

Information Security - II
Prof. V. Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 36
Public Key Cryptography - Week 7

So, that is basically what public key cryptography actually does, where the sender and receiver do not share the secret key, but shares the public key which is actually known to everybody and then private decryption key is only known only to the receiver.

(Refer Slide Time: 00:13)

Public Key Cryptography

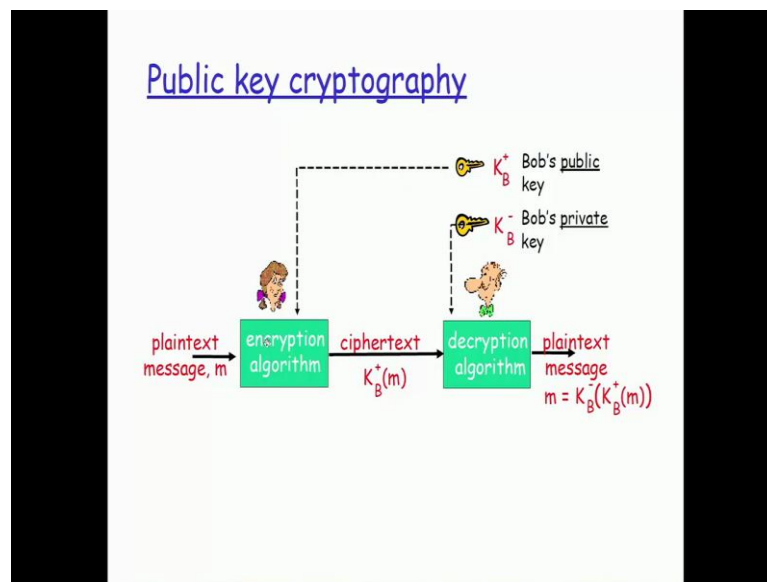
| <u>symmetric key crypto</u> | <u>public key cryptography</u> |
|---|--|
| <ul style="list-style-type: none">□ requires sender, receiver know shared secret key□ Q: how to agree on key in first place (particularly if never "met")? | <ul style="list-style-type: none">□ radically different approach [Diffie-Hellman76, RSA78]□ sender, receiver do <i>not</i> share secret key□ <i>public</i> encryption key known to <i>all</i>□ <i>private</i> decryption key known only to receiver |

The slide features a stick figure holding a glowing lightbulb on the right side of the public key cryptography column.

So, as I was telling that is something which absolutely confidential which no other person should be able to know. If by chance that is compromised, you basically go through the exercise again of completely rebuilding the whole pair of keys. Do not try to have the private key compromise. So, I will just change the private key alone, but continue to have my own the previous public key that does not work, because both of them are actually generated by certain mathematical theorems from one large random prime number. So, although I say I am generating two different keys; they are not generated in a standalone manner, but they are actually generated from the same source.

So, what is the source? That source is a very large it is a random and it is to be a prime number. So, from this large random prime number, there are certain mathematical theorems that are actually applied and then I get two different numbers: one number I hold it to my chest, another number I scream out side. Right? Whatever I am holding it to me becomes my private key, whatever I am screaming outside becomes the public key for everybody to make use of whenever they want to communicate in a very secure manner to me. So, that basically is what is the public key cryptography.

(Refer Slide Time: 01:48)



So, I have a plain text message, encryption algorithm is using Bob's public key because this is Alice. She wants to communicate with Bob, so she knows the Bob's public key. She uses that key on the encryption algorithm with the plain text I get the cipher text and then Bob, will use the private key which is known only to him on the cipher text and it will magically generate back the original plain text message. Here although both the keys are totally different it will still generate back the original plain text message as long as these two keys satisfies certain mathematical principles.

(Refer Slide Time: 02:30).

Public key encryption algorithms

Requirements:

- ① need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
$$K_B^-(K_B^+(m)) = m$$
- ② given public key K_B^+ , it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

So, that is basically how the public key cryptography basically works right. So, what should be the requirements? I should require, I need these two keys, one which is the public key in the private key such that, if I apply both of them one after the other I still get back the original message. That is the first requirement. The second requirement is given the public key, because it is something which is publicly known. Given the public key if somebody is very intuitive very, very, very curious wants to actually try to compute the private key that should be feasibly impossible. So, what is technically refer to as a NP heart. Have you heard of a NP heart problem, right. So, given a public key K_B , if somebody wants to be very smart and try to compute the corresponding private key, that should be basically be a NP heart problem which essential means that it is infeasible to calculate.

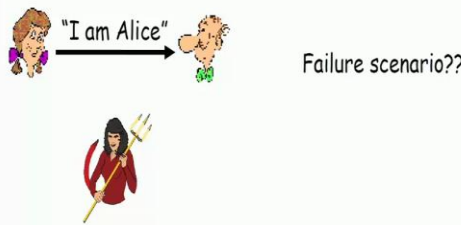
So, the very popular public key encryption algorithm is the RSA algorithm which, we would have heard of in different context ah and it different points in time by different applications. The RSA actually stands for Rivest, Shamir and Adelson because these 3 are the people who contributed towards coming out of this algorithm.

(Refer Slide Time: 03:55)

Authentication

Goal: Bob wants Alice to "prove" her identity to him

Protocol ap1.0: Alice says "I am Alice"



"I am Alice"

Failure scenario??

So, Authentication. Now the next aspect of the security that is actually needing to be done is authentication.


