

Datenbank-Grundlagen

Kapitel 10: Privacy-Schutz in relationalen Datenbanken

Prof. Dr. Stefan Böttcher
Universität Paderborn

Agenda:

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 1 / 36

Zugriffskontrolle und Privacy

Zugriffskontrolle

→ regelt, wer bestimmte Daten zugreifen darf, und wer nicht

 auf Datei-Ebene → Zugriffsrechte für bestimmte Nutzer(gruppen)


Wichtiger Spezialfall:
zugreifbare und nicht zugreifbare Daten sind vermischt
Zugriffskontrollsystem nötig (in SQL-Datenbanken vorhanden)


Privacy - Verletzungskontrolle
→ kontrolliert, wenn Zugriff gewährt wurde,
ob Daten unzulässig an Dritte weitergegeben werden konnten

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 2 / 36

Datenbank-Zugriffskontrolle - Aufgabe

Benutzer 1
Datenbank-Zugriffs-Programm

user	password	Zugriffsrecht
uid	pw	Z 



verbotene Daten geschützt

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 3 / 36

Zugriffskontrolle – warum in SQL?

SQL ist flexibler als

zusätzliche Felder für Tupel, die sagen, für wen Zugriff erlaubt ist, weil


1. Alle pro Tupel gespeicherten Rechte auch in SQL formulierbar sind
2. Rechte zudem über Views inhaltsbezogen definiert werden können

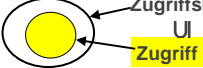
```
create view BluttestPB as
select * from Bluttest where ort = 'Paderborn'
```

```
Grant select on BluttestPB to XYZ ;
```

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 4 / 36

Zugriffskontrolle – verworfene Lösung

 Zugriffsschutz durch **Beweis:**



Zugriffsrecht

u


Zugriff

- langsam/teuer
- im allgemeinen nicht trivial zu berechnen, weil der Zugriff auch qualifizierende Tupel umfasst, also solche Tupel, deren Existenz (oder nicht-Existenz) das Ergebnis beeinflusst, obwohl sie nicht selektiert wurden

$\{ t \in R \mid \text{not } \exists u \in S (t.a = u.b) \}$ liest auch Teile von Schema(S), nämlich $\{ u \in \text{Schema}(S) \mid \exists t \in R (t.a = u.b) \}$

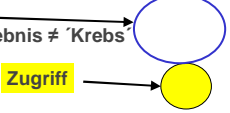
Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 5 / 36

Zugriffskontrolle – wie nicht?

 „Zugriffsschutz“ durch Test und Meldung von Verletzungen

```
select * from Bluttest
where Name = 'Alice'
```

„Sie haben Ihr Zugriffsrecht verletzt“



```
select * from Bluttest where Ergebnis ≠ 'Krebs'
```

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 6 / 36

Zugriffskontrolle – wie?

„Zugriffsschutz“ durch Modifikation der Query

select * from **Bluttest**
where Name = 'Alice'

objektiv falsch = subjektiv richtig !

select * from
(select * from **Bluttest**
where **Ergebnis ≠ Krebs**)
where Name = 'Alice'

Zugriffsrecht →

select * from Bluttest where Ergebnis ≠ 'Krebs' →

Zugriff →

**Query-Modifikation:
Relation ersetzen durch View für das Zugriffsrecht**

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 7 / 36

Datenbank-Zugriffskontrolle - Architektur

Zugriffskontrollsystem ändert: Anfrage → eingeschränkte Anfrage

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 8 / 36

Motivation für Privacy-Schutz

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 9 / 36

Privacy - Informationslecks finden (2)

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 10 / 36

Ziel

DBS zeichnet auf *wer wann was mit welchen* Daten gemacht hat

Lösungsvorschlag: alle Ergebnistupel aufzeichnen.

- ⊕ hoher Speicherbedarf
- ⊕ Informationsgewinn nicht aus Ergebnis ablesbar

Besser: Anfragen aufzeichnen

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 11 / 36

Privacy - Sicherheits-Architektur

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 12 / 36

Hintergrund-Datenbank

Hintergrund-Datenbank der Privacy-Sicherheitsarchitektur

- speichert von wann bis wann Datensätze gültig sind
- markiert gelöschte Datensätze als bis zur aktuellen Zeit gültig
- markiert eingefügte Datensätze als ab aktueller Zeit gültig.

