

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

Lecture - 38
Key Management

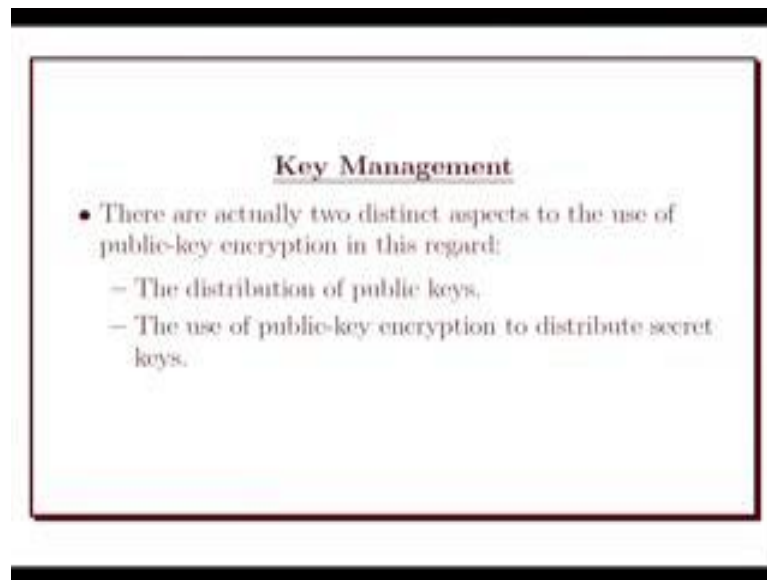
So, we talk about key management, how we can distribute the public key if you are in public key setup how we can distribute the public key. So, in public key setup, everybody has a pair of key, public key and private key or public key and secret key.

(Refer Slide Time: 00:40)



So, if party is Alice, Bob, Bimol, Palash. So, if they are in public key setup, so everybody has their public key, private key. So, this is the Bob public key then e_p, d_p . So, everybody is having their public key private key period. And now suppose Alice want to send a message to Palash. So, Alice has to get Palash public key. So, now the question is how Alice will get Palash public key. How the public key? This is a part of the key management how it distribute the public key that is one issue. Now suppose Alice want to send a message to Bob. So, this is d_A . So, Alice need to get Bob's public key that is e_b . So, how Alice will get Bob public key, so this is the part of the key management.

(Refer Slide Time: 01:51)

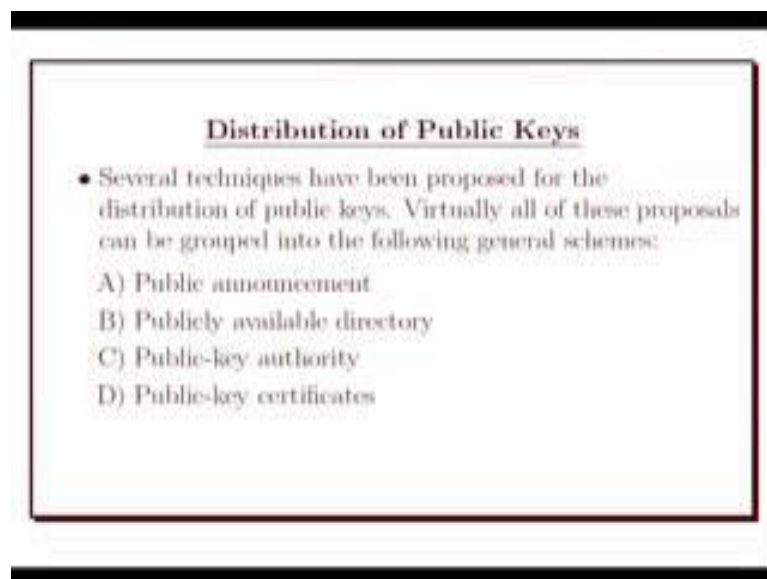


Key Management

- There are actually two distinct aspects to the use of public-key encryption in this regard:
 - The distribution of public keys,
 - The use of public-key encryption to distribute secret keys.

So, the key management in this area will discuss two distinct aspect like how to distribute these public key, and we will see how we can, if you are in a public key setup how we use the public key to share the secret key.

(Refer Slide Time: 02:14)



Distribution of Public Keys

- Several techniques have been proposed for the distribution of public keys. Virtually all of these proposals can be grouped into the following general schemes:
 - A) Public announcement
 - B) Publicly available directory
 - C) Public-key authority
 - D) Public-key certificates

So, first part, distributing of public key, how we can distribute the public key? There are basically four techniques we will discuss. The first one is public announcement. So, once I have a public key, I will keep on announce. Suppose I have a mike, so I will announce this is my public key this that is some sort of public announcement. And the second

method is publicly available directory; instead of public announcement, we can maintain a directory where I can put my public key and that directory should be in a public domain, so that everybody should be able to access it.

And the public key authority, so if we keep the directory so public then anybody can change their and do some changes in that public key, so that will create some problem for the communication. So, for that, we can have a third party which is public key authority and we can ask this third party to maintain these directory. So, some sort of trusted authority - the third party trusted party will be there.

And the next one is public key certificate. So, if we always ask this third party a give me the public key of Bimol, I want to send a message to Bimol. Then Bob is telling ok, give me a public key of Palash, I want to send a message to Palash. So, there is a bottleneck in at the third party end at the public key authority end who is having the public key directory because he has to answer all the queries. So, there will be a bottleneck bandwidth problem will be there.

