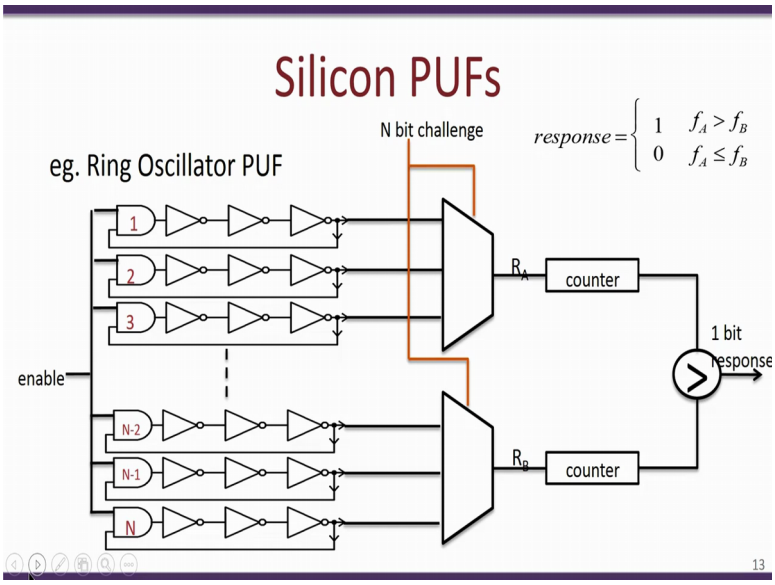
Information Security - 5 - Secure Systems Engineering
Professor Chester Rebeiro
Indian Institute of Technology, Madras
Physically Unclonable Functions (part 2)

So hello and welcome to this video lecture, so this video lecture continues from the last one where we spoke about Physically unclonable functions and how they could possibly used for authentication and for other security aspects, especially they would be important for lightweight devices like edge devices that are used in IOT. We also mentioned that there are 2 types of very commonly used PUFs, one was the arbiter PUF and other was the ring oscillator PUF, so in this lecture we will continue from where we stopped. We will look more in detail about the ring oscillator and arbiter PUF, we will compare the 2 of them and then we will actually see how we could build authentication schemes, what are the current drawbacks of PUFs and how potentially they can be mitigated.

(Refer Slide Time: 1:09)



So recollect that we actually looked at ring oscillator PUF which was something like this, so there were a series of ring oscillators, over here we had N oscillators and each ring oscillator had in this figure at least has 3 inverter gates, and output of the 3rd one is actually looped back and connected to the 1st inverter. Now, we have all of these ring oscillator outputs connected to multiplexers, so this is shown as two N by 2 bit multiplexers but we can actually have them as N bit multiplexers, and then you have counters and response which is of 1 bit. The ring oscillators PUF is something like this, to actually start the ring oscillator PUF the user

would have to give an enable signal essentially, essentially change the enable from 0 to 1 and also specify a challenge.

So a challenge over here would essentially pick choose 2 ring oscillators out of the N oscillators. So out of the N ring oscillator present, the challenge would essentially pick 2 of them for example, let us say for discussion it picks say the ring oscillator 2 and ring oscillator N. Now as we know, when the enable signal is 1, there is an initial state of the PUF which gets fed back and as a result there is a square waveform which gets present at the output of the PUF. As discussed in the previous lecture the frequency of this waveform is a function of these manufacturing process of this PUF. So for example, we mentioned that there is capacitance that is involved; there is that threshold voltage and various other nanoscale aspects that could actually change the frequency of the ring oscillator PUF.

Therefore what is expected is that each of these N ring oscillators would produce a frequency that is different. Now as we mentioned, what is done is that a challenge would actually pick 2 ring oscillators at random, so we had considered in a our discussion the ring oscillator 2 and N. And because of these intrinsic properties the frequency of the waveform generated by the ring oscillators 2 and N would be different. Now what is done over here is that there is a multiplexers to multiplex the ring oscillator 2 and the ring oscillator N therefore what we obtain is RA and RB both are square waveforms. One is this RA is due to this ring oscillator 2, and the multiplexers which has switched 2^{nd} ring oscillator into its output and this multiplexer has switched the Nth ring oscillator to RB.

