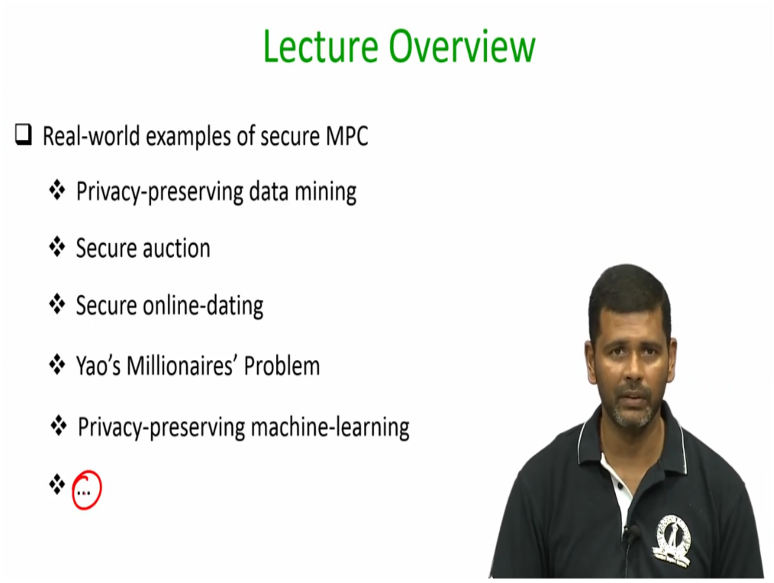
Secure Computation - Part I
Prof. Ashish Choudhury
Department of Computer Science
International Institute of Information Technology, Bangalore

Module - 1
Lecture - 2
Real-World Examples of Secure MPC

(Refer Slide Time: 00:31)



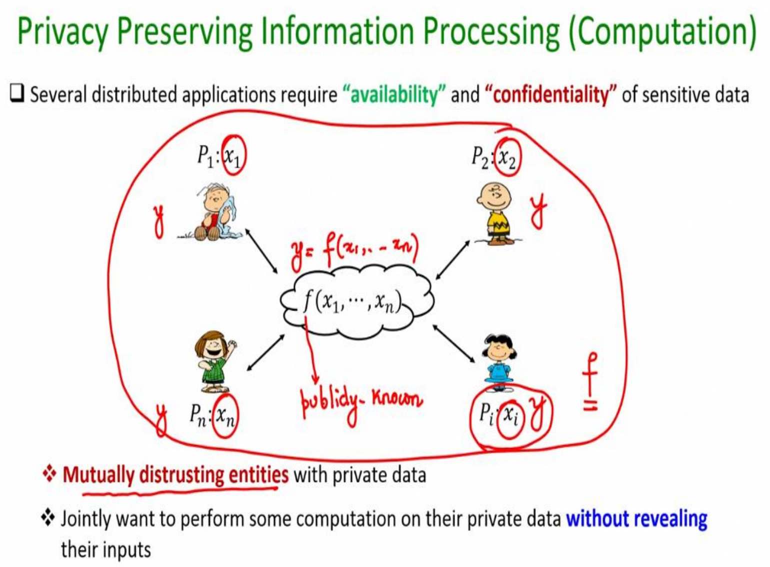
Lecture Overview

❑ Real-world examples of secure MPC

- ❖ Privacy-preserving data mining
- ❖ Secure auction
- ❖ Secure online-dating
- ❖ Yao's Millionaires' Problem
- ❖ Privacy-preserving machine-learning
- ❖ …

Hello everyone. Welcome to this lecture. The plan for this lecture is as follows: In this lecture, we will see various real-world examples of secure multi-party computation. So, this is not an exhaustive list of examples, but just for a sake of interest, I have selected these 5 applications. But there are thousands of applications, thousands of real-world problems which can fall under the umbrella of secure multi-party computation.

(Refer Slide Time: 01:05)



Privacy Preserving Information Processing (Computation)

❑ Several distributed applications require **"availability"** and **"confidentiality"** of sensitive data

❖ **Mutually distrusting entities** with private data

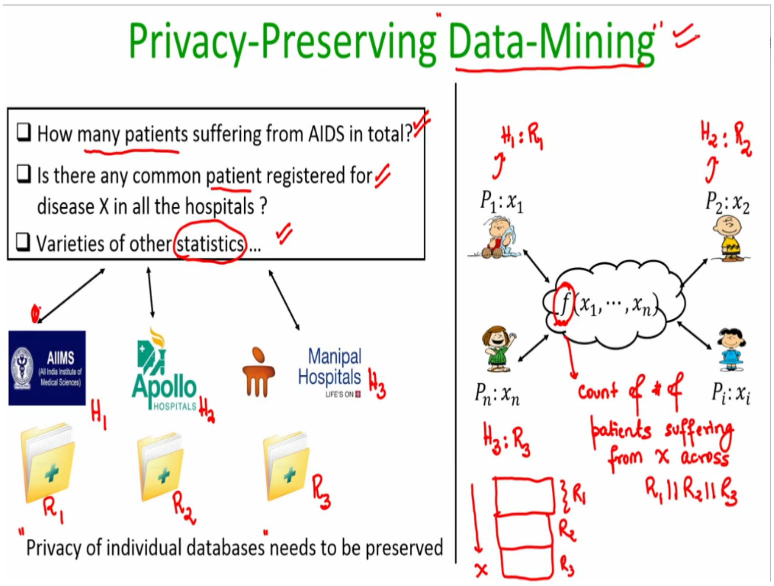
❖ Jointly want to perform some computation on their private data **without revealing** their inputs

So, let us first recall what exactly is the problem of a secure multi-party computation, how we have abstracted it out. So, we have a set of mutually distrusting entities or parties, and each party has some private data available with it. And there is some publicly known function f. And the goal of the parties is to somehow compute the output of the function on the inputs of all their parties, without revealing anything additional about their inputs.