So, to solve this, what we can do? We can issue a certificate. So, this third party or the trusted party or authorized party, they can issue a certificate to the each users and that certificate will contained the public key. So, if I have to communicate with Bimol. So, I will ask Bimol to show his certificate. So, Bimol will give me his certificate. So, from that certificate, I will come to know Bimol's public key. So, this is the way we can avoid to communicating with these third party to avoid that bottleneck. So, this is the public key certificate.

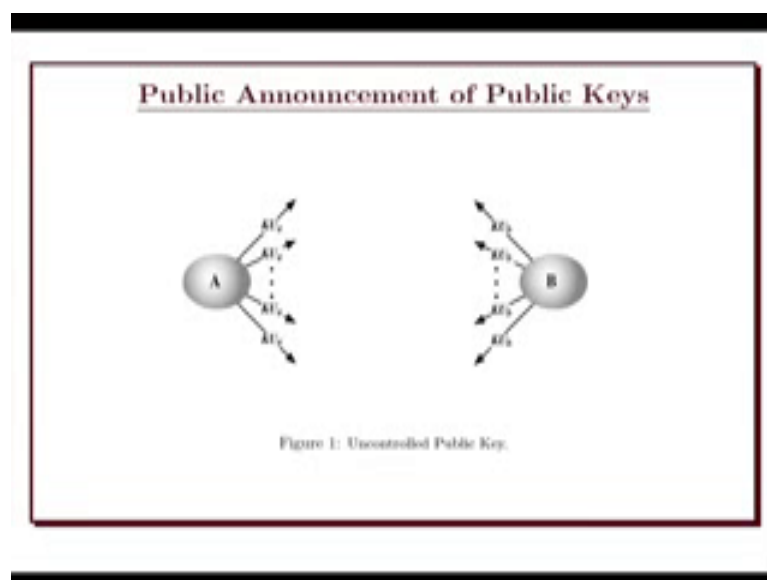
(Refer Slide Time: 04:42)

Public Announcement of Public Keys

- The point of public-key encryption should be that the public key is public.
- Thus, if there is some broadly accepted public-key algorithm, such as RSA, any participant can send his or her public key to any other participant or broadcast the key to the community at large (figure 1).
- Although this approach is convenient, it has a major weakness. Anyone can forge such a public announcement.

So, let us come to this is yeah. So, this is the first part the public announcement of the public key. So, here the public key encryption should be that public key should be public for the public key encryption. So, it could be RSA, we can setup the RSA and we run the RSA key generation, and we got the public key private key pair.

(Refer Slide Time: 05:05)



And then everybody can announce their public key. So, this is A's public key, so he keep on announcing public key. So, this is the uncontrolled public key. So, it is having the drawback that. So, if I keep on announcing then when I mean I may not sending a

message now, and the receiver is announcing the public key he is our public key now. So that is the problem I may send later on, so that time maybe the person is the receiver is not announcing yeah his or her public key.

(Refer Slide Time: 05:43)

Publicly Available Directory

- A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys.
- Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organisation (figure 2).

So, to work on this problem what we can do? We can maintain a publicly available directory.

(Refer Slide Time: 05:51)



So, one can maintain a directory like, so it is basically a file this is a public key directory, so PKD. So, it is basically contained the name and the corresponding public key. So,

Alice, Bob, Bimole, Palash and once we have a public key, so we will publish their will go to that directory and put that value suppose this is the public key of Alice; this is the public key of Bob like this. So we will maintain this directory which is called PKD public key directory.

Now, suppose this directory is available on a public domain. For example, it could be a website in a public website we keep this directory I mean this file has been maintained. So, suppose now Alice wants to send a message to Palash. So, what Alice will do? So, Alice will access this directory and Alice will so this is e Palash - e P. So, Alice will get Palash public key and Alice will encrypt the message using Palash public key and send it to Palash. So, this is the way we have to get the public key of the receiver. So, sender has to get the public key of the receiver. So, what sender will do? Sender will access that public key directory, and go to that corresponding field where the receiver public key is there. So, get the receiver public key and encrypt the message using the receiver public key and send the ciphertext to the receiver.

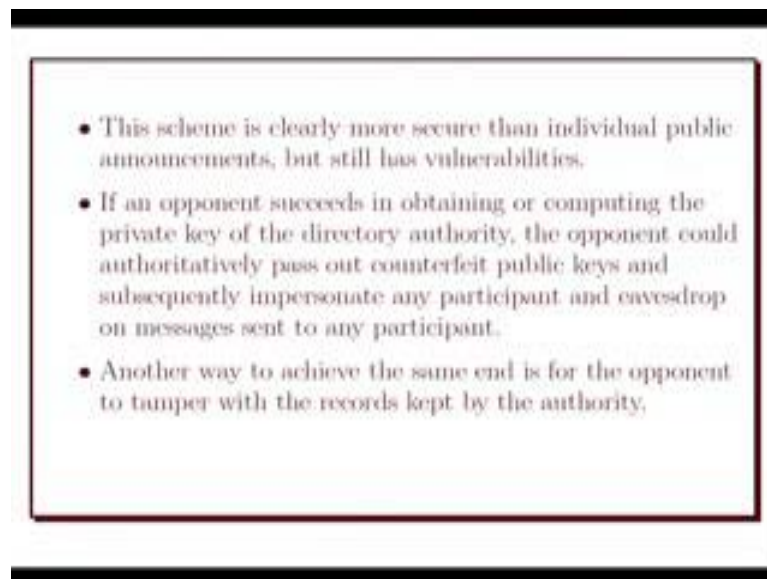
Now, if we keep this just like this then there is a problem. The problem is so if it is openly accessible to anybody. So, the attacker can go and can change the public key of a person. So, attacker change the public key of this Bob say, so we can change just two bit then whole system will be disturb. Then if Alice encrypt a message using this wrong public key of Bob, then Bob will not able to decipher it, Bob will not able get the plaintext which is sending by the Alice. So, this we cannot keep just public just like this.

