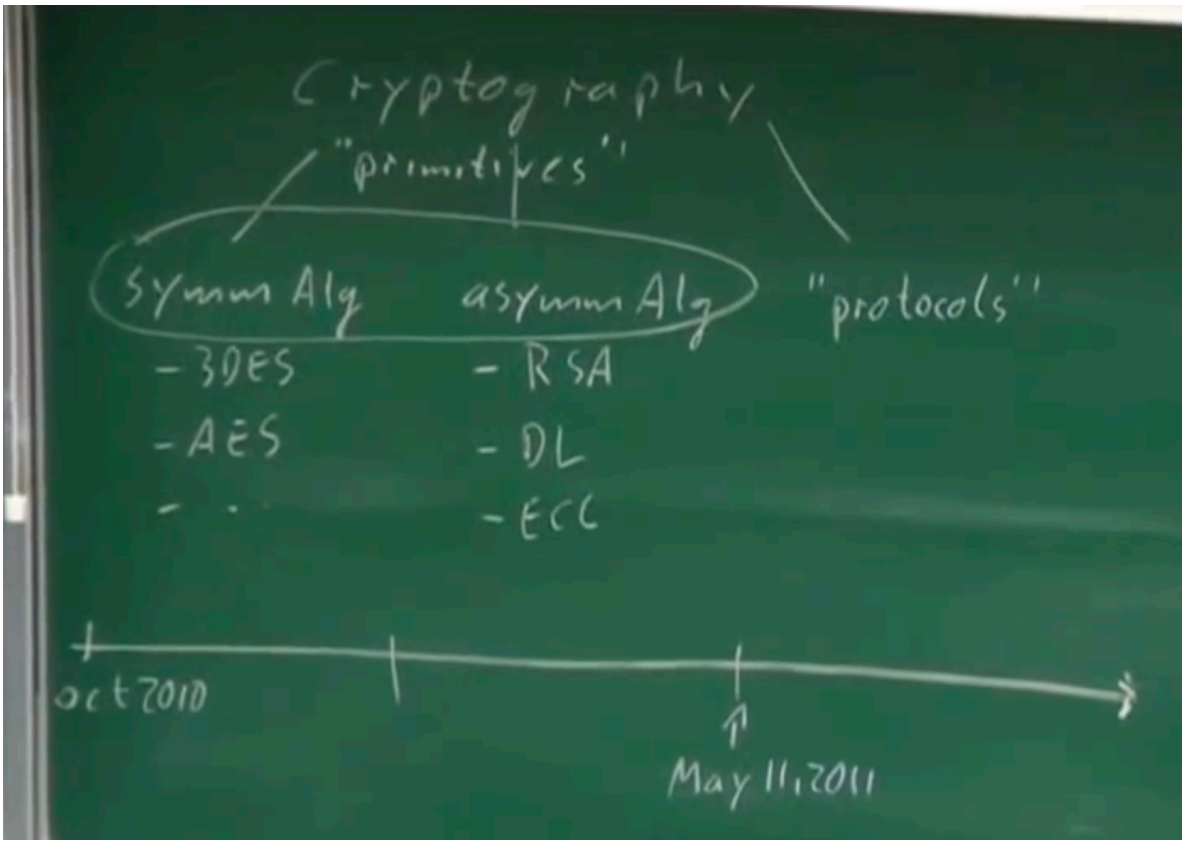
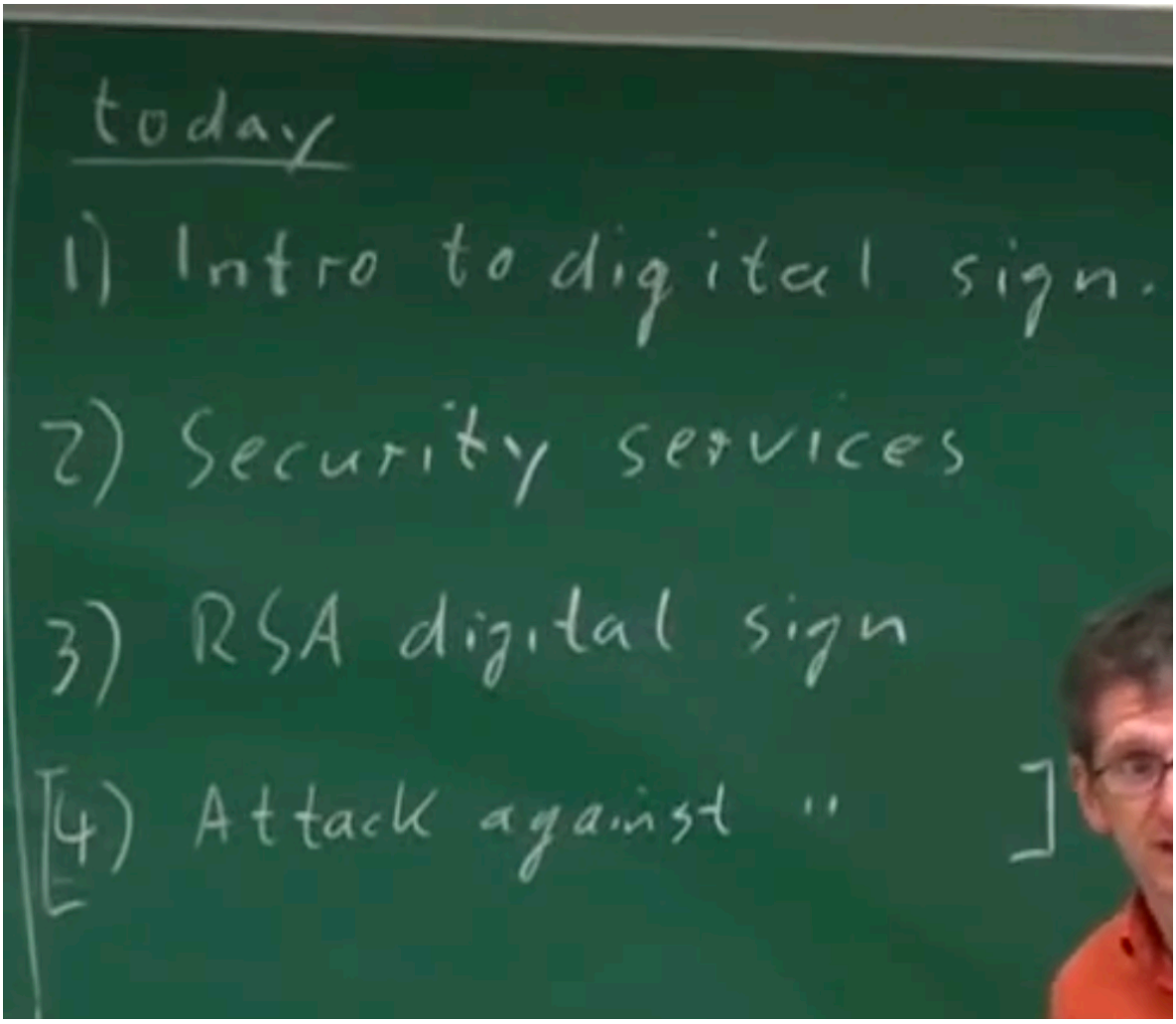
Lecture 18. Digital Signatures and Security Services
Where are we ? Till now we have done symmetric and asymmetric encryptions mainly
DES/AES ... and RSA/DH/ECC