Originaltabelle			Gültigkeitsintervall	
Name	...	Bluttest	Startzeit	Endzeit
Eva Stark	...	o.K.	3.5.2001	10.6.2005
...
Herbert Meier	...	Krebs	10.6.2005	'null'

← gelöscht
← eingefügt

Zugriffskontrolle - Privacy Protection - Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 13 / 36

Aufdecken von Privacy-Verletzungen

Zugriffskontrolle - Privacy Protection - Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 14 / 36

Privacy-Verletzungen aufdecken

Zugriffskontrolle - Privacy Protection - Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 15 / 36

Anfrage-Protokoll

- als Tabelle organisiert
- speichert alle Anfragen

Benutzer	Zeit	SQL-Statement

Zugriffskontrolle - Privacy Protection - Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 16 / 36

Audit-Anfrage : Beispiel

Alice möchte wissen, ob ihr Name **und** ihre Adresse **zusammen mit** ihrem Diabetes-Risiko weitergegeben wurden.

```

DURING t1 TO t2
AUDIT name, adresse, diagnose
FROM Bluttest
WHERE name = 'Alice'

```

Zugriffskontrolle - Privacy Protection - Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 17 / 36

von Audit-Anfrage gehütetes Geheimnis : Beispiel

Der Audit-Ausdruck selektiert folgende DB-Feld-Kombination

name	adresse	diagnose
Alice	Paderborn			Diabetis

Das dadurch beschriebene **Geheimnis ist die Kombination** aus Name='Alice', Adresse='Paderborn' und Diagnose='Diabetis'

Zugriffskontrolle - Privacy Protection - Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 18 / 36

Unentbehrliches Tupel - Idee

- Ein (virtuelles) Tupel ist unentbehrlich für eine Anfrage Q, wenn das Weglassen des Tupels das Ergebnis von Q verändern würde
- Ein (virtuelles) Tupel ist unentbehrlich für eine Audit Query A (also wesentlich für das durch A beschriebene Geheimnis) wenn das Weglassen des Tupels das Ergebnis von A verändern würde
- Ein für das durch A beschriebene Geheimnis wesentliches Tupel t, beeinflusst das Ergebnis von Q, wenn t für A und für Q unentbehrlich ist

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 19 / 36

Unentbehrliches Tupel

$\pi_{..}$ = Duplikate entfernt $\bar{\pi}_{..}$ = Duplikate behalten

- Ein (virtuelles) Tupel $v \in T$ ist unentbehrlich für eine Anfrage $Q = \pi_{CQ}(\sigma_{PQ}(T))$, wenn das Weglassen von v das Ergebnis von Q verändern würde:

$$\text{unentbehrlich}(v, Q) \Leftrightarrow \pi_{CQ}(\sigma_{PQ}(T)) \neq \pi_{CQ}(\sigma_{PQ}(T - \{v\}))$$

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 20 / 36

Unentbehrlichkeit - DISTINCT

$\pi_{..}$ = Duplikate entfernt $\bar{\pi}_{..}$ = Duplikate behalten

- Ein (virtuelles) Tupel $v \in T$ ist unentbehrlich für eine Anfrage $Q = \pi_{CQ}(\sigma_{PQ}(T))$, wenn es unentbehrlich für $Q' = \bar{\pi}_{CQ}(\sigma_{PQ}(T))$ ist.

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 21 / 36

Unentbehrlichkeit - Beispiel

Student: hört: Kurs:

Name	MNr
Alice	123456

MNr	KNr
123456	0001
123456	0002

KNr	Titel	Semester
0001	Mod	1
0002	SWE	1

$Q = \pi_{S.Name}(\sigma_{K.Semester=1}(S |X| h |X| K))$ $\pi_{..}$ = Duplikate entfernt

$t_1 = (0002, SWE, 1)$ unentbehrlich für Q ?

Ja, denn t_1 ist unentbehrlich für Q'

$Q' = \bar{\pi}_{S.Name}(\sigma_{K.Semester=1}(S |X| h |X| K))$ $\bar{\pi}_{..}$ = Duplikate behalten

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 22 / 36

Unentbehrlichkeit – Aggregation

- Ein (virtuelles) Tupel $v \in T$ ist unentbehrlich für eine Anfrage Q mit Aggregation, wenn es unentbehrlich für die aggregationslose Version Q' ist.

Q =

```
SELECT name, avg(duration)
FROM Doctor d, Treatment t
WHERE d.did=t.did AND
disease='diabetes'
GROUP BY name
```

Q' =

```
SELECT name, duration
FROM Doctor d, Treatment t
WHERE d.did=t.did AND
disease='diabetes'
```

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 23 / 36

Audit-Anfrage : Beispiel

Alice möchte wissen, ob ihr Name **und** ihre Adresse **zusammen mit** ihrem Diabetes-Risiko weitergegeben wurden.

