COMP3632 A3

20306527

SIM, Kyu Doun

kdsim@connect.ust.hk

1. Data Privacy

(a)

|  |  |
| --- | --- |
| K-anonymity | No. Complementary release attack can reveal sensitive information. Tw o companies |
| Differential Privacy | No. Without knowing the exact location of the user, the application would be useless. You need to know |
| SMP | Yes. You’re the patient, not giving the exact location. I have my own data, I have no idea about the other’s data, so we could. |
| PIR | No. Data is private  <yes>: company queries. Company has a database, patient user send query, were you here? |

(b)

|  |  |
| --- | --- |
| K-anonymity | No. Data is not private. Protects that data. Data is not sensitive |
| Differential Privacy | No. No data is not private |
| SMP | No. No data is not private. Only query |
| PIR | Yes. Query, protecting the query is a way to private. |

(c)

|  |  |
| --- | --- |
| K-anonymity | No. Too imprecise. Has only 1 DB. You don’t know anyways. Data is not sensitive |
| Differential Privacy | No. The query is private |
| SMP | No. The query is private. Person needs to talk to each other |
| PIR | Yes. the query can be made in a privacy-preserving manner to stop the DNS server from knowing what you asked for (This is assuming that the DNS server will cooperate in supporting such a query.) |

(d)

|  |  |
| --- | --- |
| K-anonymity | Yes. |
| Differential Privacy | Yes. Noise will |
| SMP | Yes. Each person has its own data. Compare SMP |
| PIR | No. |

Lecture: iOS, has great interest in privacy, if we search something on the web, apple can collect what we googled, using noise.

2 Multi-level Security Model

(a)

1. Finn tries to read the file.

**Failed operation. Finn can only read all Finance and Basic files.**

**Level: Stays in Accounting**

2. Batu tries to write to the file.

**Successful operation. Batu is a lower security subject and can write on higher security level objects.**

**Level: Stays in Accounting**

3. Finn tries to write to the file.

**Successful operation. It is not so clear whether Finance level is placed at a higher or lower security level than Accounting. According to the definition of the High-water mark Bell-Lapadula Model, *“****If a person at security level A writes to a file at security level B, then the security of the file must be raised to a level that is allowed to read both A and B; if there are multiple such levels, choose the lowest one.”* **Therefore, when Finn writes to the file, the security level elevates to Management.**

**Level: Elevated to Management**

4. Ace tries to write to the file.

**Successful operation. The security level of the file stays at the Management level.**

**Level: Management**

5. Ace tries to read the file.

**Fails. Ace can only read files on Accounting and Basic level.**

**Level: Management**

6. Ma tries to read the file

**Successful operation. Ma can read all files. Stays at management.**

**Level: Management**

2 (b)

Proof: After Finn writes a file, regardless of its original security level, Acc can never read the file.

If Finn was able to successfully write on a file, according to the high-water mark Bell-Lapuadula, the file’s security level must be raised to a level that can read files on Accounting and Finanace.

Finn to write on. Therefore, regardless of any file, being it on the Basic level or at the Finance level even in Accounting level.

The two security levels that Finn can write on, after Finn writes on the file, it will be placed or remain in the Finance security. No matter what operations come, Ace cannot read the file since the file is under the Finance

2 (c)

1. Finn tries to read the file.

**Failed operation. Finn can only read all Finance and Basic files.**

**Level: Stays in Accounting**

2. Batu tries to write to the file.

**Successful operation. Batu is a lower security subject and can write on higher security level objects.**

**Level: Stays in Accounting**

3. Finn tries to write to the file.

**Successful operation. It is not so clear whether Finance level is placed at a higher or lower security level than Accounting. According to the definition of the High-water mark Bell-Lapadula Model, *“****If a person at security level A writes to a file at security level B, then the security of the file must be raised to a level that is allowed to read both A and B; if there are multiple such levels, choose the lowest one.”* **Therefore, when Finn writes to the file, the security level elevates to Management.**

**Level: Elevated to Management**

4. Ace tries to write to the file.

**Successful operation. The security level of the file stays at the Management level.**

**Level: Management**

5. Ace tries to read the file.

**Successful operation. Ace can read all Accounting, Basic, and Management files.**

**Level: Management**

6. Ma tries to read the file

**Successful operation. Ma can read all files. Stays at management.**

**Level: Management**

2 (d)

Although the proof in 2 ( c ) does not hold anymore, it proves a crucial point about condition for the multi-level security system using the high-water mark Bell-Lapadula model.

For the model to sustain the property where the security of any file could only increase or stay at the same, the Multi-level system the top layer must be able to read all files, for each security level, it should be able to read everything below it.

3 Bitcoin

(a)

1 block per 10 minute = 1 block per 600 seconds

number of bitcoins per second is 6/600 = 0.01.

Total running cost is per second.

Therefore, $20/0.01 = $2000 is the minimum price of Bitcoin to cover the running costs.

(b)

1MB = 1048576 Bytes

1MB / 166 bytes = 6316.72289157, round it to 6316 transactions per block.

Then divide it by 600 seconds per blocks = 10.52666… transactions per second.

Therefore 10.53 transactions per second.

(c)

Number of bitcoins per second is 3/600 = 0.005, so 0.005 bitcoin per second is lost.

Monetary loss needed to be covered by transaction fees = $2000∗0.005 = $100 per second.

Therefore, the mean transaction fee needs to 100/10.53 = $9.496676 = $9.50

(d) 152 devices.

Assume that x machines are needed.

The total share of the bitcoin revenue will be

The number of bitcoins per second is 0.01 and the price is fixed to $2,500.

Therefore, for each second, the revenue would be 0.01 \* 2500 = $25 per second.

Multiply the (share of revenue) \* (revenue per second) will give

$25 \* (ratio above) =

Proﬁt is revenue minus loss. Loss in this case would be the cost of running the miner, which is $ 0.002 per second per machine.

There are about 365 \* (60 \* 60 \* 24) = 31536000 seconds in a year.

Each machine costs $8000.

We are therefore solving the following inequality:

The solution for above is

Therefore, the minimum required number of machines are 152 machines.