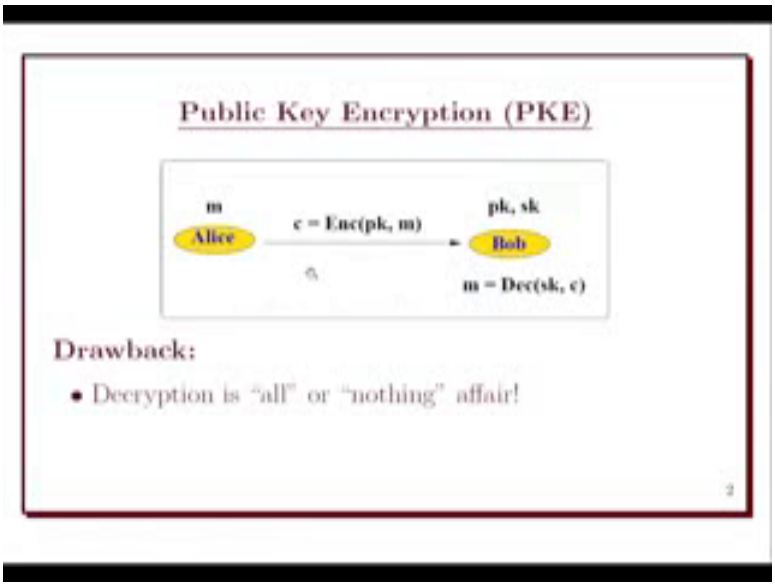
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 56
Functional Encryption (Introduction)

So, we talk about functional Encryption. So, basically the functional Encryption, we will give an introduction to function.

(Refer Slide Time: 00:28)

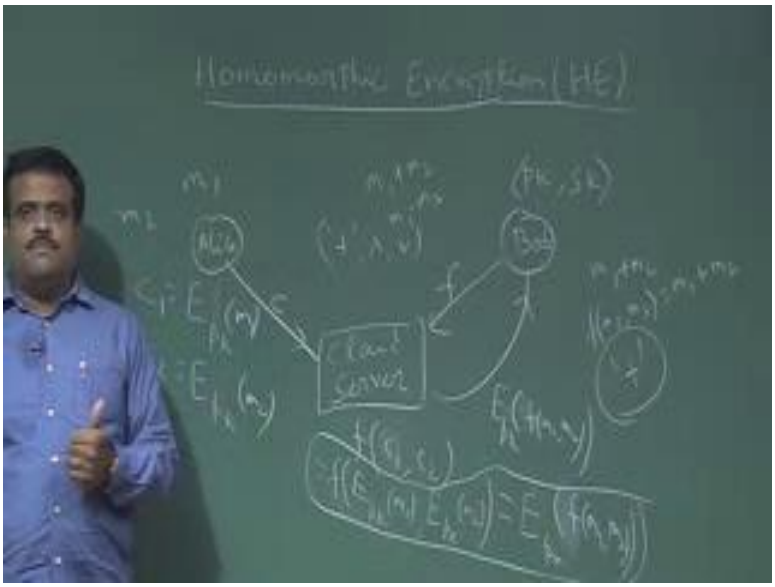


So, we have seen the public key Encryption, basically we have two party Alice and Bob. So, suppose Alice wants to send the message to Bob. So, the Bob need to generate his own public key and private key pair, and this public key has to be shared with the Alice, so that Alice can send a message to Bob. If Alice wants to send message m to Bob then Alice will encrypt this message using that Bob public key, and generate the ciphertext and send it to Bob over the public channel. So, this is what is we know the public key Encryption scheme.

And after receiving this ciphertext c what Bob will do Bob is having the corresponding secret key which is s k. So, Bob will apply the decryption algorithm on this ciphertext using this secret key and get the message. So, this is the public key Encryption scheme. And the drawback is if this is the right key s k and then Bob will give the full message otherwise nothing. So, decryption is all or nothing. So, Bob will not getting any partial

information if Bob is I mean if this is the correct key s k then Bob will get the whole message otherwise Bob will not get any information. So, this is one of the drawback of public key Encryption schemes.

(Refer Slide Time: 02:11)



Now, we will talk about Homomorphic Encryption. So, basically Homomorphic Encryption scheme means what it is called HE. So, idea is we have two party Alice and Bob; and we have a cloud server, we have a server which is referred as cloud server. So, Alice is having a message and Bob is having this Bob public key and private key pair p k and s k. So, this encrypt as so Alice is encrypting this message and generating the ciphertext. So, Alice is encrypting this message using this Bob's public key, and generating the ciphertext, and Alice is storing this in here in this c, this is say if it is m 1 this is c 1. So, c 1 is stored here. And if you have another message m 2 then c 2 will get E of p k m 2. So, c 2 is stored here. So, this is the storage in the cloud. So, we have a server which is referred as a cloud server. So, we keep this here. So, this is the, which is sending to cloud and could cloud is to storing this. So, cloud is having this ciphertext storage. This is the database maintaining by the cloud.

Now suppose Bob needs information about this m say m 1 and m 2. So, Bob will send a function to the cloud. So, suppose Bob is not required the full message m 1 and m 2. So, suppose Bob required say m 1 plus m 2. So, this is basically f of m 1, m 2, this is basically m 1 plus m 2. So, this is a plus operation, this is functionality. So, this f Bob

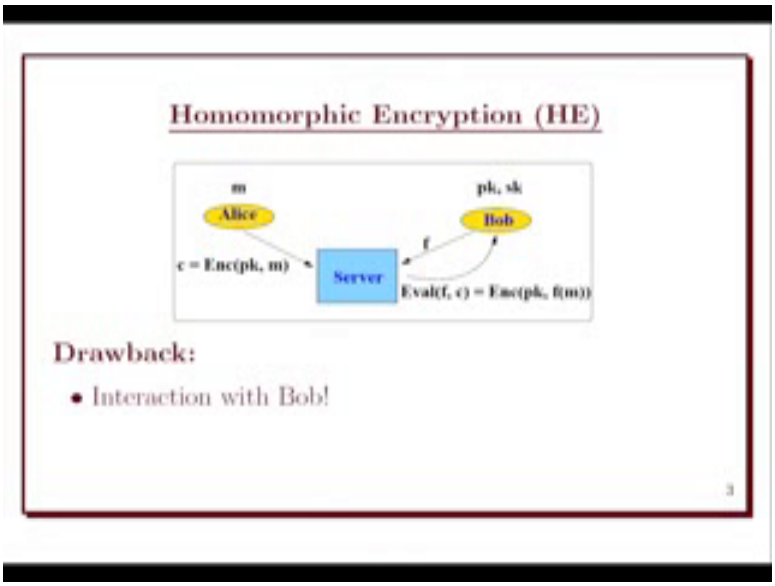
will send to the cloud so that means, Bob needs a information about m 1 plus m 2 not that particular m 1 and m 2

So, what Bob can do? So, Bob can instead of getting whole message, so cloud can send Bob to if it is just a public key Encryption. So, what cloud can do? Cloud can sends c 1, c 2 separately to Bob, and Bob will decrypt it and then Bob will compute m 1 and m 2 and then Bob will compute f of m 1, m 2. So, this is the usual the public key setup, public key scenario. But here suppose Bob is using just a small device which may not able to compute m 1 plus m 2 not only plus, we may have some other operation m 1 m into m 2 or some other circuit in general instead of plus dot we may have general circuit. So, general circuit we may have. So, that may not be possible by the Bob to calculate after getting m 1, m 2, because Bob may be caring a low power device which having some computation constant. So, if we can use the cloud to calculate that for on behalf of Bob. so that is that is the main advantage of the Homomorphic Encryption instead of calculating the shape for m 1, m 2, we ask the cloud to do that and send us to the encrypted version of m 1, m 2.

So, we want that Bob this cloud will send us the encrypted version of encryption using the bobs. So, this is the bobs public key f of m 1, m 2. So, this is we want, we want that Bob that cloud will do something such that cloud will able to send this to Bob, and then Bob is having Bob own secret key Bob will apply the decryption algorithm on this and Bob will get f of m 1, m 2. So, f is a general function general circuit not necessarily just the simple m plus. So, f could be any general circuit, it could be some complicated computation, so that computation we are doing with the help of the cloud not at this end, because Bob may caring a low power device. So, that is the main objective of this Homomorphic Encryption.

So, we want this Encryption is such that this Bob this cloud should be able to this server should be able to do this, so that means the server should able to do this means server should able to do apply f of say encrypted of this is c 1, c 2 f of c 1 c 2 this is basically what f of c 1 c 2 is basically f of encrypted version of p k m 1 and then Encryption of p k m 2 and this should be the f should go inside. So, this is E of p k of f of m 1 m 2. So, this is we want, this step should go inside. So, this is the properties of Homomorphic Encryption. So, you want these properties. So, we can have m 1, m 2, m 3 in general m k. So, Bob should this should be done and this is done usually what is called in the.

(Refer Slide Time: 08:49)



So, let us go to the slide please. So, this is usually done in this val operation. So, val means we are applying f from the encrypted ciphertext f on the encrypted ciphertext and then this f should be carry forward to the plaintext, so f of m 1. So, Alice will store all the messages in the server - cloud server then Bob will send a request f, which is the function of the plaintext, which is basically a circuit, it could be plus, it could be dot, it could be union, intersection, it could be a complicated circuit margin having all the plus dot all this things.

Now, after receiving this request f, Bob will calculate this f on this ciphertext that is the eval operation Bob will do and that eval should able to generate the encryption version of f of m using this p k. So, p k is the public key of Bob. So, that if you send this to Bob, and Bob is having his own secret key. So, Bob will apply the decryption operation on this and Bob will get back the f of m. So, Bob needs only f of m not the m may be. So, Bob can get m and then Bob can apply f of in the m, but that may be expensive operation sometimes.

So, instead of that cloud is doing that, cloud is helping us to compute f of m because Bob may carry a low power device which is not capable to compute f of m. So, that computation we are doing at the cloud server. So, this is one part of the one job of the cloud server. So, this is also called can be referred as cloud computing. So, we are doing this computing at the cloud server and then we are getting the f of m by this encrypted

way, so that Bob only can get f of m. And this has another application called this Homomorphic Encryption as another application in the area of PIR - private information retrieval.

