

Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Cryptographic Hash Functions (Part I)

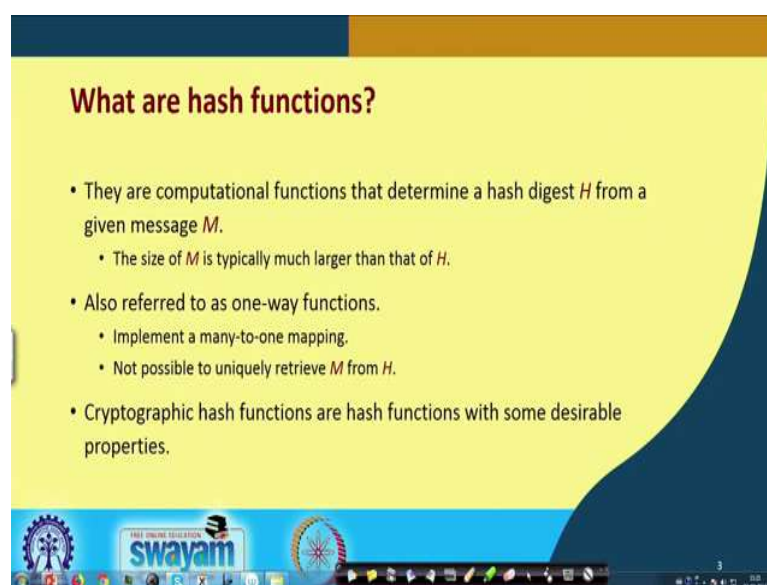
Just we mentioned earlier that in any Security Applications authentication plays a very big role and to implement authentication one of the most important functionalities that we require is something called Hash Functions or Cryptographic Hash Functions. Now in this lecture, we shall be starting our discussion on so called Cryptographic Hash Functions. We shall see what it is exactly.

(Refer Slide Time: 00:45)



So, in this lecture we shall first be talking about some of the desirable properties of a hash function and we shall see that there are broadly two types of hash functions we can use; one that uses a key which is called keyed hash function and another one which does not require a key which is the un-keyed hash function.

(Refer Slide Time: 01:08)



What are hash functions?

- They are computational functions that determine a hash digest H from a given message M .
 - The size of M is typically much larger than that of H .
- Also referred to as one-way functions.
 - Implement a many-to-one mapping.
 - Not possible to uniquely retrieve M from H .
- Cryptographic hash functions are hash functions with some desirable properties.

swayam

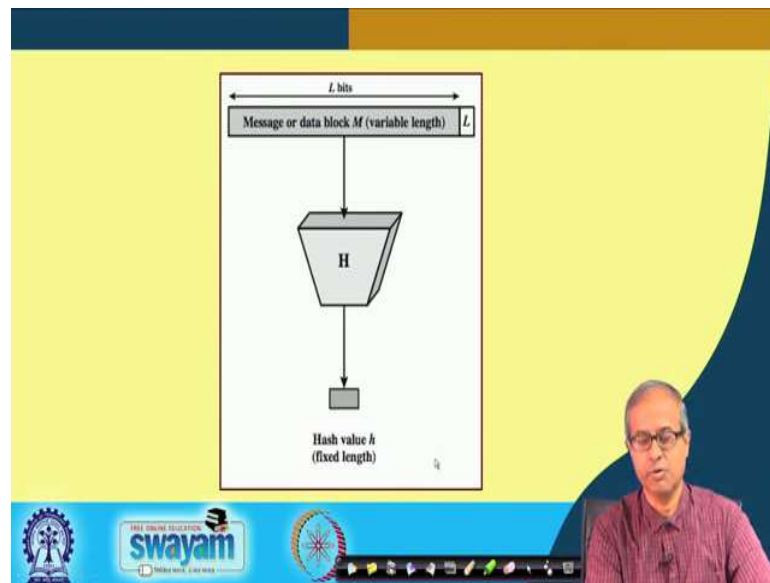
So, broadly speaking, let us first try to see what is a hash function. Well, for those of you who are coming from Computer Science background, you must have studied this topic on hash function, in some courses like data structures and algorithms. There hash functions are used in various search applications, searching data in a table and so on and so forth, ok.

But here let us see in the present context, how we visualize a hash function as? Well we visualize a hash function as, as some kind of computational function that can determine a hash digest H from a given message M which means if you are given a message M , we apply a hash function, we get something called the hash value or hash and digest H . Now the point is that, the size of this M is typically much larger than that of H ; which means we are converting a very large number into a very small hash digest.

Now, because of this kind of many-to-one mapping, this kind of function is also sometimes referred to as a one-way function. Why one way functions? Because, since we are mapping a larger set to a smaller set, it is always the possibility that multiple values of M can map into this same value of H . This is a so called many-to-one mapping, right. So, because of that, we cannot uniquely do the reverse mapping from M to H .

So, for this reason, we call this hash function as a one-way hash function, ok. And hash functions with some particular desirable properties that are more suitable for cryptographic applications; they referred to as cryptographic hash functions, ok.

(Refer Slide Time: 03:20)



So, what I have said is pictorially depicted in this diagram. We have an arbitrary message M ; this can be of any arbitrary length. Well, sometimes we add some additional bits at the end to make it a multiple of some units. To make it, let us say an L bit number the total. Then we apply a hash function, we get a small hash value, which is typically have a fixed length. It can be 128-bit; it can be 160-bit; it can be 512-bit. The size is fixed, fine.

(Refer Slide Time: 03:59)

What is authentication?

- A process through which the identity of the sender can be confirmed.
 - Often makes use of cryptographic hash functions.
- Why required?
 - Many applications require one of the parties to confirm the identity of the other party.
 - Security applications heavily use authentication.
- We discuss authentication methods using cryptographic techniques.
 - Other methods like biometric authentication require physical presence of the sender.

Now, I had mentioned one of the main or primary applications of hash functions is in the domain of authentication. So, I am again repeating what exactly authentication is? Well

authentication is a process through which we are trying to uniquely identify the person who is sending the message; that means, the identity of the sender has to be confirmed through this process.

So, here as I had said, we make use of cryptographic hash functions. Now why do we need authentication; because you see in the Internet scenario, we really cannot see the other end of a communication. We are carrying out the communication supposedly with some other person who I think that person is, but I have no way to confirm that exactly that same person is at the other end.

So, this authentication plays a big role in that respect, ok. So, there are many applications which actually required this kind of thing to be done. Require one of the parties that are involved in the communication must confirm the identity of the other party with whom he or she is communicating, ok.

So, most of the security applications starting from password based authentication to many others. They very heavily rely on authentication, ok. We shall be discussing some of these methods which are based on cryptographic techniques. Of course, here in this lecture, we shall not be talking about something like biometric authentication where do we give our fingerprint or iris, but there the physical presence of the person is required; but here we are talking of the Internet where some messages are being transmitted and we have to carry out authentication, ok. So, we cannot carry out biometric authentication.

(Refer Slide Time: 06:00)



Approaches to Message Authentication

- a) Authentication using conventional encryption.
 - Only the sender and receiver should share a key.
- b) Message authentication without message encryption.
 - An authentication tag is generated and appended to each message.
- c) Message authentication code.
Calculate the MAC as a function of the message and the key:
$$MAC = F(K, M)$$

The slide features a yellow background with a dark blue curved border on the right. At the bottom, there is a blue banner with logos for 'THE OPEN UNIVERSITY', 'swayam', and 'MOOCs'. A small video inset in the bottom right corner shows a man with glasses speaking.

So, this I have already mentioned earlier also. Broadly message authentication can be carried out using conventional encryption that we shall see that using either symmetric key or public encryption, we can carry out some authentication; or even without encryption we can do message authentication. We shall see these schemes. How we can do so? Here we shall be generating some kind of an authentication tag and we will be appending it to the message and then the other side, some kind of a verification can be carried out, ok.

And there is something called a message authentication code which is very general that also we can use; where generally as this equation shows, we apply some kind of a function F on a message M with the help of a secret key value K just like encryption, similar to that and you generate something called a message authentication code or MAC that is appended with the message, so with the help of which we can authenticate, ok.

(Refer Slide Time: 07:10)

Hash Functions: Classification

- a) Unkeyed hash function or *modification detection code* (MDC):
 - Used to preserve integrity of message.
- b) Keyed hash function or *message authentication code* (MAC):
 - Used to authenticate the source of a message in addition to preserving integrity of the message.