Now, what do you mean by authentication is, Bob want Alice to prove her identity to him. So, when somebody is trying to walk to me, I first need to convince myself that, it is that guy alone who is right down trying to talk to me and not somebody else. So, it is not like ah the social media today, where I can actually have any account right and try to communicate and the other person might feel that it is basically the same person is talking to me, but I could actually have another person's photo. Which should essential convey that boss this is the guy whom I am talking to right. So, anybody can create account with Narendra Modi's photo that does not mean that he is another Modi. So, I need to authenticate. So, how do I go about doing an authentication. So, very basic principle I just say boss I am so and so, but we already know the problem here.

(Refer Slide Time: 04:59)

Authentication

Goal: Bob wants Alice to "prove" her identity to him

Protocol ap1.0: Alice says "I am Alice"



The diagram shows three characters: a woman with brown hair (Alice), a man with a beard (Bob), and a woman in a red dress with a spear (Trudy). An arrow points from Alice to Bob with the text "I am Alice".

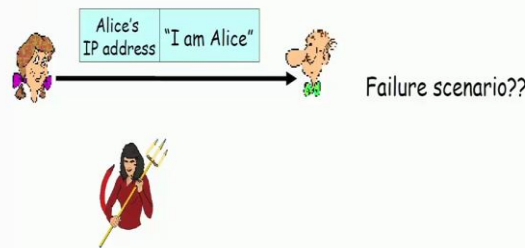
in a network, Bob can not "see" Alice, so Trudy simply declares herself to be Alice

Anybody can also announce that; I am the same person. So, that is the fundamental problem there.

(Refer Slide Time: 04:05)

Authentication: another try

Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address

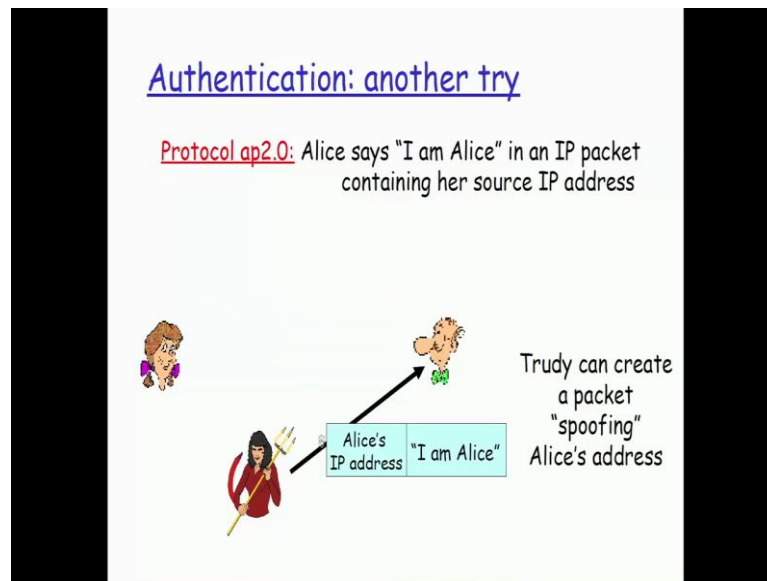


The diagram shows Alice sending a message to Bob. The message is represented as a box with two parts: "Alice's IP address" and "I am Alice". Trudy is shown below the message, indicating a potential failure scenario.

Failure scenario??

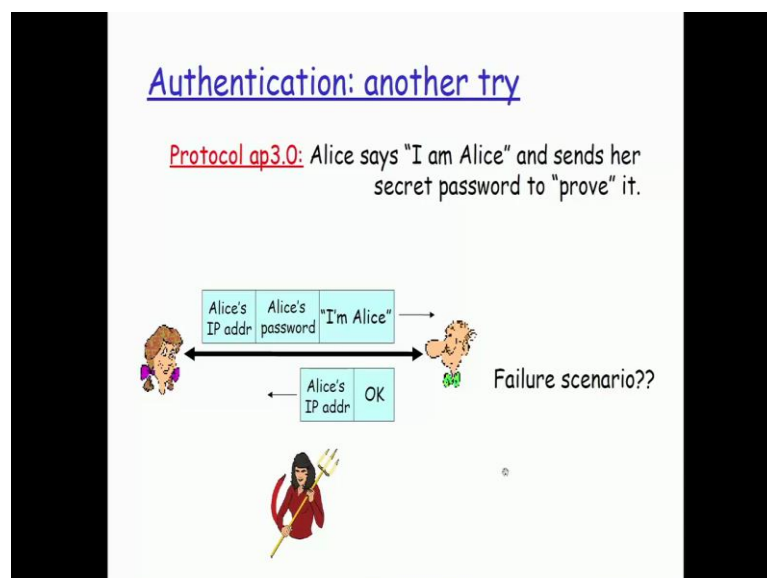
So, the next thing is I basically also have the IP address, of that particular user appended to the message, right. So, what is the failure scenario here? Yes.

(Refer Slide Time: 05:22)



So, this person can also spoof Alice IP address and then put the Alice IP address and then say I am Alice. So, this again has the problem.

(Refer Slide Time: 05:30).



The next thing is I can actually put the password along with the IP address and then send the message that I am Alice and then Bob can send back I understand that you are the person and then send it out. So, is there a problem here?

Student: (Refer Time: 05:46)

What could be a problem here?

Student: (Refer Time: 05:54)

Correct.

Student: (Refer Time: 05:56)

No, how can trudey know the password?

