrences of the same join value on the side “many” of the join to a different encrypted value using a different salt
  - replicate each tuple on the side “one” of the join and combine replicas with different salts to guarantee the matching
- Buckets
  - insert dummy tuples on the side “many” of the join to guarantee flat frequency distribution of join attribute values

Salts and buckets – Example

- number of salts: 2
- maximum number of occurrences: 3

⇒ buckets with 2 tuples each

$R_l{}^*$

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer

J^*

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3

Salts and buckets – Example

- number of salts: 2
- maximum number of occurrences: 3

⇒ buckets with 2 tuples each

$R_l{}^*$

	I	Attr
l_1	a	Ann
$l_1{}'$	a′	Ann′
l_2	b	Beth
$l_2{}'$	b′	Beth′
l_3	c	Cloe
$l_3{}'$	c′	Cloe′

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
d_1	b	*dummy*$_1$
r_4	e	hernia
r_5	e	flu
r_6	e′	cancer
d_2	e′	*dummy*$_2$

J^*

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3
l_2	b	Beth	b	*dummy*$_1$	d_1

Query evaluation

The client shares with each server S_i a symmetric key k_i

- The client send the computational cloud a request to execute a join between the relations produced by S_l and S_r

- The relations to be produced by S_l and S_r are represented as two strings, encrypted with keys k_l and k_r, respectively, and to be forwarded by the computational cloud to the respective storage server, containing:
  - subquery to be executed by the storage server
  - query key k (on-the-fly encryption) to be used by the storage server to encrypt the relation sent to the computational cloud
  - number m of markers
  - percentage p_t of twins
  - number s of salts

Join execution – Example

R_l

	I	**Attr**
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe

R_r

	I	**Attr**
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer

Storage servers

Join execution – Example

$R_l{}^*$

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe
$\bar{l}_2$	b̄	Beth

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer
$\bar{r}_3$	b̄	ulcer

Storage servers

Join execution – Example

$R_l{}^*$

	I	Attr
l_1	a	Ann
l_2	b	Beth
l_3	c	Cloe
$\bar{l_2}$	$\bar{\text{b}}$	Beth
m_1	x	*marker*$_1$

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
r_4	e	hernia
r_5	e	flu
r_6	e	cancer
$\bar{r_3}$	$\bar{\text{b}}$	ulcer
m_2	x	*marker*$_2$

Storage servers

Join execution – Example

R_l^*

	I	Attr
l_1	a	Ann
l_1'	a′	Ann′
l_2	b	Beth
l_2'	b′	Beth′
l_3	c	Cloe
l_3'	c′	Cloe′
$\bar{l}_2$	$\bar{\text{b}}$	Beth
$\bar{l}_2'$	$\bar{\text{b}}'$	Beth′
m_1	x	*marker*$_1$

R_r^*

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
d_1	b	*dummy*$_1$
r_4	e	hernia
r_5	e	flu
r_6	e′	cancer
d_2	e′	*dummy*$_2$
$\bar{r}_3$	$\bar{\text{b}}$	ulcer
$\bar{d}_1$	$\bar{\text{b}}$	*dummy*$_1$
m_2	x	*marker*$_2$
d_3	x	*dummy*$_3$

Storage servers

Join execution – Example

$R_l{}^*$

	I	Attr
l_1	a	Ann
$l_1{}'$	a′	Ann′
l_2	b	Beth
$l_2{}'$	b′	Beth′
l_3	c	Cloe
$l_3{}'$	c′	Cloe′
$\bar{l}_2$	$\bar{\text{b}}$	Beth
$\bar{l}_2{}'$	$\bar{\text{b}}'$	Beth′
m_1	x	*marker*$_1$

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
d_1	b	*dummy*$_1$
r_4	e	hernia
r_5	e	flu
r_6	e′	cancer
d_2	e′	*dummy*$_2$
$\bar{r}_3$	$\bar{\text{b}}$	ulcer
$\bar{d}_1$	$\bar{\text{b}}$	*dummy*$_1$
m_2	x	*marker*$_2$
d_3	x	*dummy*$_3$

Storage servers

$R_{l_k}^*$

$\mathbf{I_k}$	$\mathbf{L.Tuple_k}$
α	λ_1
α'	$\lambda_1{}'$
β	λ_2
β'	$\lambda_2{}'$
γ	λ_3
γ'	$\lambda_3{}'$
$\bar{\beta}$	$\bar{\lambda}_2$
$\bar{\beta}'$	$\bar{\lambda}_2{}'$
χ	μ_1

$R_{r_k}^*$

$\mathbf{I_k}$	$\mathbf{R.Tuple_k}$
α	ρ_1
α	ρ_2
β	ρ_3
β	δ_1
ε	ρ_4
ε	ρ_5
ε'	ρ_6
ε'	δ_2
$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\delta}_1$
χ	μ_2
χ	δ_3

Computational cloud

Join execution – Example

$R_l{}^*$

	I	Attr
l_1	a	Ann
$l_1{}'$	a′	Ann′
l_2	b	Beth
$l_2{}'$	b′	Beth′
l_3	c	Cloe
$l_3{}'$	c′	Cloe′
$\bar{l}_2$	b̄	Beth
$\bar{l}_2{}'$	b̄′	Beth′
m_1	x	*marker*$_1$

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
d_1	b	*dummy*$_1$
r_4	e	hernia
r_5	e	flu
r_6	e′	cancer
d_2	e′	*dummy*$_2$
$\bar{r}_3$	b̄	ulcer
$\bar{d}_1$	b̄	*dummy*$_1$
m_2	x	*marker*$_2$
d_3	x	*dummy*$_3$

Storage servers

$R_{l_k}^*$

I_k	$L.Tuple_k$
α	λ_1
α'	$\lambda_1{}'$
β	λ_2
β'	$\lambda_2{}'$
γ	λ_3
γ'	$\lambda_3{}'$
$\bar{\beta}$	$\bar{\lambda}_2$
$\bar{\beta}'$	$\bar{\lambda}_2{}'$
χ	μ_1

$R_{r_k}^*$

I_k	$R.Tuple_k$
α	ρ_1
α	ρ_2
β	ρ_3
β	δ_1
ε	ρ_4
ε	ρ_5
ε'	ρ_6
ε'	δ_2
$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\delta}_1$
χ	μ_2
χ	δ_3