(Refer Slide Time: 11:16)

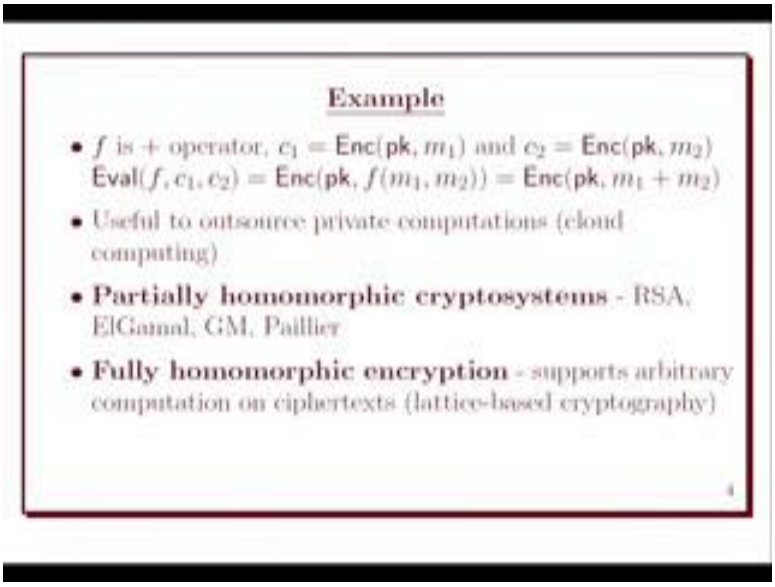
So, let us just talk about what is PIR - private information retrieval; in short, it is called PIR. So, what is basically the private information retrieval? So, suppose we have a database and we keep this database in a cloud server. So, this is the cloud server. So, this is the user. So, cloud is holding the user database, in many users are there. So, this is the database and since we have storage constant. So, we keep our data with the cloud, so that we do not need to maintain the data here because we have storage constant or something.

Now, all also we want to compute this some functionality at the cloud side in order to avoid the computation at our end, because we may be carry some mobile device which is having memory constant as well as computational constant. So, but sometimes we need to check whether cloud is really holding our all data or not; cloud may delete some of the data because if it can delete some half of the data then it is saving some storage. So, cloud can cheat us in that way, so that sometimes we need to send a query to the cloud to check whether cloud really storing our whole data, also sometimes we need to query some functionality on our data and we should able to a get the this function value on the data. So, these area is called if PIR.

And in this area, the homomorphic encryption is used very much because that is needs because the data cloud is having all having encrypted form, so encryption of m 1 because data are secret. So, called the data store in the cloud is encrypted form. So, if everything have if some function we ask then the cloud has to compute that function on this, so that function should if we takes the function of E of m 1, E of m 2 then sometimes we need this property f of m 1, m 2. So, for this we need this encryption function should be homomorphic encryption because this is the property homomorphic encryption is having. So, this is application of this PIR. So, there are also we need the Homomorphic encryption in order to have our query successfully.

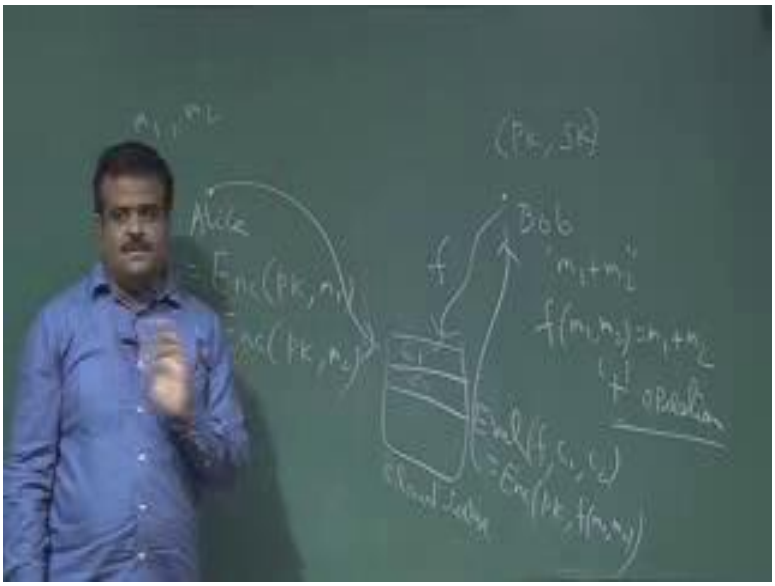
So, this is the general structure of the homomorphic encryption. So, the drawback of this is basically every time we need to interact with Bob. So, this is one of the drawback. So, we will talk about more general version of it, which is called functional encryption.

(Refer Slide Time: 14:38)



So, let us talk about homomorphic encryption in example. Suppose the function is basically the addition function or multiplication function. So, if it is addition function what we are doing our, we have a c 1 which is basically encryption of m and by using the Bob public key.

(Refer Slide Time: 15:00)



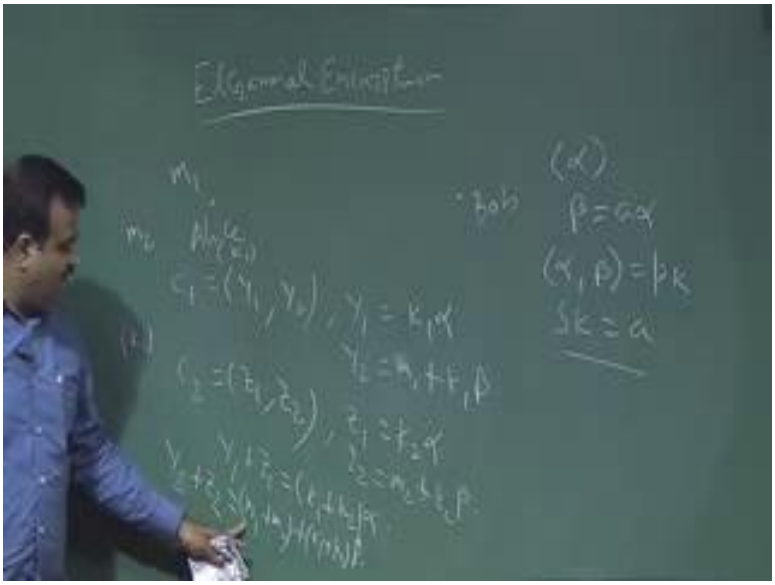
So, basically so we have Alice and Bob and this is P k, S k, so Bob public key and private key. And we have two message m 1 and m 2, and this message are encrypted using, so this is encrypted using Bob public key m 1; and c 2 is basically encrypted using Bob public key m 2. And this c 1, c 2 is stored in a cloud server; this is the cloud server.

Now, suppose Bob wants the; so Bob is not looking for the message m 1, m 2; suppose Bob is looking for the value of m 1 plus m 2, this value. So, this is basically our f; our f is just plus f of m 1, m 2 is basically m 1 plus m 2, so this plus operation. So, basically we have a plus operation. So, this Bob is looking for this thing m 1 plus m 2. So, usually in public key setup, what Bob can do, Bob can get the c 1 and can get the c 2 then decrypt it get m 1, m 2 then computing this power, but we want this to be done at the cloud server. So, cloud should able to do this some eval operation. So, this is send to cloud server and Bob send this f which is basically plus operation, now cloud should able to apply some eval operation on this c 1, c 2 and it should get this encrypted version of f of m 1 m 2.

And then cloud will send this to Bob, and Bob is having the corresponding secret key Bob will decrypt it and get m 1, m 2. So, this is the simply plus operation. So, this can achieve by using the; what is called Elgamal encryption scheme. So, in Elgamal encryption scheme what we have. So, this in Elgamal encryption scheme is a

homomorphic, but it is a partial homomorphic because it can only support this plus, it cannot support say dot.

(Refer Slide Time: 17:54)



So, let us a let us talk about Elgamal encryption scheme recap. So, in Elgamal encryption schemes what we have the key generation is so basically we have a group. Here we are taking the additive group, because we want this plus operation. So, we are taking additive group and then this is the secret key this is the generator of that group and we choose beta which is basically a alpha, where then this is basically b beta is basically public key of Bob. And then what is the Encryption. If m is the message m 1 say m 1 is the message, so the Encryption is basically c 1 is basically having two part say y 1, y 2. And what y 1 is basically we choose a Alice choose a k, k 1, so basically k 1 alpha to the power k. So, here to the power means the additive multiplicative sense here k 1 plus alpha, sorry it is basically alpha to the power k. So, k 1 alpha

Now, alpha is public, a is secret; this alpha beta is public key - this is p k; and s k is basically a is secret. So, and then y 2 is basically m 1 plus k 1 beta. So, this is y 1, y 2. And c 2 is basically z 1, z 2; c 2 is the encryption on the m 2. So, z 1 is basically some k 2 we need to choose the random number. So, k 2 alpha and z 2 is basically m 2 plus k 2 beta.

So, if you just add this, we will get basically, so if we add y 1 and z 1, so this will give us k 1 plus k 2 alpha and y 2 plus z 2 this basically m 1 plus m 2 and then plus k 1 plus k 2

beta. So, this is the val operation, where this cloud server can do. So, this is the addition can be done, but if you want to take the multiplication may be we have to take the multiplicative group, but that may not be satisfy this operation - val operation. So, that is why it is called partially homomorphic; even the RSA cryptosystem is also partially homomorphic. So, it can only support only the multiplication.

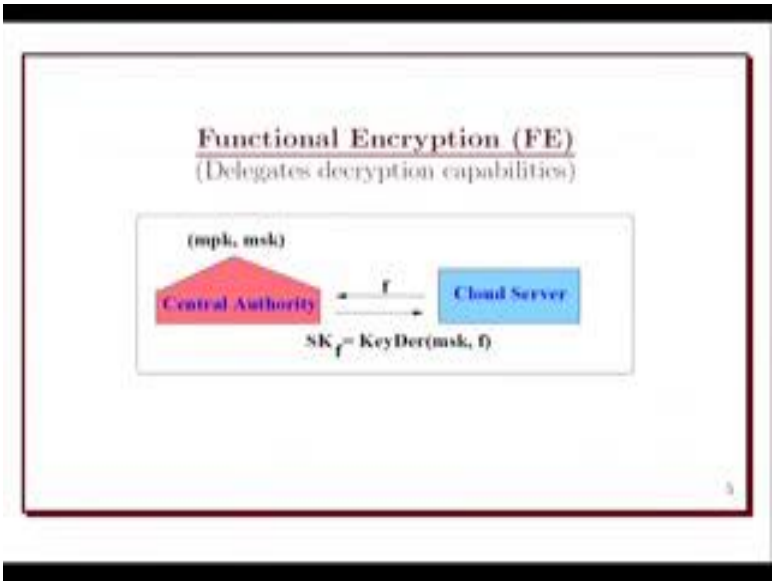
(Refer Slide Time: 21:11)

Because if you look at the RSA cryptosystem, so Alice and Bob. So, Bob is having this is the public key p k; this is the secret key s k. And RSA what we are doing in order to calculating m to the power the, this is the c. So, if you have two message c 1 and c 2, so m 1 to the power e, m 2 to the power e. So, if our f operation is the dot – multiplication, then we can just get the product c 1 and c 2 is basically m 1 m 2 to the power e. So, this is the val operation which the sever can do.

But this cannot be RSA, this is the RSA, RSA cannot be apply for plus, if the f is plus then RSA cannot be apply. So, that is why this RSA, Elgamal, and all the Paillier cryptosystem they are partially homomorphic; that means, we can support only one of the operation like dot or plus. So, it cannot support the any general circuit. So, if you have combination of this arbitrary circuit like combination of plus, dot, or, and like this, so that is called fully Homomorphic encryption which support the any general circuit instead of just the dot or plus. So, fully Homomorphic encryption scheme all almost all fully homomorphic encryption scheme is a lattice based scheme. So, lattice based

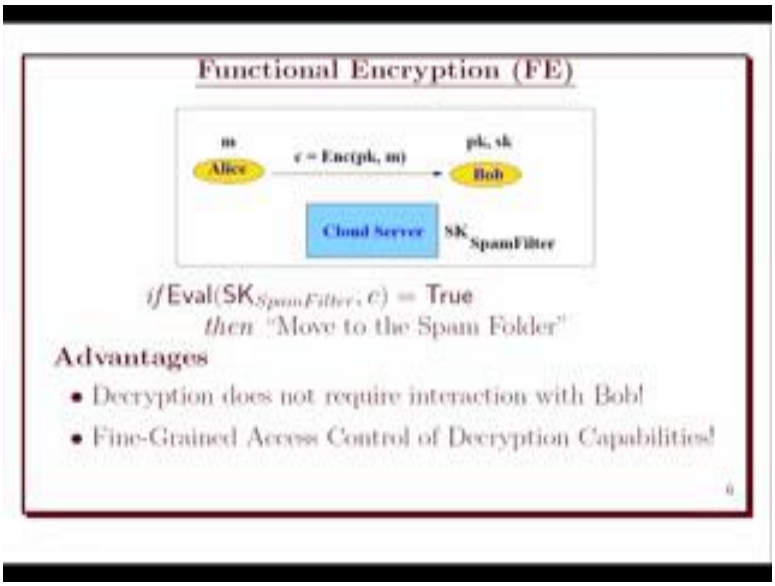
scheme. So, if you have interest, you can look at those scheme, but those for those scheme we need to have a knowledge of lattice.

(Refer Slide Time: 23:00)



So, now let us come to the functional encryption scheme, which is the generalized version of the homomorphic encryption. So, here what we are doing? We have a central authority, and these authority we are - Cloud server is sending a functionality f which is and this authority is having a at the setup phase we may having a public keys master secret key and master secret key and master public key. And this after receiving this f functionality this is applying a key derivation algorithm which is using the master secret key and this functionality f and getting the secret key corresponding to that functionality. So, if you have the ciphertext, and if you use this secret key to decrypt it and that will able to give us that functionality information and this is usually done by the cloud server. So, here the involvement is Bob is not required. So, we will take some example of this functional encryption.

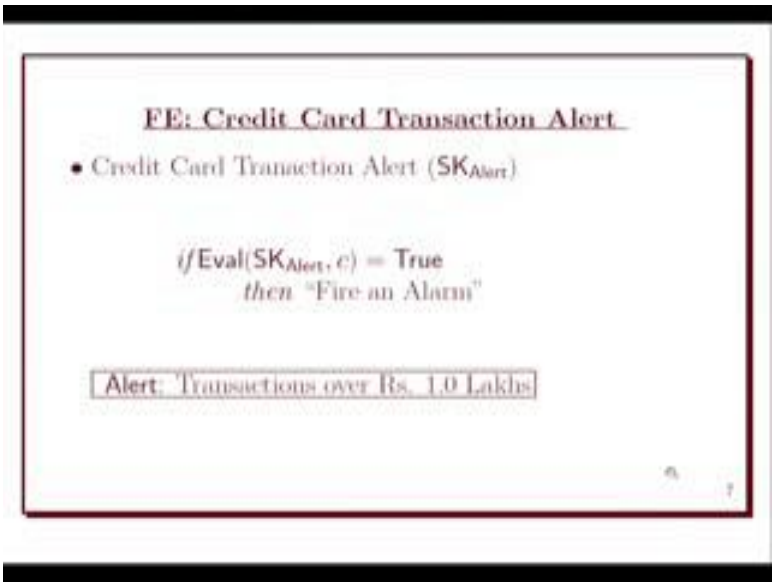
(Refer Slide Time: 24:08)



Suppose, we choose this functionality as the spam filter, so we request the cloud server, request the central authority by sending this functionality spam filter and ask to get the and request for a secret key corresponding to the that functionality. So, that clouds that trusted party will give issue this S K spam filter by applying this key derivation algorithm using the master key this master secret key and this functionality.