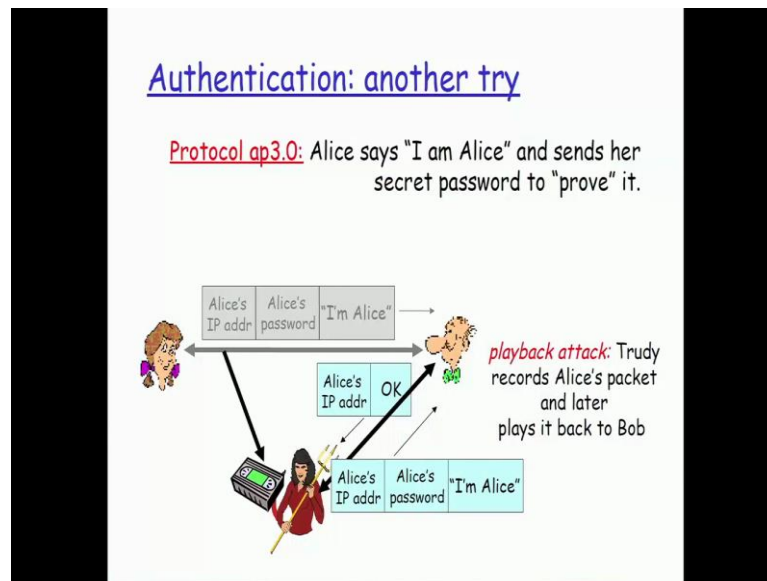
Student: (Refer Time: 06:03)

So, you assume for this purpose that Bob has access to Alice password from a central password database. Let say he can go and query from an active directory kind of database, some centralized authentication mechanism

Student: (Refer Time: 06:24)

Any other possibility? Have you heard of something called as a replay attack?

(Refer Slide Time: 06:43)



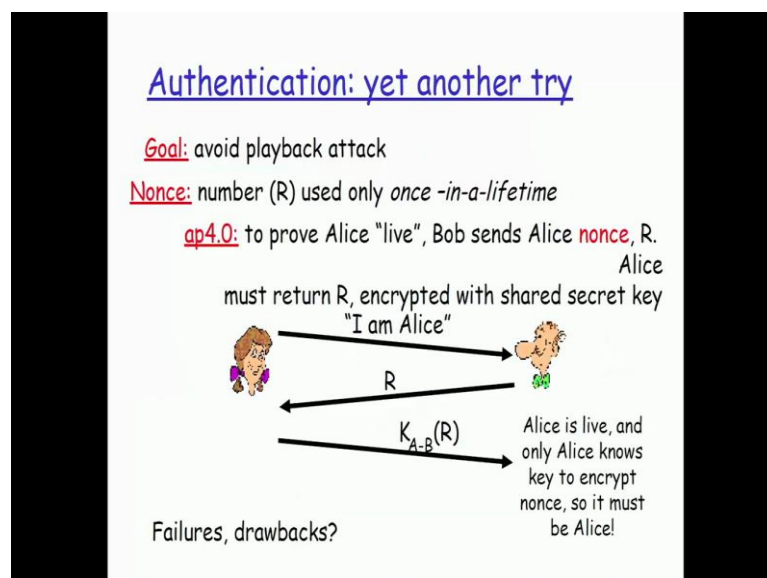
So, what exactly happens in a plain store; Store and play attack or a replay attack is I said ev's drop and I say impersonate right. So, when I ev's drop I can also parallely store the packets that is actually going on right. So, even if you look at some tool like a wire shark, you have a mechanism by which you can actually capture all the packets dumped into a file locally on your hard disk. So, when I store those packets, later on I can replace those packets. So, if I have basically store the packet of what has actually gone from Alice to Bob in this manner. After half an hour, I can basically send the same packet to Bob and Bob will think that this guy has actually again come back, Alice is again actually come back and she is trying to initiate a conversation with me right. So, that could be the problem here.

(Refer Slide Time: 07:45)



Now, if I do an encrypted password again, I have the same issue where I can store it and then, replay it at a later point in time.

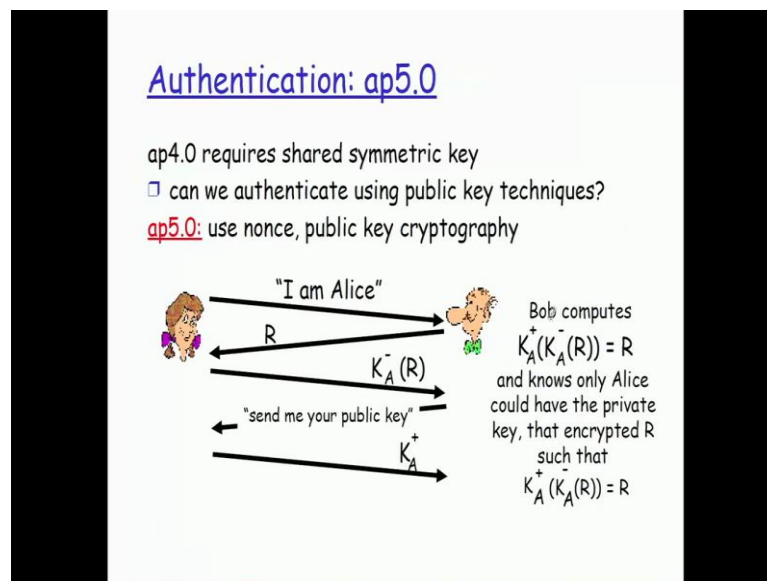
(Refer Slide Time: 07:56)