J_k^*

$L.I_k$	$L.Tuple_k$	$R.I_k$	$R.Tuple_k$
α	λ_1	α	ρ_1
α	λ_1	α	ρ_2
β	λ_2	β	ρ_3
β	λ_2	β	δ_1
$\bar{\beta}$	$\bar{\lambda}_2$	$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\lambda}_2$	$\bar{\beta}$	δ_1
χ	μ_1	χ	μ_2
χ	μ_1	χ	δ_3

Computational cloud

Join execution – Example

Storage servers

R_l^*

	I	Attr
l_1	a	Ann
l_1'	a′	Ann′
l_2	b	Beth
l_2'	b′	Beth′
l_3	c	Cloe
l_3'	c′	Cloe′
$\bar{l}_2$	$\bar{\text{b}}$	Beth
$\bar{l}_2'$	$\bar{\text{b}}'$	Beth′
m_1	x	*marker*$_1$

R_r^*

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
d_1	b	*dummy*$_1$
r_4	e	hernia
r_5	e	flu
r_6	e′	cancer
d_2	e′	*dummy*$_2$
$\bar{r}_3$	$\bar{\text{b}}$	ulcer
$\bar{d}_1$	$\bar{\text{b}}$	*dummy*$_1$
m_2	x	*marker*$_2$
d_3	x	*dummy*$_3$

Client

J^*

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3
l_2	b	Beth	b	*dummy*$_1$	d_1
$\bar{l}_2$	$\bar{\text{b}}$	Beth	$\bar{\text{b}}$	ulcer	$\bar{r}_3$
$\bar{l}_2$	$\bar{\text{b}}$	Beth	$\bar{\text{b}}$	*dummy*$_1$	d_1
m_1	x	*marker*$_1$	x	*marker*$_2$	m_2
m_1	x	*marker*$_1$	x	*dummy*$_3$	d_3

Computational cloud

$R_{l_k}^*$

I_k	L.Tuple_k
α	λ_1
α'	λ_1'
β	λ_2
β'	λ_2'
γ	λ_3
γ'	λ_3'
$\bar{\beta}$	$\bar{\lambda}_2$
$\bar{\beta}'$	$\bar{\lambda}_2'$
χ	μ_1

$R_{r_k}^*$

I_k	R.Tuple_k
α	ρ_1
α	ρ_2
β	ρ_3
β	δ_1
ε	ρ_4
ε	ρ_5
ε'	ρ_6
ε'	δ_2
$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\delta}_1$
χ	μ_2
χ	δ_3

J_k^*

L.I_k	L.Tuple_k	R.I_k	R.Tuple_k
α	λ_1	α	ρ_1
α	λ_1	α	ρ_2
β	λ_2	β	ρ_3
β	λ_2	β	δ_1
$\bar{\beta}$	$\bar{\lambda}_2$	$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\lambda}_2$	$\bar{\beta}$	δ_1
χ	μ_1	χ	μ_2
χ	μ_1	χ	δ_3

Join execution – Example

$R_l{}^*$

	I	Attr
l_1	a	Ann
$l_1{}'$	a′	Ann′
l_2	b	Beth
$l_2{}'$	b′	Beth′
l_3	c	Cloe
$l_3{}'$	c′	Cloe′
$\bar{l}_2$	b̄	Beth
$\bar{l}_2{}'$	b̄′	Beth′
m_1	x	*marker*$_1$

$R_r{}^*$

	I	Attr
r_1	a	flu
r_2	a	asthma
r_3	b	ulcer
d_1	b	*dummy*$_1$
r_4	e	hernia
r_5	e	flu
r_6	e′	cancer
d_2	e′	*dummy*$_2$
$\bar{r}_3$	b̄	ulcer
$\bar{d}_1$	b̄	*dummy*$_1$
m_2	x	*marker*$_2$
d_3	x	*dummy*$_3$

Storage servers

J^*

	L.I	L.Attr	R.I	R.Attr	
l_1	a	Ann	a	flu	r_1
l_1	a	Ann	a	asthma	r_2
l_2	b	Beth	b	ulcer	r_3

Client

$R_{l_k}^*$

$\mathbf{I_k}$	$\mathbf{L.Tuple_k}$
α	λ_1
α'	$\lambda_1{}'$
β	λ_2
β'	$\lambda_2{}'$
γ	λ_3
γ'	$\lambda_3{}'$
$\bar{\beta}$	λ_2
$\bar{\beta}'$	$\bar{\lambda}_2{}'$
χ	μ_1

$R_{r_k}^*$

$\mathbf{I_k}$	$\mathbf{R.Tuple_k}$
α	ρ_1
α	ρ_2
β	ρ_3
β	δ_1
ε	ρ_4
ε	ρ_5
ε'	ρ_6
ε'	δ_2
$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\delta}_1$
χ	μ_2
χ	δ_3

J_k^*

$\mathbf{L.I_k}$	$\mathbf{L.Tuple_k}$	$\mathbf{R.I_k}$	$\mathbf{R.Tuple_k}$
α	λ_1	α	ρ_1
α	λ_1	α	ρ_2
β	λ_2	β	ρ_3
β	λ_2	β	δ_1
$\bar{\beta}$	$\bar{\lambda}_2$	$\bar{\beta}$	$\bar{\rho}_3$
$\bar{\beta}$	$\bar{\lambda}_2$	$\bar{\beta}$	δ_1
χ	μ_1	χ	μ_2
χ	μ_1	χ	δ_3

Computational cloud

Markers and twins: Integrity guarantees

- The guarantee offered by markers and twins can be measured as the probability of the computational cloud to go undetected when omitting tuples

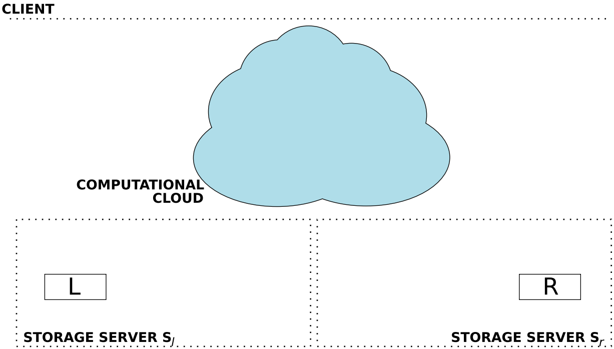
- Markers and twins offer complementary protection:
  - Twins are twice as effective as markers, but loose their effectiveness when the computational cloud omits a large fraction of tuples (extreme case: all tuples omitted)
  - Markers allow detecting extreme behavior (all tuples omitted) and provide effective when the computational cloud omits a large fraction of tuples