So today we are going to talks about "**Protocols**" the main thing in protocol is that we are not going to introduce new algorithms but instead we take the stuff from last 18 lectures and build stuff with that.
More towards application of Symm and Asymmettric algorithms and what else we can do with these primitives(Symmetric and Asymmetric algorithms)

Today:
we will try to cover following topics in todays lecture:

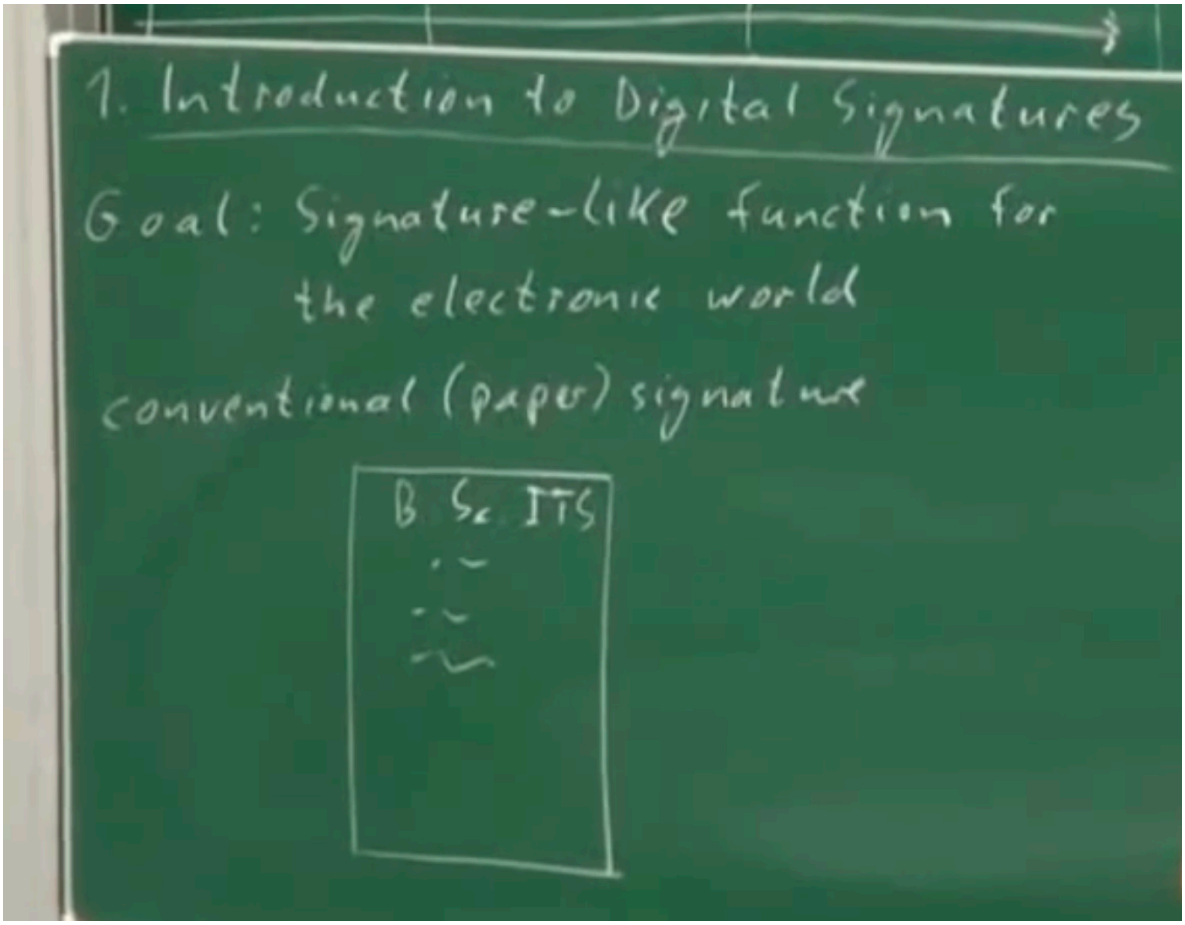
1. **Introduction to Digital Signatures.**
2. **Security services**
3. **RSA digital signatures**
4. **[Attack againts RSA DSA] = in braces because we are not sure if we have time or not.**

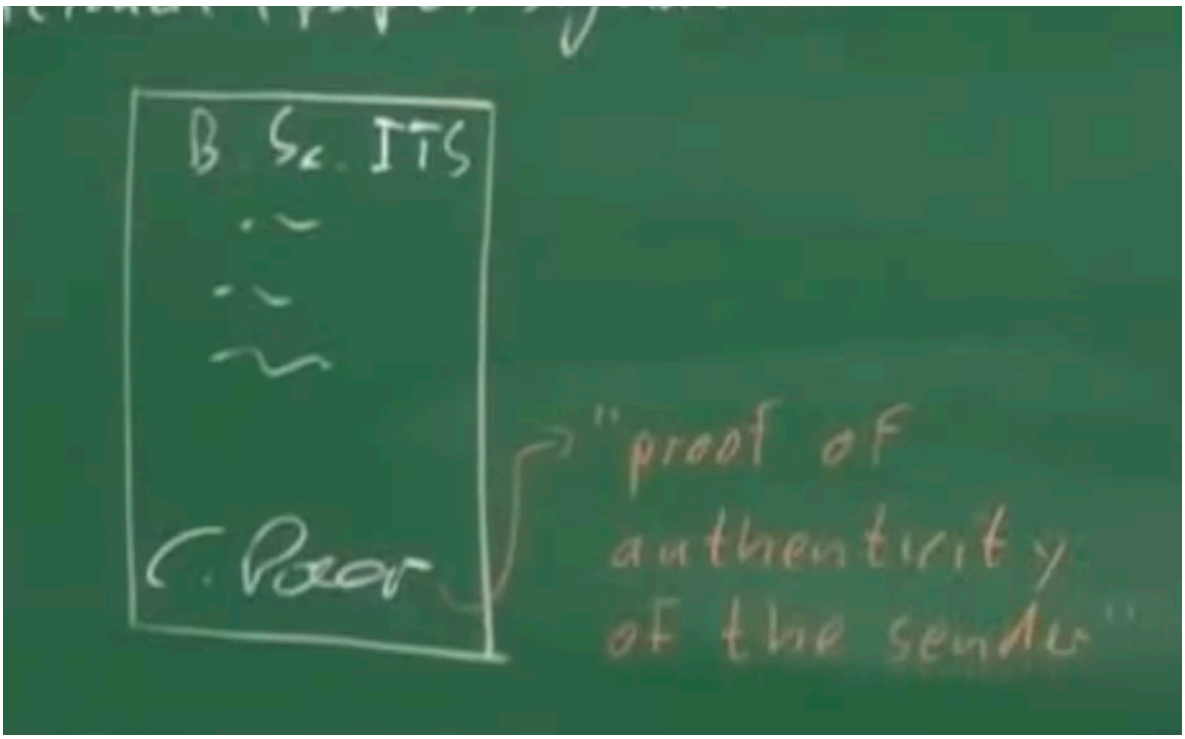


Introduction to Digital Signatures:
Objective is: we want to have signature like Function for Electronic world.
Signature like is “Signature on Piece of Paper”

We want to talk about the Signature on Physical/real world and what happens when we do the same in digital world.

Conventional (Paper) signature:
Say after completing this course or after completing Diploma, we all get the bachelor of diploma/course with Name and grade.
We can do the photoshop and take that course but for stoping that we sign the certificate ie we can get the fake diploma so society/school sign that certificate.



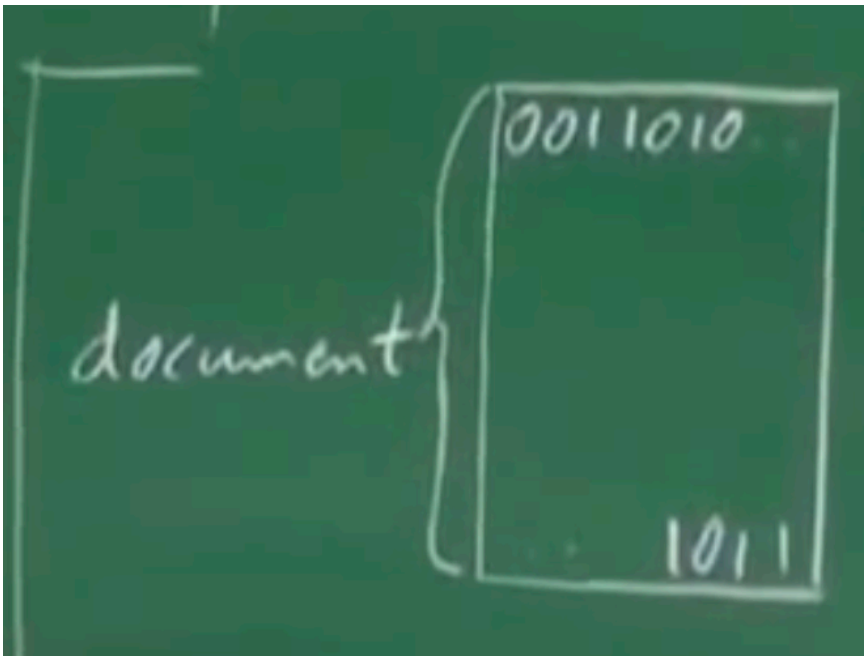


We assume that it is difficult to generate the signature. Signature is a proof of Authenticity of the document ie Who created document is found by the signature.

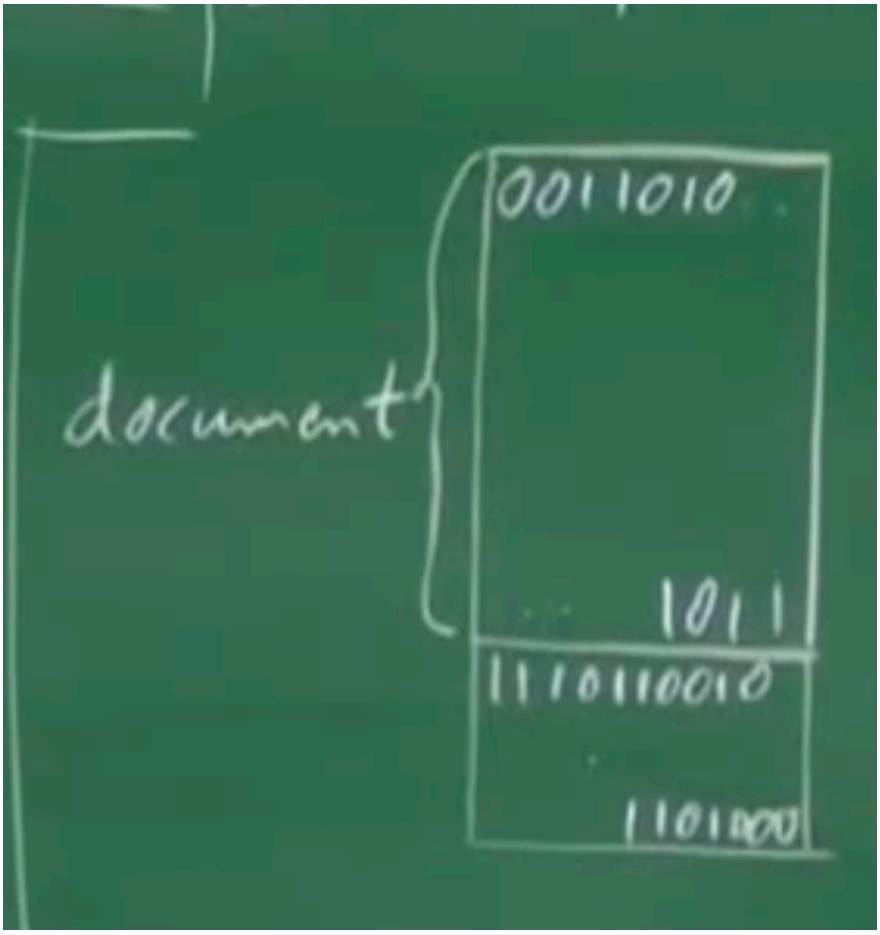
In real word, signatures works fine however they are not that really hard to break ie it is easy to break.

In Old days we are only having piece of paper but in Digital/electronics world we want a similar functionality.

Now lets try to create the same Digitally:
Now we are not having paper document but we are having electronic document like a PDF file or a word document etc. Now this document contains binary 0,1 ’s



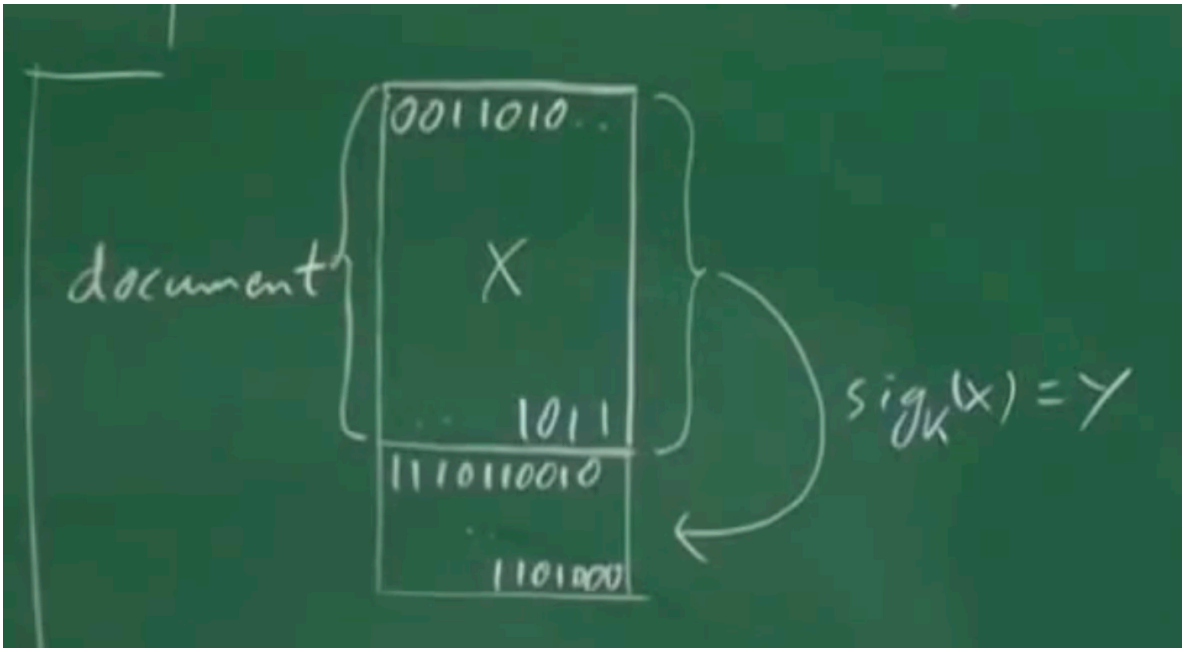
Now in the paper document we have added the signatures so if we do the same with digital document ie we pick some random digits, my own digits and put under neath document eg



Why the above way is a bad/stupid idea ?
Answer is anybody can easily copy these bits and recreate them.

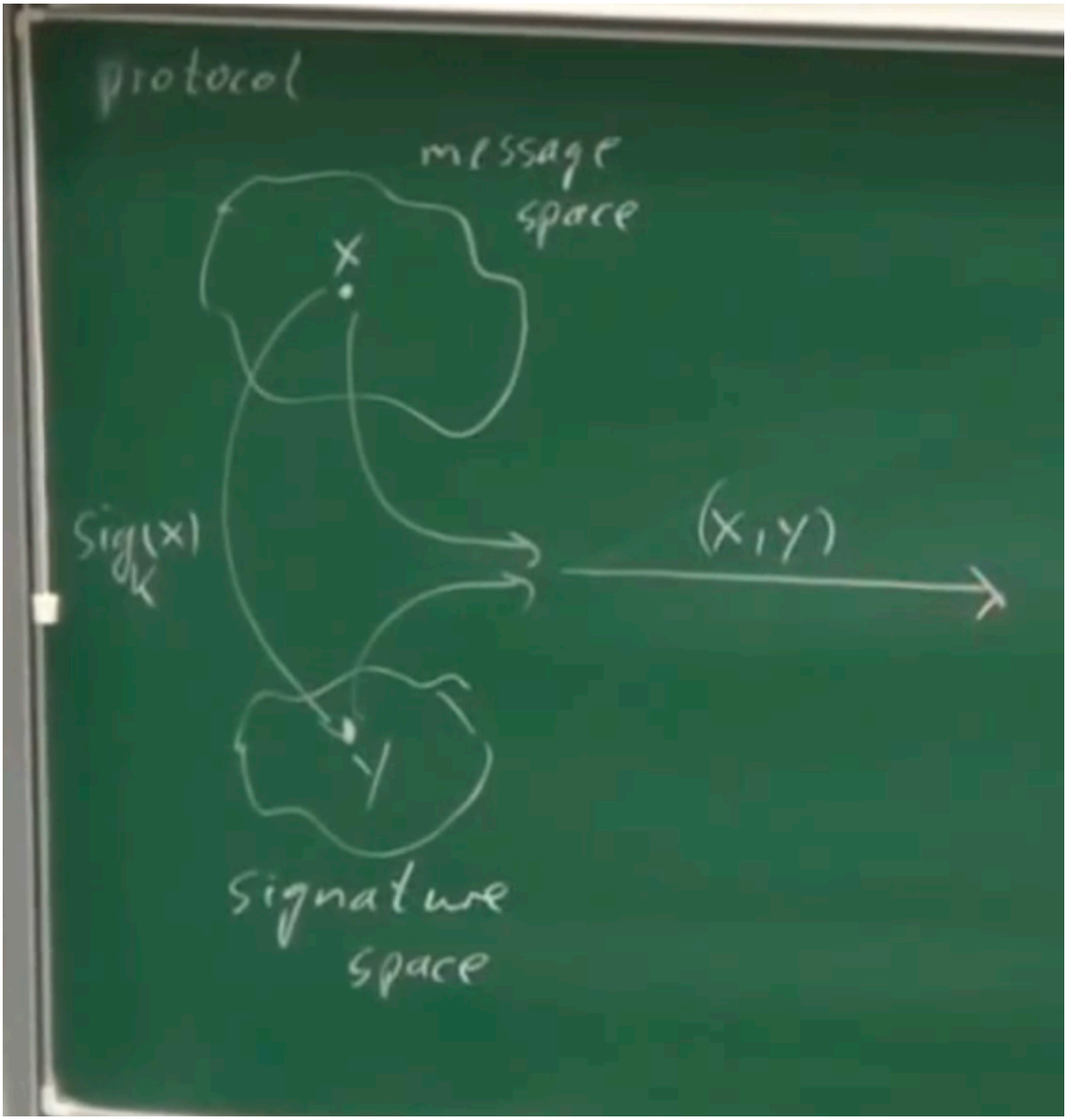
So the idea is to use the cryptography for doing the same.
We know that Alice and Bob where Alice encrypt and Bob decrypt and this works because both of them share the key.

Now if you have the cryptographic key then other people who don't have are not capable of doing certain things so now **try generating the signature using Cryptographic algorithm with the key so if you have the key you can generate the signature and if you don't have the key then you cannot generate the signature.**



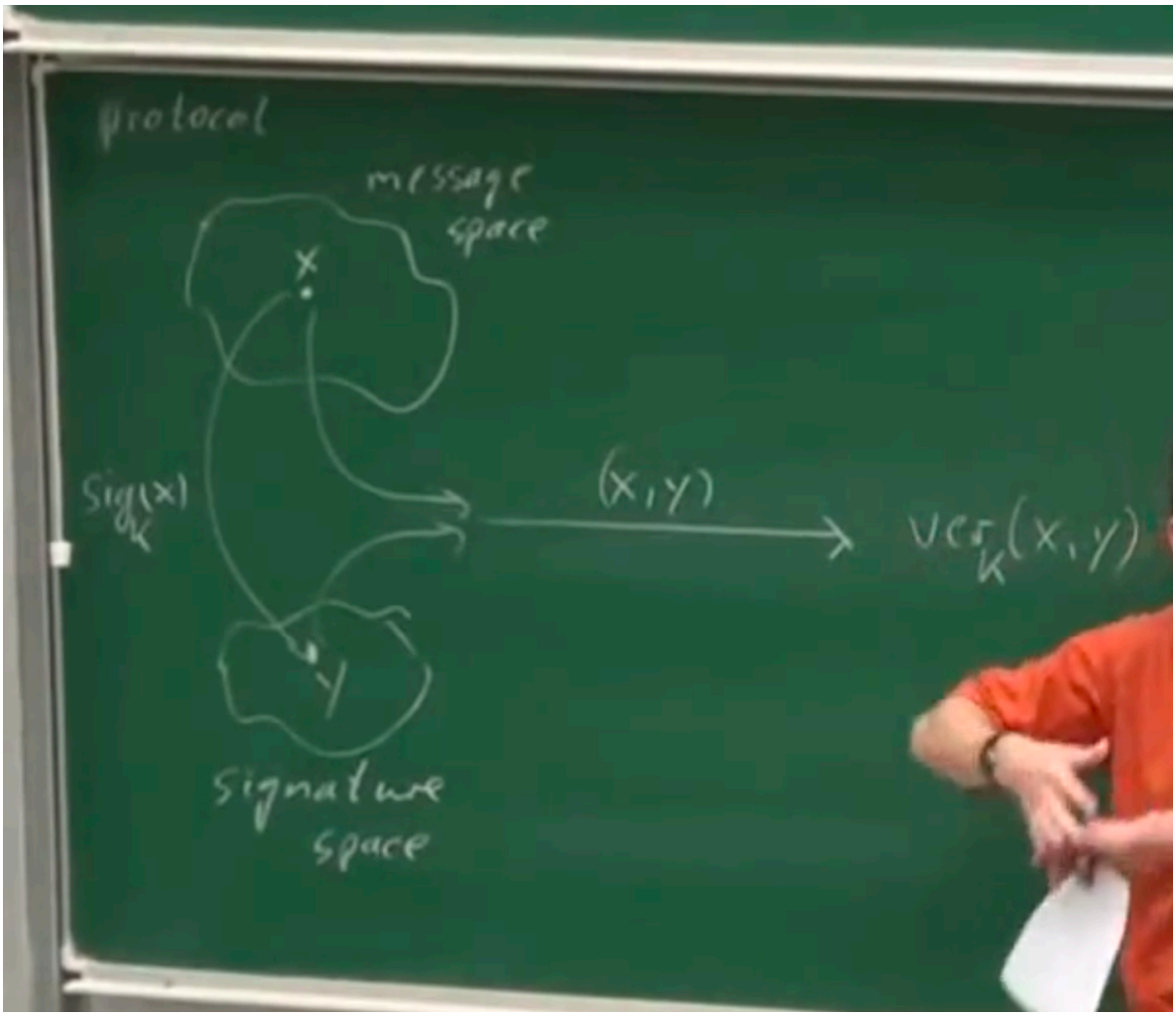
This is not the entire thing, we are half way through.

