University of New Brunswick
Faculty of Computer Science
CS4413/6413: Foundations of Privacy
Theory Homework Assignment 1, ***Due Time, Date*** *5:00 PM, February 28, 2019*

Student Name: ____________________________ Matriculation Number: ________________

Instructor: Rongxing Lu
The marking scheme is shown in the left margin and [100] constitutes full marks.

[**30**] 1. Please answer the following questions.

[5] (a) Why data privacy matters to us? Please elaborate your view as detailed as possible in terms of the General Data Protection Regulation (GDPR).

• Answer:

- We care - we are responsible for handling people's most personal information.
- This is an opportunity to make privacy central to what we do.
- By not handling personal data properly we could put individuals at risk and the entitys reputation at stake.
- Getting it wrong could result in significant fines.
- We need robust systems and processes in place to make sure we use personal information properly and comply.

[5] (b) What are k-anonymity, l-diversity, and t-closeness in database privacy?

• Answer:

- k-anonymity: Table T satisfies k-anonymity with regards to quasi-identifier QI if each tuple in (the multiset) $T[QI]$ appears at least k times.
- l-diversity: a table T satisfies l-diversity with regards to quasi-identifier QI if each of its $T[QI]$ group contains at least l well-represented values for the sensitive attributes.
- t-closeness: a table T has t-closeness with regards to quasi-identifier QI if the distance between the distribution of sensitive attribute values in each of its $T[QI]$ group is no more than threshold t.

[5] (c) What is differential privacy technique? Please describe the steps on how to add the proper Laplace noise to obtain the desirable privacy for the released dataset.

• Answer: The differential privacy technique can guarantee that the privacy risk should not substantially increase as a result of adding or deleting operations in a statistical database. For example, there are two datasets X and X', and X is a neighbor of X' because they differ in one row. However, from the released statistics, it is hard to distinguish X and X'.

Adding Laplace noise follows the following steps.

- According to the statistics function $F(D)$, compute sensitivity of function $F(D)$, i.e., SF.
- Choose the privacy level of the database ε and set the parameter of Laplace distribution λ to be $\frac{SF}{\varepsilon}$.

- When computing the statistics function $F(D)$, add a random noise x to $F(D)$, i.e., $F(D)+x$, where x is from $Lap(\lambda)$ distribution.

[5] (d) Describe the Big Data 4V's characteristics, including volumn, velocity, variety, and veracity, as detailed as possible.

• Answer:

- Volume: Data volume is increasing exponentially, e.g., 44x increase from 2009 to 2020 and from 0.8 zettabytes to 35zb.
- Velocity: Data are generated fast and need to be processed fast.
- Variety: Different types of data are involved, including relational data, text data, graph data, etc., which become more complex. All these types of data need to be linked together.
- Veracity: The data inconsistency and incompleteness, ambiguities etc. will bring some uncertainty in big data. Veracity will consider some security issues in big data in order to protect data veracity.

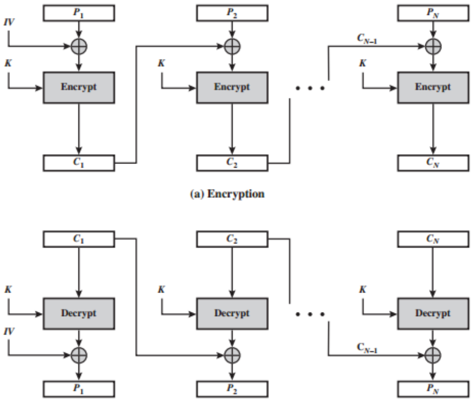
[5] (e) Describe the birthday attack in hash function.

• Answer: Birthday attack is to find two people with the same birthday. For a hash function H, the birthday attack is to find a collision, i.e., find two numbers x and y such that $H(x) = H(y)$.

[5] (f) Describe the homomorphic encryption technique as detailed as possible.

• Answer: Homomorphic encryption is a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintexts. For example, if an encryption technique $E(\cdot)$ satisfies homomorphic property, $E(x) \circ E(y) = E(x \odot y)$, where "$\circ$" is the operation over ciphertext data and "$\odot$" is the operation over plaintext data.

[**6**] 2. Suppose that a message has been encrypted using DES in ciphertext block chaining mode. One bit of ciphertext in block C_i, and another bit of ciphertext in block C_{i+1} are accidentally transformed from 0 to 1 during transmission. How much plaintext will be garbled as a result?



(a) Encryption

(b) Decryption

• Answer: As shown in the above figure, the decryption algorithm in CBC model is $P_i = Dec(C_i) \oplus C_{i-1}$. Thus, if C_i and C_{i+1} are changed during transformation, the plaintexts $P_i = Dec(C_i) \oplus C_{i-1}$,

$P_{i+1} = Dec(C_{i+1}) \oplus C_i$ and $P_{i+2} = Dec(C_{i+2}) \oplus C_{i+1}$ will be affected and other plaintexts will not be affected. Therefore, three plaintexts, i.e., P_i, P_{i+1} and P_{i+2} will be garbled.

[**6**] 3. Using the extended Euclidean algorithm to find the multiplicative inverse of

[3] (a) $12345 \bmod 54321$

• Answer: According to the extended Euclidean algorithm,

$$\begin{align} 54321 &= 12345 * 4 + 4941 \tag{1}\\ 12345 &= 4941 * 2 + 2463 \tag{2}\\ 4941 &= 2463 * 2 + 15 \tag{3}\\ 2463 &= 15 * 164 + 3 \tag{4}\end{align}$$

Thus, $gcd(12345, 54321) = 3$, so the inverse of $12345 \bmod 54321$ does not exist.

[3] (b) $350 \bmod 1769$

• Answer: According to the extended Euclidean algorithm,

$$\begin{align} 1769 &= 350 * 5 + 19 \tag{5}\\ 350 &= 19 * 18 + 8 \tag{6}\\ 19 &= 8 * 2 + 3 \tag{7}\\ 8 &= 3 * 2 + 2 \tag{8}\\ 3 &= 2 * 1 + 1 \tag{9}\end{align}$$

Thus,

$$\begin{align} 1 &= 1 - 2 * 1 \tag{10}\\ &= 3 - (8 - 3 * 2) * 1 = 3 * 3 - 8 * 1 \tag{11}\\ &= (19 - 8 * 2) * 3 - 8 = 19 * 3 - 8 * 7 \tag{12}\\ &= 19 * 3 - (350 - 19 * 18) * 7 = 19 * 129 - 350 * 7 \tag{13}\\ &= (1769 - 350 * 5) * 129 - 350 * 7 = 1769 * 129 - 350 * 652 \tag{14}\end{align}$$

Therefore, $350^{-1} \bmod 1769 = -652 \bmod 1769 = 1117 \bmod 1769$.

[**6**] 4. In a public-key system using RSA , you intercept the ciphertext $C = 9$ sent to a user whose public key is $e = 5, n = 35$. What is the plaintext M?

• Answer: According to $n = pq = 35$, $p = 5$ and $q = 7$. So, $\varphi(n) = (p-1)(q-1) = 24$. At the same time, $e = 5$ and $ed \equiv 1 \bmod \varphi(n)$, so $d = 5$.

According to the decryption algorithm $M = C^d \mod n$, so $M = C^d = 9^5 \mod 35 = 4$.

[**6**] 5. Use the Chinese Remainder Theorem (CRT) to solve x, where

$$\begin{cases} x & \equiv & 1 \bmod 3 \\ x & \equiv & 2 \bmod 5 \\ x & \equiv & 3 \bmod 7 \end{cases}$$

• Answer: Let $m_1 = 3, m_2 = 5, m_3 = 7$, $a_1 = 1, a_2 = 2, a_3 = 3$. We have $M = m_1 \cdot m_2 \cdot m_3 = 3 \times 5 \times 7 = 105$.

Then, $M_1 = M/m_1 = 35, M_2 = M/m_2 = 21, M_3 = M/m_3 = 15$.

$\alpha_1 = M_1^{-1} \mod m_1 = 35^{-1} \mod 3 = 2, \alpha_2 = M_2^{-1} \mod m_2 = 21^{-1} \mod 5 = 1, \alpha_3 = M_3^{-1} \mod m_3 = 15^{-1} \mod 7 = 1$.

Therefore, $x = (a_1 \cdot \alpha_1 \cdot M_1 + a_2 \cdot \alpha_2 \cdot M_2 + a_3 \cdot \alpha_3 \cdot M_3) \bmod M = (1 \times 2 \times 35 + 2 \times 1 \times 21 + 3 \times 1 \times 15) \bmod 105 = 52$

[6] 6. Consider an Elgamal encryption scheme with a common prime $q = 11$ and a primitive root $\alpha = 2$. If B has public key $Y_B = 7$ and A choose the random integer $k = 3$, what is the ciphertext of $M = 9$?

• Answer: The ciphertext $C_1 = \alpha^k \bmod q = 2^3 \bmod 11 = 8$ and $C_2 = M * Y_B^k \bmod q = 9 * 7^3 \bmod 11 = 7$.

[10] 7. Let $F(x)$ be the true answer on input x, and $Geom(\alpha)$ be the noise sampled from Geometric distribution with parameter $\alpha = e^{-\epsilon/S(F)}$. Please prove that the release of $F(x) + Geom(\alpha) = F(x) + Geom(e^{-\epsilon/S(F)})$ can obtain ϵ-DP guarantee.

Proof: Suppose that $A = F(x) + Geom(\alpha)$, and D_1 and D_2 are any two adjacent DBs. Thus, $A(D_1) = F(D_1) + x_1$ and $A(D_2) = F(D_2) + x_2$, where x_1 and x_2 are $Geom(\alpha)$ distributed. Since $\alpha = e^{-\epsilon/S(F)}$, the probability density for x_1 is proportional to $\frac{\alpha-1}{\alpha+1}\alpha^{|x_1|}$. Similarly, the probability density for x_2 is proportional to $\frac{\alpha-1}{\alpha+1}\alpha^{|x_2|}$. Therefore, for any $T \in range(A)$

$$\frac{Pr[A(D_1) = T]}{Pr[A(D_2) = T]} = \frac{Pr[F(D_1) + x_1 = T]}{Pr[F(D_2) + x_2 = T]} = \frac{Pr[x_1 = T - F(D_1)]}{Pr[x_2 = T - F(D_2)]} = \frac{\frac{\alpha-1}{\alpha+1}\alpha^{|x_1|}}{\frac{\alpha-1}{\alpha+1}\alpha^{|x_2|}}$$

$$= \frac{\alpha^{|x_1|}}{\alpha^{|x_2|}} = \alpha^{|x_1| - |x_2|} = \alpha^{|T - F(D_1)| - |T - F(D_2)|} \leq \alpha^{|F(D_1) - F(D_2)|} = (e^{-\epsilon/S(F)})^{|F(D_1) - F(D_2)|}$$

where the inequality follows form the triangle inequality. By the definition of sensitivity,

$$S(F) = Max_{D_1, D_2 : |D_1 - D_2| = 1} |F(D_1) - F(D_2)|$$

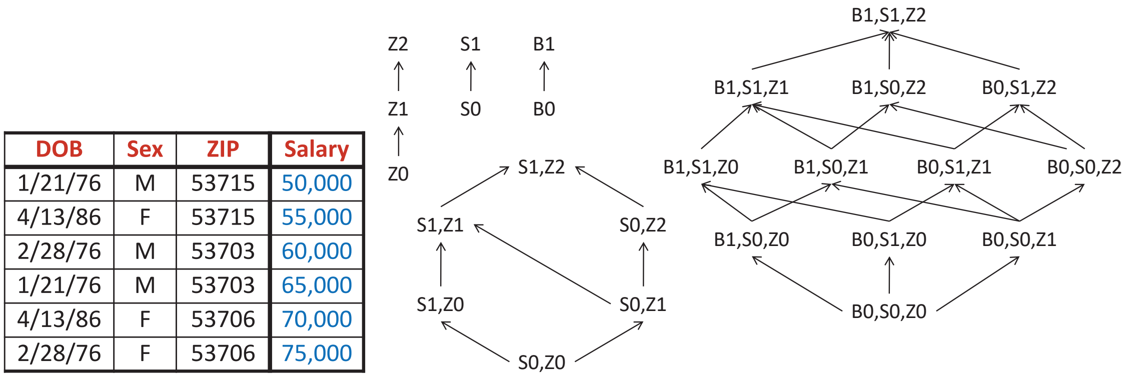
.

Thus, $(e^{-\epsilon/S(F)})^{|F(D_1) - F(D_2)|} \leq e^{-\epsilon}$. The ration is bounded by $e^{-\epsilon}$, yields ϵ-Differential Privacy.

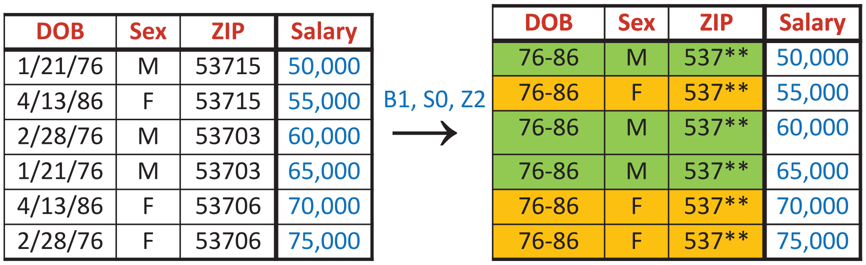
[10] 8. Incognito is one of approaches to implement the k-anonymity. Given a table below, the full-domain generalizations described by "domain vectors" are represented as follows.

- $B_0 = \{1/21/76, 2/28/76, 4/13/86\} \rightarrow B_1 = \{76 - 86\}$
- $S_0 = \{M, F\} \rightarrow S_1 = \{*\}$
- $Z_0 = \{53715, 53710, 3706, 53703\} \rightarrow Z_1 = \{5371*, 5370*\} \rightarrow Z_2 = \{537**\}$

Based on the Lattice of Domain Vectors, we can achieve the k-anonymity for the original table. For example, with (B_1, S_0, Z_2) generalization, we can achieve $k = 3$-anonymity.



DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

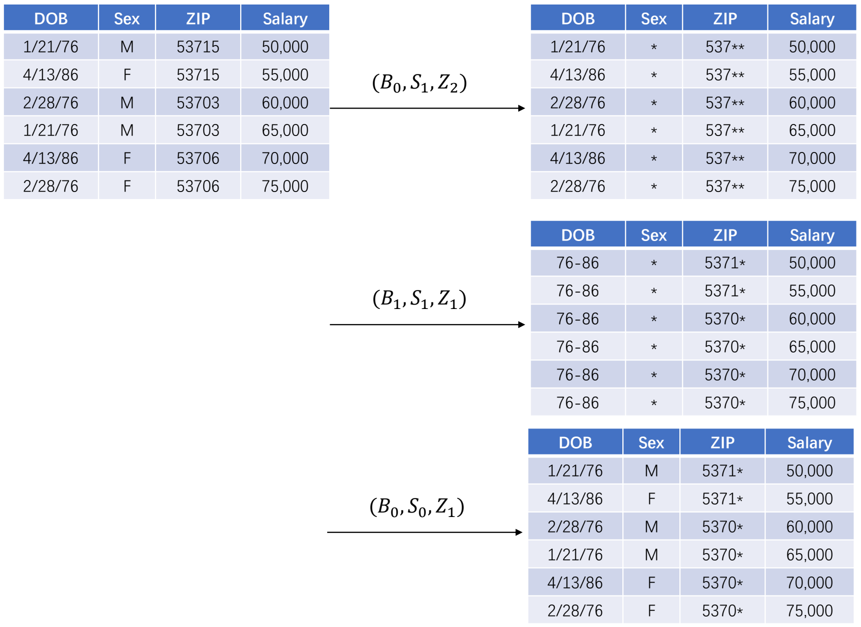


DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

DOB	Sex	ZIP	Salary
76-86	M	537**	50,000
76-86	F	537**	55,000
76-86	M	537**	60,000
76-86	M	537**	65,000
76-86	F	537**	70,000
76-86	F	537**	75,000

Please follow the example to apply (B_0, S_1, Z_2), (B_1, S_1, Z_1), (B_0, S_0, Z_1) to generalize the original table, and discuss what is the value of k in each generalized case?

- Answer: The generalized tables are shown as the following figure.



DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

(B_0, S_1, Z_2)

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	65,000
4/13/86	*	537**	70,000
2/28/76	*	537**	75,000

(B_1, S_1, Z_1)

DOB	Sex	ZIP	Salary
76-86	*	5371*	50,000
76-86	*	5371*	55,000
76-86	*	5370*	60,000
76-86	*	5370*	65,000
76-86	*	5370*	70,000
76-86	*	5370*	75,000

(B_0, S_0, Z_1)

DOB	Sex	ZIP	Salary
1/21/76	M	5371*	50,000
4/13/86	F	5371*	55,000
2/28/76	M	5370*	60,000
1/21/76	M	5370*	65,000
4/13/86	F	5370*	70,000
2/28/76	F	5370*	75,000

From the figure, we know that (1) when applying (B_0, S_1, Z_2) to the original table, $k = 2$; (2) when applying (B_1, S_1, Z_1) to the original table, $k = 2$; (3) when applying (B_0, S_0, Z_1) to the original table, $k = 1$.

[**10**] 9. Alice and Bob are good friends, they have shared a secret key sk in advance. Now, Alice wants to send 20 messages $x_1, x_2, \cdots, x_{20}$ to Bob, because there may be errors occurring in communication channel and also possible injection false data attack from external attackers, Alice hopes to use the bloom filter to enhance the security of these messages in term of source authentication and data integrity. Can you help Alice and Bob to design an efficient bloom filter?

[5] (a) For all hash functions $h_1, h_2, \cdots, h_k$ used in the bloom filter, we define each hash function h_i : $\{0,1\}^* \rightarrow \{2^0, 2^1, 2^2, \cdots, 2^{159}\}$, which means we are using an array D with length $n = 160$ as a bloom filter. Then, with the bloom filter D, Alice just needs to use 160 bits overheads for authenticating 20 messages. Can you compute the optimal number of hash functions, k , and the corresponding false positive FP?

- Answer: The optimal number of hash function $k = \ln 2 * \frac{n}{m} = \ln 2 * \frac{160}{20} \approx 6$ and the false positive is $(1 - e^{-\frac{km}{n}})^k = (1 - e^{-\frac{6*20}{160}})^6 = 0.0216$.

[5] (b) For the <u>same values</u> of k, n in Question 9(a), we now design two bloom filters $(D1, D2)$ as the authenticator as follow: the authenticate overhead is still $n = 160$ bits in total, but each hash function is $h_i : 0, 1^* \rightarrow \{2^0, 2^1, 2^2, \cdots, 2^{79}\}$, which means the bloom filters $D1$ and $D2$ are with length $n/2 = 80$ bits. Among k hash functions, half of them, i.e., $k/2$ hash functions, are used for $D1$, the rest $k/2$ hash functions are used for $D2$. With these settings, can you compute the new false positive FP? Compared with the bloom filter D in Question 9(a), which one is better, $(D1, D2)$ or D?

- Answer: For the bloom filter D_1, $n_1 = 80$, $k_1 = \frac{k}{2} = 3$ and $m = 20$.

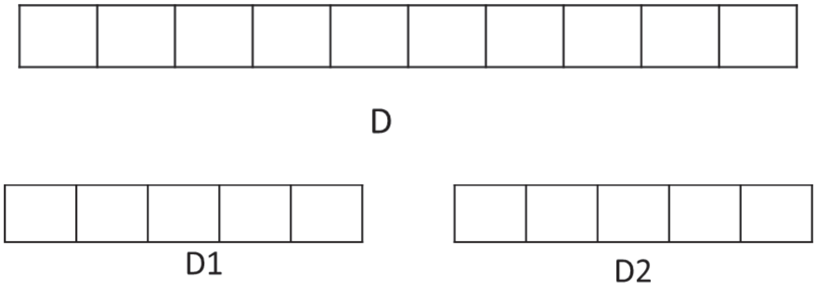
Thus, the false positive in bloom filter D_1 is $p_1 = (1 - e^{-\frac{k_1 m}{n_1}})^{k}_{1} = (1 - e^{-\frac{3*20}{80}})^3 = 0.1489$.

Similarly, for the bloom filter D_2, $n_2 = 80$, $k_2 = \frac{k}{2} = 3$ and $m = 20$.