Semi-join execution strategy – 1

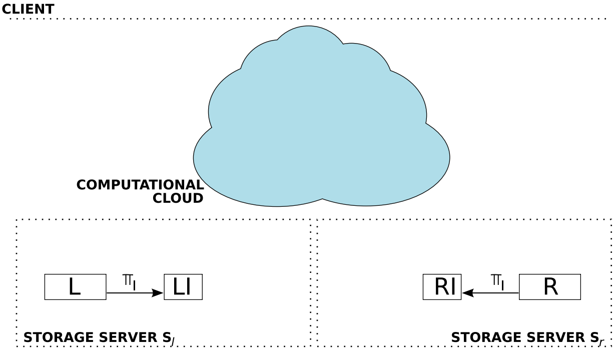
- Salts and buckets introduce computation and communication overhead

- Semi-join execution strategy [DFJLPS-14b]
  - protect the join profile without the need of introducing salts and buckets
  - support one-to-one, one-to-many, many-to-many joins and join sequences

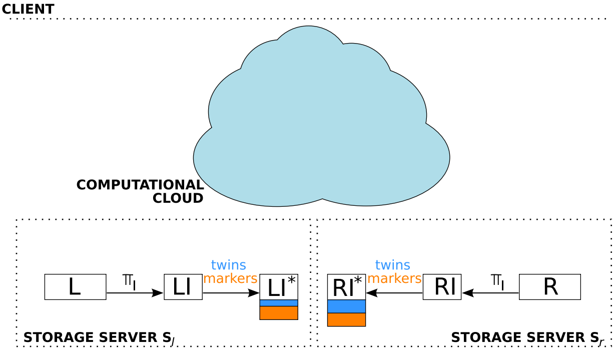
Semi-join execution strategy – 2



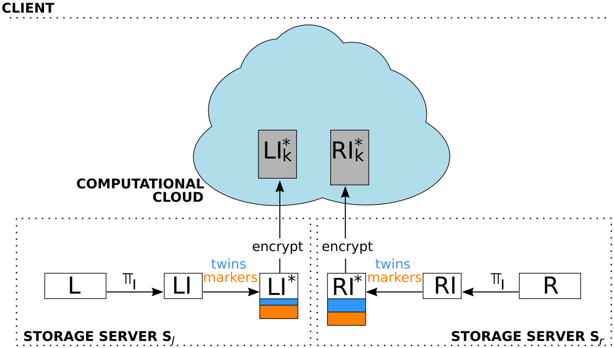
Semi-join execution strategy – 2



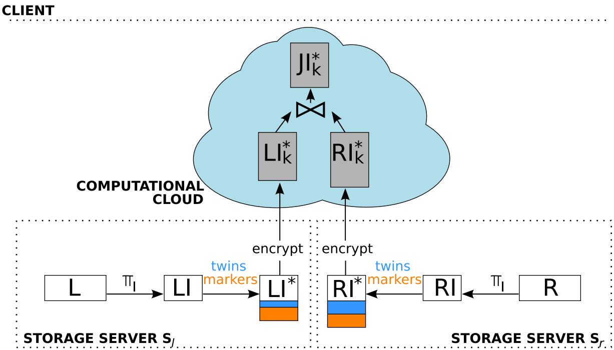
Semi-join execution strategy – 2



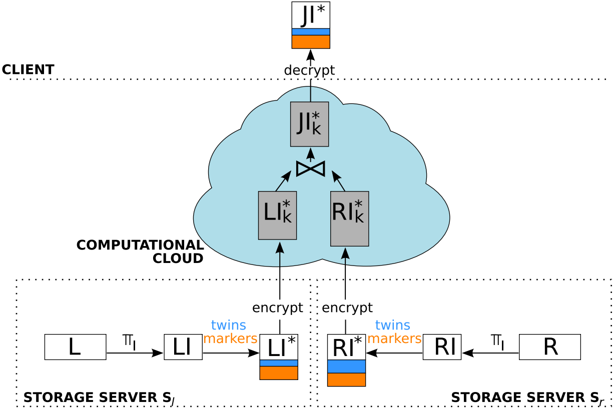
Semi-join execution strategy – 2



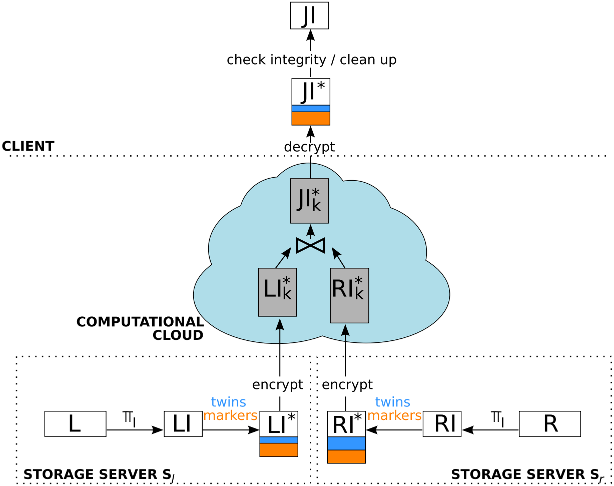
Semi-join execution strategy – 2



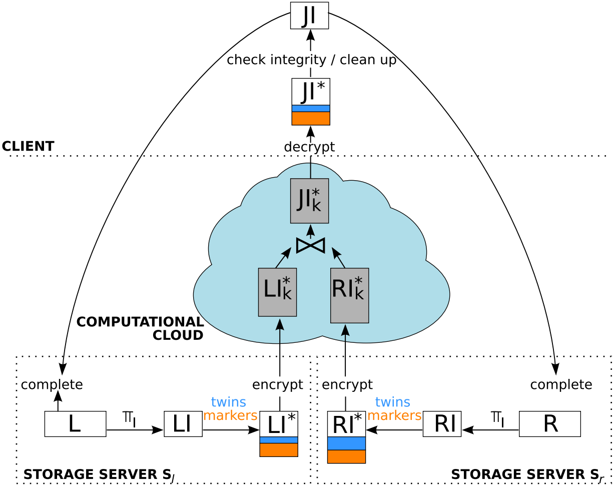
Semi-join execution strategy – 2



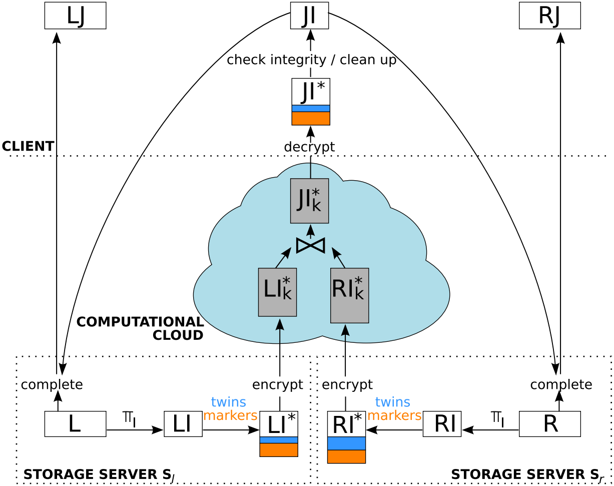
Semi-join execution strategy – 2



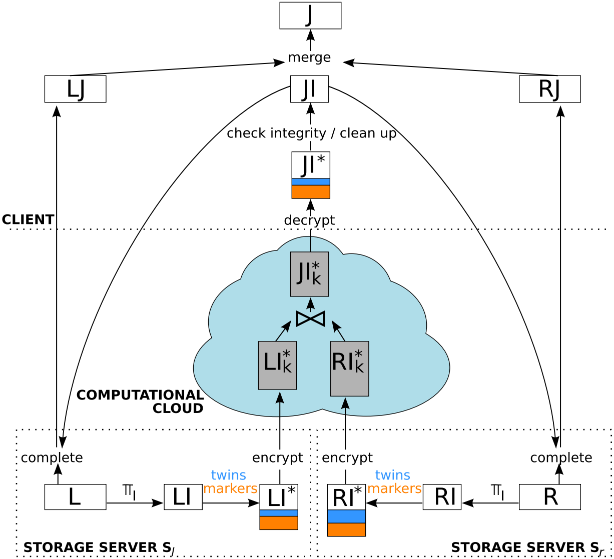
Semi-join execution strategy – 2



Semi-join execution strategy – 2



Semi-join execution strategy – 2



Distributed computational cloud

Some computational cloud scenarios support the processing of a vast amount of data in parallel on a large number of nodes (e.g., MapReduce)

