NATIONAL UNIVERSITY OF SINGAPORE

Semester 1, 2018/2019

CS5322 Database Security

Time Allowed: 2 Hours

INSTRUCTION TO CANDIDATES

1. This is a CLOSED book assessment.
2. This assessment paper contains **SIX (6)** questions and comprises THIRTEEN **(13)** printed pages.
3. Answer *ALL* questions within the spaces provided in this booklet.
4. You are allowed to use the back of the paper but please remember to state "P.T.O."
5. *Cross out any draft* or otherwise we will mark the poorer answers.
6. Please write your student number below, but NOT your name.
7. We reserved our rights to deduct marks if your writing is too untidy or unrecognizable.

STUDENT NUMBER:________________________

(This portion is for examiner's use only)

Question	Max. Marks	Score	Check
Q1	4		
Q2	3		
Q3	7		
Q4	7		
Q5	9		
Q6	10		
Total	40		

Question 1 (2 + 2 marks)

Suppose that we outsource the following two tables, Employees and Orders, to a service provider using CryptDB:

- Employees(EID, Name, Department), where the types of EID, Name, and Department are integer, string, and string, respectively.
- Orders(OID, SalesPerson, Price), where the types of OID, SalesPerson, and Price are integer, string, and string, respectively. In addition, SalesPerson is a foreign key referencing Employees.EID.

(a) State whether CryptDB can answer the following query:

- SELECT Department, SUM(Price) FROM Employees E, Orders O
  WHERE E.EID = O.SalesPerson AND Price > 100
  GROUP BY Department

If your answer is yes, then explain what encryption schemes should be used for the attributes involved. If your answer is no, then explain how CryptDB should be extended to support the query.

Answer:

Cannot cause E.EID is an integer and O.SalesPerson is a string.

- Maybe to allow this query, there is a need to change SalesPerson to Employee ID, as an integer, or just add a column for Employee ID

(b) State whether CryptDB can answer the following query:

- SELECT Department, SUM(Price) AS total FROM Employees E, Orders O
  WHERE E.EID = O.SalesPerson AND Price > 100
  GROUP BY Department
  HAVING total > 1000

If your answer is yes, then explain what encryption schemes should be used for the attribute involved. If your answer is no, then explain how CryptDB should be extended to support the query.

Answer:

Question 2 (3 marks)

Suppose that we are to construct a secure outsourced database using Intel SGX. Explain the major challenges that we would need to address to build such a database system.

Answer:

The size of the Enclave of Intel SGX is small and will be significantly slowed down if the database is too large. Another challenge is that Intel SGX has a few speculative attacks which it is vulnerable against, hence more hardware implementation needs to be added to securely use Intel SGX. However, this would create a tradeoff of speed and utility of the database.

Question 3 (4 + 3 marks)

Assume that we have the following two tables, Student and Enrollment:

Student

Name	Gender
Helen	Female
Cath	Female
Dave	Male
Alice	Female
Gill	Male
Emily	Female
Bob	Male
Fred	Male

Enrollment

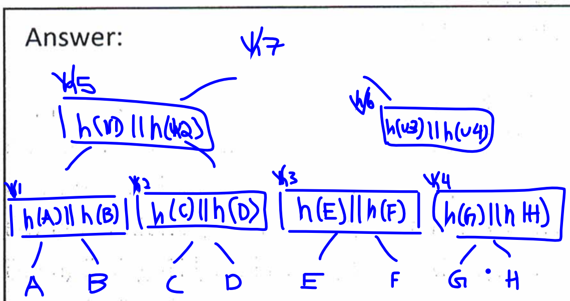
Name	Course
Alice	DB
Helen	AI
Bob	DB
Alice	AI

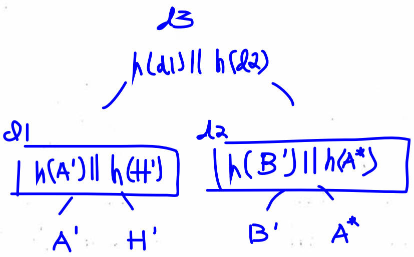
Suppose that we outsource Student and Enrollment to a service provider, and allow users to query the two tables based on Student.Name and Enrollment.Name. To ensure the integrity of query results, we construct Merkle hash trees on Student and Enrollment, respectively, based on their Name columns, using a cryptographic hash function *h*. For convenience, we use A, B, C, D, E, F, G, and H to denote the tuples in Student that correspond to Alice, Bob, Cath, Dave, Emily, Gill, and Helen, respectively, and we use A', H', B', and A* to denote the first, second, third, and fourth tuples in Enrollment, respectively.

(a) Illustrate the Merkle hash trees on Student and Enrollment, respectively, and explain how they can be used to ensure the integrity of the answers of the following two queries:

- SELECT * FROM Student WHERE Name = 'Dave'
- SELECT * FROM Enrollment WHERE Name = 'Dave'

Answer:





SELECT * FROM Student WHERE Name = 'Dave'

Send = v1, C, D, E, h(F), v4, v7

SELECT * FROM Enrollment WHERE Name = 'Dave'

Send A', H', B', A*, d3

(b) Explain how the Merkle hash trees can be used to ensure the integrity of the answer of the following query:

- SELECT * FROM Student, Enrollment WHERE Student.Name = Enrollment.Name

Answer:

Need to send all the leaf nodes and the root node hashes,
so that able to correctly verify why some are able to join and why some cannot

Question 4 (3+4 marks)

Suppose that we have a dataset D that we would like to share with 5 users: u_1, u_2, u_3, u_4, and u_5. Before sharing the data, we divide D into four parts: D_1, D_2, D_3, D_4, and we watermark each part independently using the AHK approach. Let D_i+ denote the watermarked version of D_i (i = 1, 2, 3, 4). Let D_i– denote a modified version of D_i+ obtained by flipping each watermarked bit in D_i+.