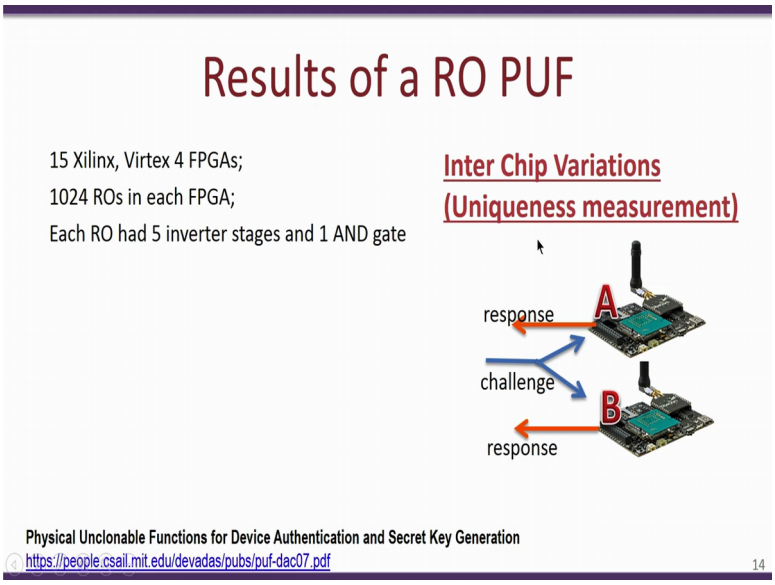
Now we have two counters; both the counters start with 0 and they begin counting each periodic or each positive pulse that is obtained from the ring oscillator. So therefore, what we know is that since the 2 ring oscillators are operating at different frequencies due their intrinsic properties, these 2 counters would after a period of time say like 1 or 2 seconds would obtain a different count value. Now based on this we would make a comparison after say 1 or 2 seconds and determine which of these 2 counters is higher and according to that we would give output of 1 or 0. So the response of the challenge is one order 0 in our example, if the frequency F A corresponding to this counter has a higher value than we provide an output 1, else we provide an output 0.

What we see over here is that the hardware device would implement this ring oscillator PUF and later when deployed, an application could unable this ring was B PUF, choose a challenge, essentially choose a pair of these ring oscillators, provide an output which is either

1 or 0. Now the entire purpose of this ring oscillator PUF or the entire uniqueness of this ring oscillator PUF as mentioned in the previous video is that this output response is going to be a function of that particular device. If I have two exactly identical devices, each of these devices having a ring oscillator PUF, each would give me a different response for a particular challenge.

So what is done is that this one challenge gives me one bit of response, so if we actually run this ring oscillator with multiple different challenges we could build larger output response so for example, if I continuously choose different challenges I would get a larger string of responses, each response is independent of the other.

(Refer Slide Time: 6:31)

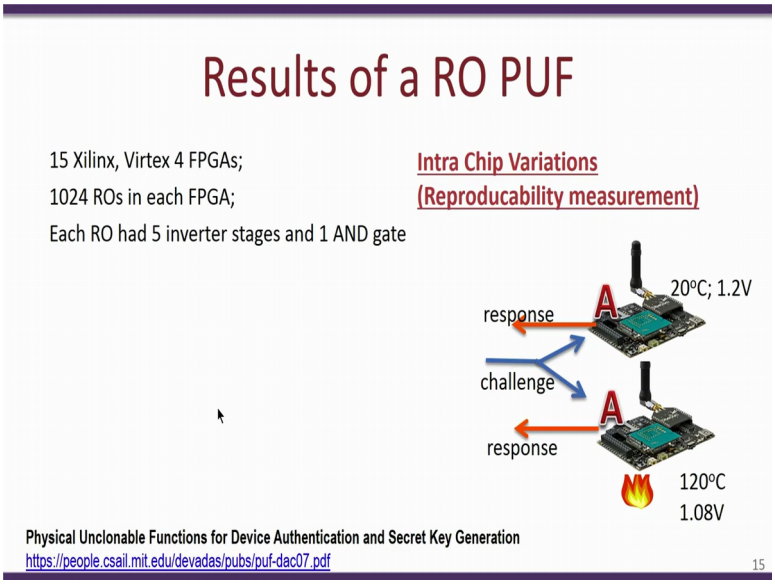


So we will take an example and we will actually look at various properties of the ring oscillator PUF, so we will be referring to this particular paper which was by Professor Devdas in DAT 2007. So this paper shows the implementation of a ring oscillator PUF vertex 4 FPGA, so they compared 15 such devices, all of these devices are exactly identical and they implemented 1024 ring oscillators in each FPGA, this means that the N over there was 1024. Further, they also used that instead of 3 inverters they had used 5 inverters, so **one** each ring oscillator had 5 inverters and an AND gate. This particular figure shows the inter-chip variation or the uniqueness of the PUF or the uniqueness of the PUF response.

For a given challenge, the same challenge sent to the device A and in other device B, the responses are connected and the hamming distance between the 2 responses is computed. This particular craft shows the response for 128 bits. Now what is expected is that for a good

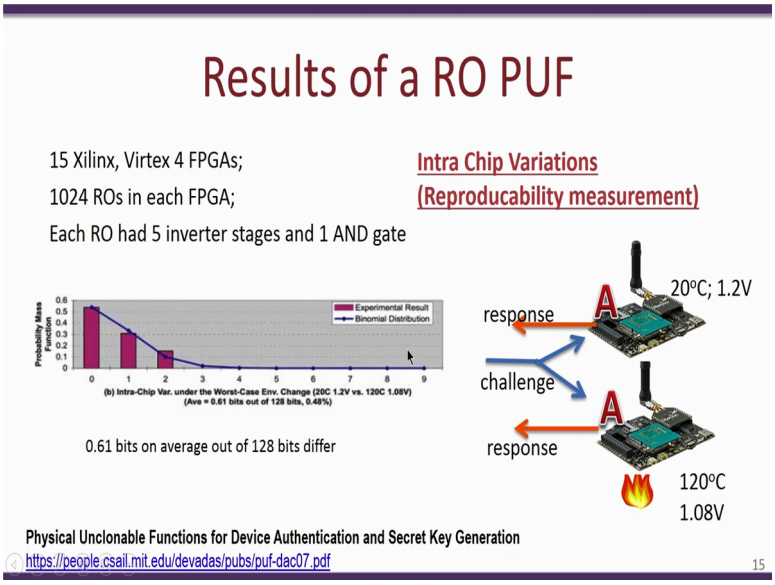
PUF the inter-chip variation should be maximum, so this means that the response of A should be as different as possible from the response of B. So if we are considering that each response of 128 bits in order to have maximum difference between A and B, ideally A and B should vary in 64-bits. And as we see over here, on the X-axis it shows the Hamming distance between A and B and the Y-axis shows the probability.

(Refer Slide Time: 9:01)



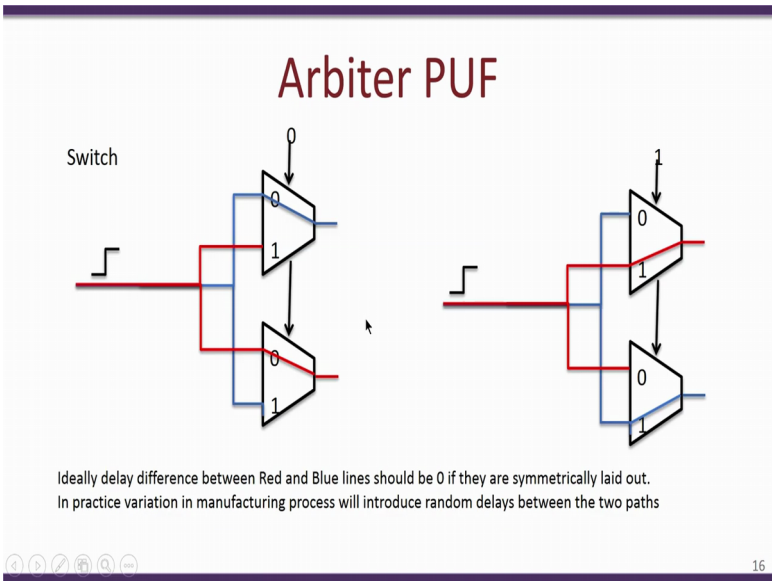
We see that in a large this waveform for this distribution of the Hamming distance is around having an average of 59.1, so ideally it should have been 64 but 59.1 is also a very good Hamming distance. Another check which we also require was the intra-chip variation, so as we mentioned in the previous lecture the intra-chip variation shows the reproducibility or the robustness of a PUF. So as we mentioned, we provide the same challenge to the device which is having the PUF, obtain the response and in a different condition maybe after a day or after a month or after a year, we send exactly the same device and obtain the response. So we could also have different aspects like we could send one challenge at a specific temperature say 20 degree and with a voltage of 1.2 volts and the other challenge to exactly the same device at which was heated to 120 degrees and having a voltage of 1.08 volts in this particular example.

