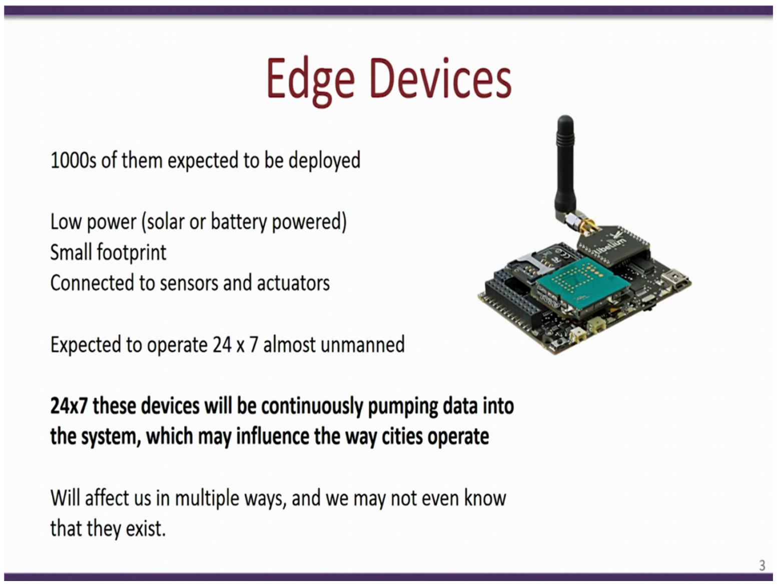
Information Security - 5 - Secure Systems Engineering
Professor Chester Rebeiro
Indian Institute of Technology, Madras
Physically Unclonable Functions (part 1)

Hello and welcome to this lecture, so we will take a slight deviation from what we have done so far. And we will actually look at something known as hardware security. So hardware security is quite an upcoming field and it actually deals with security in chips, processors and so on. Since this is at the base of your computer systems, the security of these hardware could actually impact all the things which come above. So for example, a flaw in your hardware could actually be used to steal or compromise everything about that hardware like the BIOS operating system and the various applications. So there are hardware security itself is a very broad field, so we will be starting this video with a very something known as physically unclonable functions or PUFs.

So there will be multiple such videos, this is the 1st part of it. Essentially we will be actually looking at this paper or we will be discussing this paper called physically unclonable functions and applications tutorial, so you can download this tutorial from this IEEE website. A major application for PUFs is for authentication.

(Refer Slide Time: 1:38)



Edge Devices

1000s of them expected to be deployed

Low power (solar or battery powered)
Small footprint
Connected to sensors and actuators

Expected to operate 24 x 7 almost unmanned

24x7 these devices will be continuously pumping data into the system, which may influence the way cities operate

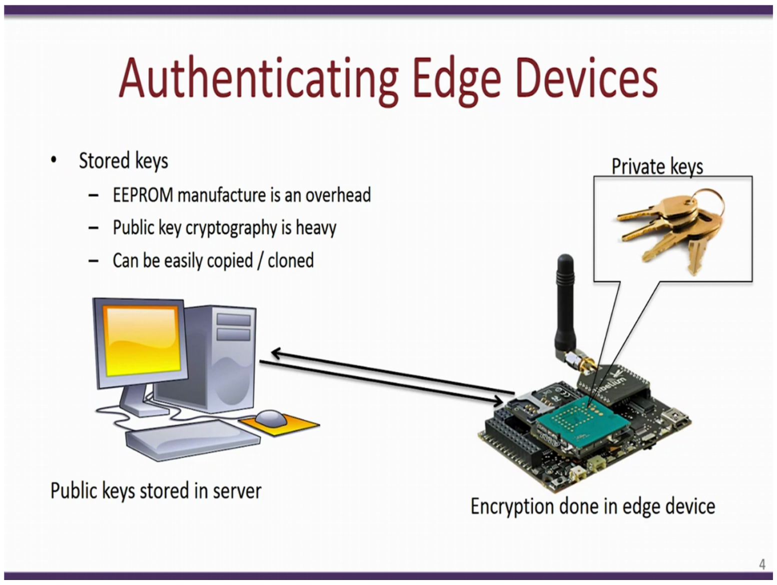
Will affect us in multiple ways, and we may not even know that they exist.

3

So authentication especially for edge device, devices like which I use for IOT is extremely important, the reason being that there are like thousands of IOTs that are expected to be deployed in the next few years, all of these devices are low powered, they have small footprints, memory footprints and quite often connected to sensors and actuators in process control plants. Now a problem with having such other characteristics, low-profile is that a lot of cryptographic algorithms will not work on this, especially the public cryptography algorithms will not work on this. Several of especially the Public cryptographic algorithms quite complex and therefore would require considerable amount of compute power and memory in order to execute therefore authenticating these devices is actually a problem.

The 2nd issue is that authentication of these devices cannot be ignored, the fact is that these edge devices are quite critically connected to various components of process control system it could for example be connected to some of the actuators or monitoring elements in nuclear thermal plants, and therefore the data-processing which is actually collected by this device is actively important for the functioning or the correctness of the entire plant. Other features of these devices is that it has to operate 24 by 7 and it is completely unmanned and therefore the security of these devices are extremely important.

(Refer Slide Time: 3:34)



Authenticating Edge Devices

- Stored keys
  - EEPROM manufacture is an overhead
  - Public key cryptography is heavy
  - Can be easily copied / cloned

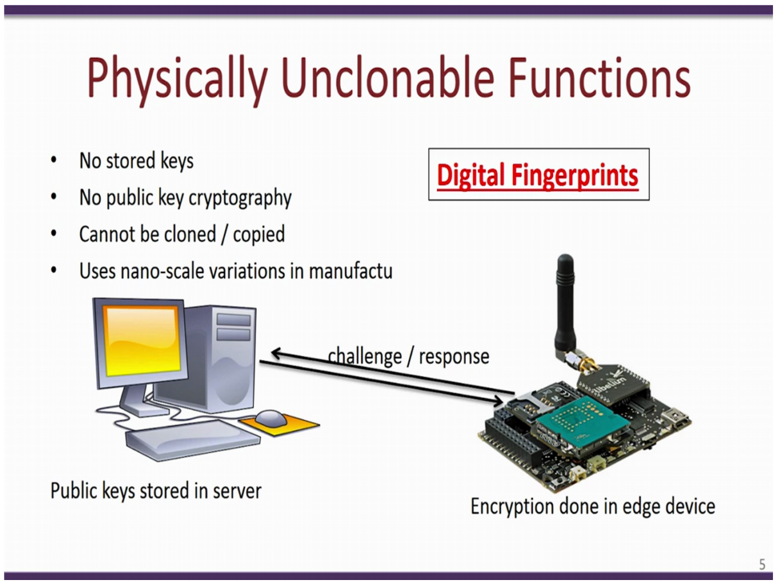
Private keys
Public keys stored in server
Encryption done in edge device
4

Now the typical way to actually authenticate these is to store a secret key in Flash device or a ROM device present in this device, so this secret key would then be used to with lightweight ciphers, there are several lightweight ciphers which have been developed. So one of these ciphers would be used to actually do encryptions and then there would be an authentication protocol with a server. The server would also have the same private keys for all the devices that is connected to and it would send, during the handshake or during the connection between the server and this edge device, the private keys would be used to authenticate the validity of this device.

So a similar type of private keys that you would see also in various other devices like smart cards, credit cards and so on and all of these devices essentially have a large random number which is stored, which is then used to identify the device. The problem with this approach is that this private key is stored in the device requires special memory known as EEPROM, which is considerably overhead especially to manufacturer. Further, as many of you would know smart cards and other such low-end devices can be easily cloned or copied, so this means that I could actually clone this particular hardware, create another hardware which has exactly the same private key.

The issue with this is that now I would have two devices and both are can be authenticated by the same server because they would have the same private key in them, essentially both are clones of each other. Quite recently there has been a new technique which was suggested to replace the use of private keys as a mean to authenticate device, so this is known as physically unclonable functions.

(Refer Slide Time: 5:28)



Physically Unclonable Functions

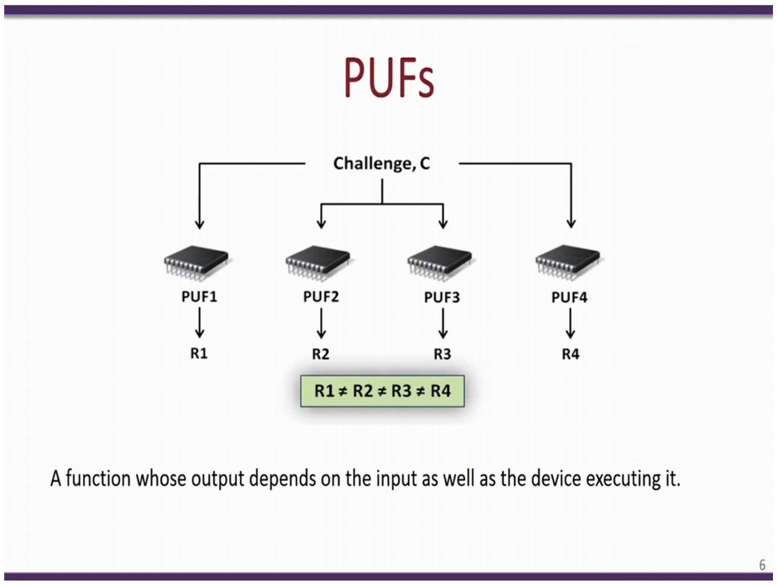
- No stored keys
- No public key cryptography
- Cannot be cloned / copied
- Uses nano-scale variations in manufactu

Digital Fingerprints

challenge / response
Public keys stored in server
Encryption done in edge device
5

So physically ungovernable functions can be used for authenticating devices, it does not have require any stored keys, typically does not require any public key cryptography, and furthermore it also (())(5:39) the drawbacks of authentication with the stored keys. So using physically and clone able functions, it is not possible to actually clone I device and therefore, you get much better security. Now what are physically unclonable functions? So essentially this physically unclonable functions are something like digital fingerprints which can uniquely identify a device. They are based on nanoscale variations in which are present during the manufacturing of the device, so we will see this in little more details in our later slide.

(Refer Slide Time: 6:23)



PUFs

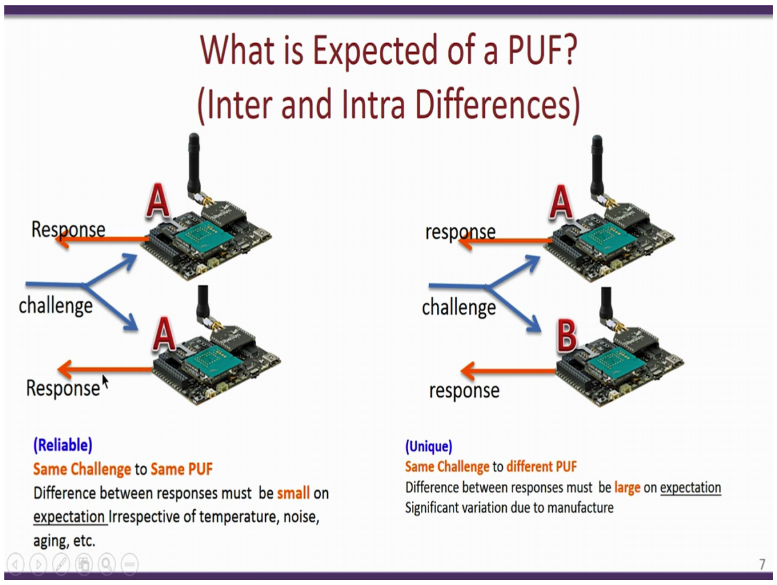
A function whose output depends on the input as well as the device executing it.

6

So basically a PUF is a function, it takes an input known as challenge and provides an output which is known as the response. However, the way it is different from other functions is that the response not only depends on the input challenge that is given but also on the device where the function is executing in. So for example, let us say that I have a function and this exactly same function is implemented in 4 different devices as shown over here. If I give the same input to all of the 4 functions, what I would expect is 4 responses and all of the responses would be different. So what this means is that given a particular challenge I can use the response to essentially identify which device has executed that particular function.

So note that this is very different from any other function like the sine function or cos function or any other functions that we typically implement as a C program or any software or hardware. Over there for a given input independent of the device you would get the same output however, with a PUF the output will differ based on which device is executing that PUF function. So this feature of PUF is what is actually used to authenticate a device, now there have been several PUFS which have been proposed over the last 10 to 15 years or so and therefore there has been various classification and ranking of these pufs to actually sign some PUFS as good PUFs or some as bad PUFs and so on.

(Refer Slide Time: 8:15)



What is Expected of a PUF? (Inter and Intra Differences)

(Reliable)
Same Challenge to **Same PUF**
Difference between responses must be **small** on expectation Irrespective of temperature, noise, aging, etc.