Now, for this what is generally used is there is something called as nonce. Now what is this nonce is, the nonce essentially means that a value is legal, only when it is actually

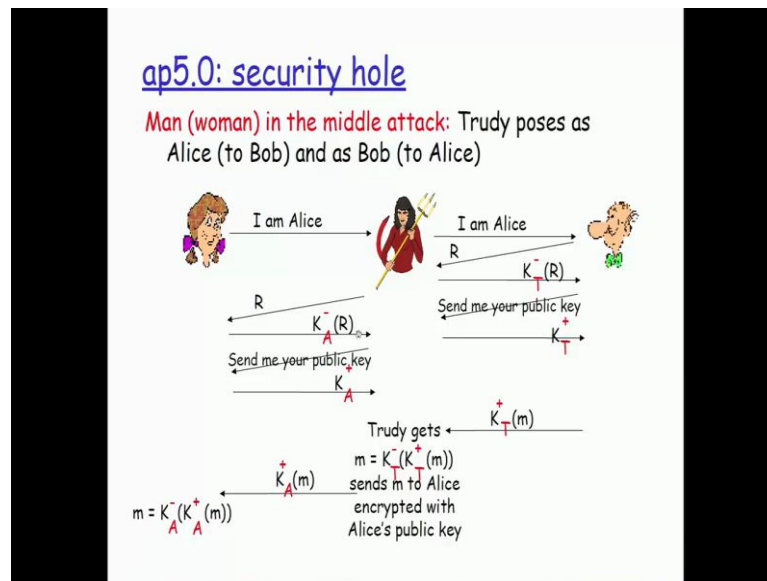
used once. So, nonce actually stands for not ones. So, the movement I tried get the same value again, Bob can immediately find out that boss somebody is playing fishy here. Right? So, I basically have a nonce value also generated, but that basically getting encrypted with the shared secret key to be additionally productive. So, that is the next mechanism that is there for an authentication.

(Refer Slide Time: 0:42).



So, again when I use a symmetric key, I have a problem. So, for the encryption of this nonce value I will go head and use the public key technique. So, that will be that much more safer as far as the authentication part is concern right.

(Refer Slide Time: 09:00)



So, here again I might be subject to a man in the middle attack. Where I would be able to have this recorded, but what is going to possibly happen is, when I when this person is going to store and replay it at later point in time, Bob is going to have that value already received. Right? And when he finds that the same value is actually coming in right now, he knows that it is something which is getting replayed, after storing and through that mechanism he will be able to prevent this kind of an attack right.

(Refer Slide Time: 09:42)

Hash functions

- A *hash function* is defined as "a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length"
- Hash functions are implementations of *one-way* functions
- The probability of two messages with the same hash is minimized

```
graph LR; MD1[Message Data] --> HA1[Hash Algorithm]; HA1 --> H1((Hash)); H2((Hash)) --> HA2[Hash Algorithm]; HA2 --> MD2[Message Data]; style MD2 stroke:#f00,stroke-width:2px
```

So, that is basically from the authentication point of view, what kind of ah functionality is there. Then the next thing that is actually required is what is called as a Hash function. So, a Hash function have you heard of any kind of Hash algorithm still now?

Student: (Refer Time: 10:01)

Sha anything else? So, with in SHA itself you have different versions. So, you have SHA 1, SHA 256, SHA 512. So, what are the first algorithm? MD 5, right? What is basically Hash function actually do is? So, Hash function is basically a computationally efficient function, mapping binary strings of arbitrary length to binary strings of some fixed length. Now, what I basically tell is if I have a text, let say a string that I want to send it to another party. We said that somebody one of the possible types of attackers, somebody trying to do a modification of the content, right? So, I want to have a mechanism by which the receiver will come to know, that boss somebody has modified the data here. And that is what I actually use a Hash function for.

So, when I have a Hash function right. So, is going to be a algorithm that is there. When I send my entire data that I want to send, through that algorithm, he is going to generate me Hash value. So, when I am sending the message across I will also send the Hash

value along with the message. Now when I send the Hash value along with the message what will happen? The other side will receive the entire thing, right? What is it the other side is expected to do? The other side is actually expected to run the same algorithm, right? On the data that they have received, and if they find out the value that has actually come there, and the value that has been received as part of my message to be same. What does that mean?

Student: (Refer Time: 12:10)

Message has not been tampered with. Because the beauty of this Hash function is that it is a computationally efficient function mapping, binary strings of arbitrary length to binary strings of some fixed length right. So, when you have a Hash function one of the requirements of the Hash function is going to be that, if I basically change, one bit of my input text and then give the modified text to my Hash function, the Hash value that this guy is going to generate to me should be as much as possible completely different as compared to my previous Hash value, right?