(a) Suppose that the dataset that we shared with each user is as follows:

u_1: {D_1+, D_2+, D_3+, D_4+}
u_2: {D_1–, D_2+, D_3+, D_4+}
u_3: {D_1+, D_2–, D_3–, D_4+}
u_4: {D_1–, D_2–, D_3–, D_4–}
u_5: {D_1+, D_2+, D_3+, D_4–}

Analyze whether this watermarking scheme can detect all possible ways of user collusion.

Answer:

No, because u2, u3, and u5 can take the majority rule and this will produce ++++ which hence will be able to frame u1

(b) Suppose that the dataset that we shared with each user is as follows:

u_1: {D_1+, D_2+, D_3−, D_4+}
u_2: {D_1−, D_2−, D_3+, D_4−}
u_3: {D_1+, D_2+, D_3+, D_4+}
u_4: {D_1+, D_2−, D_3−, D_4−}
u_5: {D_1−, D_2+, D_3−, D_4+}

Analyze whether this watermarking scheme can detect all possible ways of user collusion.

Answer:

No either, u3, u4 and u5 can collude and choose the majority and this will result in ++-+, which will be able to frame u1

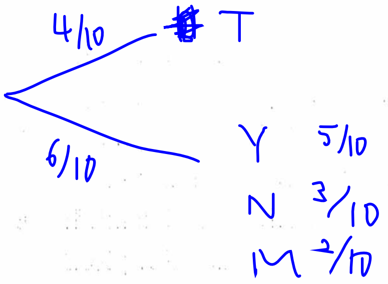
Question 5 (3+6 marks)

Suppose that we ask n users a question Q with three possible answers: "Yes", "No", and "Maybe". For privacy protection, each user adopts randomized response to perturb her answer as follows:

1. She flips a coin that heads with 40% probability;
2. If the coin heads, then she submits her true answer;
3. If the coin tails, then she submits "Yes" with 50% probability, "No" with 30% probability, and "Maybe" with 20% probability.

(a) Explain how we may estimate the fraction of users whose true answers are "Yes", based on the perturbed answers that we collect from the users.

Answer:



Out of all answers, 60% are fake

- out of the 60%, half are perturbed yes
- So out of all the total yes, deduct 30%

H

(b) Suppose that Alice is one of the users. Assume that before observing Alice's perturbed answer, our prior belief about Alice's true answer is 30% "Yes", 30% "No", and 40% "Maybe". Further assume that Alice's perturbed answer is "Yes". Derive our posterior belief after observing Alice's perturbed answer, and show your steps.

H D D H|D

Answer:

$$H|D = \frac{H \wedge D}{D} = \frac{D|H \times H}{D}$$

Y 30
N 30
M 40

60 / \ 40

Y N M T

5/10 3/10 2/10

30 18 12

Y N M

12 12 16

$$\frac{42 \times 30}{D} = ~~42\%~~$$

$$D = D \wedge H + D \wedge H'$$
$$= D|H \times H + 30 \times 70$$
$$= 12.6 + 21 = 33.6$$

$$\frac{\frac{42}{100} \times \frac{30}{100}}{\frac{42 \times 3}{1000} + \frac{210}{1000}} = \frac{126}{1000} \div \frac{336}{1000} = \frac{126}{\cancel{1000}} \times \frac{\cancel{1000}}{336}$$

$$= \frac{126}{336} = \frac{63}{118}$$

Question 6 (5+5 marks)

Let $S = \{v_1, v_2, \ldots, v_n\}$ be a set of values, such that $v_1 \leq v_2 \leq v_3 \leq \ldots \leq v_n$. Let MEDIAN(S) be a function that returns the *median* of the values in S. That is, MEDIAN(S) = $v_{(n+1)/2}$ if n is odd; otherwise, MEDIAN(S) = $(v_{n/2} + v_{n/2+1})/2$.

Consider a table GradesTable(Name, Grade) that stores information about students' grades for CS5322, and the data type of Grade is floating-point number. Suppose that we have a statistical database only allows users to issue queries on GradesTable in the following form:

```
SELECT MEDIAN(Grade) FROM GradesTable
WHERE [some conditions on Name]
```

That is, the users can only issue median queries on subsets of the records.

(a) Suppose that the statistical database requires that each query set size must be at least 3, i.e., each query must cover at least 3 records. Provide an example to illustrate how it is possible that a user may infer the grade of a specific student by issuing multiple queries to the statistical database.

Answer:

(b) Suppose that the statistical database does not impose any query set size control, but denies a user's query *Q* whenever the user can infer the exact grade of a specific student based on the answer of *Q* and the answers of some previous queries (if any). Provide an example to illustrate how the denial of queries may leak information, i.e., how it is possible that a user may infer the exact grade of a specific student when his query is denied. (Note: We assume that no two users collude.)

Answer: