Computer Science COMP4670/8640

Assignment 3

Due: End of Sunday, November 24, 2019

Question 1 (20%)

There are two main approaches to use in privacy-preserving techniques, secure computations, and anonymization & randomization.

- What would be the main trade-off in each of these approaches, in terms of efficiency, accuracy, utility, and privacy?
- Based on your answers to the previous question, bring an example in real-world applications, such that the first approach, secure computation, is more suitable and/or applicable to be used to preserve data privacy.
- Bring another example, such that the second approach, anonymization & randomization, is more suitable and/or applicable to be used to preserve data privacy.
- **Secure Computation: Privacy <-> Efficiency**
- **Anonymization & Randomization: Privacy <-> Accuracy (& Utility)**
-
- **Secure Computation: Data Sensitive applications, such as some Health-related applications.**
- **Anonymization & Randomization: Realtime system with low data sensitivity.**

Question 2 (20%)

Suppose there are two parties, P_A and P_B, each of which owns a matrix of private integer values. We illustrate these matrices as follows:

$$P_A:\ A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}_{m \times n} \quad , \quad P_B:\ B = \begin{pmatrix} b_{11} & \cdots & b_{1r} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nr} \end{pmatrix}_{n \times r}$$

Now, suppose these two parties want to multiply their matrices, and get the **final result as the addition of two separate matrices**, each of which owns by one party. What is your suggested solution, such that no individual input values of one party are disclosed to another party, and each of the two final matrices will only be disclosed to one party? No third party is allowed to participate for this secure two-party computation. (Show your complete solution.)

As each of the cells in the result is computed by the dot product of one row and one column, it could be done using secure dot product.

Question 3 (20%)

Suppose we already have **Secure Two-party Production** protocol as follows:

$$\boldsymbol{Prod2Sum}: \ \boldsymbol{x_1 * x_2 = z_1 + z_2}$$

In the above equation, $\boldsymbol{x_i}$ is the private input and $\boldsymbol{z_i}$ is the private output, both belong to party $\boldsymbol{i}$.

Clearly show how we can use $\boldsymbol{Prod2Sum}$ protocol to generate the final private outputs for the parties in the following **Secure Multiparty Production** protocol.

$$\boldsymbol{x_1 * x_2 * \ldots * x_n = z_1 + z_2 + \cdots + z_n}$$

$$\boldsymbol{x_1 * x_2 * \ldots * x_n = (y_1 + y_2) * x_3 * \ldots * x_n}$$
$$\boldsymbol{= (y_1 * x_3 * \ldots * x_n) + (y_2 * x_3 * \ldots * x_n)}$$

This can be continued until we have a set of two-party multiplications that can be solved using the above Prod2Sum protocol.

Question 4 (20%)

One of the set operations used in some applications is "**Secure Cardinality of Set Intersection**" or "**Secure Set Intersection Size**", in which two or more parties, each of which has a set of values, want to jointly and securely compute the size of the set intersection of their private sets, without revealing the set items to each other. Especially, for two parties, ***Alice*** and ***Bob***, for instance they have the following sets:

$$\boldsymbol{Alice}: \boldsymbol{S_A} = \{\mathbf{5, 12, 9, 4, 8, 1}\} \quad , \qquad \boldsymbol{Bob}: \boldsymbol{S_B} = \{\mathbf{6, 9, 13, 5, 1, 7}\}$$

Therefore: $\boldsymbol{S_A \cap S_B = \{5, 9, 1\} \Rightarrow |S_A \cap S_B| = 3}$

Propose a secure method, using any known secure building blocks (such as Secure Addition, Secure Dot Product, etc.), by which ***Alice*** and ***Bob*** are able to reach to the final value of set intersection size, **without disclosing their set values or the size of their sets to each other**.

Show all the steps of your work.

Note: You can assume that **they already now the range of the values inside the sets**, i.e. the minimum and maximum values for the set items. It means that there are two, publicly known, numbers $\boldsymbol{m}$ and $\boldsymbol{M}$ such that:

$$\forall \boldsymbol{s} \in (\boldsymbol{S_A} \vee \boldsymbol{S_B}) : \boldsymbol{m} < \boldsymbol{s} < \boldsymbol{M}$$

Alice and Bob know the minimum and maximum values from their sets. Therefore, Alice creates a vector of binary values, such that the each item of the vector is 1 if the value of its index exists in Alice's private set. For instance, in the above example, m=1 and M=13. Therefore, Alice's vector would be:

$$V_A = \langle 1,0,0,1,1,0,0,1,1,0,0,1,0 \rangle$$

Bob will do the same:

$$V_B = \langle 1,0,0,0,1,1,1,0,1,0,0,0,1 \rangle$$

Now, that can run a secure dot product to find out the set intersection size of their private sets.

Question 5 (20%)

Process of **variable type checking** could not be the same in different programming languages. Discuss about handling this task in two programming languages, Java and PHP, in terms of program security. Bring an example to show how the program security will be at risk because of the type of handling this task in one of those two languages.

The process of type checking may occur either at compile time/static check or run time/dynamic check. If a language specification requires its typing rules strongly (i.e., more or less allowing only those automatic type conversions that do not lose information), one can refer to the process as strongly typed, if not, as weakly typed.

Type checking in Java: Java is Strongly Typed. Let's consider the following example:

int a;
a = 2+ 2.5;

In Java, the variable ‘a’ has been declared as having type ‘int’. We consequently examine the assignment statement. We know that ‘a’ has type ‘int’ so we check that the assignment which is being assigned to ‘a’ also has type ‘int’. For an addition to have type ‘int’ both of the values must also be of type ‘int’. First we check 2, which is an integer as we expect, but then we check 2.5, which is a real number, and therefore we have a type error.

Type checking in PHP:

PHP is Weakly Typed. In PHP, it automatically converts data of any type into the expected type. This feature very often masks errors by the developer or injections of unexpected data, leading to vulnerabilities.

Let’s consider the following example:

One might want to write the following code to allow a user to view a calendar that

displays a specified month in UNIX environment-

```
$month = $_GET['month'];

$year = $_GET['year'];
```

This code has a security flaw, since the $_GET[month] and $_GET[year] variables are not validated. The application works perfectly, as long as the specified month is a number between 1 and 12, and the year is provided as a proper four-digit year. However, an attacker might append ";ls -la" to the year value and thereby see a listing of the Website’s html directory. A malicious attacker could append ";rm -rf *" to the year value and delete the entire Website.