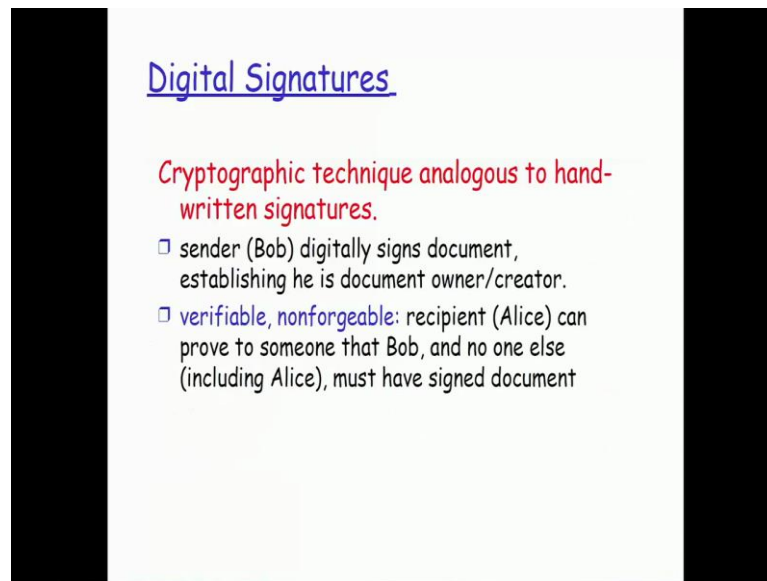
So, let us say that I have actually sending a text of 1, 2, 3, 4, 5 to the remote side, right? This number 4 has basically become 5. So, somewhere either because of corruption of some memory access or somebody was actually trying to really play it very smart, he was actually trying to modify that value, whatever it is, out of this 5 numbers only one number has got changed and there is actually go on and it is a modified string to the other side. What I would essentially want is when the same Hash algorithm is being subjected to with the received text, the Hash value that it is generating should not be anywhere closely resembling to whatever I have generated and then sent along with my message, right So, that is one very important requirement of my Hash algorithm.

So, what we typically call as a Avalanche effect. So, we all know Avalanches right. So, the weather disturbance, when it comes, it comes really very bad people will not be able to sustain it like what very recently we had in Chennai also. So, what is that Avalanche effect is that, if I basically change even one bit of my input text, the corresponding number of bits that has to be changed in my Hash value should be significantly higher.

So, if I change one bit, one bit alone should not be getting changed in my Hash value. In which case I will have a problem in finding out whether there has been actually any modifications on n root or not. Right? So, the requirement of an Hash algorithm this actually, has this requirement of fit having and good Avalanche effect and also that with the Hash value being given into the Hash algorithm I should not be able to get back the original message data. So, that is the basically reason why the Hash functions are typically referred to as a one-way function.

So, I cannot get the reverse part of it. With the Hash value input into the Hash algorithm I cannot get the message data. Only when the message data that I want to actually be checked for my integrity, is given as an input into my algorithm. I will be able to get a Hash value. So, that is basically reason why we typically refer to as a one-way function right. So, the probability of 2 messages with same Hash is minimized. So, please note that it is not eliminated is not used here, it is only minimized right. So, what we technically call as a collision. So, the idea here is that the number of collisions that could typically happen, has to be extremely less. So, that is going to be one very important criteria based on which somebody is going to measure the efficiency of the Hash algorithm. So, we are talking of this has an efficient function here, one of the parameters of efficiency is going to be measured by is there going to be a collision or is there going to be a good Avalanche effect, by using this algorithm to generate the Hash value.

(Refer Slide Time: 16:26)



Digital Signatures

Cryptographic technique analogous to handwritten signatures.

- sender (Bob) digitally signs document, establishing he is document owner/creator.
- verifiable, nonforgeable: recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

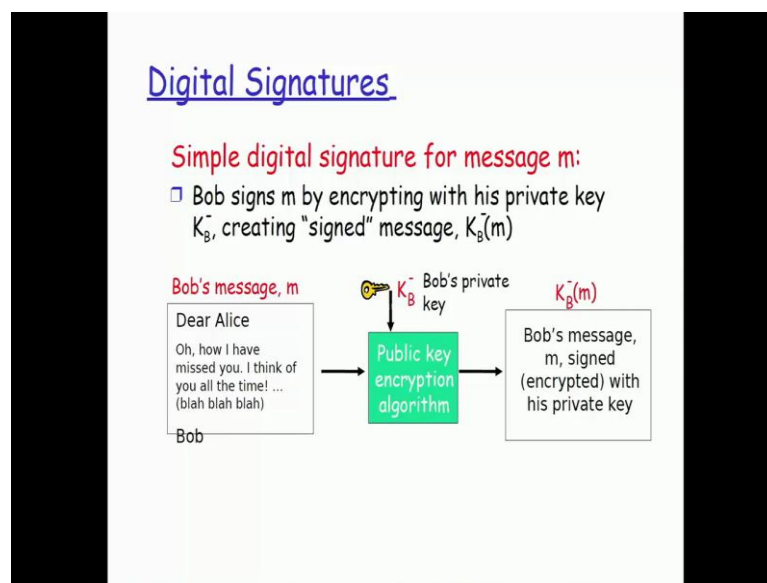
So, then the next aspect that you are going to be very commonly be getting refer to is what is called as a Digital Signature. So, we all know a normal signature that we do on piece of paper. Now what is the digital signature? A person is actually sent you some data. So, today he has sent you some data he has communicated something to you. Tomorrow you do not want him to basically say that boss, I was never the one who actually send you the data itself I do not know who sent it.

So, most of the politicians today do [FL]. So, advertently, inadvertently they come out with some statement out of their mouth. Unfortunately, in this world of social media every single letter every single word is all recorded. Then after sometime they have to do a back track saying that I never said it, media misquoted me, I meant that only, media use wrong words, all those kind of things.

