

תרגול נוסף של שאילתות מסוג GROUP BY, הצפנה - המשך

אלה התרגילים שניסינו בכתה (עם הפתרונות הפעם)

נתונה הטבלה הבאה: sales

OrderID	OrderDate	OrderPrice	OrderQuantity	CustomerName
1	12/22/2005	160	2	Smith
2	08/10/2005	190	2	Johnson
3	07/13/2005	500	5	Baldwin
4	07/15/2005	420	2	Smith
5	08/10/2005	210	3	Johnson
6	12/22/2005	1000	4	Wood
7	12/22/2005	820	4	Smith
8	11/03/2005	2000	2	Johnson

0. כתוב שאילתה שתמצא כמה כסף בזבז כל לקוח:

```
SELECT CustomerName, SUM(OrderPrice)
FROM Sales
GROUP BY CustomerName
```

1. כתוב שאילתה שתתן עבור כל לקוח (CustomerName) את סכום ההזמנות שלו עבור כל תאריך בו הזמין מוצר, בסדר יורד של סכום ההזמנות. התוצאה בהפעלת השאילתה על הטבלה הנ"ל תהיה:

CustomerName	OrderDate	Sum of orders
Johnson	11/03/2005	2000
Wood	12/22/2005	1000
Smith	12/22/2005	980
Baldwin	07/13/2005	500
Smith	07/15/2005	420
Johnson	08/10/2005	400

```
select CustomerName, OrderDate, sum(OrderPrice) as 'Sum of orders'
from sales
group by CustomerName, OrderDate
order by 3 desc
```

2. כתוב שאילתה שתתן עבור כל לקוח (CustomerName) את סכום מספר ההזמנות שלו עבור כל תאריך בו הזמין מוצר, בסדר עולה של סכום מספר ההזמנות.

```
select CustomerName, OrderDate, sum(OrderQuantity) as 'Quantity of orders'
from sales
group by CustomerName, OrderDate
order by 3
```

3. כתוב שאילתה שתתן עבור כל לקוח (CustomerName) את סכום מספר ההזמנות הכללי שלו.

```
select CustomerName, sum(OrderQuantity) as 'Quantity of orders'
```

```
from sales
group by CustomerName
```

4. כתוב שאילתה שתיתן עבור כל תאריך את סכום מחירי ההזמנות עבור אותו תאריך.

```
select OrderDate, sum(OrderPrice) as 'Total sum'
from sales
group by OrderDate
```

5. כתוב שאילתה שתתן את שמות הלקוחות שלהם מעל הזמנה אחת, יחד עם מספר ההזמנות של כל אחד. כל שורה בטבלה: sales נחשבת להזמנה אחת.

```
select CustomerName, count(*)
from sales
group by CustomerName
having count(*) > 1
```

הצפנה (המשך)

כדי לבצע עליו התקפה מסוג collision (כלומר למצוא 2 הודעות שונות שיתורגמו לאותו digest, נירשום אלגוריתם פשוט כלהלן: לייצר הרבה הודעות ולחפש עבורם diget שווה).

```
for (i=1; i<k; ++i) // k is the list of messages we create
{
  create message M[i]
  D[i] = h(M[i]); //D holds the digests, h is a hash function
  for (j=1; j<= i-1; ++j) // Check equality of previously generated digests
  {
    if (D[i] == D[j]) return M[i], M[j]
  }
}
return failed;
```

ע"מ לעשות את האלגוריתם יותר שימושי, אנחנו יכולים לייצר 2 הודעות בעלות משמעות ולהוסיף להן רווחים או דברים לא משמעותיים בצורה מתודית ולתת רשימה של : M1, M2,.... ושל: M'1, M'2,.... ואז להשתמש באלגוריתם הבא:

```
for (i=1; i<k; ++i) // k is the list of messages we create
{
  D[i] = h(M[i]);
  D'[i] = h(M'[i]);
  for (j=1; j<= i-1; ++j) // Check equality of previously generated digests
  {
    if (D[i] == D'[j]) return M[i], M'[j]
  }
}
```

```
}
return failed;
```

ההסתברות להצלחת האלגוריתם היא כמו במקרה של ימי ההולדת (המשתנה הראנדומלי הבלתי תלוי הם ה-digests שמיוצרים על ידי האלגוריתם). ולכן P בקירוב הוא: $1 - e^{-k(k-1)/2N}$

N הוא מספר האפשרויות עבור ה-digest ומכיוון שבביטים עסקינן, אם נניח שה-digest בעל n ביטים (16 בדוגמה הנ"ל), מספר הצירופים הוא: 2^n . ולכן מכאן נובע שמספר ה-digests שהאלגוריתם יצטרך לייצר (k) יהיה: $1.18 \times 2^{n/2}$ שזה 76 במקרה דנן.