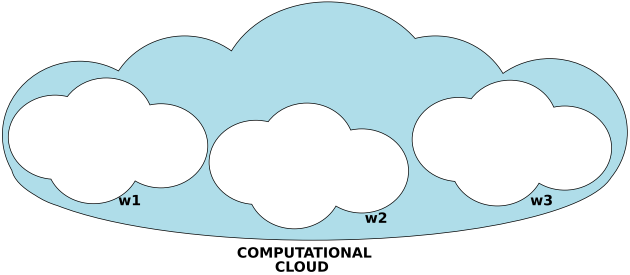
- need to reason about different nodes involved in the enforcement of integrity controls and ensure
  - control is well distributed among different nodes
  - ability to recognize misbehaving nodes (accountability)

Computational cloud working: MapReduce

A MapReduce framework supports execution of tasks over large amount of data in parallel by multiple nodes (worker), coordinated by a manager

- A user-defined map function translates the input (tuples to be joined) in a set of pairs $\langle \mathit{key}, \mathit{value} \rangle$
- An assignment function f assigns pairs $\langle \mathit{key}, \mathit{value} \rangle$ to workers $\Longrightarrow$ all pairs with the same *key* go to the same worker $w = f(\mathit{key})$
- A user-defined reduce function (join operation) is executed by each worker, and the result is then combined by the manager

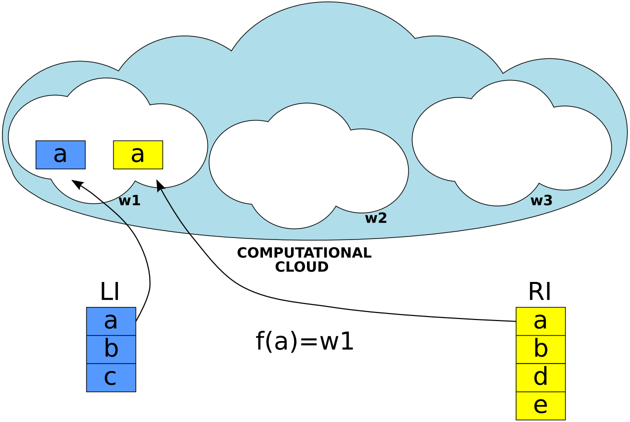
Computational cloud working: MapReduce – example



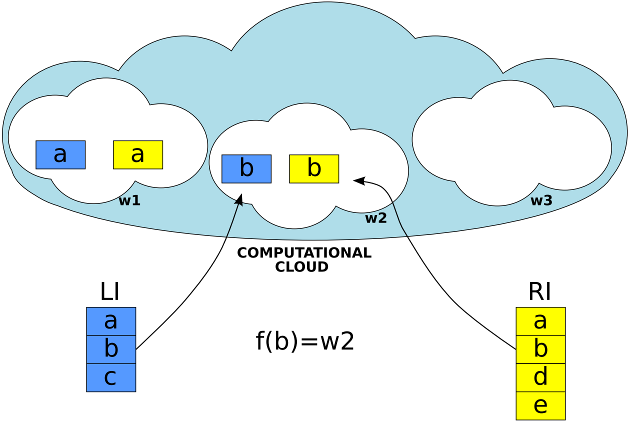
LI
a
b
c

RI
a
b
d
e

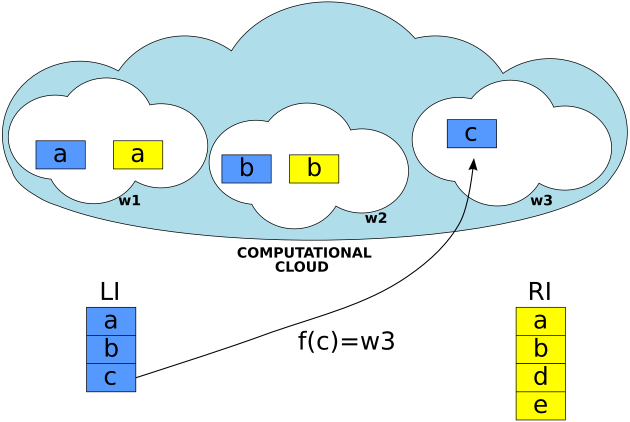
Computational cloud working: MapReduce – example