DURING t_1 TO t_2

AUDIT name, adresse, diagnose

FROM Bluttest

WHERE name = 'Alice'

Zugriffskontrolle Privacy Protection Vergleich mit externen Daten - Prof. Dr. Stefan Böttcher - SS 2005 24 / 36

von Audit-Anfrage gehütetes Geheimnis : Beispiel

Der Audit-Ausdruck selektiert folgende DB-Feld-Kombination

name	adresse	diagnose
Alice	Paderborn			Diabetis

Das dadurch beschriebene **Geheimnis** ist die **Kombination** aus Name='Alice', Adresse='Paderborn' und Diagnose='Diabetis'

Benutzer-Query : Beispiel

```
SELECT name, adresse
FROM Bluttest
WHERE diagnose='Diabetes' AND plz='33100'
```

name	adresse	plz	...	diagnose
Bob	Dörenhagen	33100		Diabetes
Alice	Paderborn	33100		Diabetes

Benutzer-Query ist Candidate Query - Bsp.

Benutzer-Query umfasst alle Spalten der Audit-Query

name	adresse	diagnose
Alice				

name	adresse	plz	...	diagnose
Bob	Dörenhagen	33100		Diabetes
Alice	Paderborn	33100		Diabetes

Candidate-Query

- Eine Anfrage Q ist eine *candidate query* zu einem gegebenen Audit-Ausdruck A , wenn Q auf alle Spalten C_Q zugreift, die als Spalten C_A in A vorkommen:

$$Q \text{ ist candidate Query für } A \Leftrightarrow C_A \subseteq C_Q$$

- Candidate Queries können leicht durch statische Tests ermittelt werden

Benutzer-Query ist verdächtig - Beispiel

- Benutzer-Query umfasst alle Spalten der Audit-Query
- Es gibt ein für beide Queries unentbehrliches Tupel

name	adresse	diagnose
Alice				

name	adresse	plz	...	diagnose
Bob	Dörenhagen	33100		Diabetes
Alice	Paderborn	33100		Diabetes

Benutzer-Query ist verdächtig - Beispiel

name	adresse	plz	...	diagnose
Bob		33100		Diabetes
Alice	Paderborn	33100		Diabetes

- Es gibt gemeinsames unentbehrliches Tupel:
→ candidate query ist verdächtig (=suspicious)

Verdächtige Anfrage

Eine candidate query Q ist verdächtig bei einem gegebenen Audit-Ausdruck A , wenn beide ein unentbehrliches (maximal virtuelles) Tupel v teilen:

$$\text{suspicious}(Q,A) \Leftrightarrow \overbrace{C_A \subseteq C_Q}^{Q \text{ ist candidate query}} \wedge \exists v : (\text{unentbehrlich}(v,Q) \wedge \text{unentbehrlich}(v,A))$$

Anfrage-Erzeugung

Existiert gemeinsames unentbehrliches Tupel?

- Selektionsbedingungen beider Anfragen verschachteln
- Sicherstellen, dass gleiche DB-Sicht zugrunde liegt
- Effizienz (kleinere Datenmenge für 2. Selektion)

Anfrage verdächtig, wenn Ergebnismenge *nicht* leer:

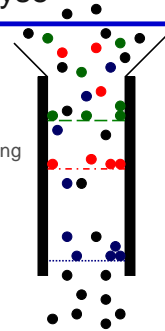
$$\sigma_A(\sigma_Q(T_\tau)) \neq \emptyset$$

Audit-Anfragen-Generator

- 3 Schritte:
 1. Statische Analyse
 - ermittelt candidate queries
 2. Zeitintervall-Prüfung
 3. Verdächtigkeits-(Suspicious-)Test
 - Kombiniert candidate query und audit query zu einzelner SQL-Anfrage

Statische Analyse

1. Candidate-Test:
Vergleich der Attribut-Namen:
 $C_A \subseteq C_Q$
 2. Zeitintervall-Test: DURING-Bedingung
 $\tau \in [t_1 ; t_2]$
 3. Widersprüche
→ es gibt kein gemeinsames unentbehrliches Tupel
- übrig bleiben: verdächtige Queries



Zusammenfassung

1. Anfragen aufzeichnen
2. DB-Zustände wiederherstellen
3. Zu prüfende Daten spezifizieren → Audit-Ausdruck
4. Audit-Anfrage generieren
 - Verdächtige Anfragen erkennen
 - Candidate queries ermitteln
 - Gemeinsames unentbehrliches Tupel finden

Offene Fragen / Ausblick

- Externe Verknüpfung von Anfrage-Ergebnissen
- Verknüpfungen mit externen Daten
- Manipulation von Anfrage-Protokoll und Hintergrund-DB