Protocol:
Say there is a message space and there is a signature space and a message from message space is mapped to one of the signature in signature space. As shown below:

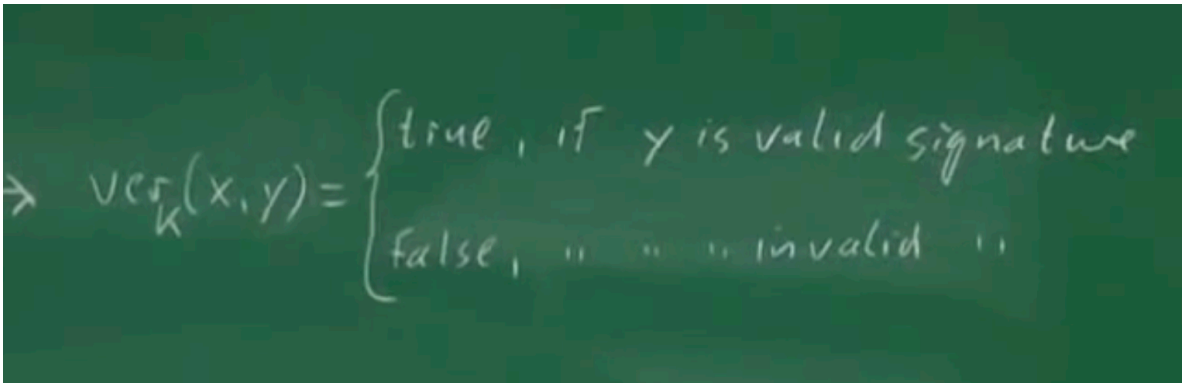


This is somewhat similar to the physical world analogy but one important question is **how in physical world we verify the signature ie what will happen at bob side ?**
We generally compare the signatures or we might have seen the signatures before etc but that is not true with Digital signatures ie we want to check/verify its authenticity.

so at bob side we will have verification function which takes input as VERkey(Message, Signature)



Now the question is what is the output of verify function ? it is only 1 bit ie go or no go/true or false.

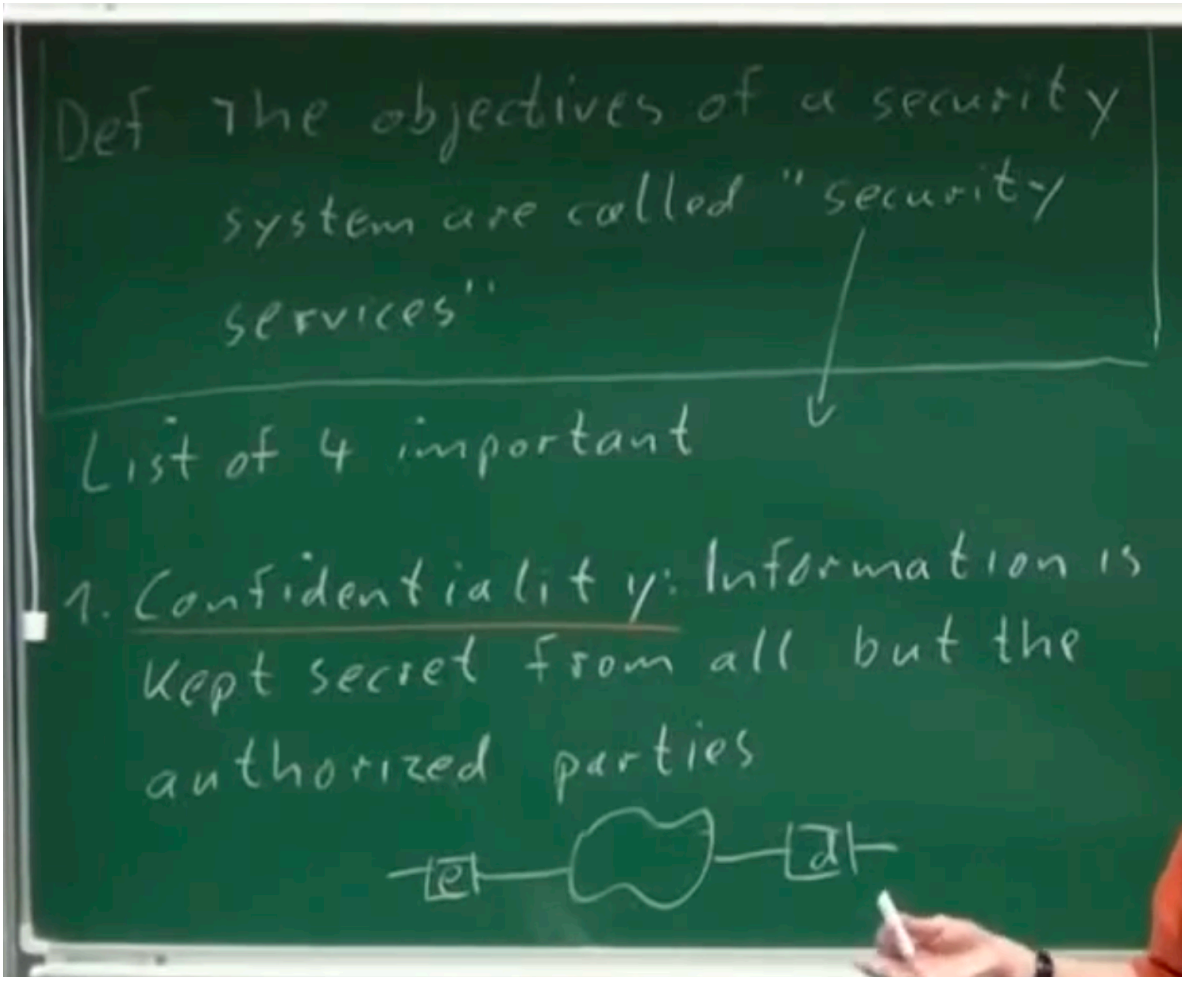


Now this is not the complete truth and there is some detail missing but what we do now is:
Say we have build this system than what do we achieve, what are the functions which we achieve?