Similarly, when you accepted to a particular message right when you have said that boss I confirm, and I accept and I take this responsibility whatever it is. Tomorrow you should not basically be in a position to say that, boss somebody has spoofed as me and then send it. Somebody has impersonated as me and then send it to you right that is simply not acceptable right. So, the digital signature basically helps you to do that. So, sender digitally signs the document establishing he is a document owner or the creator. And then

it is verifiable and non-forgable. So, recipient can prove to someone that Bob was the one who actually has sent this particular document or this particular message. And tomorrow bob can also not say suddenly that I am not the one who have actually had sent it. So, in order to prevent this, you have the digital signature. So, there is a message in the way that I will use the Bob's private key. This is the really the [FL], what I do for an encryption.

(Refer Slide Time: 18:43)



So, when I actually had to send a for my encryption and decryption purpose what I use to do was, I use to use the public key of the recipient, right. Here what I do here is that, I use the my private key. The moment I use my private key, what will the other guy do? The other guy will use

Student: (Refer time: 19:03)

Other guy wants to get back the original message, what will he do? But he does not have my private key no.

Student: (Refer time: 19:20)

Have to use my public key, which any way he knows correct. Now, what has this proved indirectly.

Student: (Refer time: 19:18)

How is it proved?

Student: (Refer time: 19:32)

Because, remember I told you the private key and public key is a pair. So, if it has been encrypted with your private key and somebody uses my public key to decrypt it. They will get only junk, right? But here what is actually happening is, Bob is going to be using his public key, yes. So, the signed message is going to be actually encrypted with the private key. The other person will basically be using the public key. The moment he actually gets back. Alice gets back the original message by using Bobs public key indirectly it is communicated it is accepted that Bob only has sent it. Why only bob has sent it and not Vasan has sent it? Because Bob's private key is known only to Bob. So, tomorrow with that cipher text with that encrypted text Bob cannot claim, that boss I was not the one who sent it, right.

So, in the encryption and the decryption for confidentiality we actually encrypt to the public key and decrypt with my private key, but when I have to use it as a digital signature, what I do is, I do the reverse. So, I encrypt it with my private key and then when somebody decrypts it, they are going to use my public key and thereby it basically proves that I have sort of attested whatever, the contents of the documents or message or whatever it is. So, this is basically concept of a digital signature.

(Refer Slide Time: 21:33)

Digital Signatures (more)

- Suppose Alice receives msg m , digital signature $K_B^-(m)$
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$.
- If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that:

- ➔ Bob signed m .
- ➔ No one else signed m .
- ➔ Bob signed m and not m' .

Non-repudiation:

- ✓ Alice can take m , and signature $K_B^-(m)$ to court and prove that Bob signed m .

So, there are details on how... as I was telling that non repudiation is essentially is the one there the Bob cannot later on claim that he had not actually send the message and not standing by that right.

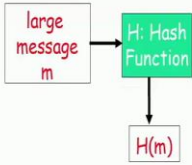
(Refer Slide Time: 21:46)

Message Digests

Computationally expensive to public-key-encrypt long messages

Goal: fixed-length, easy-to-compute digital "fingerprint"

- apply hash function H to m , get fixed size message digest, $H(m)$.



```
graph TD; A[large message m] --> B[H: Hash Function]; B --> C[H(m)];
```

Hash function properties:

- many-to-1
- produces fixed-size msg digest (fingerprint)
- given message digest x , computationally infeasible to find m such that $x = H(m)$

So, with this he will be able to prove that. Now message digest is basically a Hash function, but computationally expensive basically because when I have a public key encryption that I have to do for very long messages. See the whole process of encryption and decryption is not really very cheap. So, when we say it is not very cheap what do we essentially mean here? Time, time for doing what? Because whether, you are running at des algorithm. Whether you are running a RSA algorithm. So, des algorithm for example, goes through 16 rounds

Student: (Refer time: 22:29)

Yes, K R Anurag, Neelet any clarifications?

Student: (Refer time: 22:54)

Any other groups, any clarifications?

Student: (Refer time: 23:12) No sir.

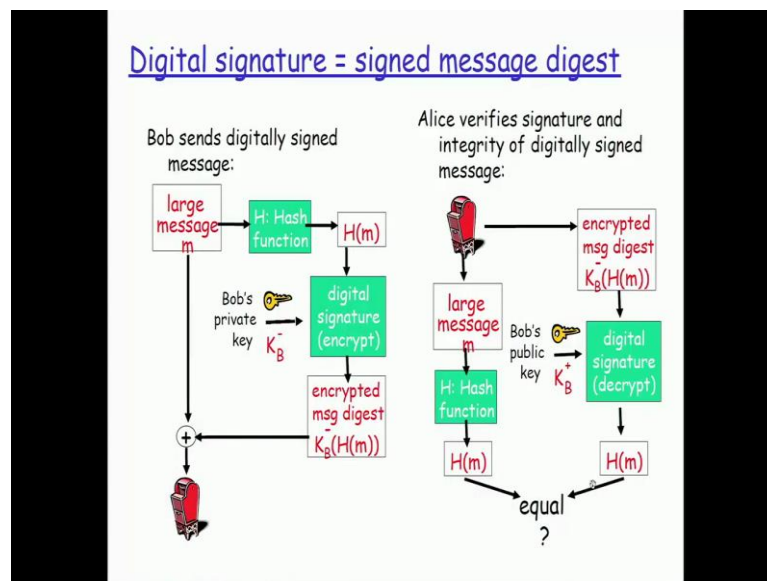
Essentially, when I have to do an encryption or a decryption of a very long message it is going to take a lot of CPU cycles on that particular encryption and decryption. Which I would possibly want to avoid at least in situations where it is avoidable. So, for that what is actually done is, a fixed length is easy to compute digital finger print, which is essentially as I told you it is a Hash function. So, I get a fixed size message digest. And I add that message digest to the large message and then send it out, right.