מה זה אומר למשל עבור digest באורך של 64 ביטים? $k = 1.18 \times 2^{32}$ כדי לייצר מספר כזה של אפשרויות בהנחה שהמחשב יכול לייצר כמליון אפשרויות בשנייה, ייקח קצת יותר משעתיים. זה אומר ש-digest באורך 64 אינו ממש בטוח. ל-MD5 ששימשה הרבה זמן, היה digest באורך 128 ביטים - בהינתן מצב שאפשר לייצר כמליארד אפשרויות בשנייה (2^{30}) נזדקק ל- 2^{34} שניות (כי יש לנו לחשב בערך 2^{64} אפשרויות) וזה יותר מ-500 שנה. SHA-1 משתמשת ב-160 ביט (יותר מ-10,000 שנים לפצח בקצב הנ"ל) ו-SHA-512 בהחלט בטוחה (512 ביטים), כלומר 2^{256} אפשרויות כלומר 6.4×10^{60} שנים לפיצוח.

אז כעת, כאשר אנו יודעים שקיים אלגוריתם די בטוח לייצר digest שקשה לחקות אותו, וניראה איך זה יעזור לאמזון לזהות את עצמה.

קודם ציינו שהייתה לנו בעיה גם אם אמזון בחרה הודעה (x) וגם אם בוב בחר אותה (כיוון שמאלורי יכל לזייף) לכן צריכים שההודעה תהיה ראנדומלית. אפקט זה יושג על ידי פונקציה ההאש h , וכך יראה התהליך:

1. בוב בוחר את x כלשהו
2. בוב שולח את x לאמזון
3. אמזון מחשבת את $w = h(x)$ (h ידועה לכל)
4. אמזון מבצעת "פיענוח" של w בעזרת מפתחה הפרטי d כך: $y = D(d, w)$
5. אמזון שולחת את y לבוב
6. בוב מפעיל את המפתח הציבורי של אמזון (e) על y כך: $z = E(e, y)$
7. בוב בודק אם $z = h(x)$ אם זה מתקיים, הרי שבוב אכן קיבל הודעה מאמזון.

בכך השגנו מצב שבוב התחיל את התהליך עם הודעת דמה ראשונית אבל פונקציה ה- h יצרה את הדבר הראנדומלי שעליו חתמה אמזון (כלומר אף אחד מהצדדים לא יזם את הודעת הבדיקה)

בתאוריה, שיטה זו עובדת, אבל במציאות היא איטית מאוד. שיטת הצפנה סימטרית מהירה פי 1000 אבל לא בטוחה. מה עושים? משלבים את שניהם. אמזון תשתמש בקוד סימטרי, תצפין אותו עם הקוד הציבורי (א-סימטרי) של בוב, כך שרק בוב יוכל לפענח את הקוד הסימטרי. וכמו כן תשלח את ההודעה מוצפנת בעזרת הקוד הסימטרי. בוב יפענח את הקוד הסימטרי ובעזרתו יפענח את הודעה.

בנוסף לזיהוי המוחלט של מקור השולח, זו גם שיטה לחתום על מסמך (טביעת אצבע), משלוח של המסמך ביחד עם החתימה x ו- $h(x)$. כל אחד יכול שוב להפעיל את h ולהיווכח אם המסמך המקורי שונה או לא (כי המסמך והחתימה מגיעים ביחד).

שיעורי בית

לכתוב שאילתה שתיתן את רשימת מספרי זיהוי הספרים (isbn) של הספרים שנישאלו על ידי שלושה סטודנטים לפחות, מהספריות שנימצאות בניו יורק.