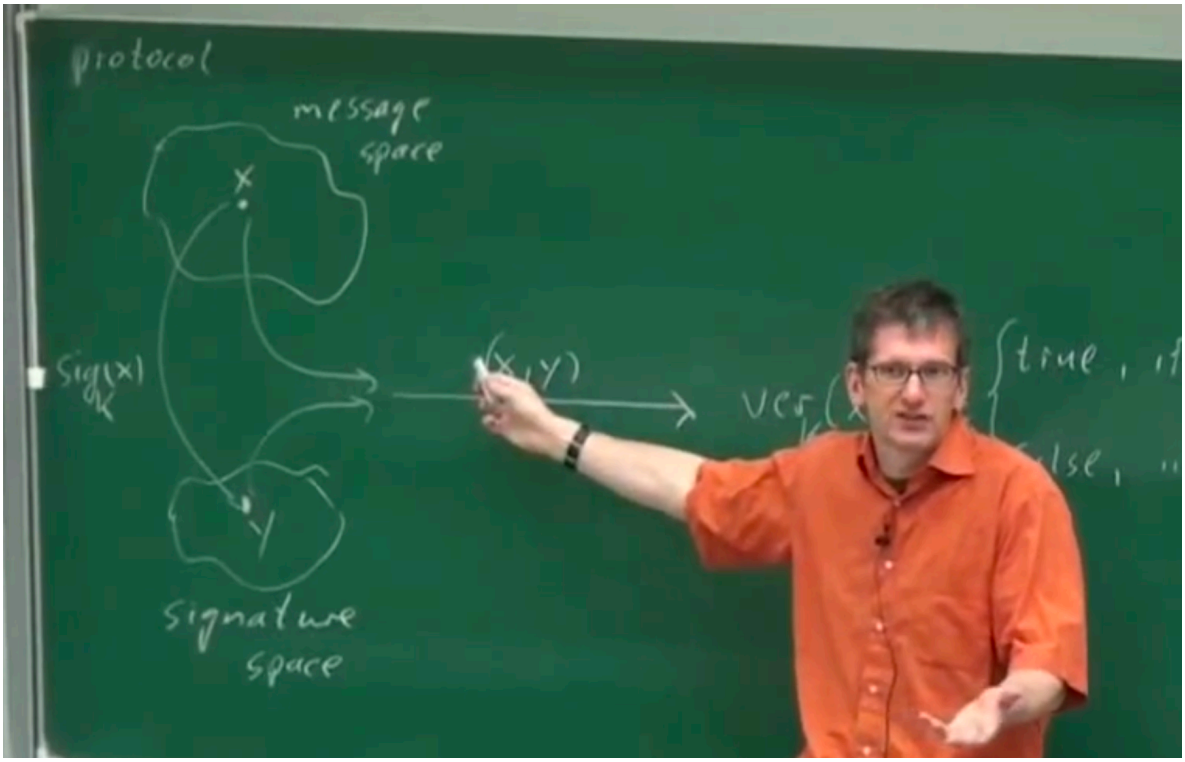
We are going to visit this chapter 1 again to complete it but before that we will

go with chapter 2.
Chapter 2: Security Services:
Definition: The Objectives of a Security System are called Security Services

There are many Security services but lets talk about the 4 important security services:
Confidentiality: - Keeping things secret. Information kept secret from all but authorised parties.



Question: we are currently reading about Digital Signatures and we have defined a protocol so question is does this provides confidentiality ?
Answer: No why ?



Because we are sending Message as well as Signature both i.e. we are sending the plain text only so it is not confidential. we get degree from college it is not confidential.

Message Authentication: The sender of a message is authentic.
Question: in the protocol does bob after verifying that it is correct, knows that this is the message really coming from Alice ?
answer: is yes because bob has the key to verify and only alice and bob have the key because may be they used the key exchange protocol and assuming no one else has the same key.

Message Integrity: Message has not been modified during transmission.

Question: Say in the above protocol some one modified X with X` do you think it works ? i.e. if bob verifies does that works ?
Answer is not as Y is a function of X and Key and modifier doesn't have the key and as Y remains same so this will not work so the Protocol provides message integrity.

Now lets compare with the Physical work signed degree/diploma:
Do we have the message integrity there ?
Answer is no, if someone modifies something in paper degree then there is not way to verify that. Like changing 68 to 88 etc or adding more zeros.

So digitally we get both Authenticity as we as Integrity but in physical paper we are getting only authenticity.

Non-Repudiation:
Little Analogy:
Say we bought something from E-Commerce, and then you find that you don't require then what we do ? E-Commerce provides very good feature of returning and you will get you money back.

Say now we have the similar scenario i.e. say you ordered a car and configure the car … engine, navigation etc and then after a week or so Volkswagen delivered the car and now say no one in family liked it so you disagreed to pay to Volkswagen so then they are going to sue you in the court.

What is there point, there point is that you have ordered the card and now you are not paying for it.
Now Important point is:
We can say that we have not ordered the car ? So Volkswagen says that they have send the order which is digitally signed by their key as only they have the key.

So what is our legal argument? Point is Volkswagen can also fake the signature as they are also having the key.
So in this case Judge cannot decide.

Reality is both the parties can generate the signature.

The Sender of the message cannot deny the creation of the message -> This is called Non Repudiation.

Actually above is Sender Non Repudiation but we can have receiver Non Repudiation also where say you are submitting the papers before deadline but you want to prove that receiver receives before the deadline.

In above protocol we don't have Non-Repudiation.
so what we can do ? Nothing as long as we stay with Symmetric Encryption as both alice and bob have the same key so they can do the same stuff.

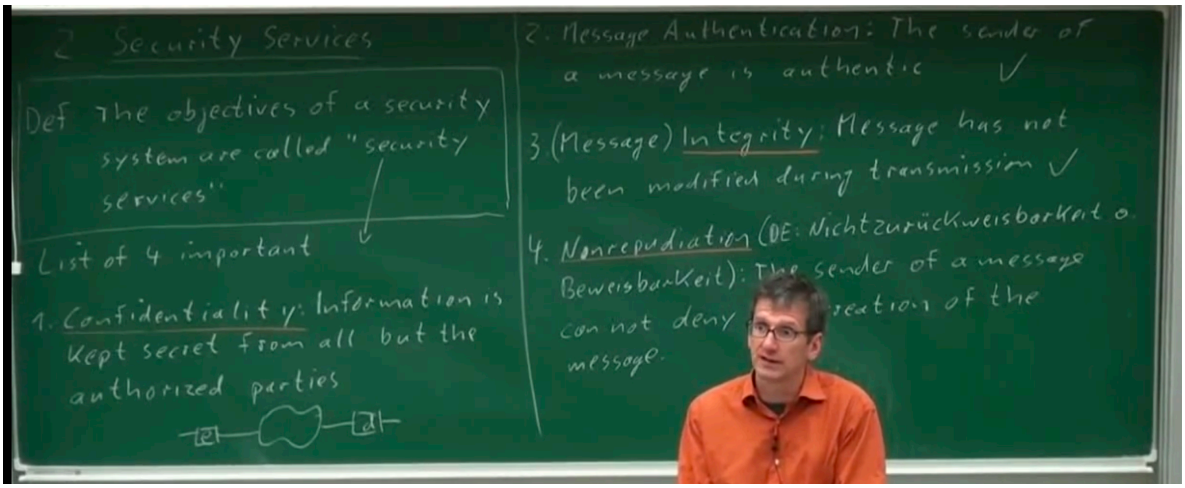
All the 3 other services like confidentiality, Integrity and authentication the attacker was oscar/third party so these services are needed if third party is a bad guy.