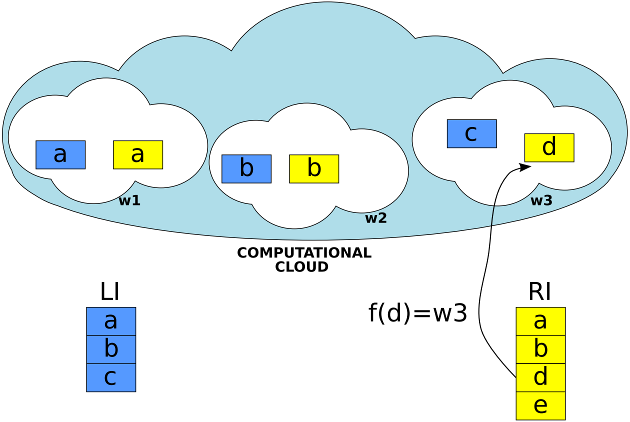
Computational cloud working: MapReduce – example



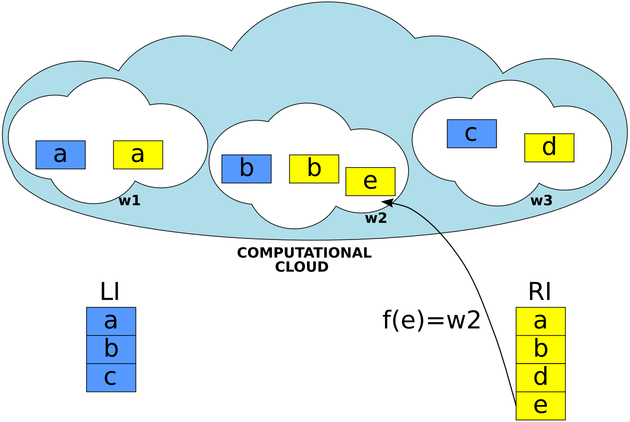
Computational cloud working: MapReduce – example



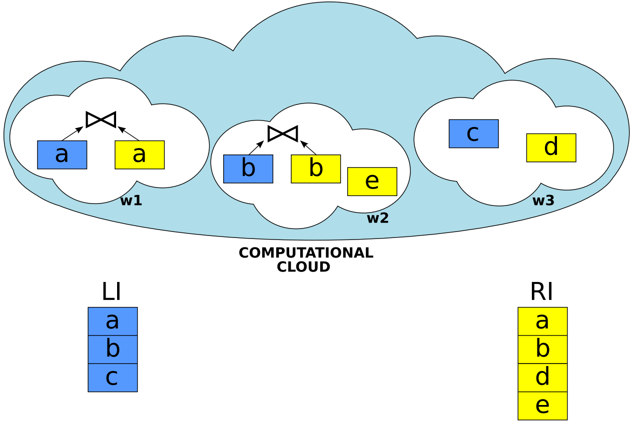
Computational cloud working: MapReduce – example



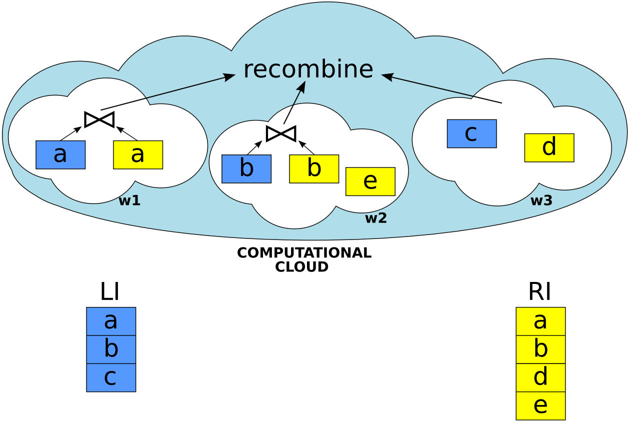
Computational cloud working: MapReduce – example



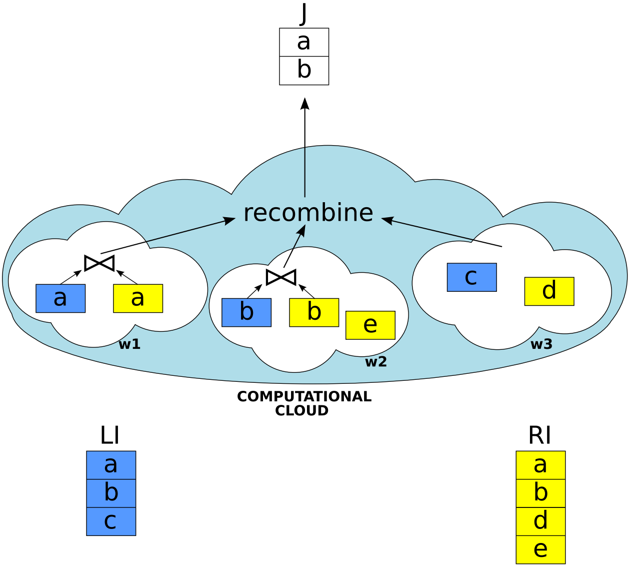
Computational cloud working: MapReduce – example



Computational cloud working: MapReduce – example



Computational cloud working: MapReduce – example



On-the-fly encryption

Encryption is applied to the join attribute of the relations involved in the join before they are passed to the computational cloud

- Every storage server encrypts its relation B obtaining $B^k(I_k)$, with I the join attribute of B
  - for each distinct $t[I]$ in B, there is τ in $B^k : \tau[I_k] = E_k(t[I])$
  - E is a symmetric encryption function with key k
  - k is defined by the client and changes at every query
- Encrypted values are translated into pairs $\langle$*key*,*value*$\rangle$ of the form $\langle \tau[I^k], B^k \rangle$
  - tuples with the same values for the join attribute are assigned to the same worker
  - no tuple is missed from the join due to an improper allocation

Markers and MapReduce (1)

- Markers should be properly distributed among all l workers (to distribute control)

- Marker distribution strategy $\langle N, N_{min}, N_{max} \rangle$ with N the total number of markers, N_{min}/N_{max} the minimum/maximum number of markers per worker
  - random $\langle N, 0, N \rangle$: no condition on the distribution of markers to workers
  - at-least-n $\langle N, n, n + (N - n \cdot l) \rangle$: every worker must receive at least n markers ($n \leq \lfloor N/l \rfloor$)
  - perfect balance $\langle N, \lfloor N/l \rfloor, \lceil N/l \rceil \rangle$: markers should be distributed evenly (the number of markers at any pair of workers can differ by at most one)

Markers and MapReduce (2)