The slide is part of a video lecture. At the bottom, there is a blue banner with the 'swayam' logo and text 'FREE ONLINE EDUCATION'. A small inset video shows a man with glasses speaking. The bottom of the slide also features a standard presentation navigation bar with icons for back, forward, and other controls.

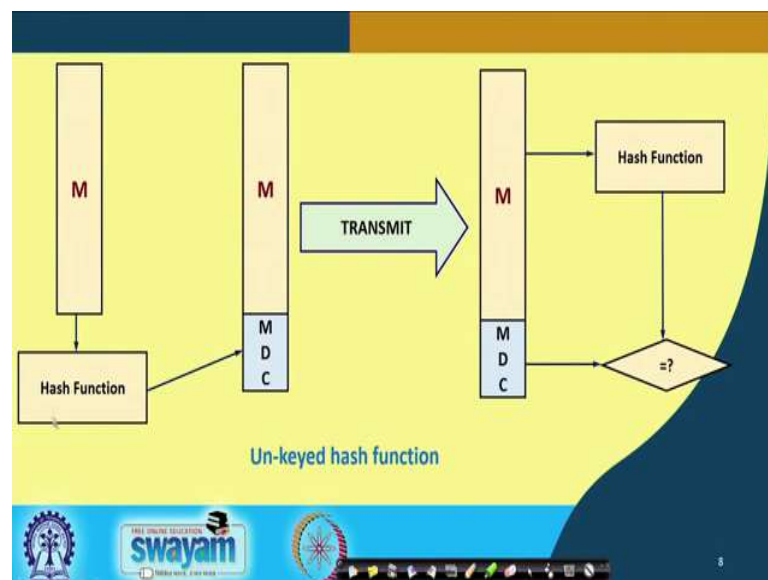
Now broadly speaking, I said that these kind of hash functions can be either keyed or unkeyed.

There can be a secret key or there can be no secret key. Based on that we can broadly classify hash functions into something called modification detection codes which are unkeyed, which do not rely on any key. So, whenever the messages modified, that can be detected, ok. So, here this kind of hash function is used to preserve the integrity of message; which means the receiving end can tell whether the message is intact or it was modified.

So, whenever the message is modified, this so called MDC which is also coming as part of the message, will get modified, ok. And the other one is that uses a secret key. As I just now said, with message authentication code here we can preserve integrity of the message as well as we can identify the sender who is sending the message.

So, it is the second one which can be used for authentication, but the first one, you could use more for the purpose of verifying the integrity of a message. So, you see this application of both kinds of hash functions are there in real security scenarios, security applications, ok.

(Refer Slide Time: 08:39)



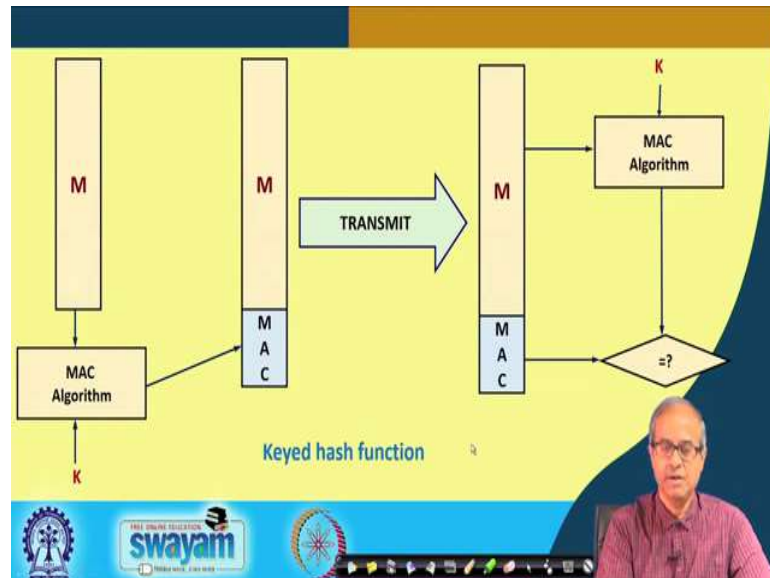
So, this is a pictorial depiction of an un-keyed hash function. So, you see, here on the left side, I have a message M that I want to transmit to a receiver. So, what I do; I first apply a hash function on this message M, I get a hash digest MDC, modification detection code and what I do? I append this MDC along with the message and I transmit this whole thing over the internet to the other side.