Question: who is the bad guy incase of Non-Repudiation?
Alice and not oscar

So in internet we want to not only communicate securely but also to protect against dishonest party.

When Alice and Bob cheat, then there is nothing we can do with Symmetric encryption as they have same keys so what Alice can do the same bob can do so

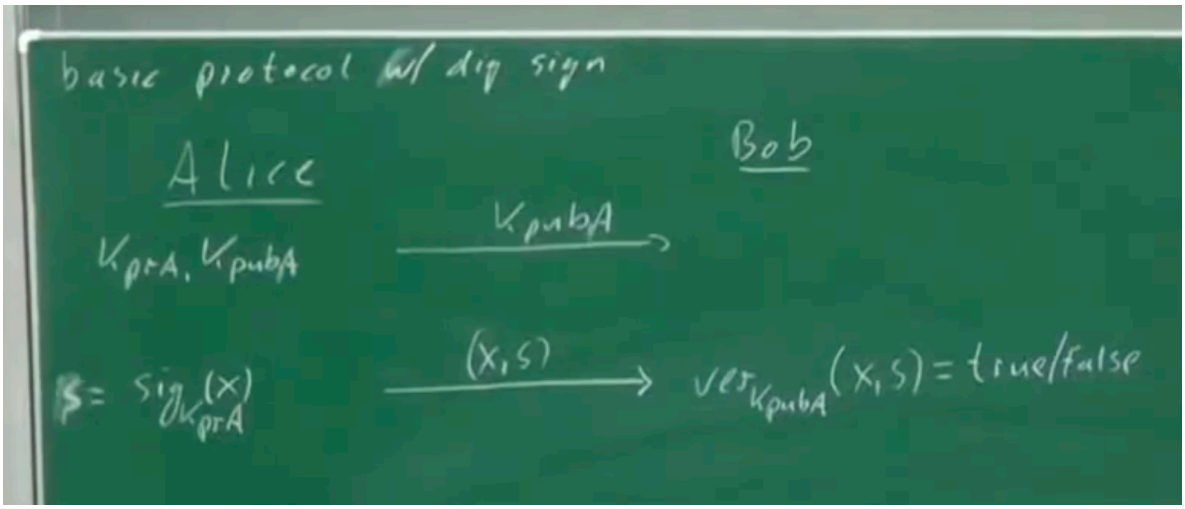
Actually in case of Symmetric algorithms both the parties have the same capability.
we need to switch algorithm families ie we need to switch to **Asymmetric Algorithms.**



So what are we going to do ?
we are going to modify the protocol. So we will use Public key cryptography.
Now question is where do we use the private key Signing or Verification ?
Answer is During signing because it is easy to publish/distribute public key and anyone can verify but private key is alice's key which is private which tells that only alice can sign the document.

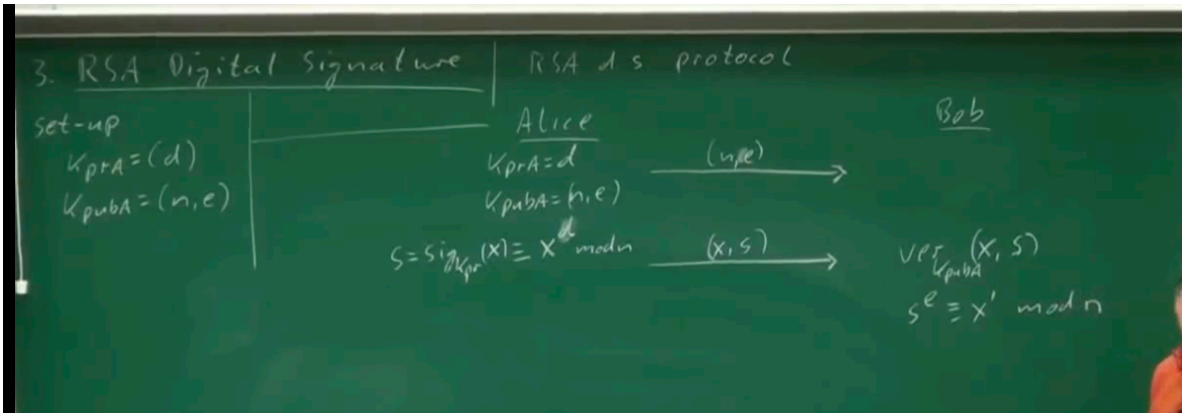
So say we go back to Volkswagen point, now if say we ordered but later we tell that we have not orders and then Volkswagen drag us in court and now that point that Volkswagen has signed it cannot be valid ie receiver can verify but cannot generate signatures by himself.

Basic Protocol with Digital Signatures:



With symmetric setup we have build a protocol earlier it also has a name called “MAC”.
MAC are very useful and web browsers use MAC’s heavily.

Chapter 3: RSA Digital Signature:
Now biggest question is what is in Signature function and what is in Verify function and it turns out that in case of RSA it is super easy.



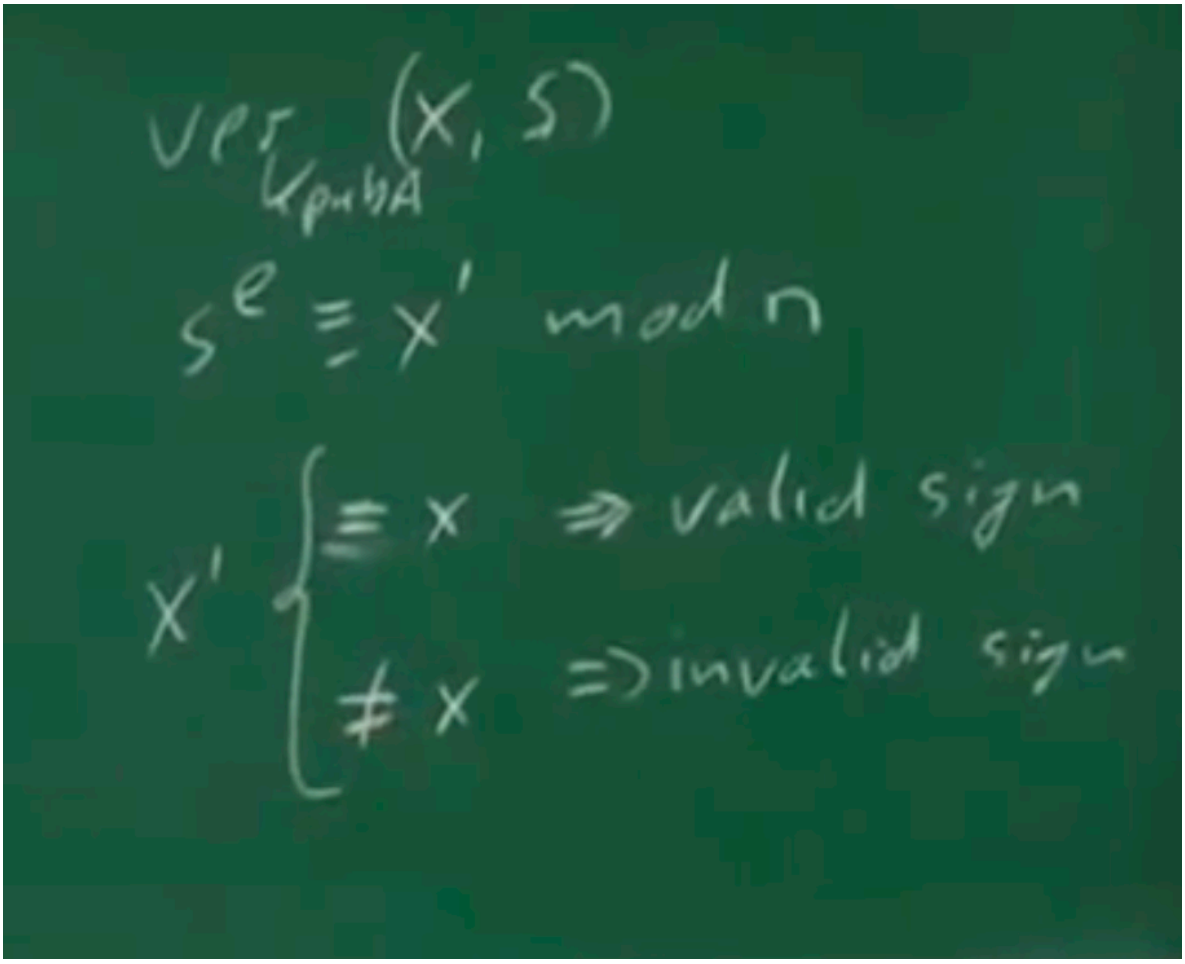
First we come to setup phase, so question is what is setup phase in public cryptography ?
it is the generation of Key pairs.