(Refer Slide Time: 10:05)



So what is expected for a good PUF is that independent of these environmental conditions like temperature, time and input voltage, the response should be as close as possible for the same challenge. This particular graph shows the estimation of the intra-chip Hamming distance variation, so it tells you how different each response looks for the same challenge under these 2 conditions; 20 degrees 1.2 volts and when the challenge was given at 120 degrees 1.08 volts. So what you see is that in most cases it is most likely that the response does not changed with the different environmental conditions, so on an average there was 0.61 bits of variation out of the 128 bits of the PUF responses that were actually reported.

(Refer Slide Time: 10:47)



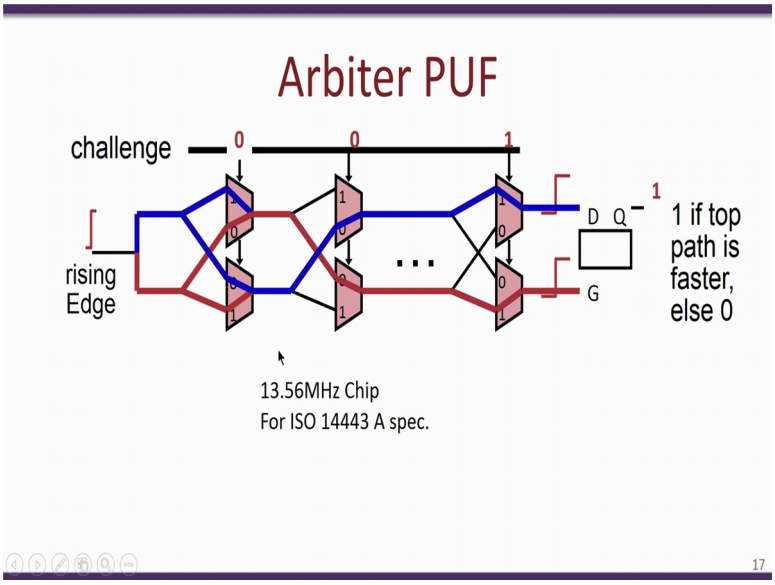
So we will now look at different kind of PUF so this is known as arbiter PUF and essentially this has certain different properties compared to the ring oscillator PUF that we have seen. So an arbiter PUF fundamentally is based on a switch, so it has 2 multiplexers which is used in the switch if you consider this particular figure, both are given the same input that is 0 over here and the same 0 is connected to this as well. And what each of these multiplexers does is that based on the input either 0 or 1, it will switch that corresponding input to the output.

So for example, if the multiplexers select line is 0 then this input at the multiplexers is sent to the output. On the other hand, if the select line of the multiplexer is set to 1 then the input present at the 2^{nd} input that is this input would be switched at the output. Now, in an arbiter switch two multiplexers are used, the same input is switched to both multiplexers present and both multiplexers have the same select line. There is a minor difference between the 2 multiplexers and what you see is that the input line is actually connected differently in each multiplexer for example over here, the blue line is connected to 0 in this multiplexer and therefore will be switched to the output when the select line is 0, while in this case the blue line is connected to 1.

Similarly the red line is connected to 1 in this case and 0 in this particular multiplexer and therefore when the select line is 0, it is a red line that can switch. So what you say is given this particular configuration of the switch, the output would be dependent on select line. If the select line is 0 then we have the blue line on top over here and the red line at the bottom. On the other hand, if the select line is set to 1 then it is the red line which goes on top and the blue line which is at the bottom. So essentially what this means is that depending on the select line we are either sending the blue line or the red line in each of the multiplexers output. So this primitive component called the arbiter switch is then used to build an arbiter PUF.

So now the fundamental feature about this particular switch that makes it interesting for the arbiter PUF is that these outputs whether the blue is sent on top or the red is sent on top depends on the characteristics of these PUFs, so this fundamental arbiter switch is used to build an arbiter PUF.

(Refer Slide Time: 13:50)