So, the function output, let us call it as y. So, say y is equal to the output of the function on the inputs $x_1, \dots, x_n$. Our goal is to help these end parties to learn y. But in the process, no party should learn anything additional other than what it can learn from y and its own input. That is what is our goal roughly. So, if I consider P_i, P_i will be knowing y; it will be knowing its own input x_i; and anyhow, the details of the function f is known.

We want a mechanism so that P_i should not learn anything additional apart from what it can infer from x_i, y and f. That means, it should not learn anything additional about $x_1, x_2, \dots, x_n$, apart from what it can learn from x_i, y, and f. That is what is our goal.

(Refer Slide Time: 03:09)



"Privacy-Preserving Data-Mining"

- □ How many patients suffering from AIDS in total?
- □ Is there any common patient registered for disease X in all the hospitals ?
- □ Varieties of other statistics…

"Privacy of individual databases" needs to be preserved

So, now, what we are going to do is the following: We will see some varieties of real-world problems which can be abstracted by this blueprint of secure multi-party computation. So, let us start with a problem of privacy-preserving data-mining. And the problem here is the following: Imagine you have individual hospitals. So, in this particular example, I am considering 3 hospitals. And each hospital has its own medical record.

So, imagine this is hospital number 1 with its medical record R_1; that is a database. Hospital number 2, with its own set of patient records; let us denote it by R_2. And hospital number 3, which has its own set of medical records or patient database, denoted by R_3. And remember that medical records of the patients is a very sensitive information. Each hospital is bound to maintain the privacy of the patient records available with that hospital.

It cannot afford to make them available in the public domain. It cannot afford to leak them in the public domain. That will be considered a severe breach of privacy. And, in fact, hospitals may face legal actions if they fail to maintain the privacy of the individual patient records. Now, suppose the hospitals together would like to perform some data-mining operation. That means they would like to perform various kind of operations on the combined database across all the 3 hospitals.

Say for instance, they might be interested to find out how many patients are suffering from AIDS across all the 3 hospitals. You might be asking - why would hospital 1, hospital 2, and hospital 3 would be interested to carry out this operation? Well, they might be forced, for example, by the government. Say, if the government is trying to perform some survey on how many patients are suffering from AIDS in a particular city. Then it can simply issue an order to all the hospitals that, okay, we do not care how do you solve this problem, we are interested to find out, for the purpose of statistics, how many patients are suffering from AIDS across all the hospitals in the city.

Or since we are facing a pandemic, the government might be interested to find out, the number of COVID patients across all the hospitals in the city. So, that is some kind of data-mining operation. And the data-mining operation needs to be performed across all the 3 hospitals. That is important here. In the same way, it might be the case that hospitals are part of a bigger network of hospitals which maintains a kind of a global database.

And we might be interested to find out if there is any specific patient who has registered across some specific disease across all the 3 hospitals. So, for instance, it might be the case that a specific patient had first gone to the hospital number 1 to get treatment for x, and the patient was not satisfied by the treatment. Then, the patient went to the hospital number 2. And then she was not satisfied with the service available by hospital number 2.

So, she moved to the hospital number 3 and so on. So, what might happen is that, when a patient goes to a new hospital, the new hospital might want to find out from the other hospital whether this specific patient has been already enrolled to the earlier hospitals or not, without knowing anything additional about the other records available with the other hospitals. That also constitutes some kind of data-mining operation.

In the same way, there could be varieties of other statistics which these hospitals would like to collaboratively compute. Now, the important point here is that, even though we want to perform these data-mining operations across all the hospital records, we need to maintain the privacy of the individual databases. What do I mean by that? So, let us take the first problem, namely, our goal to find out how many patients are suffering from AIDS across the 3 hospitals.

So, we are interested to find out the count. Our goal is to perform this computation on R_1, R_2 and R_3 and to get the count of the total number of patients suffering from AIDS. That is a computation we are interested to perform. And we want a mechanism which helps us only to get the result of this computation, without revealing anything additional about the individual databases R_1, R_2 and R_3.

So, if we are not interested to maintain the privacy of the individual databases, then, performing data-mining across the 3 hospitals is very easy. Hospital number 1, what it can do is, it can transfer all its records to hospital number 2. And now, hospital number 2 can transfer the records of hospital number 1 as well as hospital number 2 to hospital number 3.

And now, hospital number 3 will have the records of all the hospitals in clear, and it can perform whatever operations it would like to perform, and it can declare the result to hospital number 1 and hospital number 2. So, the problem is very easy to solve if we do not put the requirement that the privacy of the individual hospital records have to be maintained. But, as I said that no hospital can afford to provide its medical records to any third party. That is a breach of confidentiality.

So, our goal here is to maintain the privacy of the individual hospital records and still perform these operations. Now, let us see how this problem can be abstracted by the blueprint of privacy-preserving information processing or secure computation that I have discussed. So, in the abstract blueprint, we had n mutually distrusting parties with their individual data $x_1, x_2, \dots., x_n$.

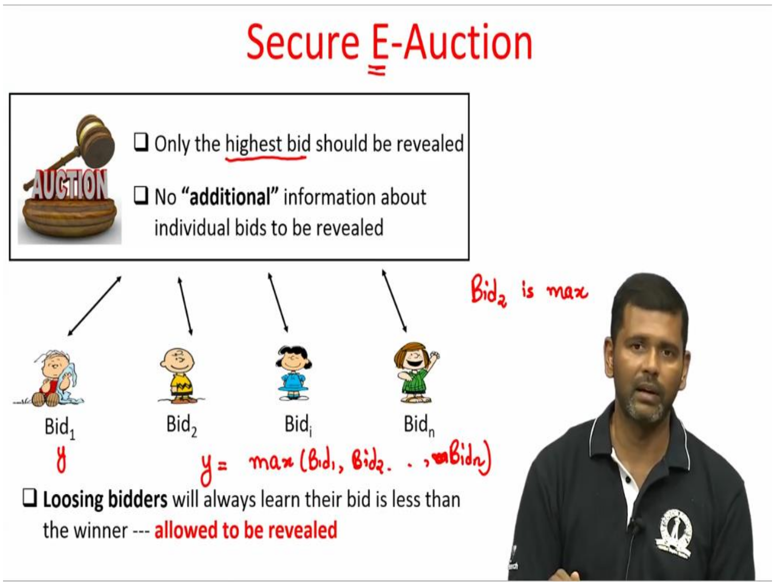
And there was a publicly known function and the goal was to compute securely$f(x_1, .., x_n)$. So, in this particular example, you can imagine that party 1 is your hospital number 1 and x_1 is nothing but the medical records. Party number 2 is nothing but H_2. And x_2 is nothing but the medical records associated with the second hospital. And in the same way, the nth party, you can imagine that it is hospital number 3.

And x_n is nothing but the records of the third hospital. And what is the function f? Well, the function f here is the count of number of patients suffering from x, disease x, across the combined database R_1, R_2, R_3. So, pictorially you can imagine that each database is a table, for simplicity.

We are interested to find out how many people are suffering from x. That is a computation. That, you can imagine as an SQL query, but as an abstract task, you can view it as a computation. In the same way, the second problem, namely, is there a common patient registered for some specific disease across R_1, R_2, R_3? That can be abstracted as another kind of function and so on.

So, depending upon what exactly is the statistics you would like to obtain, that can be abstracted by some abstract function f. And the goal is to only obtain the result of the abstract function f on the data $x_1, \ldots, x_n$, and nothing additional. So, this is a real-world example of secure multi-party computation.

(Refer Slide Time: 13:30)



Secure E-Auction

❑ Only the highest bid should be revealed

❑ No **"additional"** information about individual bids to be revealed

❑ **Loosing bidders** will always learn their bid is less than the winner --- **allowed to be revealed**

Now, let us see the problem of secure E-auction. So, imagine you have n bidders, and they are bidding for some valuable object. And each bidder has a private bid. So, since this is an auction application, the bidders cannot afford to make their bid available to the other bidders. And we would like to perform the auction in an online fashion. So, the goal here is to obtain a mechanism which allows the bidders to learn the value of the highest bid.

But in the process, no "additional" information about the individual bids should be revealed. What do we mean by additional here? So, you see, once the value of the highest bid is learnt; say for instance, everyone found that bid_2 is the highest bid.

Now, once this result is learned by everyone at the end of the E-auction protocol algorithm, then the first bidder will know, not only the first bidder, every bidder, $bidder_2, bidder_3, \ldots, bidder_i, \ldots, bidder_n$, everyone will know that the value of bid_1 is less than bid 2. That information is revealed or leaked. But that would not be considered as a breach of privacy, because, anyhow, once the value of the maximum bid is learned, everyone will know that it, apart from the bid of the person who has won, all other bid values are less than that particular value.

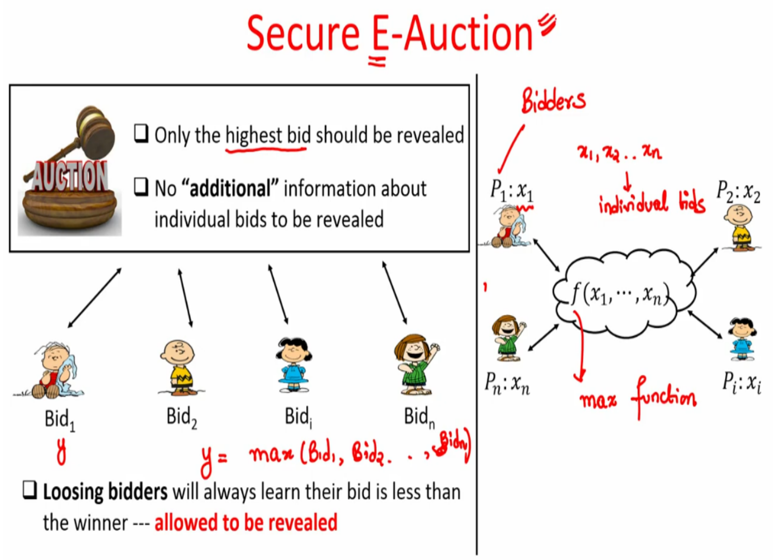
That information is anyhow allowed to be leaked. That is not something which you can prevent from getting leaked. But that would not be considered as a breach of privacy, because, remember, our goal is to ensure that each party learns the function output or the result of the computation. And the computation is basically to find out the maximum among $bid_1, bid_2, .., bid_n$. That is a function parties would like to compute. That is your y.

So, what I am saying here is that our goal for the MPC problem, our goal through secure multi-party computation is to ensure that each party learns only y and nothing additional about the other party's input, apart from what it can learn from y and its own input. So, in this particular case, if I take $bidder_1$, from bid_1 and y, it will of course learn that, okay, the value of the other bids is less than this y.

That information is anyhow inferred from the function output. And that is fine to be learnt by the parties. That is not something which we are interested to keep private. That is allowed to be leaked. But apart from the fact that the value of the other bids are lesser than y, the exact bid value should not be revealed, because, that is not something which can be inferred from the y and the individual bids.

So, that means, the losing bidders, the fact that the losing bidders learns that their bid value is less than the winner is not something which should be considered as a breach of privacy here.

(Refer Slide Time: 17:39)



Secure E-Auction

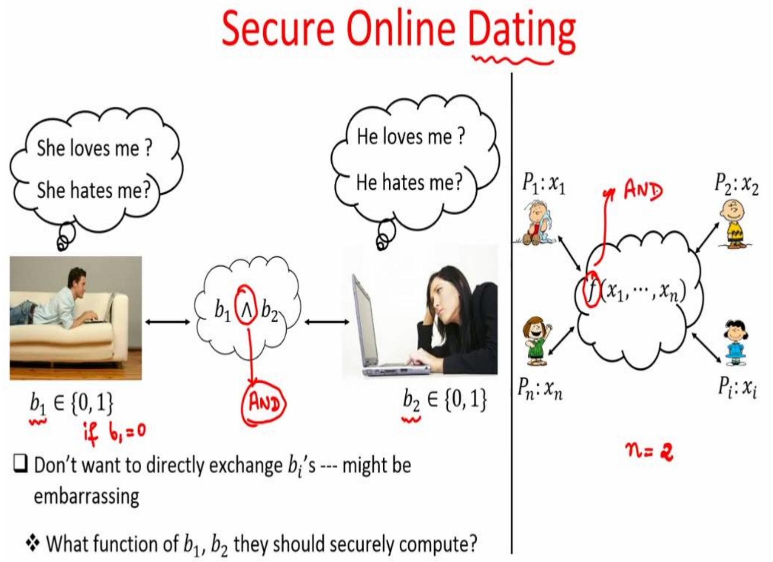
❑ Only the highest bid should be revealed

❑ No **"additional"** information about individual bids to be revealed

❑ **Loosing bidders** will always learn their bid is less than the winner --- allowed to be revealed

So, now, let us see how this specific problem can be abstracted by this blueprint of secure multi-party computation. So, here, your parties are your bidders. And these $x_1, x_2, \dots, x_n$, they are the individual bids. And the function f that they are interested to compute is the max function. So, our goal is to enable these parties, compute the max of $bid_1, \dots, bid_n$. And in the process, nothing additional should be revealed about the individual bids, other than what can be inferred from the result and the individual bids. So, this again, this problem is again a very nice example of secure multi-party computation task.

(Refer Slide Time: 18:43)



Secure Online Dating

❑ Don't want to directly exchange b_i's --- might be embarrassing

❖ What function of b_1, b_2 they should securely compute?

Now, let us consider the problem of secure online dating. And this is a 2-party problem. You have 2 parties. They might be knowing or may not be knowing each other. And each party has a choice bit, which is a value which could be either 0 or 1. Let the choice of party P_i be b_i. So, if I consider party number 1, b_1 is 0 if the party 1 does not like party 2, whereas b_1 will be 1 if party 1 likes or interested in the second party. And similarly you can interpret the bid value b 2.

Now, that is a private information associated with P_1 and P_2 respectively. And imagine that they are dating over some online dating site. The parties may not be willing to directly tell each other whether they are interested in the other party or not, because that might be embarrassing. But still they would be interested to find out whether both of them are mutually interested in each other or not.

So, the problem that we are interested to solve is the following: We do not want b_1 and b_2 to be learned by each other, but still we would be interested to find out whether party 1 and party 2 are both interested in each other or not.

That means, definitely we are interested to compute some Boolean function of b_1 and b_2. So, now, the question is, what Boolean function will help us to solve this problem? Is it the OR function or is it the AND function or is it the XOR function? What precisely is the function? Basically, we are interested here to compute the AND of b_1 and b_2. Because, if the result of the AND is 1, that means, the result of AND is 1, only when b_1 and b_2 are both 1.