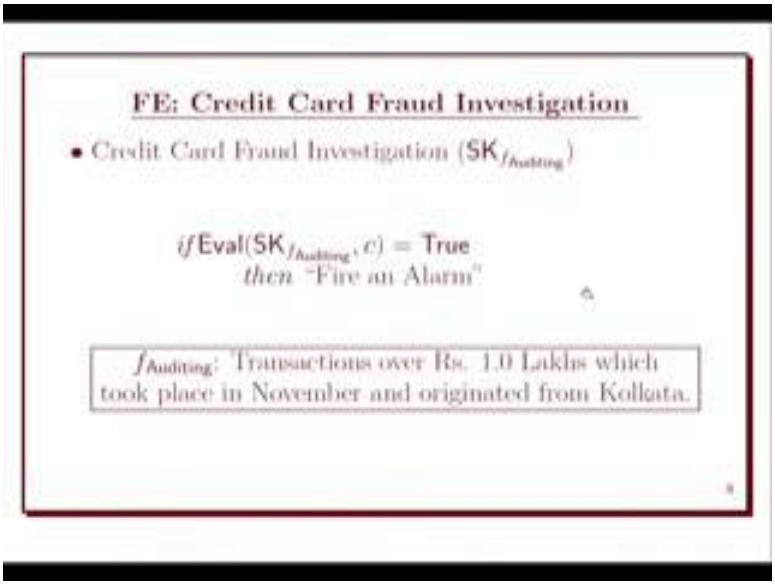
So, after receiving this, so it can apply. So, it once it has this secret key corresponding to that functionality spam filter; and if you have the ciphertext then we apply the eval operation, and if it is true then we move this to the spam folder. And here the advantage of this is it the decryption does not require interaction to with the Bob, and also we can have the fine gained access control or access structure for decryption capability. So, this is the more general version.

(Refer Slide Time: 25:33)



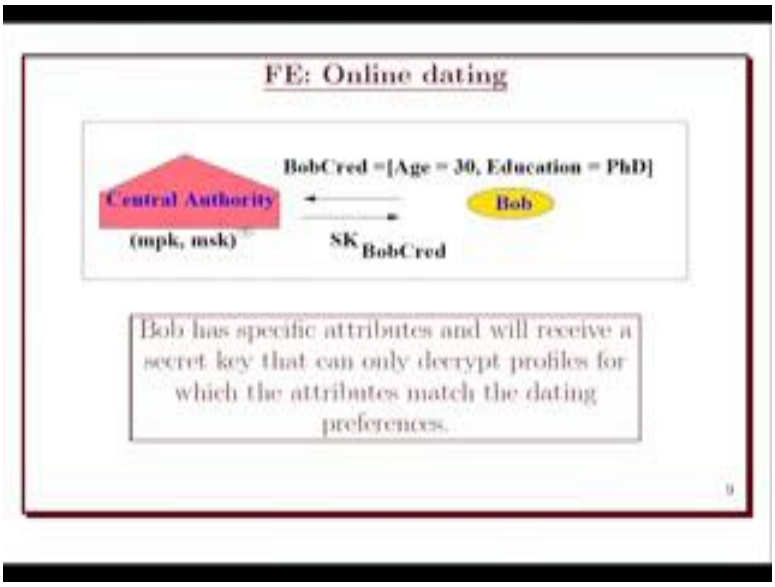
So, we have another example credit card transaction. So, this is say we have alert this is the functionality if the transaction is over 1 lakh, then we must fire an alarm. So, this is the functionality. So, this corresponding to this functionality suppose cloud server got a secret key corresponding to this functionality. So, if we have a corresponding ciphertext, if the ciphertext is corresponding to that message that then our transaction is more than 1 lakh, then it will be true and then we fire an alarm.

(Refer Slide Time: 26:10)



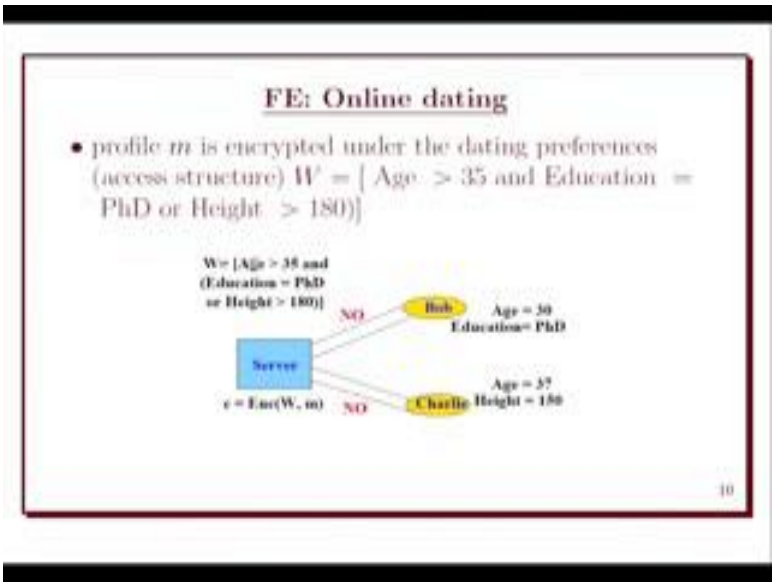
And we can we can have more refinement on this; suppose we have this functionality like auditing like credit transaction is over 1 lakh and this transaction is done in November and it is from Kolkata. So, this is a more complicated circuit. So, then the corresponding to this functionality suppose we got this cloud got this secret key. So, if that is the transaction then the corresponding ciphertext will give with this eval will give us true then we will have alarm. So, this is the applications.