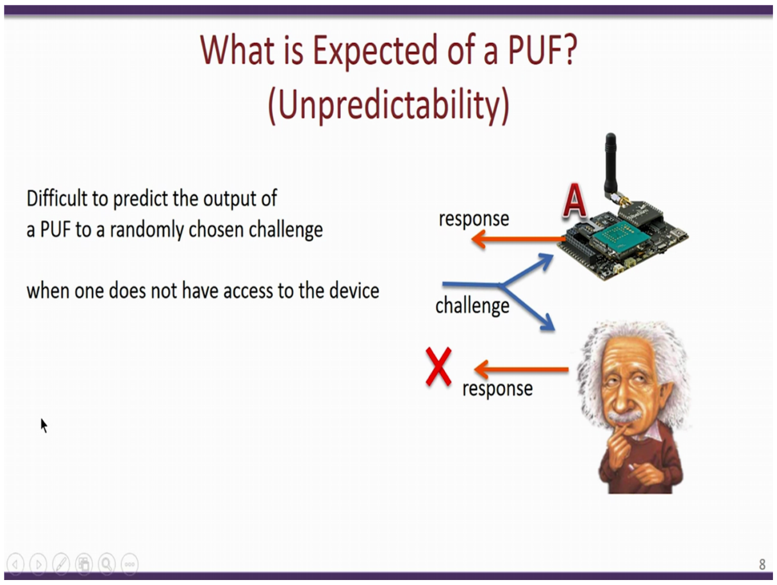
(Unique)
Same Challenge to **different PUF**
Difference between responses must be **large** on expectation
Significant variation due to manufacture

7

So typically what is expected of a PUF is that if I have two devices which are exactly identical and I provide the same challenge to both the devices, then what a PUF function should do is that it should provide a response which is different, essentially each device should provide a unique response for the same challenge. Further, the difference between this response should be should be significant, so this is known as the inter-difference, and another requirement for PUF is the intra-difference. This means that if I take a particular device say device A, send it a particular challenge and obtain its response and after some time send the challenge to the same device and collect the response.

And what is expected of a good PUF is that these 2 responses taken from the same device after a period of time should be as close as possible or should be exactly identical, so this is a feature which is known as reliability of a PUF and therefore these 2 would actually form the most critical aspects for gauging the strength of PUF. The firm should not only be reliable but also should provide unique responses with respect to different challenges and different devices.

(Refer Slide Time: 9:45)



What is Expected of a PUF? (Unpredictability)

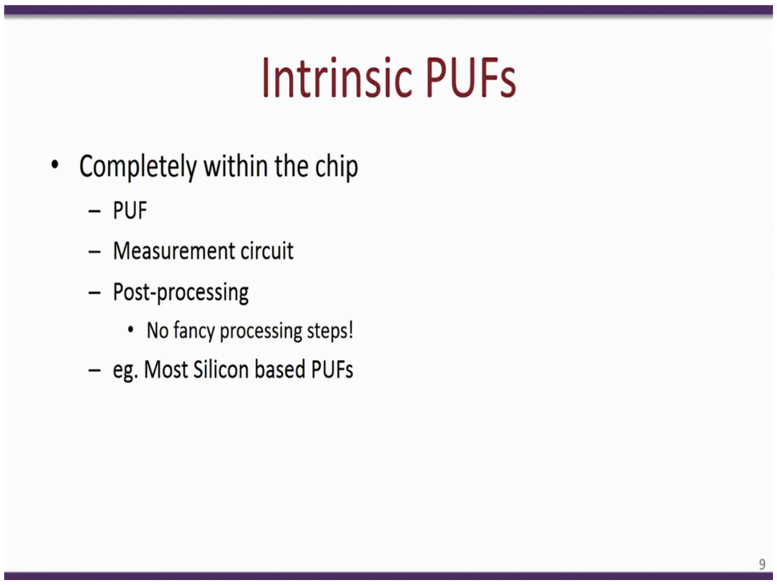
Difficult to predict the output of a PUF to a randomly chosen challenge

when one does not have access to the device

response
A
challenge
X
response

Another feature which is important in a good PUF is the unpredictability. What this means is that the ability to actually provide the current response to a challenge should require the presence of the device. So for example, if I provide a challenge and I obtain a response I should be guaranteed that this response is actually originated from that device and not from some other algorithm or some other technique, so someone could actually predict the response for the PUF for that particular challenge. So these are the 3 things the uniqueness, reliability, and the unpredictability that is required for every PUF, so many of these things are orthogonal to each other and achieving all 3 and the same PUF is extremely challenging problem and a lot of researchers are actually working on this to actually create design PUFs which could achieve all of these 3 features.