All storage servers generate markers with function μ set by the client

```
Generate_Markers($N, N_{min}, N_{max}$)
spare := $N - (N_{min} * l)$ /* spare markers */
repeat
    generate a new marker $m$ via function $\mu$
    let $w$ be $f(E_k(m))$ and $n_w$ be the number of markers already assigned to it
    if $(n_w < N_{min})$ or $(n_w < N_{max}$ and spare $> 0)$
    then retain $m$
            $n_w := n_w + 1$
            if $n_w > N_{min}$ then spare := spare $- 1$
    else discard $m$
until $N$ markers have been allocated
```

Markers and MapReduce (3)

- Every storage server generates the same set of markers
  - each server produces the same sequence of markers
  - allocation of markers to workers is deterministic
- The generated markers are correct:
  - for each worker w: $N_{min} \leq n_w \leq N_{max}$
  - the total number of assigned markers is N

Twins and MapReduce (1)

- Twins also should be properly distributed on the different workers
- Controlling twin generation like for markers is not possible
  $\Longrightarrow$ twins depend on the join attribute values at each server
  - each server can twin different tuples depending on its instance
  - each server can observe a different number of twins for a worker
  - servers cannot coordinate to regulate twin distributions
- Twin separation: a twin cannot be assigned to the same worker as its original tuple
  - property on which all servers have the same visibility
  - two-man-rule: a worker missing t would be exposed by the presence of $\bar{t}$ on a different worker (and viceversa)

Twins and MapReduce (2)

Storage servers twin tuples based on condition C_{twin} and a salt generating function σ set by the client

```
Generate_Twins($B$, $C_{\text{twin}}$)
for each $t$ in $B$ satisfying condition $C_{\text{twin}}$ do
    let $w$ be $f(E_k(t[I]))$
    repeat
        generate salt $s$ via function $\sigma$
        $\bar{t}$ := $t$
        let $\bar{w}$ be $f(E_k(\bar{t}[I] \oplus s))$
    until $\bar{w} \neq w$
```

Twins and MapReduce (2)

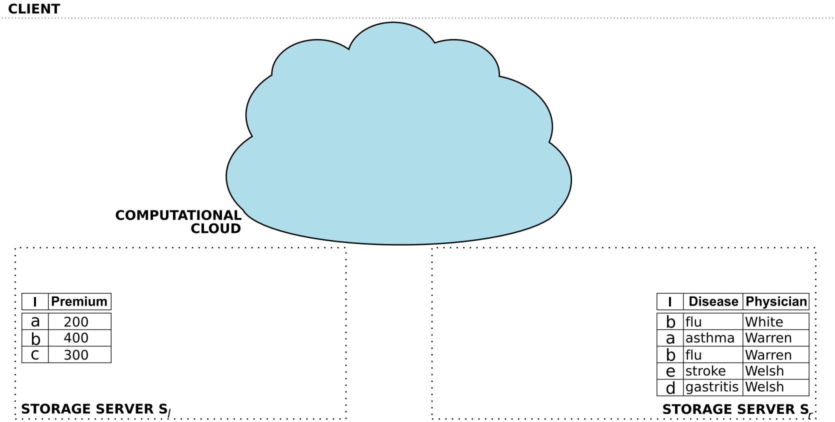
Storage servers twin tuples based on condition C_{twin} and a salt generating function σ set by the client

```
Generate_Twins($B$, $C_{\text{twin}}$)
for each $t$ in $B$ satisfying condition $C_{\text{twin}}$ do
    let $w$ be $f(E_k(t[I]))$
    repeat
        generate salt $s$ via function $\sigma$
        $\bar{t} := t$
        let $\bar{w}$ be $f(E_k(\bar{t}[I] \oplus s))$
    until $\bar{w} \neq w$
```

Twin pairs are guaranteed to participate in the join result

- all servers generate twins with the same generation function
- allocation of twins to workers is deterministic