So, receiver will receive that same thing, the message M along with the MDC appended to it. So, what the receiver will do? The receiver will take out this message, will apply that same hash function which is known to both the parties.

And here also some MDC will get generated. So, that MDC and the MDC that was received, will be compared. So, if the message was modified in transit so, immediately

there will be a mismatch. You will know that the message integrity has been lost. So, this is one simple way in which you can check the integrity of a message.

(Refer Slide Time: 09:55)

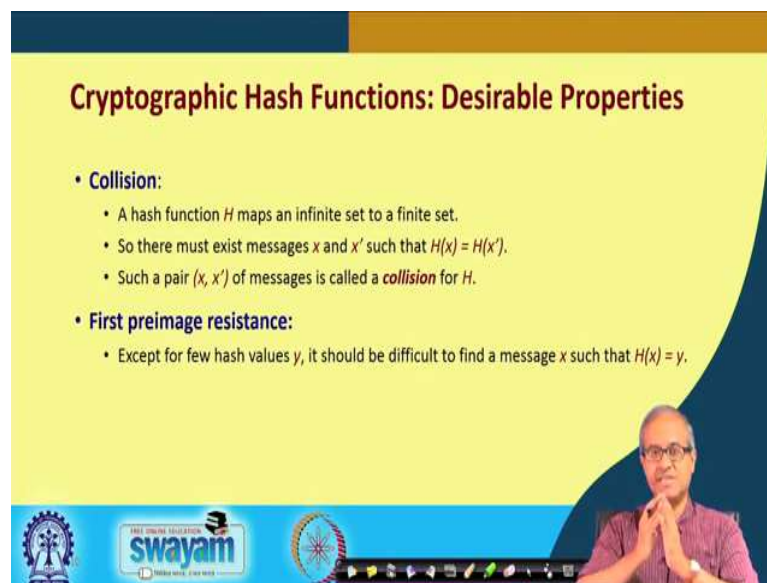


Now, talking about keyed hash function, here we are talking about a MAC algorithm which generates a message authentication code, which also takes a secret key K . I am assuming here in this diagram, that this secret key is shared by the sender and the receiver both sides. So, just like encryption and decryption.

So, the sender will generate this MAC, sorry using this key and the message M and this MAC, MAC will be appended to the message. This will be transmitted and in a very similar way at the receiver end this message will again be fed to the MAC algorithm with the same key value K . So, again a MAC will be generated and you compare these two MACs.

Now the MAC is generated using this key value and the MAC just transmitted, if they are not matching then either your key is wrong or your message was modified. So, you can identify the sender also, because if you are using the same key with the sender, that means, you know that the same secret key is also lying with that sender. It must be that sender sending it. So, this is keyed hash function.

(Refer Slide Time: 11:15)



Cryptographic Hash Functions: Desirable Properties

- **Collision:**
 - A hash function H maps an infinite set to a finite set.
 - So there must exist messages x and x' such that $H(x) = H(x')$.
 - Such a pair (x, x') of messages is called a **collision** for H .
- **First preimage resistance:**
 - Except for few hash values y , it should be difficult to find a message x such that $H(x) = y$.

Now talking about cryptographic hash function as I had said, they should have some desirable properties. Let us very quickly look at, what these desirable properties are? Well any hash function, because it maps a larger set to a smaller set, there will be occurrence of something called collision.

So, what is a collision? Collision means because you are mapping a larger set to a smaller set, there will always be two distinct messages say x and x' which will map to this same hash value; because your message can be 1000 bit, your hash can be a 100 bits, ok. So, there will always be a minute to one mapping and there will be such collisions happening.