so Alice generate the Kpub and Kprivate keys and send the public key to Bob, now Alice compute the signature S = X^Kpri mod n

Actually we are doing the same thing as what we do in RSA encryption Only difference is we are encryption it with Private key however in RSA encryption we are encrypting with Public key of Bob and reason is there we want confidentiality but here we want opposite and we don’t want confidentiality but we want integrity and non repudiation and authentication.

Now Alice sends Signature and X, we need to send both as only X and Only

Signature doesn’t signifies anything.
Now Bob has to verify so now what does bob do ?
Bob decrypts and Signature 2048 bits and get say X` so now bob compares X` with X and if they are equal then it is verified else it is no a valid signature.



Proof of correctness:
Bob computes S^e => (X^d)^e mod n = X mod n.
as d*e = 1 mod phi(n) —> **Need to look more into the proof.**

Lets step back and see what happens compared to Security services:
What happens to integrity ? say oscar is changing the bits ?
Answer: Verify fails because X != X`
What happens to Authenticity of sender ?
Answer: As Private key is only with Alice so bob knows this is really coming from Alice.
What happens to Non-Repudiation ?
Answer: this we already discussed that we get from Asymmetric encryption.

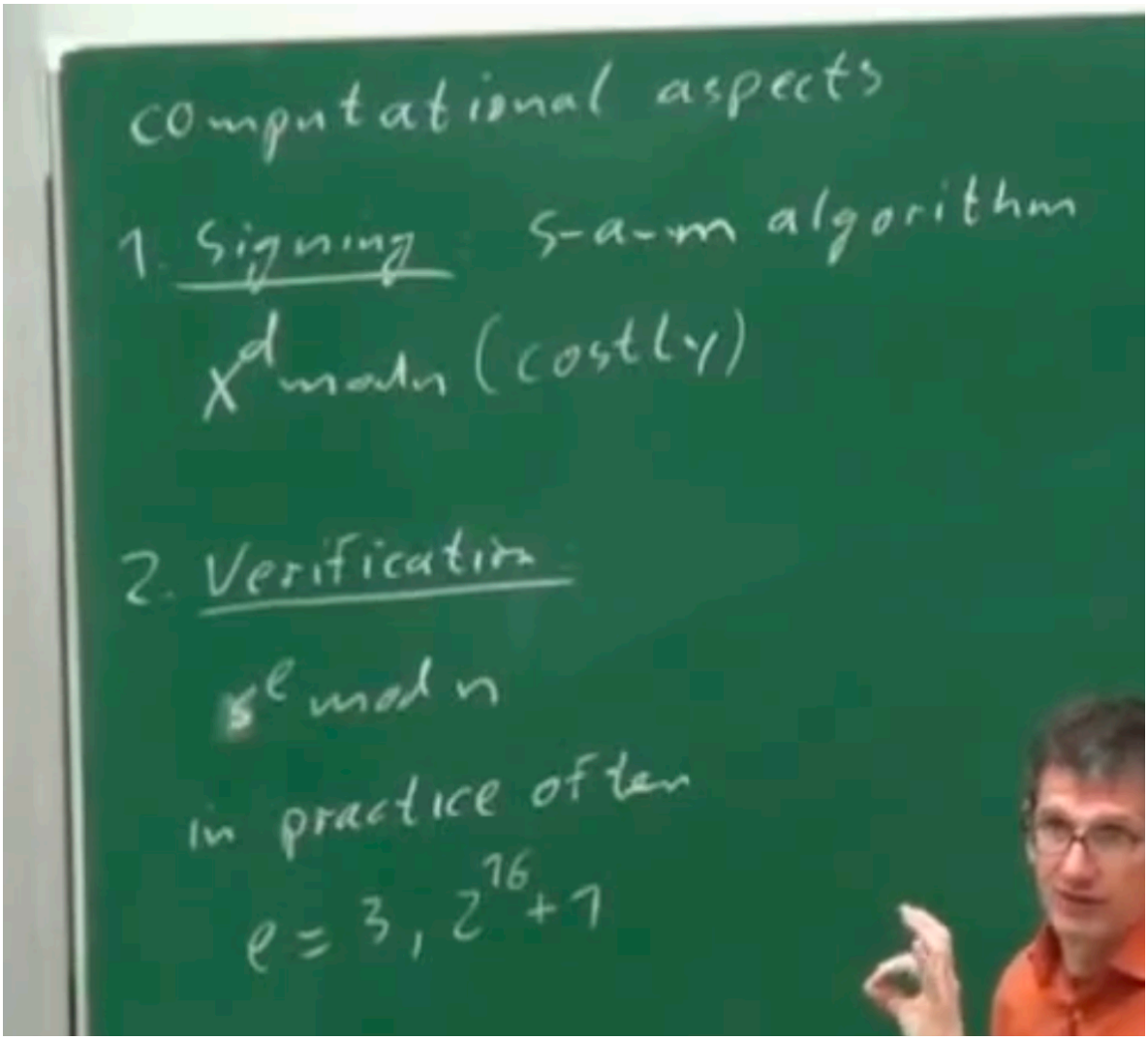
Computations Aspects:
Signing: Encryption needs exponent calculation so we use S-A-M ie Square

and Multiply algorithm which is no fun as it is quite costly and quite slow in comparison to AES.

In general Public key cryptography is 1000 times slower than Private Key(Symmetric key cryptography).

Verification:
Normally they use in Verification also Square and Multiply as there is no shortcut to that but they use one trick i.e.
in practise: often people choose “e” as small number not 2048 bit number like say 2 bit number or 2^16 + 1 which is 17 bit number so that the square multiply can become very fast. it makes verification becomes super fast.

and signing becomes slow.



In EC they both are equal speed but the above is specific to RSA.