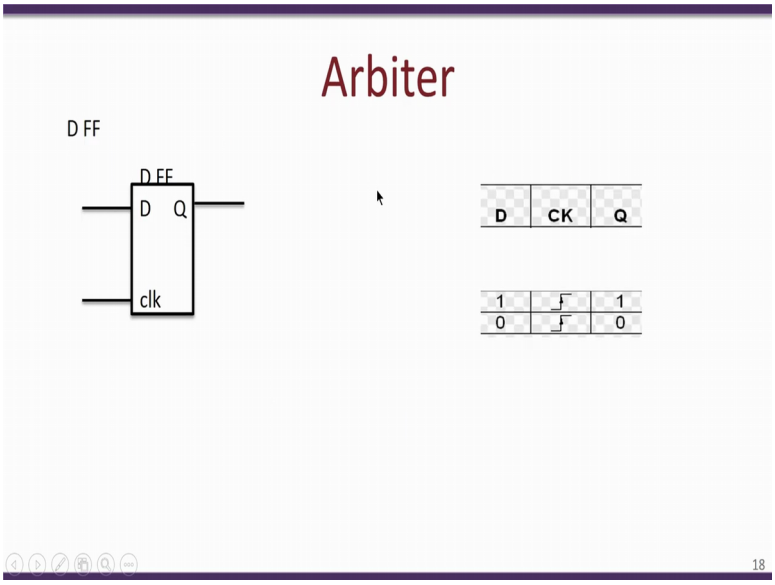


A typical arbiter PUF looks something like this, so we have actually multiple such switches present one after the other and a single input which is fed to both switches to each switch as shown in the previous slide. So we also have a challenge which is present, so we have challenges which is 0 or 1, and essentially what the challenge does is that it decides whether the blue line or the red line should be switched on top or bottom respectively. So over here for the example you see that we have poured 0 for this particular switch and therefore it switches the blue line to the bottom and the Red Line on top and so on. After a series of such switches we eventually have a flip-flop, this is a D flip-flop, this particular D flip-flop gives an output of 1 or 0.

So there are essentially 2 aspects that make this design interesting for use as a PUF. The 1st is the fact that these signals since they are propagating through all of these switches due to the nanoscale variations in the design of these switches, these 2 lines the red and the blue line would attain a different speed of transmission as we have seen in the case of ring oscillator as well the delays provided by each of these multiplexers is influenced by the manufacturing processes and the various intrinsic properties of the silicon and the process as well.

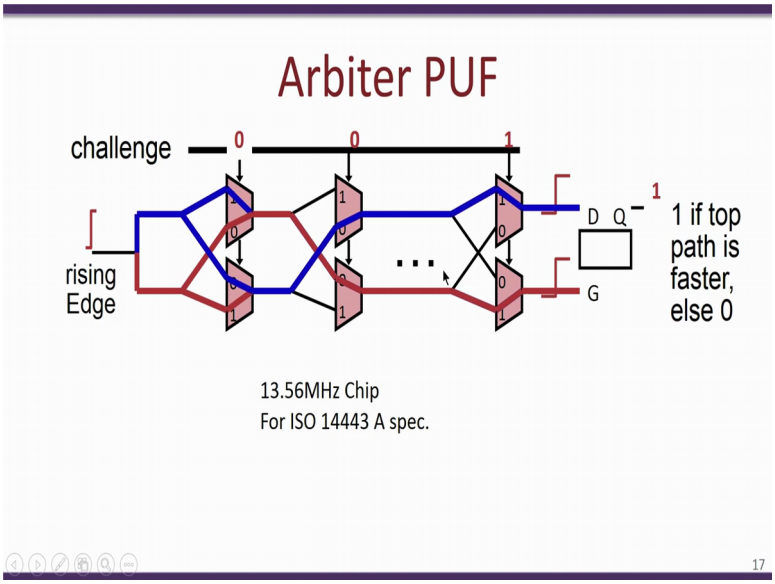
So as a result, after cascading through a number of such switches what we get is that one of these lines either the blue or the Red Line would reach the output faster than the other line. So we also now have a D flip-flop over here which measures which of these 2 lines is in fact faster and correspondingly gives output of either 0 or 1, so let us look in more details about how this D flip-flop actually is able to identify which of these 2 signals is indeed faster.

(Refer Slide Time: 15:59)



So let us consider a D flip-flop and what is done is that one of the lines let us say the blue line is connected to the D input and the Red Line is connected to the clock input and output is one bit which will give you either 0 or a 1. Now let us say that the blue line itches connected to that the input reaches 1st, as a result we would have something which looks like this so we have the D line reaching first then the clock signal reaching and as we know how a D flip-flop works when the clock transitions from 0 to 1, the input at D is then latched at the output. Since the signal at the D line has arrived first when the clock transitions from 0 to 1, the output would obtain a value of 1. On the other hand, if the clock in fact has arrived 1st when the clock transitions from 0 to 1, it would see that the D is still at the value of 0 and therefore the output Q would obtain a value of 0.