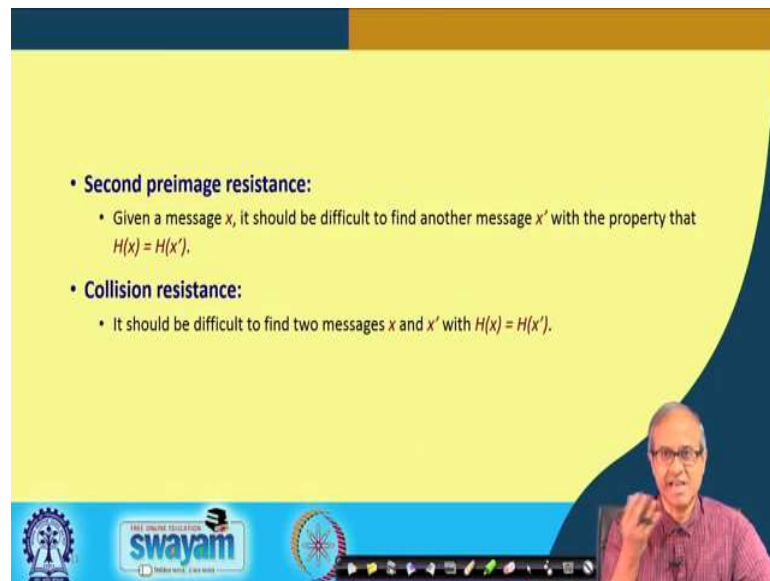
So, there must be two messages x and x' such that their hash values will be the same. This is what we mean by collision and such a pair of messages x and x' are set to result in a collision with respect to a hash function H , ok.

Now talking about the properties, the first is called first preimage resistance; well what this property says? It says that except for a few hash values y , it should be very difficult to find a message x such as $H(x) = y$; which means I have given you some hash value and I asked you find out a message for which the hash value will be equal to that, I have given you.

So, it is not easy to do that; well if it were easy then an intruder can fabricate a message, modify a message in such a way that the same hash value will result. That is difficult to do.

This is what is referred to as the first property of first preimage resistance.

(Refer Slide Time: 13:19)

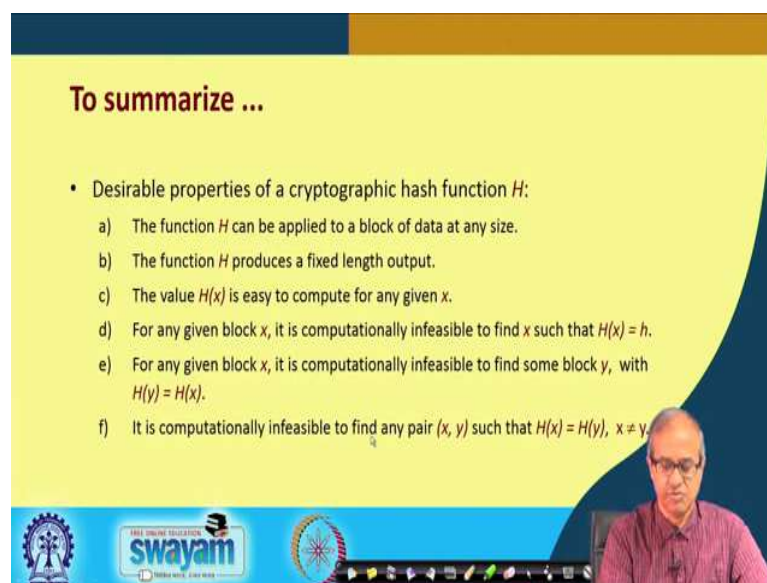


- **Second preimage resistance:**
 - Given a message x , it should be difficult to find another message x' with the property that $H(x) = H(x')$.
- **Collision resistance:**
 - It should be difficult to find two messages x and x' with $H(x) = H(x')$.

Then comes the second level, second preimage resistance what it says, here I have given a message; given a message x , it is difficult to find another message x prime for which the hash values will be the same. Like I am sending a message, the intruder cannot fabricate another different message for which the hash value will be the same as my original message. So, it is not easy to do that; that is another property.

And finally, of course, collision resistance; it should not be very easy to do. It will be very difficult to find two messages whose hash values are same. So, if all these properties hold, then you say that my hash value is good; I can use it for a real cryptographic security application, ok.

(Refer Slide Time: 14:14)



To summarize ...