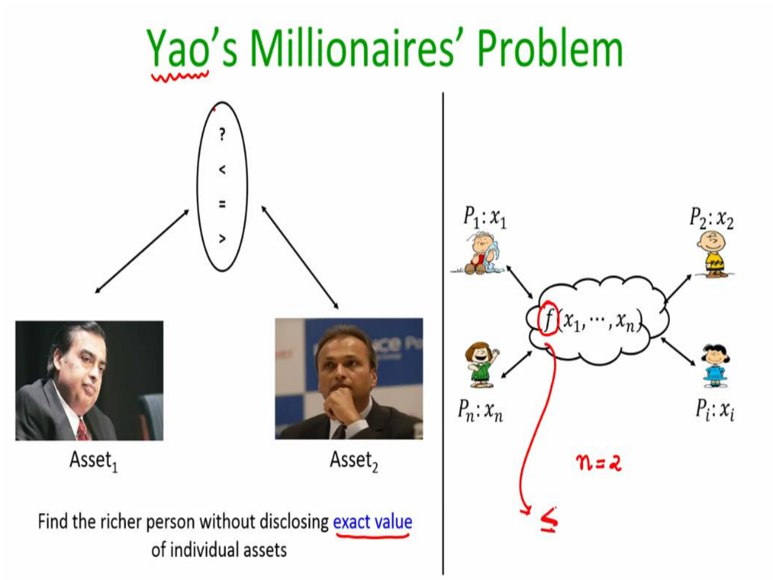
Even if one of the bits b_1 or $b_2$is 0, the result of the AND will be 0. And since we are interested to find out whether both the parties are interested in each other or not, basically, we are interested to securely compute the AND of b_1 and b_2. Again, as I said, one way of finding whether both the parties are mutually interested in each other or not is that, well, they could have directly told at the first place to each other; well, I am not interested in you, and then you tell me whether you are interested in me or not, but that might be embarrassing.

We want a mechanism which allows the parties to securely compute the AND of b_1 and b_2. And in the process, neither P_1 nor P_2 should learn anything additional apart from what these individual parties could learn from the AND of the bids and their own bid. So, for instance, what we want here is the following: If $b_1 = 0$, then anyhow party 1 knows that the overall result will be 0, irrespective of whether the second party is interested in the first party or not.
But we want a mechanism where, at the end of whatever algorithm P_1 and P_2 deploy, P_1 learns the result; it will learn the result is 0, but it should not learn whether b_2 is 0 or whether b_2 is 1.

That is what we are interested to; saw. Now, again, this can be abstracted by your blueprint of secure multi-party computation. Here you have $n = 2$. So, this secure online dating is a special case of secure multi-party computation, where there are only 2 parties are associated.

In all the previous examples, we had 2 or more number of parties. But in this specific case, we have exactly 2 parties. And the function f that they are interested to securely compute is the AND function.

(Refer Slide Time: 23:27)



Yao's Millionaires' Problem

Find the richer person without disclosing exact value of individual assets

Let us see another nice example of secure 2-party computation. This is called as Yao's Millionaires' problem attributed to the Turing Award winner Andrew Yao, who actually formulated the problem of secure multi-party computation and proposed the first protocol for secure multi-party computation. Later in the course, we will see how exactly Yao solved the problem of secure 2-party computation.

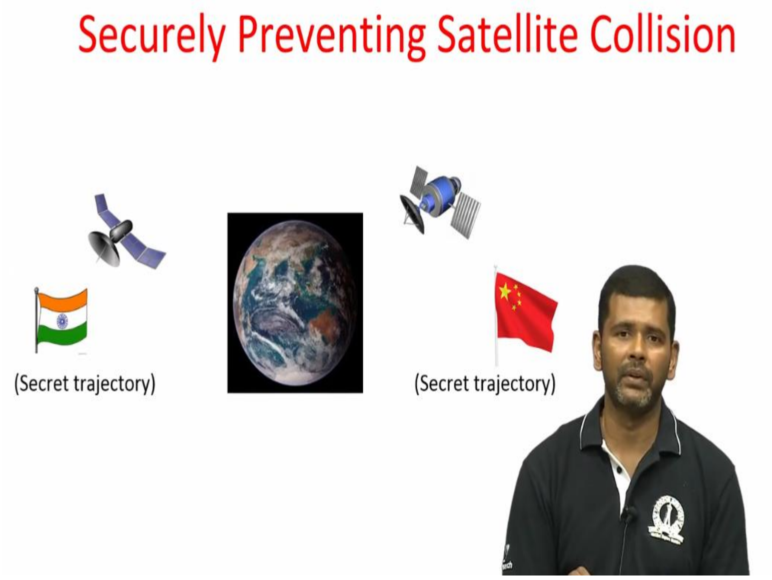
But here, we are interested to understand the Yao's Millionaires' problem and how it constitutes a real-world example of secure multi-party computation. So, the problem here is the following: You have 2 millionaires and they have their individual assets, but the millionaires are mutually distrusting. So, they are not interested in directly telling their assets to the other party or to the other millionaire.

But, still they are interested to find out who is the richer person without disclosing the exact values of the individual assets. So, basically, we want a mechanism here, where the 2 millionaires should learn whether asset 1 is greater than asset 2 or whether asset 1 is equal to asset 2 or whether asset 1 is lesser than asset 2. And in the process, nothing additional should be revealed.