So, then what is the solution? Solution is we can hire a third party which is a authority or the trusted party who can maintain this. So, a third party third party or some authority - trusted authority who can maintain this file, this is PKD dot doc say some file, who can maintain this file. And if I have to get the public key of somebody I will ask this authority he give me the, because then otherwise this file can be made read only mode then I can only read the file, I should not able to write anything on the file. So, if I have to add my public key in the file. So, suppose I have changed my public key. So, I am going to, so I was having a public key over here, I was having a public key over here now I change the public key. So, this key change is a common thing because say for example, suppose my corresponding secret key is revealed, and I am fearing that may be it got it known to somebody. So, now I am going to change this public key private key pair.

So, again run the setup phase of the public key encryption, public key cryptosystem and then I got a new public key. So, I have to change the whole public key. So, what I will do, I will contact the this trusted authority who is having this control on this who is maintaining this public key directory and I will send my public key, and the trusted authority just change that new public key, replace this replace my old public key by the new public key.

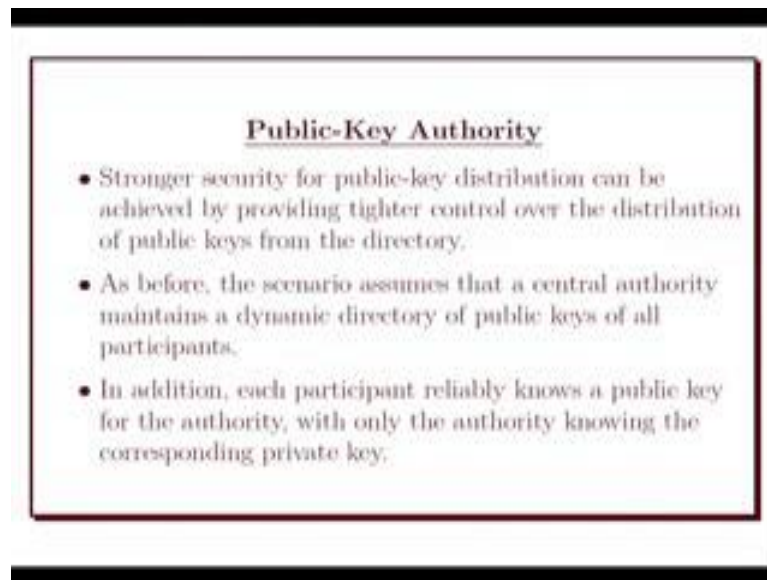
So again that so how trusted authority can publish this public key in a book form like we have a telephone directory, telephone book, the big yellow book telephone book. So, like this periodically, this trusted authority can publish this book in order to have that bottleneck like if everybody is asking - what is the public key of my sender and my receiver. So, it will be a headache for the trusted party. So, to avoid that it can publish the public key over a hard copy or some sort of pdf file or something, so those are basically read only, so nobody can change there. So, this is one way.

(Refer Slide Time: 11:41)



And another way is. So, this scheme is clearly more secure than the individual public announcement, but still it has problem.

(Refer Slide Time: 11:54)

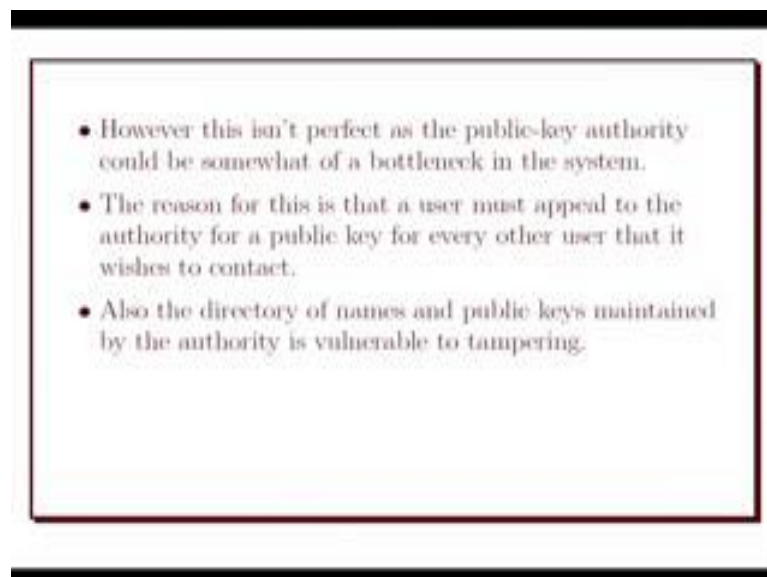


Public-Key Authority

- Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory.
- As before, the scenario assumes that a central authority maintains a dynamic directory of public keys of all participants.
- In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.

So, this is the scheme. So, we are hiring a public key authority which will be having this maintaining this public key directory.

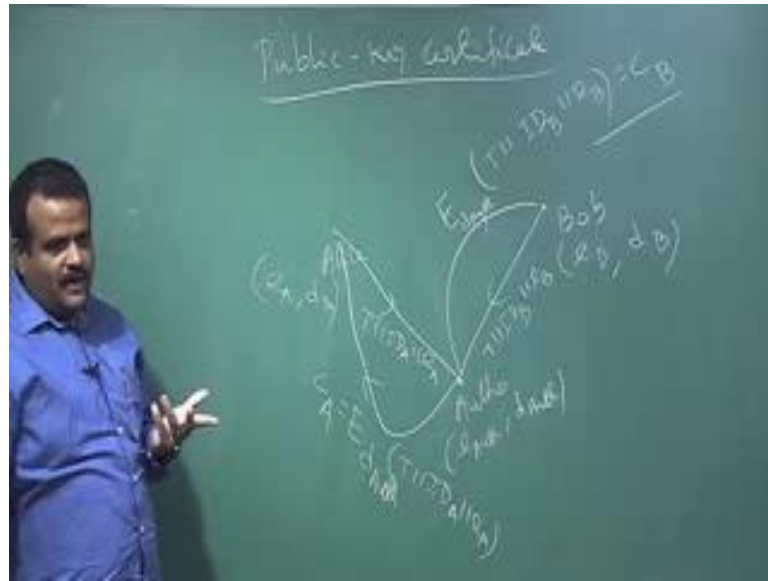
(Refer Slide Time: 12:07)



- However this isn't perfect as the public-key authority could be somewhat of a bottleneck in the system.
- The reason for this is that a user must appeal to the authority for a public key for every other user that it wishes to contact.
- Also the directory of names and public keys maintained by the authority is vulnerable to tampering.

So, this is basically we are trusting the authority, now and also authority having a bottleneck to if he has to give the public key to all the asking sender, for to avoid this, so we introduce the concept of certificate, public key certificate.

(Refer Slide Time: 12:38)



So, this is basically public key certificate. So, basically this is an alternating approach to get the public key. So, the trusted party the public key authority will issue certificate and each certificate containing a public key and the other information like some timestamp and created by a certified authority, and it is giving to the participant with the matching private key. So, suppose Alice and Bob, so Bob is having this public key private key pair. And there is a trusted party here say authorized party Autho we can say, who is giving the certificate. So, what Bob can do? Bob can send a message to this; Bob can send Bobs ID, ID of Bob along with the public key of the Bob to get the certificate. Now, the authority is having its own public key private key pair Autho, d Autho. So, this is the authorities public key private key pair.

So, now, Bob has to get its certificate public key certificate. So, what Bob will do? Bob will send along with a timestamp, some session say for today's date, it could be date; for today, this is the certificate for today, so it may valid for some time, so will come to a standard of the certificate. So, this is basically a time, and this is the ID of that person participant and this is the public key, it is send to authority. And what authority will do? Authority will generate the certificate by signing on this. So, signing means authority has to encrypt this using authorities secret key so T ID of B e B. So, this is the certificate of Bob.

So, if Alice has to get the certificate Alice has to do the same thing, Alice has to send. So, if Alice has to send a certificate from the authority Alice has to generate Alice public key private key pair, it could be RSA it could be anything. So, it is a public key cryptosystem. So, now, Alice send a timestamp along with ID of Alice and along with the public key of Alice, and then the authority will send back the certificate which is basically C of A - Alice certificate which is basically the digital signature on this message I mean, so this is basically encryption of so digital signature. So, it has to be by the private key of the authority T I D of A e of A. So, this is the certificate Alice will receive after asking from the authority. So, in this way, everybody is having their own certificate.

Now, so this is the certificate generation from the authority.

(Refer Slide Time: 16:37)



Now, after getting the certificate what Alice, now suppose, Alice wants to communicate with Bob, so Bob is having e B d B public key private key pair. Now, Bob having the certificate which is basically signed by the certified authority. So, this is the digital signature E of d auth some timestamp ID of Bob and the public key of Bob. So, now, Alice wants to send a message to Bob. So, Alice ask Bob certificate. So, Bob will send the certificate to Alice C A. And what Alice will do? Alice has to verify this is Bob certificate or not. So, for that, so this is signed by the authority. So, Alice has to check this signature. So, for that Alice need to decrypt it using authorities public key. So, what

Alice will do? Alice will decrypt it using authority public key on this C A. So, this is basically give us this part. So, this is basically D of e auth of and this is basically E of d auth T e of B. Now, these two will cancel it will give us T of I D of B and e of B.

So, now Alice will check the time, this time is it is certificate is current certificate or not some sort of timestamp is there. So, Alice will check this time and also Alice will check this ID of the Bob. So, this is the ID of the Bob. So, this is really a Bob certificate and then Alice will get this e B the Bob public key by seeing the certificate. Now, after getting the Bob public key Alice can choose a message and encrypt the message using Bob public key and send it to Bob. So, this is the way we can avoid to contact the authority in order to get the public key of the receiver just by seeing the certificate we can get this. So, similarly if there are other party like say Rob. So, everybody is having their, everybody every participant is having their certificate, which is issued by the trusted party or the authority.

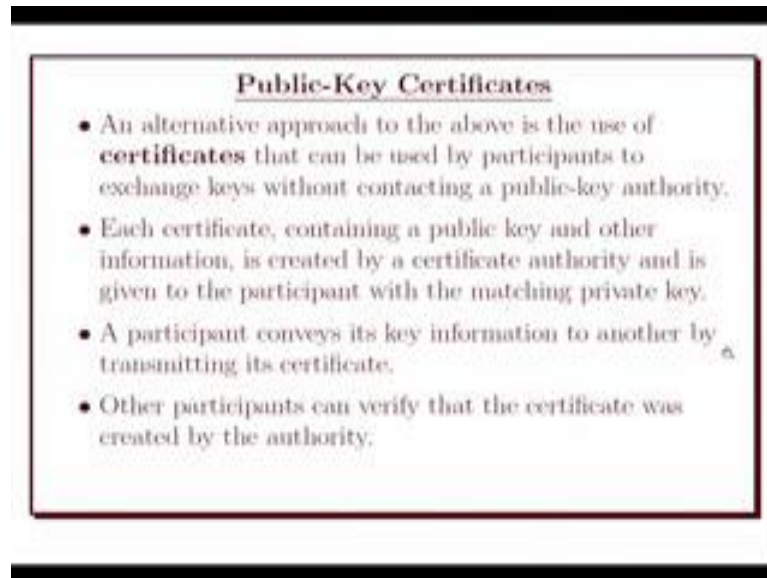
(Refer Slide Time: 20:04)



So, Alice, Bob, Palash, Rob, everybody is having their own certificate C A, C P, sorry this is C B, CA and C R. So, now this was issued by the trusted party or the authority when you ask for the certificate. And this certificate contains what? This contain the timestamp the valuation of the certificate, how long it is valid kind of thing, and also this contain the ID of the participant, and it contains the public key of that participant and it is digitally signed by the authority.

And now if a person wants to communicate, suppose Rob wants to communicate with Palash. So, Palash has to provide the certificate to Rob and Rob will check whether this is the Palash certificate by checking the time and ID and then Rob will get the Palash public key and Rob will encrypt the message using Palash public key and sending to Palash. So, this is the way we just generate the certificate.

(Refer Slide Time: 21:32)

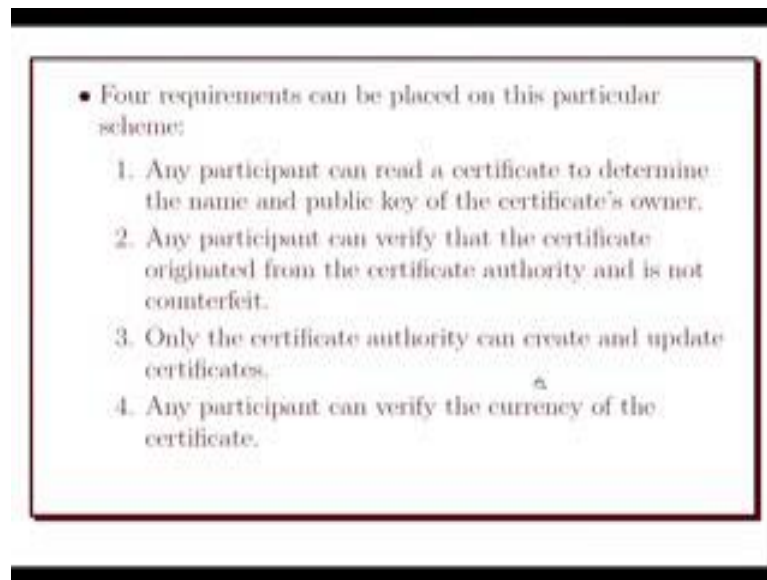


Public-Key Certificates

- An alternative approach to the above is the use of **certificates** that can be used by participants to exchange keys without contacting a public-key authority.
- Each certificate, containing a public key and other information, is created by a certificate authority and is given to the participant with the matching private key.
- A participant conveys its key information to another by transmitting its certificate.
- Other participants can verify that the certificate was created by the authority.

So, as we said each certificate containing a public key and the other information is created by the certificate authority, and it is giving to the participant and with the matching private key. So, participant conveys this key information to another by transmitting the certificate. Other participant can verify the receiver sender can verify that the certificate was created by the authority.

(Refer Slide Time: 22:03)



So, four requirements can be place here. Any participant can read a certificate to determine the name and the public key of the certificate's owner; this is the first requirement. And the second requirement is any participant can verify the certificate originated from the certified authority and it is not counterfeit. So, it is not a false certificate. So, this verification can be done should be done by the any participant. Only the certified authority can create and update the certificate. So, suppose currently I have a certificate, now I am changing the public key or I want to change the, because I am guessing that my corresponding secret key has been revealed. So, what I do?