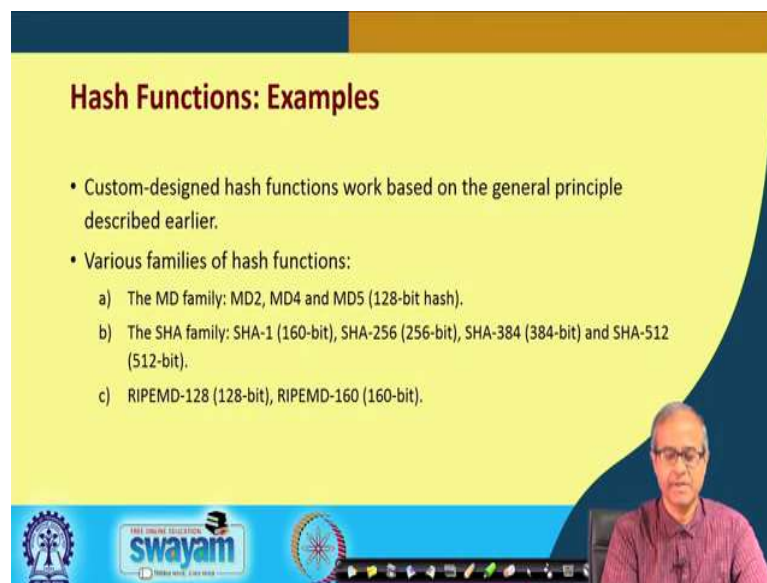
- Desirable properties of a cryptographic hash function H :
 - a) The function H can be applied to a block of data at any size.
 - b) The function H produces a fixed length output.
 - c) The value $H(x)$ is easy to compute for any given x .
 - d) For any given block x , it is computationally infeasible to find x such that $H(x) = h$.
 - e) For any given block x , it is computationally infeasible to find some block y , with $H(y) = H(x)$.
 - f) It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$, $x \neq y$.

So, whatever we had said just to summarize these things, desirable properties of a hash function H ; the first thing of course, is this H should be applicable to a message of arbitrary size. Of course, not arbitrary long size, there is of course, an upper limit; but that upper limit should be very large.

So, roughly speaking I can say, block data, block of data of any size up to some reasonable upper limit and this hash function typically will produce a fixed length of output as the hash value and these are the three properties I talked about c, d and e; the value $H(x)$ is easy to compute for any given, this is of course, in terms of computation. It should not be too computationally intensive and the last three points, these are the first preimage resistance, second preimage resistance, and the collision properties. These three properties are mentioned here; the same thing I just now talked about, right.

So, broadly speaking hash function should be able to satisfy these properties and also, they should not be too computationally expensive to, you can set, compute the hash function, it should be efficient, ok.

(Refer Slide Time: 15:38)



Hash Functions: Examples

- Custom-designed hash functions work based on the general principle described earlier.
- Various families of hash functions:
 - a) The MD family: MD2, MD4 and MD5 (128-bit hash).
 - b) The SHA family: SHA-1 (160-bit), SHA-256 (256-bit), SHA-384 (384-bit) and SHA-512 (512-bit).
 - c) RIPEMD-128 (128-bit), RIPEMD-160 (160-bit).

The slide features a yellow background with a blue header and footer. The footer includes the Swamyam logo and a video inset of a man in a red shirt speaking.

So, some examples of hash functions, ok, there are many types and classes of hash functions that have been proposed and many of them have been used. They all satisfy the properties I have mentioned, desirable properties and broadly speaking they can be classified into some families; first is the MD families: MD2, MD4, MD5 which generates a 128-bit hash. Now at one point at time MD5 was used very widely, but subsequently some weaknesses in MD5 was found out, some collision weaknesses was found, ok.

So, at present MD5 is not that widely used, in some occasional case it is used; but it is another family called SHA, we call it a SHA family that is much more widely used. And there are several variations in the SHA family which generates hash values or digest of various different sizes; 160-bits, 250-bits, 384 or 512. There are other intermediate varieties also and there is another class called RIPEMD, they also have 128 and 160-bit versions.

So, with this we come to the end of this first lecture on cryptographic hash function. Now we shall be continuing with our discussion in the next lecture, where we shall be seeing some properties of one-way hash function, the different ways we can use it and as specific case studies. We shall be looking at a couple of hash functions, how they exactly look like.

Thank you.