Join overall execution – example



CLIENT

COMPUTATIONAL CLOUD

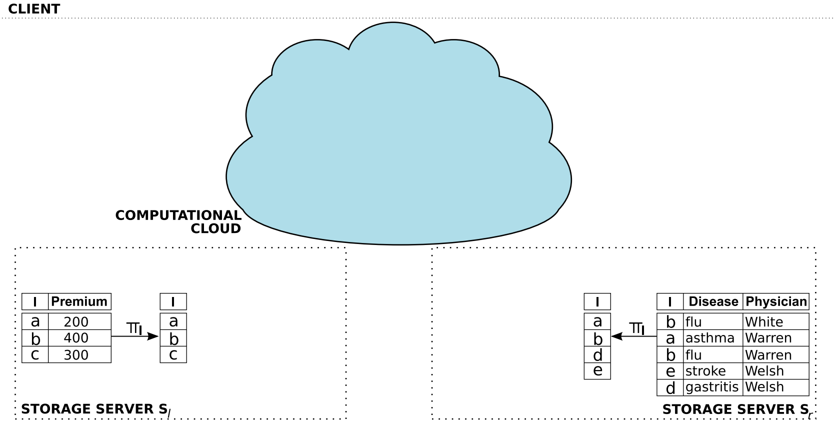
I	Premium
a	200
b	400
c	300

STORAGE SERVER S_l

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

STORAGE SERVER S_r

Join overall execution – example



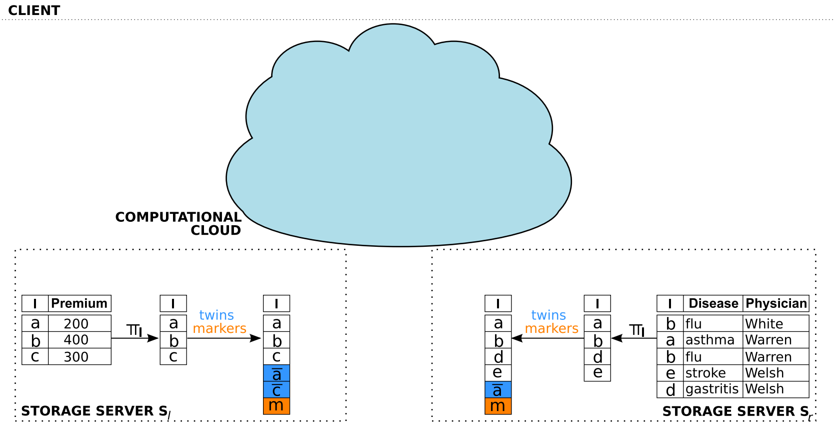
I	Premium
a	200
b	400
c	300

I
a
b
c

I
a
b
d
e

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

Join overall execution – example



I	Premium
a	200
b	400
c	300

I
a
b
c

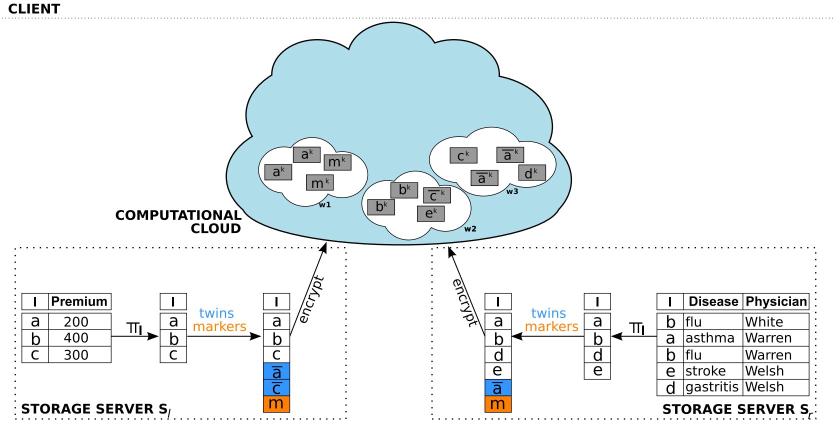
I
a
b
c
$\bar{a}$
$\bar{c}$
m

I
a
b
d
e
$\bar{a}$
m

I
a
b
d
e

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

Join overall execution – example



I	Premium
a	200
b	400
c	300

I
a
b
c

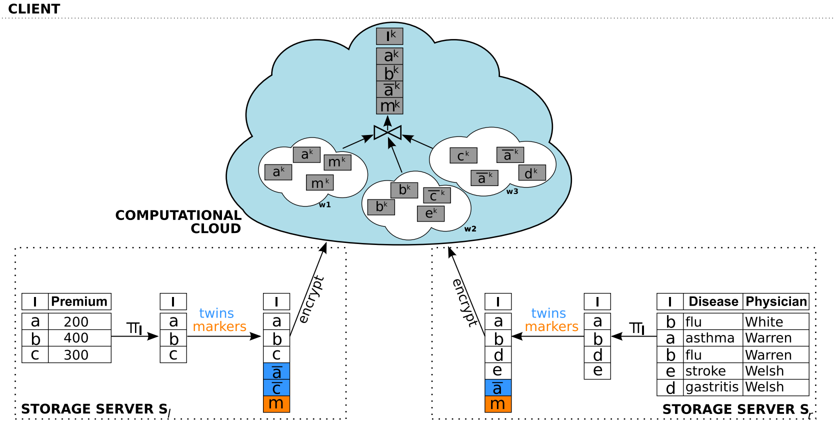
I
a
b
c
$\overline{a}$
$\overline{c}$
m

I
a
b
d
e
$\overline{a}$
m

I
a
b
d
e

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

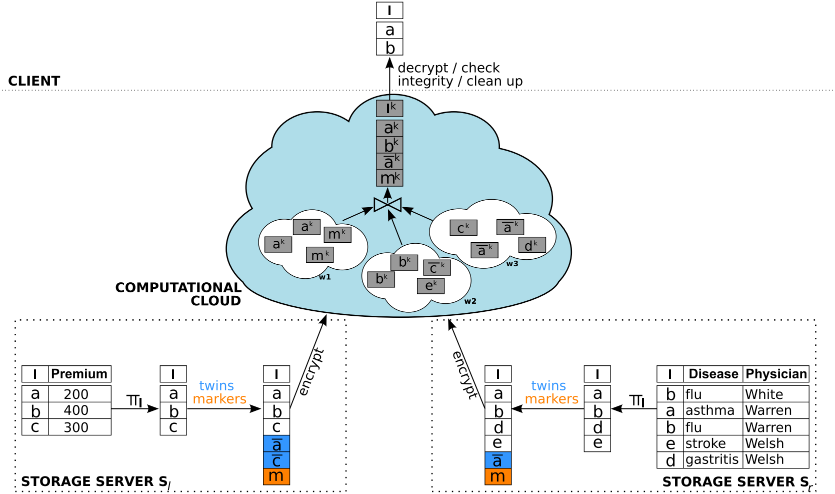
Join overall execution – example



I	Premium
a	200
b	400
c	300

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

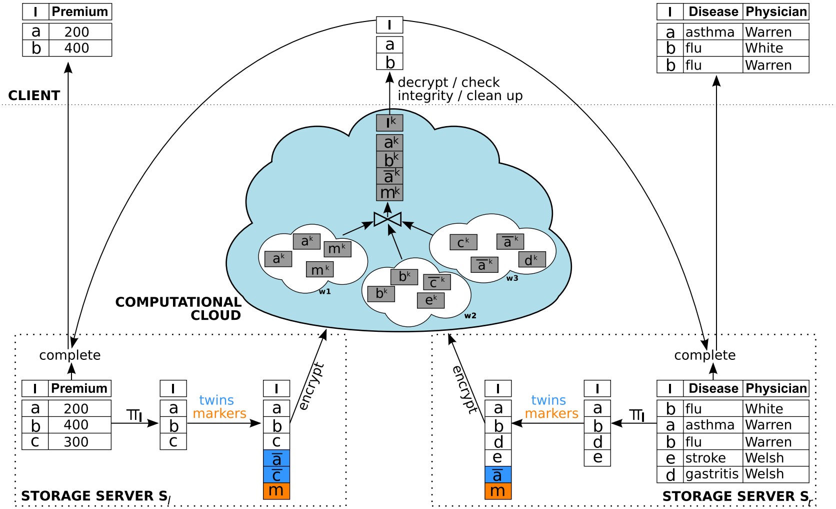
Join overall execution – example



I	Premium
a	200
b	400
c	300

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

Join overall execution – example



I	Premium
a	200
b	400

I
a
b

I	Disease	Physician
a	asthma	Warren
b	flu	White
b	flu	Warren

I^k
a^k
b^k
$\overline{a}^k$
m^k

a^k m^k a^k m^k

b^k $\overline{c}^k$ b^k e^k

c^k $\overline{a}^k$ $\overline{a}^k$ d^k

I	Premium
a	200
b	400
c	300

I
a
b
c

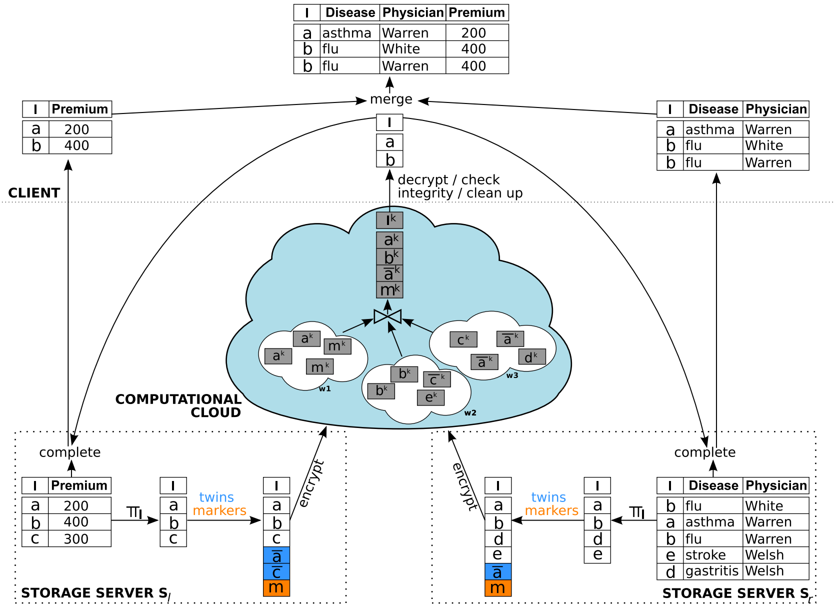
I
a
b
c
$\overline{a}$
$\overline{c}$
m

I
a
b
d
e
$\overline{a}$
m

I
a
b
d
e

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

Join overall execution – example



I	Disease	Physician	Premium
a	asthma	Warren	200
b	flu	White	400
b	flu	Warren	400

I	Premium
a	200
b	400

I
a
b

I	Disease	Physician
a	asthma	Warren
b	flu	White
b	flu	Warren

I	Premium
a	200
b	400
c	300

I
a
b
c

I
a
b
c
$\overline{a}$
$\overline{c}$
m

I
a
b
d
e
$\overline{a}$
m

I
a
b
d
e

I	Disease	Physician
b	flu	White
a	asthma	Warren
b	flu	Warren
e	stroke	Welsh
d	gastritis	Welsh

Variations/open issues …

- Execution of a join as a semi-join to support n:m joins and protect join profile [DFJPS-14]

- Application of the techniques to only a portion of the data (verification object) [DFJPS-14]

- Consideration of different trust levels

- Removal of trust assumptions in the storage servers

References – 1

- [CKGS-98] B. Chor, E. Kushilevitz, O. Goldreich, M. Sudan, “Private Information Retrieval,” in *JACM*, vol. 45, no. 6, 1998.
- [DFJLPS-14] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, G. Livraga, S. Paraboschi, P. Samarati, “Integrity for Distributed Queries,” in *Proc. of CNS*, San Francisco, CA, USA, October 2014.
- [DFJPS-14] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, “Optimizing Integrity Checks for Join Queries in the Cloud,” in *Proc. of DBSec*, Vienna, Austria, July 2014.