(Refer Slide Time: 23:09)

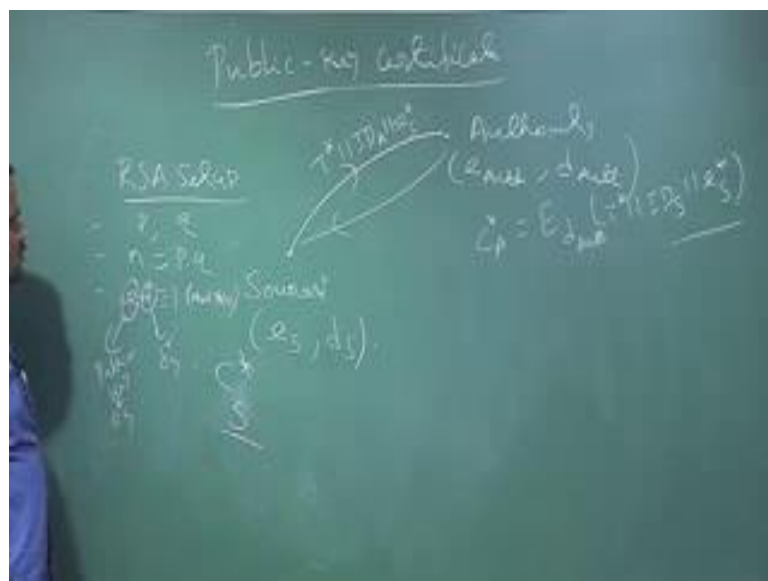


So, this is the trusted party or the authority, so which is having. So, this is the public key of the authority private key of the authority. So, suppose this is me. So, I was having a private key and public key pair. I was having also a certificate which was containing this was digitally signed by the authority and was having a timestamp along with my ID and my public key.

Now, suppose I feel to change my public key, because I am fearing that may be my symmetric secret key is revealed. So, I want to again run that setup. Suppose we are in RSA setup, RSA cryptosystem. So, I want to run RSA setup. So, again I do this p, q and then n is equal to p into q like this, then I choose two e, b such that e is congruent to $1 \bmod \phi(n)$. So, ϕ is basically p minus 1 , q minus 1 . And then this is my public key. This is basically e, n and this is my corresponding secret key new secret key.

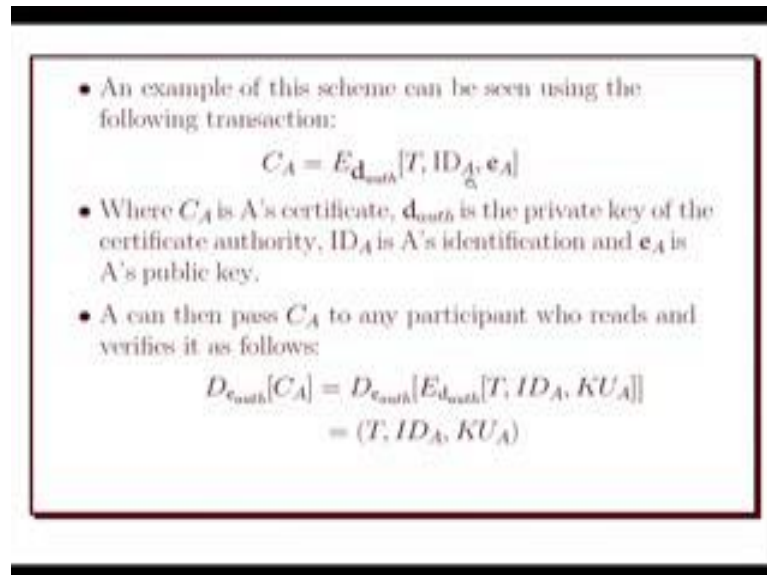
Now, this I want to so this setup I run because I feel that I should change it I should change my public key. So, I run this setup, and I got the new setup public key private key pair. Now, I send a request for a new certificate to the authority. So, I have a timestamp may be today's date along with the time and I send my ID and I send this is the say new the current new public key. So, what authority will do? Authority will do the same thing authority will digitally sign on this. So, E of d of auth along with T star ID of s along with s star and this authority will send it to me. So, now, my certificate is this C s is now replaced by C S star.

(Refer Slide Time: 25:59)



So, this way I can change the certificate, I can update the certificate and this can be only done by the authorized person.

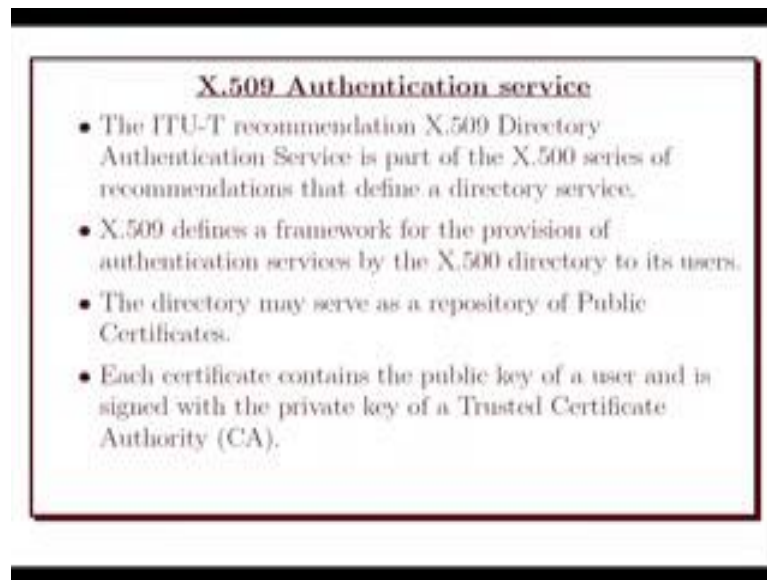
(Refer Slide Time: 26:16)



