Übungsblatt der Lehrveranstaltung “Datenschutz in vernetzten Informationssystemen”

16. Juli 2013

Aufgabe 1: Gesellschaftliche Grundlagen *(10 Minuten)*

a) Wann sind Daten “personenbezogene Daten” nach den Kriterien des Bundesdatenschutzgesetzes?

b) Welcher Zusammenhang besteht zwischen Demokratie und Datenschutz?

c) Nennen und erläutern Sie drei der acht OECD-Empfehlungen für den Datenschutz.

Aufgabe 2: Anonymitätsmaße *(15 Minuten)*

a) Erläutern Sie drei Verfahren, mit denen eine Datenbank so verändert werden kann, dass sie der l-Diversity genügt.

b) Gegeben ist folgende Tabelle:

A	B	C	Sensibles Attribut
1	2	3	a
1	3	5	b
1	4	5	c
2	3	5	d
2	4	5	e
3	5	3	f
4	5	6	g
5	2	6	h

Markieren Sie in der folgenden Auflistung die Attribute bzw. Attributkombinationen, die Quasi-Identifier für diese Tabelle sind:

- A
- B
- C
- A,B
- A,C
- B,C
- A,B,C

c) Welche Datenbank lässt sich mit geringeren Veränderungen an den Daten in eine Datenbank überführen, die der l-Diversity genügt? Begründen Sie Ihre Antwort.

1. eine Datenbank, bei der jedes sensitive Attribut unterschiedlich ist

2. eine Datenbank, in der 20% aller sensitiven Attribute den Wert true *und alle anderen den Wert* false *haben*

Aufgabe 3: Datenschutz im Internet *(10 Minuten)*

a) Welche Informationen haben Sender und Empfänger über ihren Kommunikationspartner bei der Empfängeranonymität?

b) Wie funktioniert in TOR der Aufbau einer anonymen Verbindung (nur Senderanonymität)? Benennen Sie dazu die Protokollschritte sowie die Informationen, die der Client für die jeweiligen Schritte braucht.

c) Nennen und erläutern Sie einen Angriff gegen das TOR-Protokoll. Gehen Sie auf folgende Punkte ein:

1. Welche Ressourcen benötigt der Angreifer für diesen Angriff?
2. Welches Vorwissen braucht der Angreifer?
3. Was erfährt der Angreifer bei erfolgreichem Angriff?

Aufgabe 4: Lokationsbasierte Dienste *(10 Minuten)*

Beantworten Sie die folgenden Fragen zur Peer-to-Peer-Anonymisierung:

a) Welche Eingabedaten erhält die Peer-to-Peer-Anonymisierung von jedem Teilnehmer?

b) Welche Systemarchitektur verwendet das Verfahren?

c) Welche gemeinsamen Eigenschaften müssen Teilnehmer in einer Peer-Group aufweisen?

d) Welche Schritte durchläuft der Algorithmus zur Anonymisierung?

e) Nennen Sie drei Beispiele für Lokationsbasierte Dienste, für die das Verfahren besonders gut geeignet ist.

f) Nennen Sie drei Schwachstellen oder Angriffsmöglichkeiten des Verfahrens.

Aufgabe 5: Datenschutz in Datenbank-Szenarien *(15 Minuten)*

Im Folgenden geht es um die Bucketization.

a) Streichen Sie in der folgenden Liste die falschen Aussagen durch:

- Ein Angreifer beim Honest-but-Curious-Angreifermodell darf selbst Daten in die Datenbank einfügen, d.h., als Client auftreten.
- M:N-Mapping bei ordnungserhaltender Bucket-Kodierung schützt gegen Frequenzangriffe.
- Aggregatanfragen sind bei ordnungserhaltender Kodierung effizient möglich.
- Ein N:M-Mapping reduziert die Zahl der False Positives bei Exact-Match-Anfragen.
- Ein 1:1-Mapping bei der Bucketization unterstützt effiziente Verbundanfragen.
- Die Mengenoperation “Minus” kann bei M:N-Kodierung nicht korrekt auf den verschlüsselten Daten auf dem Server berechnet werden.

b) Erläutern Sie die Unterschiede zwischen ordnungserhaltender und nicht ordnungserhaltender Kodierung auf die Effizienz von Bereichsanfragen. Bei welcher Kodierung werden mehr false positives vom Server an den Client übertragen, und warum ist das so?

c) Gegeben ist folgende Tabelle:

Klartext-Tabelle db

Name	**Arzt**	**Diagnose**
Alice	Dr. Brown	Schnupfen
Alice	Dr. Brown	Schluckauf
Alice	Dr. White	Husten
Bob	Dr. Brown	Schnupfen
Bob	Dr. Red	Schnupfen
Carol	Dr. Red	Heiser
Dave	Dr. Brown	Husten
Eve	Dr. White	Heiser

Bucketization-Tabelle encdb

X	**Y**	**Z**
b	3	i
b	3	iii
b	1	iii
a	3	i
a	2	i
c	2	ii
c	3	iii
b	1	ii

(Anmerkung: die Reihenfolge der Tupel wurde bei der Bucketization beibehalten)

Kodieren Sie folgende Anfragen so um, dass sie der Dienstleister auf der mit Bucketization kodierten Tabelle anwenden kann. Geben Sie an, wieviele false positives Ihre Anfrage zurückliefert, und beschreiben Sie auftretende Probleme.

```
SELECT Arzt FROM db WHERE Name = ‘‘Dave’’
(Gib alle Ärzte aus, die Dave behandelt haben)

SELECT count(*) FROM db WHERE Diagnose = ‘‘Schluckauf’’
(Zähle, wie oft Schluckauf diagnostiziert wurde)

SELECT * FROM db WHERE Diagnose = ‘‘Husten’’
UNION SELECT * FROM db WHERE Name = ‘‘Bob’’
(Gib alle Datensätze aus, bei denen Diagnose=Husten oder Name=Bob)

SELECT Arzt, count(*) FROM db GROUP BY Arzt
(Zähle, wie oft jeder Arzt in der Datenbank vorkommt)
```