- [DFJPS-13] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, "Integrity for Join Queries in the Cloud," in IEEE Transactions on Cloud Computing (TCC), vol. 1, n. 2, July-December 2013, pp. 187-200.
- [DFPPS-11a] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, P. Samarati, “Efficient and Private Access to Outsourced Data,” in *Proc. of ICDCS*, Minneapolis, MN, USA, June 2011.
- [DFPPS-11b] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, P. Samarati, “Supporting Concurrency in Private Data Outsourcing,” in *Proc. of ESORICS*, Leuven, Belgium, September 2011.

References – 2

- [DFPPS-13] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, P. Samarati, “Distributed Shuffling for Preserving Access Confidentiality,” in *Proc. of ESORICS*, Egham, U.K., September 2013.
- [DGMS-00] P.T. Devanbu, M. Gertz, C.U. Martel, S.G. Stubblebine, “Authentic third-party data publication,” in *Proc. of DBSec*, Schoorl, The Netherlands, August 2000.
- [GT-00] M. Goodrich, R. Tamassia, “Efficient Authenticated Dictionaries with Skip Lists and Commutative Hashing’,” in Technical Report, Johns Hopkins Information, 2000.
- [LC-04] P. Lin, K.S. Candan, “Hiding Traversal of Tree Structured Data from Untrusted Data Stores,” in *Proc. of WOSIS*, Porto, Portugal, April 2004.
- [NT-05] M. Narasimha, G. Tsudik, “DSAC: Integrity for Outsourced Databases with Signature Aggregation and Chaining,” in *Proc. CIKM*, Bremen, Germany, October-November 2005.
- [PPP-10] B. Palazzi, M. Pizzonia, S. Pucacco, “Query Racing: Fast Completeness Certification of Query Results,” in *Proc. DBSEC*, Rome, Italy, June 2010.

References – 3

- [RFKSSvD-15] L. Ren, C. Fletcher, A. Kwon, E. Stefanov, E. Shi M. van Dijk, S. Devadas, “Constants Count: Practical Improvements to Oblivious RAM” in *Proc. of USENIX*, Washington, USA, August 2015.
- [S-05] R. Sion, “Query Execution Assurance for Outsourced Databases,” in *Proc. of VLDB*, Trondheim, Norway, August-September 2005.
- [SC-07] R. Sion, B. Carbunar, “On the Computational Practicality of Private Information Retrieval,” in *Proc. of NDSS*, San Diego, CA, USA, February/March 2007.
- [SVSFRYD-13] E. Stefanov, M. van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, S. Devadas, “ Path ORAM: An Extremely Simple Oblivious RAM Protocol,” in *Proc. of CCS*, Berlin, Germany, November 2013.
- [WSC-08] P. Williams, R. Sion, B. Carbunar, “Building Castles Out of Mud: Practical Access Pattern Privacy and Correctness on Untrusted Storage,” in *Proc of CCS*, Alexandria, USA, October 2008.
- [WS-12] P. Williams, R. Sion, “Single Round Access Privacy on Outsourced Storage,” in *Proc. of CCS*, Raleigh, NC, USA, October 2012.
- [WYPY-08] H. Wang, J. Yin, C. Perng, P. Yu, “Dual Encryption for Query Integrity Assurance,” in *Proc. of CIKM*, Napa Valley, CA, USA, October 2008.

References – 4

- [XWYM-07] M. Xie, H. Wang, J. Yin, X. Meng, "Integrity Auditing of Outsourced Data," in *Proc. of VLDB*, Vienna, Austria, September 2007.