(Refer Slide Time: 10:44)



Intrinsic PUFs

- Completely within the chip
  - PUF
  - Measurement circuit
  - Post-processing
    - No fancy processing steps!
  - eg. Most Silicon based PUFs

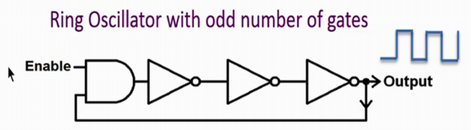
9

As mentioned before so there are lots of PUFs which have been proposed in the last 15 to 20 years, types of PUFs which are actually been used quite often these days something known as intrinsic PUFs. So these are PUFs which can be completely implemented within a chip that is you would have a PUF function, the measurement circuit to actually measure the response and also post processing is also present within that single chip, most PUFs that we used these days are with most PUFs which we actually consider these days are all intrinsic PUFs. So in this video lecture we will actually look at to PUFs; these are the 2 most common PUFs which are evaluated and used these days, so this is the arbiter PUF and something known as the ring oscillator PUF.

(Refer Slide Time: 11:37)

Silicon PUFs

eg. Ring Oscillator PUF



$$f = \frac{1}{2nt}$$

f Frequency of ring oscillator

n Number of stages

t Delay of each stage

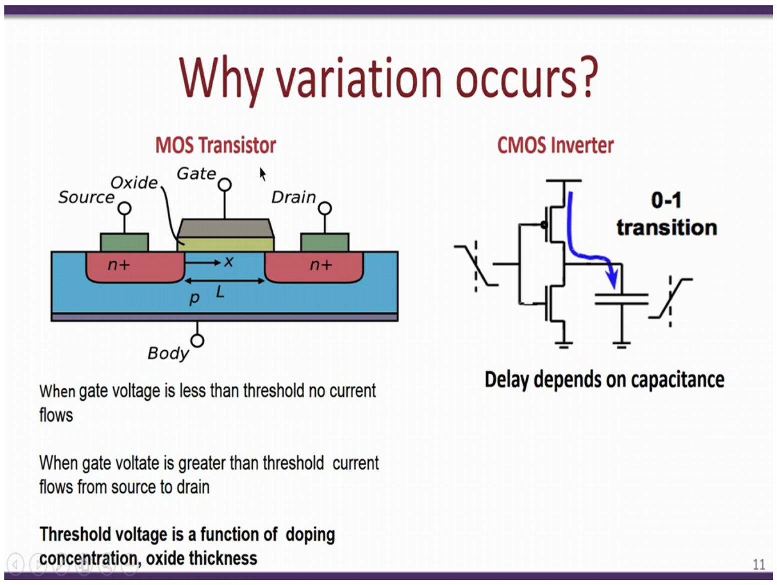
10

So we will start with ring oscillator, we will show how ring oscillator can be used as a PUF. So essentially, we could consider a ring oscillators that looks something like this, so it has an AND gate over here and an enable. So whenever there is an enable that is whenever the enable is set to 1, this AND gate would pass the other signal through right. And besides these there are odd number of inverters that are pleasant and finally it has a feedback from the output to the AND gate. So let us say that the enable bit is set to 1 and let us assume that the other input has a value 0 and therefore the output of this and gate would also be 0, so this gets 0 gets inverted to 1 then 0 and then 1 again and then there is a feedback.

So now your input to the handset has changed from 0 to 1 and the result of this is that 1 gets converted to 0 then 1 and 0 over here, and then the output changes to 0 again. So what we see is happening here is that this output continuously changes from 0 to 1 and so on, so it essentially creates a periodic output of 1 and 0. Now it would look the output would look something like this, the frequency of the output depends on multiple factors. First it depends on the number of NOT gates that are present in this ring oscillator for example, if you have more number of inverters gates then the frequency would be of this waveform would be lower because essentially the signal takes a longer time to propagate to the output.