- An example of this scheme can be seen using the following transaction:
$$C_A = E_{d_{auth}}[T, ID_A, e_A]$$
- Where C_A is A's certificate, d_{auth} is the private key of the certificate authority, ID_A is A's identification and e_A is A's public key.
- A can then pass C_A to any participant who reads and verifies it as follows:
$$D_{e_{auth}}[C_A] = D_{e_{auth}}[E_{d_{auth}}[T, ID_A, KU_A]] \\ = (T, ID_A, KU_A)$$

So, this is the example. So, this is the timestamp, this is the ID of A, this is the public key, and this is digitally signed by the authority. And so this CA is the certificate of a and d auth is the public private key of A, this, the authority is digitally signing on this a message. So, this message is containing timestamp ID and the public key of the participant. And then this can be verified by this way which we have discussed. So, this is just so this is the public key of the authority which is public, anybody can have the access of this. So, he or she will take this public key and decrypt this certificate of A and check getting this and check whether this is timestamp is ok and ID and get the secret key of this.

(Refer Slide Time: 27:14)

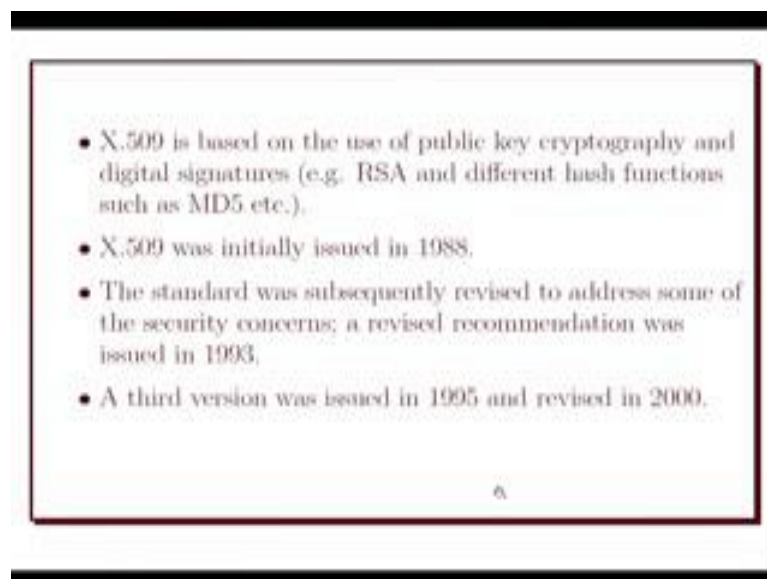


X.509 Authentication service

- The ITU-T recommendation X.509 Directory Authentication Service is part of the X.500 series of recommendations that define a directory service.
- X.509 defines a framework for the provision of authentication services by the X.500 directory to its users.
- The directory may serve as a repository of Public Certificates.
- Each certificate contains the public key of a user and is signed with the private key of a Trusted Certificate Authority (CA).

Now, we will come to a standard certificate standard, which is called X 509. So, it is basically I mean standard. So, since it is a standard, it must have some criteria like what should be the, what are the field it should have? So, we will just go to that.

(Refer Slide Time: 27:43)

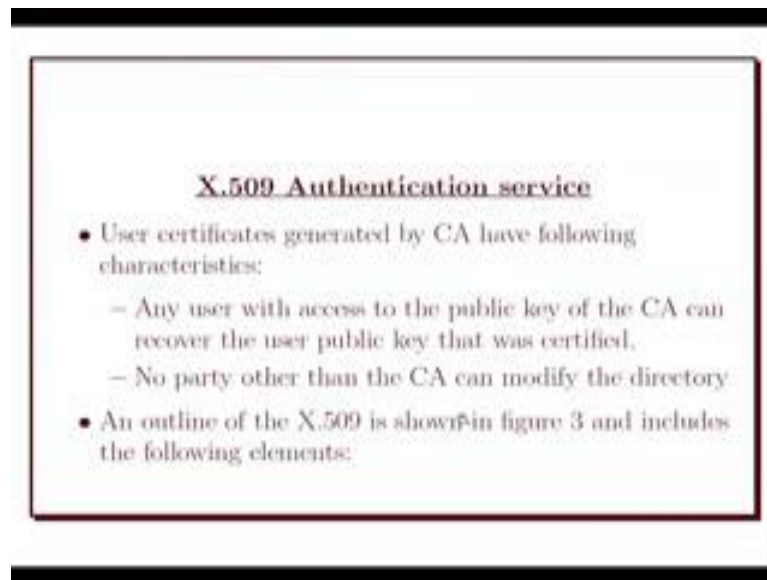


- X.509 is based on the use of public key cryptography and digital signatures (e.g. RSA and different hash functions such as MD5 etc.).
- X.509 was initially issued in 1988.
- The standard was subsequently revised to address some of the security concerns; a revised recommendation was issued in 1993.
- A third version was issued in 1995 and revised in 2000.

