CS458 Assignment 3
Insung Chung
i4chung
20576619

Written Response Questions

Question 1. Cryptography: Textbook RSA

a) Compute m_1^e, m_2^e, m_3^e and compare it with c, expecting for an identical result.

b) Three-digit number from 000 to 999 is not a large space. Eve can simply compute $(m_1||x)^e$, $(m_2||x)^e$, $(m_3||x)^e$, for x in [000, 999], and see if any of the result is identical to c'. As this is only 3,000 repetitions, it will not take a long time.

c) Mallory sends $c' = (2^e * c) = (2^e * (r^e \, mod \, n))$ to Bob to decrypt. Bob returns $\left(2^e * (r^e \, mod \, n)\right)^d mod \, n = 2^{e^d} * r^{e^d} \; mod \, n = 2r$ as a result. Mallory divides the resulting value in half and retrieves r.

Question 2. Cryptography Applications: TLS certificates

a) The attack would succeed. TLS ensures that the website you are accessing is really the website of its address, and that the website itself is assured by a trusted CA. Although Mallory's website delivers malicious service, its certificate is from a trusted CA with matching address.

b) This attack would not work. TLS also verifies if the certificate of the website matches its domain.

c) The attack would succeed as the requested address and the host address from the certificate would match.

Question 3. Privacy-Enhancing Technologies: Anonymous Communications

a) Alice is sending messages to server Cyan, j=3. This can be inferred by observing first row of U and Y. Since the first row of Y indicates that all messages are being sent to Cyan and knowing that Alice has sent out 1 message from the first row of U, the particular server Alice is talking to must be Cyan.

b) Calculate sending profile of all except Dave:

$$(9 + 6 + 6) * \text{ProfileExceptDave} = \begin{bmatrix} 6 \\ 2 \\ 13 \end{bmatrix}$$

$$\Rightarrow \quad \text{ProfileExceptDave} = \begin{bmatrix} 6 \\ 2 \\ 13 \end{bmatrix} / 21$$

Estimate Dave's profile based on above profile:

$$(2 + 1 + 1) * \text{ProfileExceptDave} + 5 * \text{ProfileDave} = \begin{bmatrix} 4 \\ 1 \\ 4 \end{bmatrix}$$

⇨ $\text{ProfileDave} = \left(\begin{bmatrix}4\\1\\4\end{bmatrix} - \frac{4}{21}\begin{bmatrix}6\\2\\13\end{bmatrix}\right)/5$

⇨ $\text{ProfileDave} = \left(\begin{bmatrix}60\\13\\32\end{bmatrix}\right)/105$

⇨ $\text{ProfileDave} = \left(\begin{bmatrix}0.571\\0.124\\0.305\end{bmatrix}\right)$

c) Alice sends to Cyan. (From part a)
Carol sends to Yellow. (From row 2: since Alice sends 2 to Cyan, so Carol must be sending 1 to Yellow)
Bob sends to Cyan (From row 1) and Magenta (From row 3: Alice sends 1 to Cyan, Carol sends 1 to Yellow, so Bob must be sending 1 to Magenta)
Dave sends to Yellow (From row 4) and Cyan (From row 9: Alice sends 1 to Cyan, Carol sends 1 to Yellow, so Dave must be sending 1 to Cyan).

Senders	**Recipients**	
Alice	Cyan	
Bob	Magenta	Cyan
Carol	Yellow	
Dave	Yellow	Cyan

d) One way to make profiling difficult is to increase the threshold. This will increase the anonymity of the mix by increasing the candidate size itself. Furthermore, it will reveal less rows with a single sender, revealing less obvious giveaways.

Another way is to add dummy packets as a result of mixes. This will add noise to profiles, making it less accurate, increasing anonymity.

Question 4. Privacy-Enhancing Technologies: Anonymous Communications

a) First, get result from following queries: Select Sum(Salary) From Employee Where …
1. Not Name = Alice
2. 1=1
Then Alice's Salary is (2) – (1).

b) Select Sum(Salary) From Employee Where …
1. Type = "Full Time" And Not Name = Alice
2. Type = "Part Time" And Not Name = Alice
3. 1=1
Then Alice's Salary is (3) – {(1) + (2)}

c) Select Count(*) From Employee Where …
1. Type = "Full Time" Or (Name = Alice And Salary = 50000)
2. Type = "Part Time" Or (Name = Alice And Salary = 50000)
3. 1=1
If (3) = (1) + (2), then Alice's salary is not $50,000. But if (3) = (1) + (2) + 1, this implies that the tuple of

Alice has been counted twice, meaning that the tuple with the condition (Name = Alice And Salary = 50000) exists.

d) Select Count(*) From Employee Where …

1. Type = "Full Time" Or (Name = Alice And Salary >= 50000)
2. Type = "Part Time" Or (Name = Alice And Salary >= 50000)
3. 1=1

Just like part c, if (3) = (1) + (2), Alice's salary is less than $50,000 but if (3) = (1) + (2) + 1, Alice's salary is greater than or equal to $50,000.

e) Consider following sequence of operations:

```
// returns true if Alice.type = "Full Time"
Boolean isFullTime(){
  Select Count(*) From Employee Where …
  1. Type = "Part Time" Or (Name = Alice)
  2. Type = "Part Time"
  return (2) - (1)
}

// If Alice is full time, we need to use "Part Time" as query type
String q_type = isFullTime() ? "Part Time" : "Full Time"
Int TOTAL = Select Count(*) From Employee Where Type = [q_type]

Boolean isInRange(low, high){
  Int counts = Select Count(*) From Employee
      Where Type = [q_type]
      Or ( Name = Alice
          And Salary >= [low]
          And Salary <= [high]
      )
  return TOTAL – counts
}
**Perform binary search using isInRange(low,high) as its comparison operator
```

As this function returns True if Alice's salary is in the range of [low, high] and false otherwise, if we adapt this as a comparison operator for a binary search algorithm from salary range $10,000 to $500,000, which is a search space with n=490,000, this will return in after $3 + \log_2(490{,}000) \fallingdotseq 22$ queries.

Even if we consider salary range including cents, the algorithm will return in $3 + \log_2(49{,}000{,}000) \fallingdotseq 29$ queries.

Therefore, $\mathrm{N}_{\mathrm{q}}^{max} \leq 30$.