Another aspect which influences the frequency of this ring oscillator output is the delay of each stage. So for example, if the delay of each inverter present is a very short then the signal would take a shorter time to reach the output and as a result you will get a higher frequency. On the other hand, if the delay of each inverter is long than the frequency of the ring oscillator output will be much smaller, so we could actually write the frequency of ring oscillator PUF like this, so as is equal to 1 by 2 n t. As you increase n, the number of stages in the ring oscillator or t the delay of each stage, the frequency of the output of the ring oscillator would reduce and vice versa. So while it is easy to understand that n that is the number of stages in ring oscillator upends the frequency, the delay of each inverter that is t actually depends upon the fabrication process of these gates.

(Refer Slide Time: 14:48)



Why variation occurs?

MOS Transistor
Source
Oxide
Gate
Drain
n+
x
n+
p
L
Body
CMOS Inverter
0-1 transition

When gate voltage is less than threshold no current flows

When gate voltate is greater than threshold current flows from source to drain

Threshold voltage is a function of doping concentration, oxide thickness

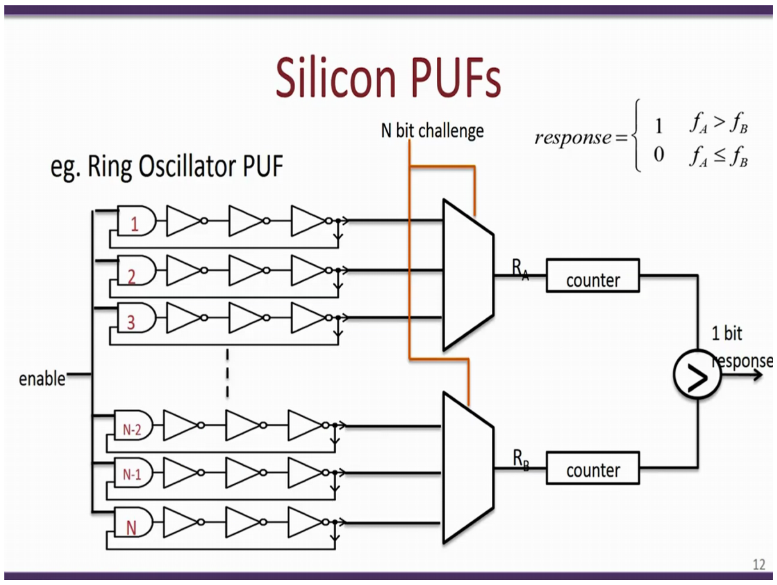
Delay depends on capacitance

11

So if we look at a typical MOS gate as many of us know, so it has a source, drain and gate and there is a channel and established between the source and drain depending on the voltage available at the gate. A CMOS inverter look something like this, you have and NMOS and a PMOS transistor which are connected in this manner and what you see is that when the input at the gate, when the input goes from 1 to 0, the output changes from 0 to 1. So during the fabrication of these transistors and gates, that could be multiple different ways, one CMOS inverter differs from another CMOS inverter. This could be due to silicon wafers that are used because no silicon wafers would be exactly identical, it could also be due to other factors such as the doping concentration or the oxide thickness and so on which is going to be unique for each transistor.

As a result of this, there will be minor variations in that threshold voltage for each and every gate that is manufactured. So what I mean by this is that if I take 2 transistors which have been fabricated by exactly the same process and you could also assume that these transistors are manufactured one after the other, still there are minor nanoscale variations between these transistors and as a result of this the behavior of these transistors when added to circuit such as CMOS inverter would vary very marginally, so it is this marginal difference that causes delay in the oscillator due to the T part and this is what is used by PUFs to extract the signature for that particular device.

(Refer Slide Time: 16:45)



Silicon PUFs

$$response = \begin{cases} 1 & f_A > f_B \\ 0 & f_A \le f_B \end{cases}$$

So this is how a typical ring oscillator PUF is used, so note that we have looked at ring oscillator that is this part over here comprising of the AND gate and multiple odd number of inverters. And in order to build a ring oscillator pub, we would use many such ring oscillators and 2 multiplexers, so in this particular slide we have shown that we have used N ring oscillators, we have 2 multiplexers and a counter and 1 bit response. So depending on the characteristics of each PUF the response would be either 1 or 0, so we will not go into details about this and how this ring oscillators PUF actually works, this we will actually keep for the next video. In the next lecture we will also look at arbiter PUF and we will also compare the ring oscillator PUF and the arbiter PUF and see what are the characteristics and where each one can be used, thank you.