6

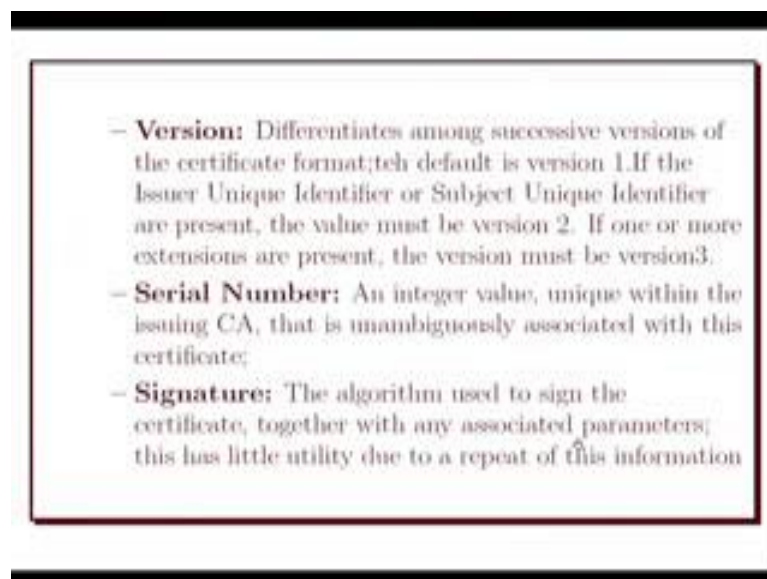
So, this is a certificate standard. So, this is basically used RSA and MD 5. And it was initially issued in 1988. And then it has some variation. So, this is the next variation 1993, and then this is the revised variation again after 1997.

(Refer Slide Time: 28:01)



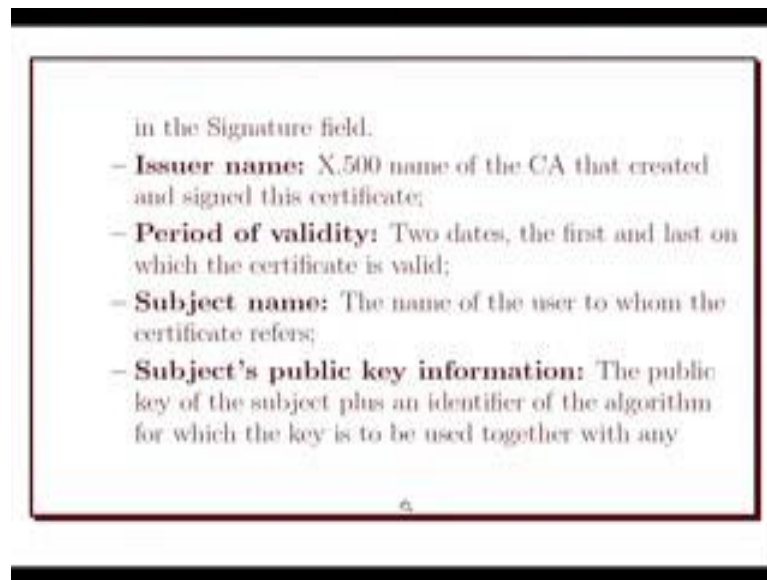
So, this is the user certified generated by the CA of the following characteristic. Any user with access to the public key of the CA can recover the, so this is the basic criteria of a certificate. So, no party other than CA can modify this directory.

(Refer Slide Time: 28:23)



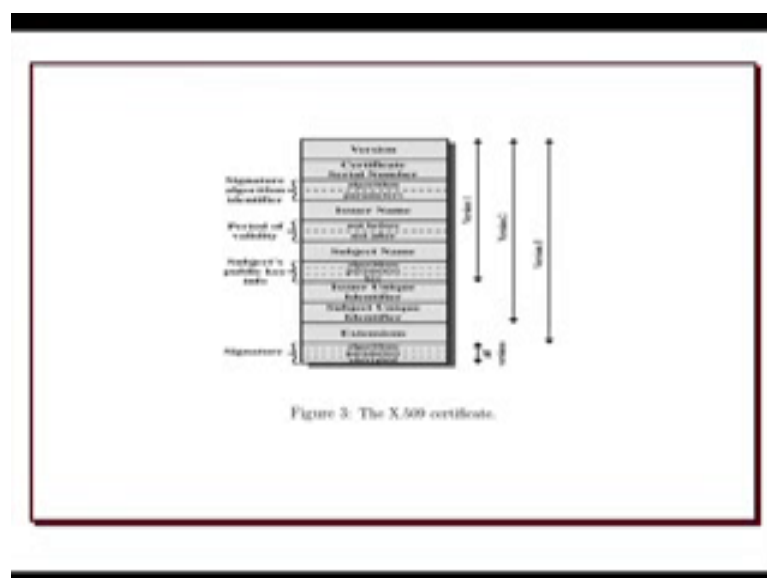
So, let us go to the figure. So, it has this, these are the field it has. So, version, which of the version, serial number, signature, signature of the trusted party, and then issuer name, this is certificate is X 500, so that is the name of the issuer we can say.

(Refer Slide Time: 28:36)



Period of the validity, so how many days you want the certificate to be valid, so that field also be needed. Then the subject name, the name of the user or name of the participant, then the public key information we need to keep the public key into the certificate. So, that is the public key of the participant, and the issuer unique identification, then the subject unique identification, extension, so then the signature.

(Refer Slide Time: 29:16)



So, this is the picture. So, this is basically having these are the fields, the version, certificate serial number, the signature algorithm we are using whether we are using the

RSA signature, Elgamal signature that should be mentioned there, issuer name, period of validity. So, this is basically and we should have the public key of the participant. So, this is basically called X 509 certificate standard. So, this is the certificate standard and everybody can have their own certificate in order to communicate in a public key cryptosystem, in order to encrypt decrypt in a public key cryptosystem.

Thank you.