(Refer Slide Time: 26:45)



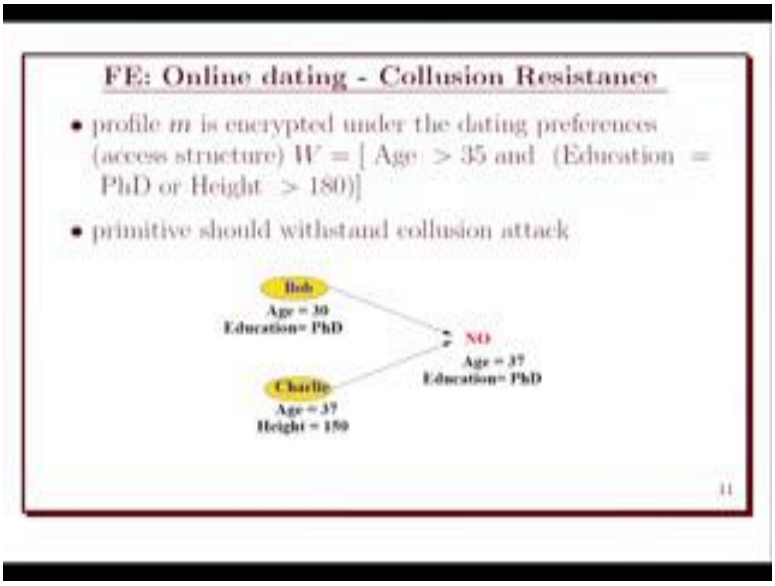
So, another application could be online dating. Suppose, this is the Bob credential Bob age is 30, education is PhD. So, based on this, Bob will send this to the trust central authority and central authority give the secret key corresponding to the Bob credential using this central the master secret key. And then the Bob has the specific attributes and we will receive a secret key that can only decrypt profile for which the attribute match the dating preference.

(Refer Slide Time: 27:21)



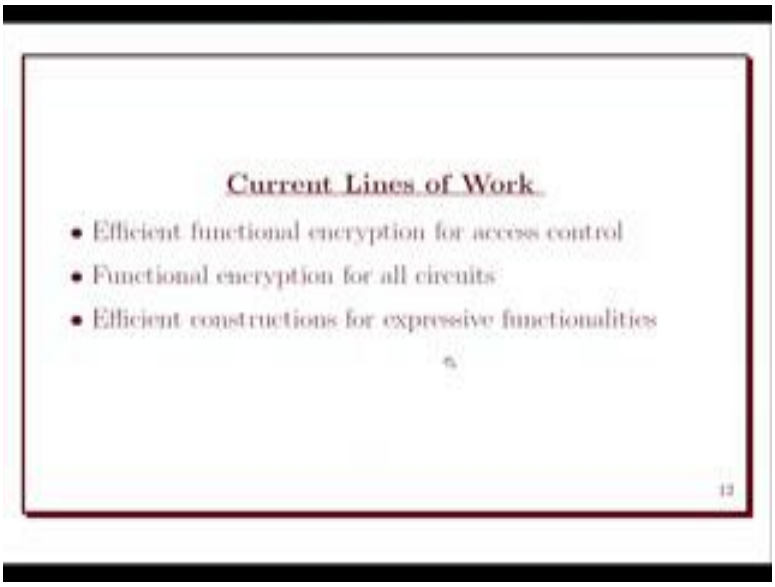
Now, suppose, we have a message for which the age is greater than 35 and this is the circuit and education is PhD or height is greater than this. Now, suppose Bob is having this age is 30 and education is PhD, so here we need age is greater than 35 and this is and. So, Bob cannot, so eval will that if you apply eval using the Bob credential it will be no. So, similarly for the Charlie, because the Charlie age is more; although the height is also not, so this is one general circuit.

(Refer Slide Time: 28:02)



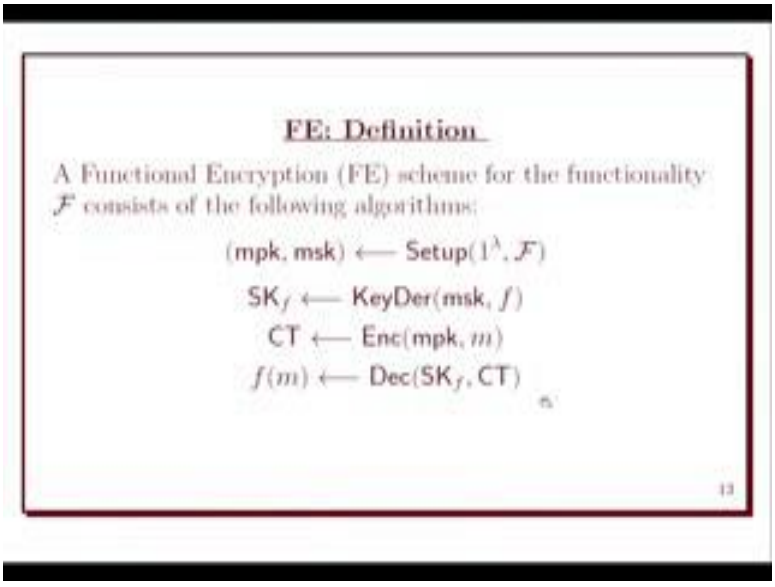
And then if they both combined do you want this property also. If they both combined Charlie and Bob then they are satisfying all that properties, but we do not want this to be. So, this is the collision attack. So, we want to prevent the collision attack in our functional encryption scheme.