So, in addition if I want to really make sure that this Hash value is also not modified on the way I just encrypt the Hash message alone. Hash of the that message alone, before sending it out to the other party. So, I am not really required to do an encryption of my complete long message because of the message size is long and I am going to end up spending a lot of time on that. But if I basically do an encryption of a smaller Hash value and just append that encrypted Hash value with my message and then send it out, I will possibly be able to satisfy quieter a majority of my requirements. Then I would prefer to

do it for saving up some of the cpu cycles which I would otherwise be requiring to be spending on that.

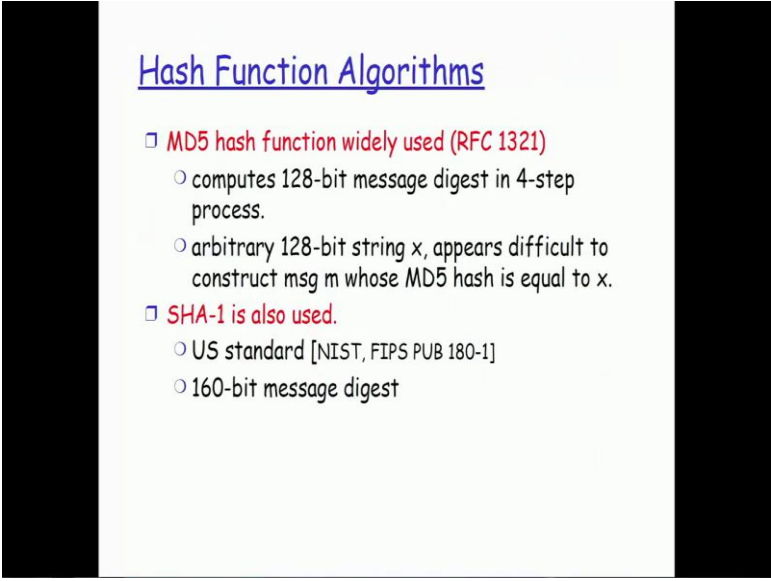
So, similarly to just how the Hash function is the primitive Hash value is used I will basically calculate the Hash function on both sides and find out whether the Hash message that I am getting on the recipient side and the value that I am getting out of the ah calculation, right? So, this is basically the calculation that I am myself made on the text, on the message and what I have actually got after the decryption part of it if they are equal. Then I will be confirmed that my message had not got actually got modified enroot.

(Refer Slide Time: 25:15).



So, instead of actually trying to spend so much of time in trying to calculate my digest value for the entire message, I only calculate the value for a small sized Hash value alone and then encrypt it and then, send it out. And at the receiving end I do the decryption and then do the comparison. The same comparison that I will do otherwise if I find it both matching then, I am still convinced that yes that the message that I have actually, right now got is something which is unadulterated so to say.

(Refer Slide Time: 25:58)



Hash Function Algorithms

- MD5 hash function widely used (RFC 1321)
 - computes 128-bit message digest in 4-step process.
 - arbitrary 128-bit string x, appears difficult to construct msg m whose MD5 hash is equal to x.
- SHA-1 is also used.
 - US standard [NIST, FIPS PUB 180-1]
 - 160-bit message digest

So, MD5, SHA-1 SHA 256, 512 or the other common Hash function algorithms that are typically used. So, you have typically the different RFC documents that are there for each of these algorithms also. So, we understand what is RFC right? What is RFC stand for?

Student: (Refer Time: 26:19)

Request for comments.

So, these are all actually available on iatf. So, you go to iatf dot org that is basically the standardization body for all the internet protocols. Each of the protocols has a corresponding standard document that is basically referred to as the RFC document. So, one protocol will be back to multiple RFC documents typically; one or more RFC documents which will give you the complete details about that particular protocol on what is that. Where is it being used? How is the protocol to be implemented? Which all the vendors are supposed to follow for their respective implementations? So, all these algorithms are actually available as a RFC document that we could actually make use of right.

(Refer Slide Time: 27:05)

Public Key Certificates

- Binds a public key with an identity using a digital signature
- Signature generated by a trusted source
 - Certificate Authority (CA)
- The Freescale Reference CST tools use OpenSSL as the CA
 - Full control over code signing private keys and certificates

| |
|-----------------|
| Version |
| Issuer |
| Subject |
| Validity Period |
| ⋮ |
| Public Key Data |
| Signature Data |

So, now there is something called as the public key certificate. Now, we talked the primary reason for us to come out to the public key cryptography algorithm was with the symmetric key I need to actually going through the trouble of sharing. But in public key cryptography also I am doing the sharing. I expected at least one of you would ask me the questions sir in public key cryptography I have to share the public key? So, the same problem is existing here also. This is the same problem or is it the different problem in symmetric key cryptography we said it is a same key because, it is the same key used by both sides we said that I have to share the key and I said that is the problem. So, because of that problem we reasoned out saying public key cryptography is a way to go. Right? In public key cryptography, we said that there are two different keys. One key is not shared only known to me, the other key is shared across to everybody else. Now you should have immediately asked the question

Student: (Refer Time: 28:18)

why.