Of course, if asset 1 is lesser than asset 2, that is allowed to be learned by both the parties. But that would not help to find out the exact value of asset 1 or exact value of asset 2. So, we are interested to learn the output of this function, namely this less than equal to function. And nothing additional apart from what we can infer or what the millionaires could infer from the result and their own individual assets should be revealed in the process.

Again, this can be abstracted by this blueprint of secure multi-party computation. You have precisely 2 parties here. x_1 stands for asset number 1, x_2 stands for asset number 2 and your function f is nothing but your less than equal to function.

(Refer Slide Time: 26:01)

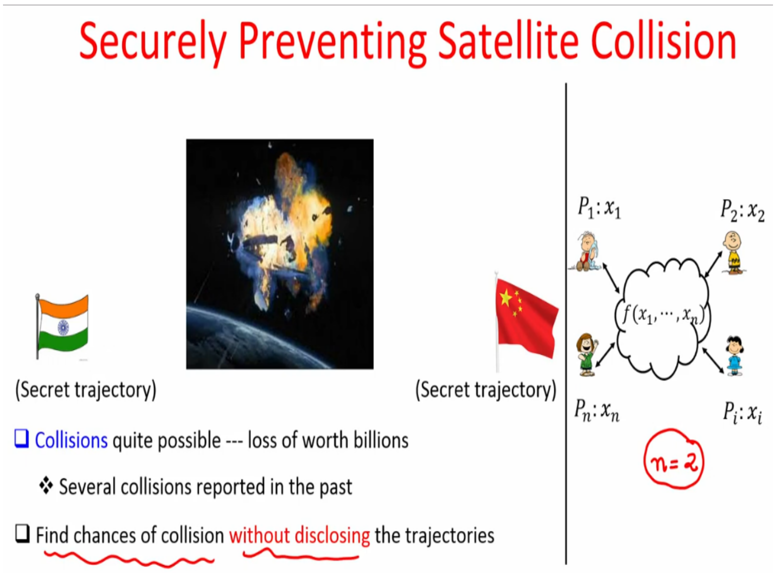


Securely Preventing Satellite Collision

Let us see another very interesting application of secure multi-party computation. This is securely preventing satellite collisions. So, imagine you have 2 countries who do not have very good relationship. And each country launches its own spy satellite. Very often, countries launch their spy satellite to keep track of what other countries are doing and so on. Now, since it is a spy satellite, a country cannot afford to make public the details of the trajectory information of the spy satellite.

So, if India is launching its spy satellite, there will be some secret trajectory information which its spy satellite might be following. And in the same way, if China is launching its own spy satellite, its satellite will have its own secret trajectory information, which will be known only to China. But since these 2 countries are arbitrarily launching the spy satellites, without knowing each other's trajectory information, it is quite possible that collision happens between the 2 satellites.

(Refer Slide Time: 27:23)



Securely Preventing Satellite Collision

(Secret trajectory)

(Secret trajectory)

❑ Collisions quite possible --- loss of worth billions

❖ Several collisions reported in the past

❑ Find chances of collision without disclosing the trajectories

And if it so happens, then it will be a loss worth of billions. And it is quite possible, because your space is not infinitely large, it is enormously large. But since India is launching its satellite arbitrarily, without knowing the direction or the trajectory information of China; and same way, China is also launching its spy satellite, without knowing the path followed by India's satellite, it is quite possible.

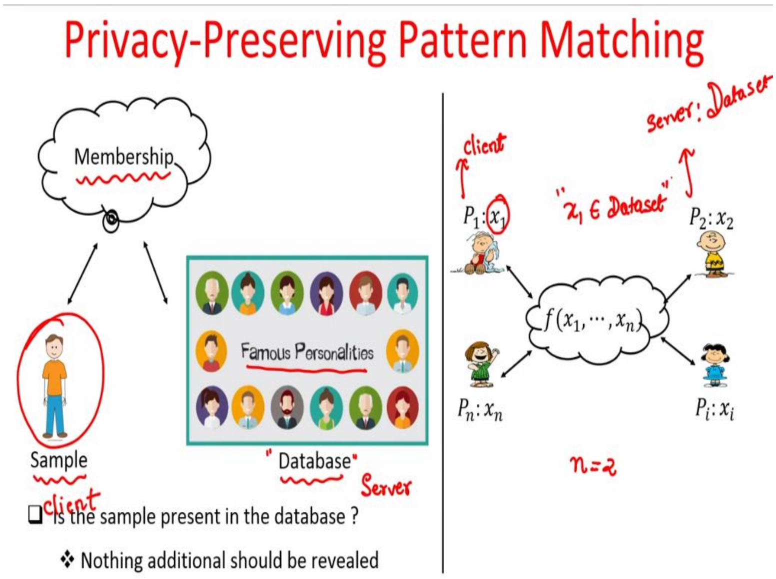
The probability that the 2 satellites collide is definitely non-zero. It might be small, but it is non-zero. And if the collision happens, then that will be loss to both the countries. And if you search in Google, you can find out that several such collisions have been reported in the past, not between the satellites of India and China, but satellites of 2 mutually distrusting countries.