(Refer Slide Time: 28:20)



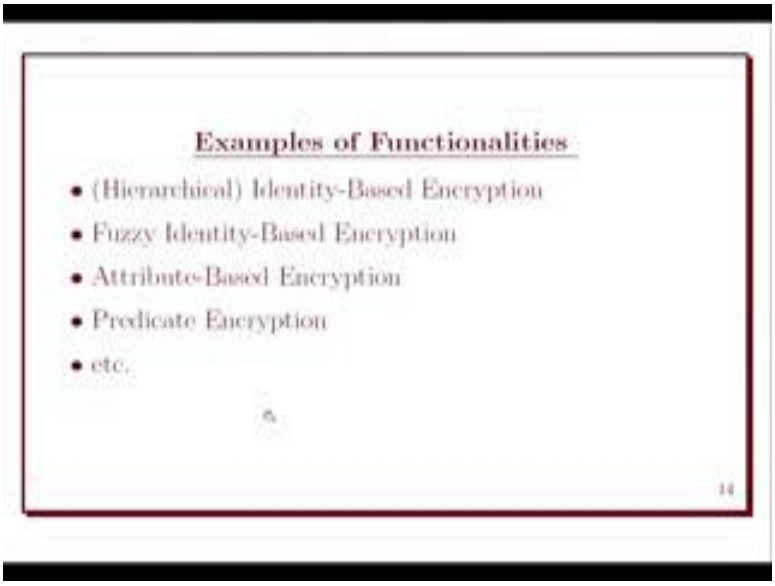
So, the current line of work in this functional encryption scheme: so efficient functional encryption for access control, functional encryption for all circuit general circuit; so efficient construction for expensive functionalities.

(Refer Slide Time: 28:35)



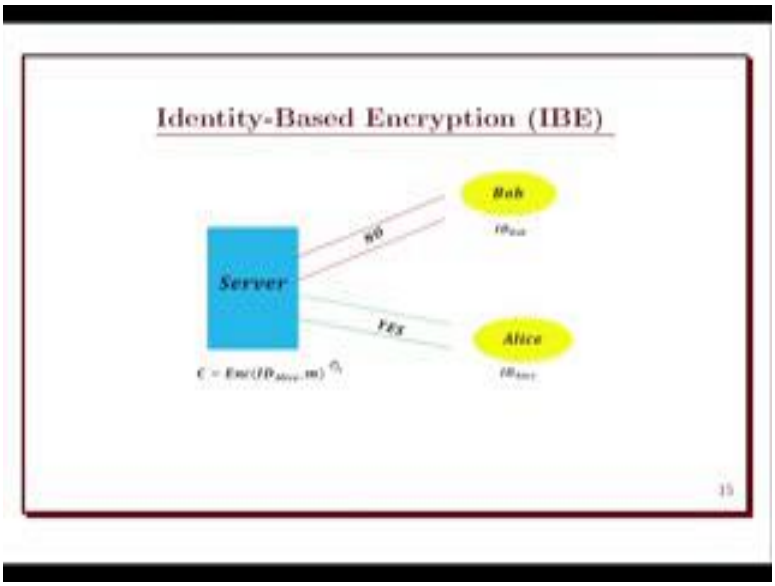
So, this is the general overview of any functional encryption scheme. So, we have a setup phase, setup phase will generate the master public key, and the master secret key for the central authority. And there is a key derivation algorithm which takes the input of that functionality of the circuit, and that will use this master secret key so that means, this algorithm can only run by the trusted authority and it will give us the secret key based on that functionality. And this is the ciphertext or ciphertext header. So, the ciphertext is encrypted encryption of the plaintext using the master public key. Now, this is the val operation. Now, if this is satisfying this decryption is true then we will able to get the f of m. So, this is the more general structure of the functional encryption.

(Refer Slide Time: 29:41)



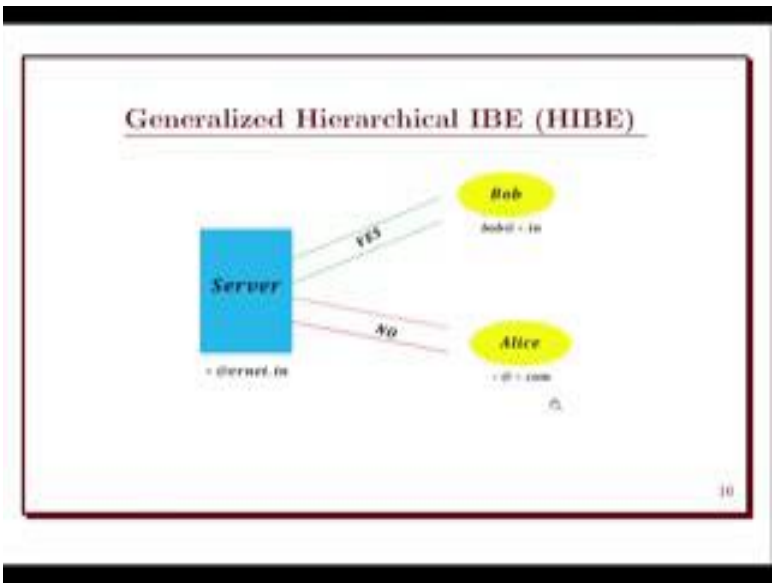
So, we have some example of this functionality id-based encryption, fuzzy id-based encryption, attribute-based encryption, predicated encryption.

(Refer Slide Time: 29:50)



So, this is the id-based Encryption. So, if we have used the id of the Alice as functionality then only Alice should able to decrypt it not the Bob.

(Refer Slide Time: 30:03)



And this is the generalized hierarchical id-based Encryption, so HIBE. So, here suppose the functionality is dot star at the rate internet dot in. Now, suppose if Bob is having this credential then Bob can able to do, because Alice is having dot com. So, this is the hierarchical IDBE. So, these are the example of this functional encryption scheme. So, we have, we can use the inner product based inner product for having a functional

encryption scheme. So, those are some current, this is the area of current research. So, if you have interest, you can look at the some of the research paper on the functional encryption scheme.

Thank you.