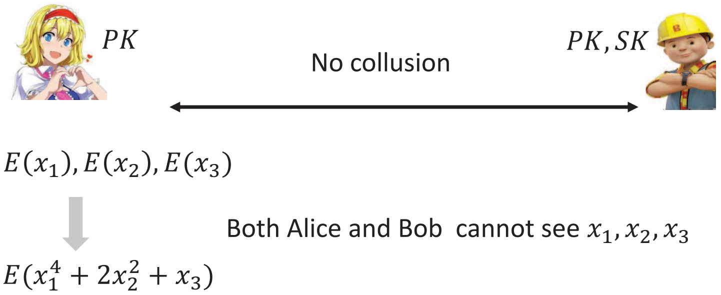
Thus, the false positive in bloom filter D_2 is $p_2 = p_1 = 0.1489$.

So the overall false positive probability $p = p_1 * p_2 = 0.1489 * 0.1489 = 0.0216$.

Since the false positive probability in two bloom filters (D_1, D_2) is the same as that in bloom filter D, the bloom filter D is the same as (D_1, D_2).



[**10**] 10. Let $E()$ be a BGN homomorphic encryption scheme. Assume Bob has the public/private key pair (pk, sk) of $E()$, and Alice only has the public key pk of $E()$. Consider there is no collusion between Alice and Bob. When Alice has three ciphertexts $E(x_1)$, $E(x_2)$ and $E(x_3)$, please design a protocol run between Alice and Bob. With the protocol, Alice can finally obtain the ciphertext $E(x_1^4 + 2x_2^2 + x_3)$ while both Alice and Bob have no idea on the plaintexts x_1, x_2, and x_3.


PK
No collusion
PK, SK
$E(x_1), E(x_2), E(x_3)$
Both Alice and Bob cannot see x_1, x_2, x_3
$E(x_1^4 + 2x_2^2 + x_3)$

• Answer:

Solution 1:

- Compute $E'(x_1^4)$:
  - Alice chooses a random number r_1 and computes $E(x_1 + r_1) = E(x_1) * E(r_1)$. Then, Alice sends $E(x_1 + r_1)$ to Bob.
  - Bob recovers $(x_1 + r_1)$ with the private key sk, and then computes $E((x_1 + r_1)^2) = E(x_1 + r_1)^{x_1 + r_1}$. Then, Bob sends $E((x_1 + r_1)^2)$ to Alice.
  - After receiving $E((x_1 + r_1)^2)$, Alice computes $E(x_1^2) = E((x_1 + r_1)^2 - 2x_1r_1 - r_1^2) = E((x_1 + r_1)^2) * [E(x_1)^{2r_1}]^{-1} * E(r_1^2)^{-1}$.
  - Alice computes $E'(x_1^4) = e(E(x_1^2), E(x_1^2))$.
- Compute $E'(x_2^2)$: Alice first computes $E(2x_2) = E(x_2)^2$ and then $E'(2x_2^2) = e(E(2x_2), E(x_2))$.
- Compute $E'(x_3)$: Alice computes $E'(x_3) = e(E(x_3), E(1))$.
- Compute $E'(x_1^4 + 2x_2^2 + x_3) = E'(x_1^4) * E'(2x_2^2) * E'(x_3)$.

Solution 2:

- Compute $E'(x_1^4)$:
  - Alice chooses a random number r_1 and computes $E(x_1 + r_1) = E(x_1) * E(r_1)$. Then, Alice sends $E(x_1 + r_1)$ to Bob.
  - Bob recovers $(x_1 + r_1)$ with the private key sk, and then computes $E((x_1 + r_1)^2) = E(x_1 + r_1)^{x_1 + r_1}$. Then, Bob sends $E((x_1 + r_1)^2)$ to Alice.
  - After receiving $E((x_1 + r_1)^2)$, Alice computes $E(x_1^2) = E((x_1 + r_1)^2 - 2x_1r_1 - r_1^2) = E((x_1 + r_1)^2) * [E(x_1)^{2r_1}]^{-1} * E(r_1^2)^{-1}$.
  - Similarly, Alice computes $E(x_1^4)$ by $E(x_1^2)$ with the help of Bob.
- Alice computes $E(x_2^2)$ using the same algorithm that computes $E(x_1^2)$.
- Alice computes $E(x_1^4 + 2x_2^2 + x_3) = E(x_1^4) * E(x_2^2)^2 * E(x_3)$.