So, now, the question is, is it possible for these 2 countries to find out what is the probability of the 2 satellites to collide without actually disclosing the individual trajectories? Because, if India discloses the trajectory information to China and China discloses its trajectory information to India, then of course, they ensure that, okay, they launch their respective satellites in such a way that that they never collide.

But then, the whole purpose of a spy satellite is lost. If India tells China that, okay, this is the path I am going to follow, then China will ensure that, okay, it sees, secretly follows India's satellite and then ensure that all the spying activities are nullified, right? So, the goal here is to find out the chances of collision without actually disclosing the individual trajectory information.

And again, this is a specific example of 2-party computation, where $n = 2$. Of course, you can bring in multiple parties. Instead of saying 2 countries, you can bring the third country, say Pakistan; the fourth country, Bangladesh, Maldives and so on. But the special case of $n = 2$ is more realistic. And this can be abstracted by your, this specific problem of preventing satellite collision in a privacy-preserving fashion is a special case of secure 2-party computation.

(Refer Slide Time: 30:01)



Privacy-Preserving Pattern Matching

❑ Is the sample present in the database ?

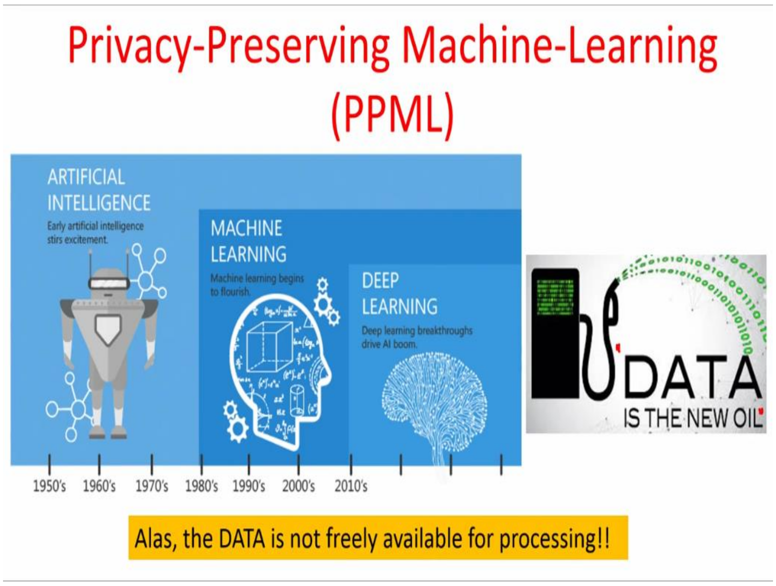
❖ Nothing additional should be revealed

Another real-world application of secure multi-party computation is that of privacy-preserving pattern matching. And what is the problem we want to solve here? Say we have a client with some sample, and say there is a database which is hosted in some server. And that database is the database of famous personalities. We want to find out whether this sample is present in the database or not.

Namely, we are interested to solve the membership problem. But in the process, we do not want the client to learn anything additional about the database. Namely, the client will learn whether the sample is present in the database or not. That is the result of the computation. But, that should not help the client to find out what are the other entries in the database, how large is the database and so on.

These are the additional information which should not be revealed from the result of the computation. And at the same time, we do not want the database to learn anything about the sample, because the privacy of the client's sample is also important. So, again, this is a special case of 2-party computation, where party 1 is your client with some sample x_1 and party 2 is your server which has a huge database, which is its data x_2. And we want to find out whether x_1 is present in the data set or not. That is the computation we are interested to perform in a secure way.

(Refer Slide Time: 32:07)



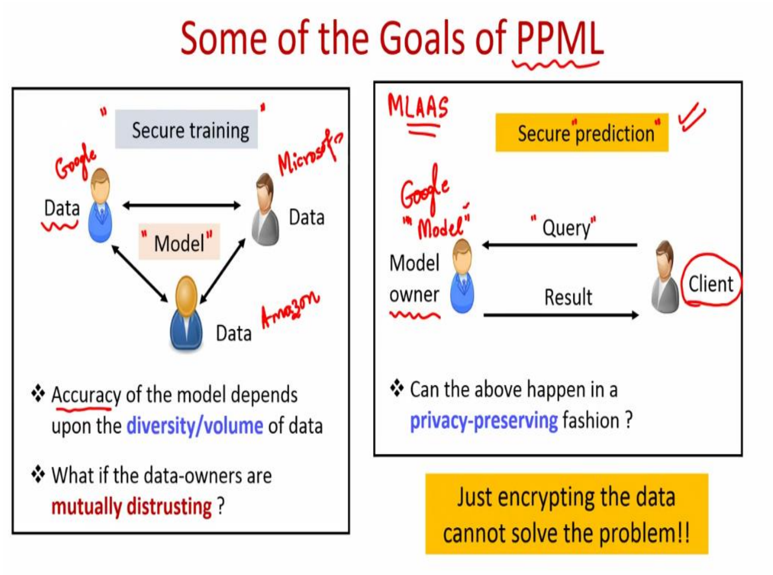
Privacy-Preserving Machine-Learning (PPML)

ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's
DATA
IS THE NEW OIL

Alas, the DATA is not freely available for processing!!