Student: (Refer Time: 28:23)

So, is that the only reason why it is not a problem? Why is sharing this symmetric key a problem in symmetric key cryptography, but why I sharing a public key is not a problem in asymmetric key cryptography both are sharing know?

Student: (Refer Time: 28:51)

The very nature of it being public I really do not care, whether everybody comes to know of it. But what is happening in symmetric key? That is the most important part if somebody comes to know of that then whatever, messages or data that is going for that particular user or node is gone. So, public key I have got to share. So, how do I share a public key. So, that is where by public key certificates come into play.

So, I basically have the public key data along with certain information like what is the version of the certificate who is the issuer. So, there are people called as Certificate Authority, shortly called as CA. So, these are all licensed people who can give a public key certificate right. So, that is the reason why, if you go to some websites. Let say it basically a site that is actually promoting e commerce. So, the moment it is a e commerce site it will always be only https correct? Will not be a http, have you observed the difference? So, what is that s in the https? Secure. So, when you say https what is actually happening?

So, if you are very curious, what you should you do is you should try to capture packets in wire shark. When you are trying to access your banking site for example or amazon site or whatever, right? And you have to capture packets in wire shark when you try to access sites with http protocol, then look at the difference between the two of them, right. When you are captured the packages the https protocol, the content will completely encrypted; you cannot make head or tail of the content. When you have captured the package with the http protocol, you can get to know the complete details because it will be in a plain text form right. So, the encryption is going to happen first from your browser to the server.

So, let us say that you are going to icici bank dot com, because you want to access your online account, your browser is IE. So, IE because it is an https connection has got to encrypt the content before sending it to the web server of icici bank dot com. For encrypting the connection, what does it require? Public key of icici banks web server right? So, he has to basically get the public key from him. How he is going to get his public key? Using this certificate.

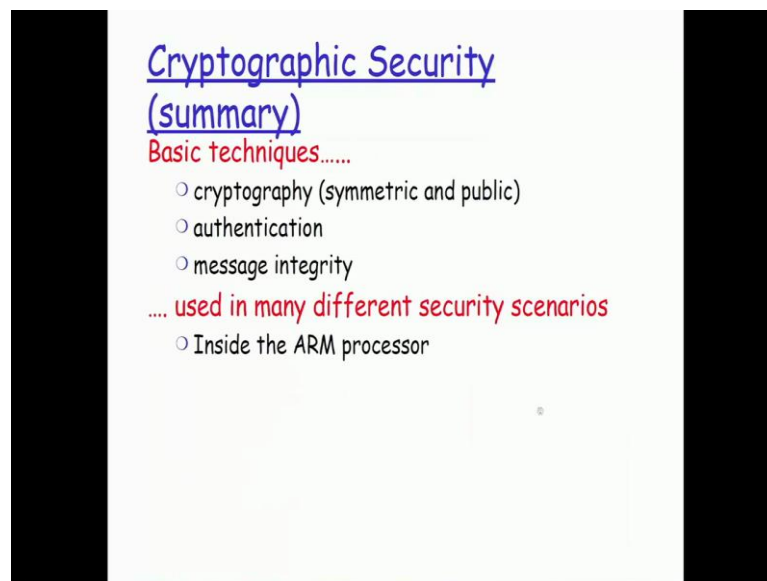
Now anybody can tell somebody else's public key has some other public key. Correct [FL], right? I can go and spoof and say your public key is this to your client. It will become a problem know? Correct? So, somebody has got to authenticate boss icici's public key is only this. No problem. That person is what is called as the Certificate Authority.

Now what we do is the certificate authority is actually squeezes extreme amount of money. A small sized group or an organization cannot afford to pay, this kind of money. So, what they do is they use the open ssl kind of a tool set which also generates the certificate, and then pushes into the web server. So, that is the reason why when you go and access certain web servers certain web sites, you might have seen a pop up box opening up saying that, this digital certificate is not signed by a trusted authority. Do you still want to continue? Correct? Why has that happened? Because I have the digital certificate in the form that I have given here, but the signing authority right the issuer all those things are non-recognized fields. The browser is basically throwing a problem back at you, boss if you accept it, I will continue. If you just try pressing no there, your connection will not be getting set up.

So, I am going to use the public key, but I need to ensure the public key is actually the one for whom it is being claimed for and there is no impersonation there for that I am looking out for somebody to give me a authorized this thing. And in cases where I am not able to buy that kind of a certificate, I have generated myself and then put it into a web server, which is not signed certificate, then you will have a pop up box thrown like this and the browser will basically put the owners on you to decide right whether you want to accept it or not. If you find it fishy you can say no I do not want to take a risk. If you say

that yes I can do it, then it will further go down and then get the certificate, extract the public key from there. Because you see here this is only the entire certificate this is the only thing which is related to the public key everything else is paraphernalia. It will extract the public key from here, use that public key now for the encryption part. So, that is basically where I have to use the public key certificate.

(Refer Slide Time: 34:58)



So, in summary I have gone through the high level overview of symmetric and public cryptography. We have looked at the authentication, we looked at message integrity. Message integrity is basically nothing but the Hash function and the digital certificate. So, now, going forward what you are going to do is, you are going to look at how these different techniques and algorithms are actually made use of by ARM processor as an example right.

So, what kind of components you have inside? Where do you actually make use of each of that? What are the benefits of that? Then I will finally, take you through one device that we are actually build with the ARM processor for you would to get an idea.