Now, a very recent application of secure multi-party computation is this very fascinating and very hot area of privacy-preserving machine-learning. Now, I do not need to tell you the importance of machine learning. Everyone these days is talking about machine learning, artificial intelligence. And we have this new mantra that data is the new oil. We have, in fact a new area of science called as data science, dealing with huge data and so on. However, machine learning and AI applications need enormous amount of data for running those algorithms, but the data is not available freely for processing in these applications.

(Refer Slide Time: 33:06)



Some of the Goals of PPML

❖ Accuracy of the model depends upon the diversity/volume of data

❖ What if the data-owners are mutually distrusting ?

❖ Can the above happen in a privacy-preserving fashion ?

Just encrypting the data cannot solve the problem!!

So, some of the goals of privacy-preserving machine-learning are the following: The first problem that we want to solve here is that of secure training. So, imagine you have individual data owners. And when I say individual data owners, they are basically tech giants, say Google, Microsoft, Amazon and so on. They have enormous data available with them. And together they would like to perform some ML computation to prepare what in ML jargon we call us model.

And it turns out, the accuracy of the model depends upon the diversity and the volume of data. Because, if you have enormous amount of data and you are training on that data, then your model will be very accurate to do the future predictions. So, typically in ML, what we do is, we have enormous amount of data available, we run some ML algorithm and do some training and prepare what we call a model.

And once the model is ready, then whenever some new data comes, a new query comes, we find out or we classify the new data with respect to the model that we have already prepared. So, the result of the future queries, of the future computations depends upon the accuracy or how good is my model, how nicely I have performed my training. And that further depends upon what is the volume of data, what is the diversity of the data on which I have performed my training algorithm.

So, for example, if I am preparing my model for a medical application, then, if I have done the training over a huge amount of data spread across the entire country, then the model will be very accurate. If the model is for, say, predicting cancer patients or AIDS patients. On contrary, if I do the training only over the data of say Karnataka patients, then my model may not be very accurate, because, a model which is trained with respect to the Karnataka patients may not give you very accurate answers with respect to a patient query coming from say, Uttar Pradesh or other states, because of the diversity and the population, pollution issues and so on.

So, the point here is that, if the amount of data and the diversity of the data is large, then the accuracy of the model will be very good. So, now, what we are envisioning here is the following: We have data owners who are kind of distributed, and they do not trust each other. But still they would like to run the ML training algorithm, prepare a model without disclosing their individual data to the other data owners.

If I considering the application where the data owners are tech giants, then Google may not be interested to reveal the data that it is owned by Google to Microsoft or Amazon. Similarly, Microsoft may not be interested to reveal its data to Google and Amazon and so on. But still they would like to run the ML training algorithm and prepare the model in a privacy-preserving fashion. So, that is one of the goals of privacy-preserving machine-learning.

And this can be abstracted by your blueprint of secure multi-party computation, where the parties are the data owners. And the function that they would like to compute is basically the ML training algorithm. The second problem that we would like to solve in privacy-preserving machine-learning is that of secure prediction. So, there are various servers, huge tech giants who offers what we call as machine learning as a service.

And the scenario here is the following: You have a model owner who has some private model available with itself. And the model has been prepared by the model owner by performing the training algorithm on enormous amount of data. Now, it is willing to provide this model as a service to individual clients. What do I mean by that? If a client comes with a query, then this model owner is willing to provide the result of the query run on its model.

That is the business model for this model owner. That means, this model owner basically is acting as a kind of a cloud service provider, where a weak client who may not have its own model but would like to learn the result of the query on the model, may provide some fees to the cloud service provider, and a query, and get the result of the query on the model. So, that is the problem of prediction or machine learning as a service.

But if we allow the client to provide its query in clear, then the query is learnt by the model owner. That breaches the privacy of the query. So, there might be applications where the client may not be willing to directly reveal the query to the model owner. Say for instance, if this is a medical application, and if this client is a potential patient suffering from AIDS, then she may not be willing to directly provide her query to a model owner.

And in the same way, model owner would like to ensure that the client learns only the result of the query. It should not learn anything about the model which is owned by the model owner. Because, if the model is also revealed to the client, then tomorrow, the client itself can just copy paste the model, and it can also start acting as a cloud service provider and run the same business which the model owner is running right now.

So, we have privacy requirements at both the ends. The model owner wants to preserve the privacy of its model; the client wants to preserve the privacy of its client. But still, they would like to perform what we call as prediction in a secure way. So, the goal of privacy-preserving machine-learning is to solve this problem. And this second problem, namely, secure prediction is similar to that of solving the membership problem.

It is not exactly the same problem, but similar. You have a client; you have a server; client had a query there; and it wanted to test whether it belongs to the database or not. Here also, instead of membership problem, the client wants to learn the result of the query. The result could be the output of any ML query. It could be, say the result of linear regression, logistic regression.

So, these are some of the sophisticated ML algorithms which are run with respect to a query and a model. So, you can imagine that the model owner is, say Google, and the client is some weak applicant. And the client has some query and say wants to learn the result of linear regression with respect to its query and the model held by the Google service provider. We want only to learn the result of the linear regression and nothing additional.

So, that is another application of secure multi-party computation. That brings me to the end of today's lecture. Just to summarise, in this lecture, we have seen several real-world examples of distributed computation where we have multiple entities. And each entity has its own private data. And the entities would like to perform some kind of computation without disclosing their private data. Thank you.