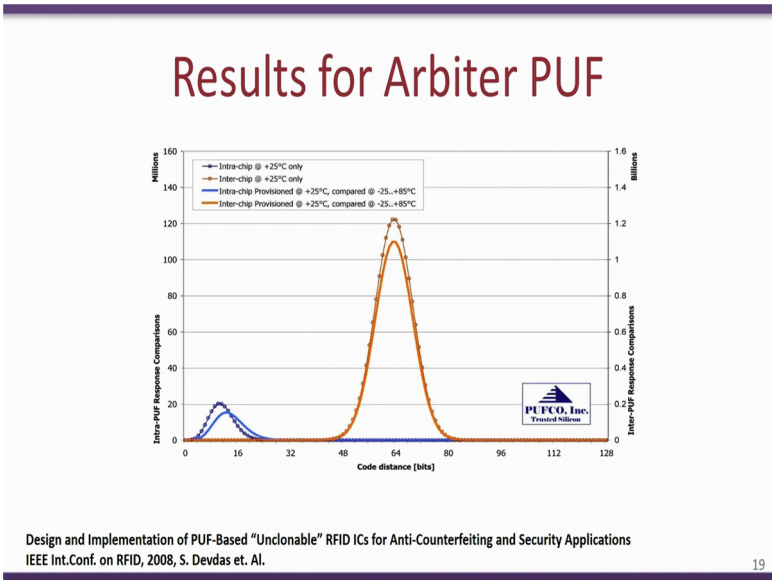
(Refer Slide Time: 17:14)



So thus we see over here that providing a rising edge 0 to 1 transition to these various switches would result in a differential path for these 2 lines and as a result, at the output at this particular point we have one path which is faster than the other. Now as we have seen the the D flip-flop which is configured in the way that we have discussed would be able to identify which of these 2 paths has arrived 1st and correspondingly we will be able to switch between 0 and 1. So the challenge in the arbiter PUF is that the select line of the multiplexers so for example over here considering that there are 3 switches, the challenge is 0, 0 and 1. So note that if we change the challenge, it essentially changes the path for the red and blue signals and as a result output may change.

Since the path is different, the choice between which of these 2 signals is faster may also be different, and therefore the output of the PUF may also change. So creating an arbiter PUF would require that we provide a challenge and correspondingly determine whether the output is 0 and 1. So the uniqueness is obtained because each of these paths for a given challenge would be different for each device and therefore on one device the same arbiter PUF for the same challenge would perhaps give an output of 0, while on other device will give output of 1. So this is essentially utilised for authentication and other purposes.

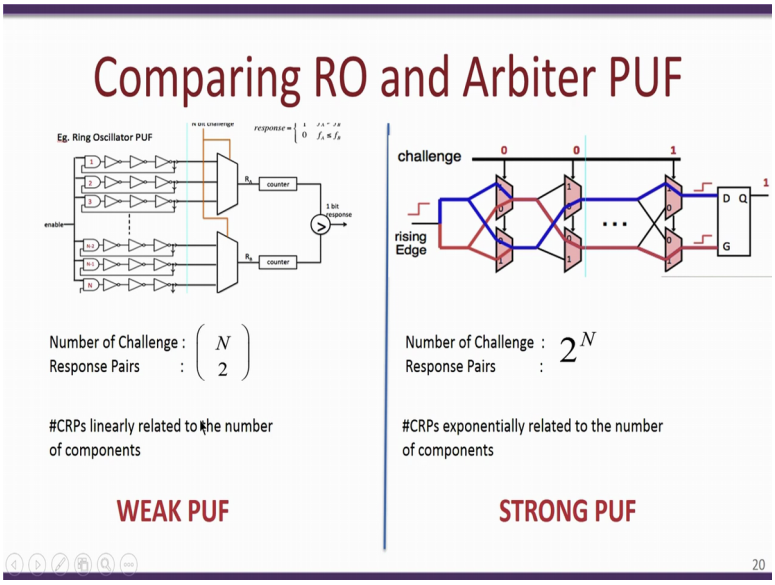
(Refer Slide Time: 18:57)



So this is a result for an arbiter PUF, it is obtained from this particular paper 2008 paper which shows both the inter and the intra chip distances at two temperatures; one is at 25 degrees and the other one is at 85 degrees ok. So the results are quite similar to that of ring oscillator PUF, so what we see is that the inter-chip distance for the same challenge is having an average which is around 64 for a 128 bit response. So this means that if I provide the same challenge to 2 different arbiter PUFs, the response would vary by roughly 64 bits. Similarly, intra-chip distance is less than 16 for a 128 bit response and input cases the temperature does not affect the response much.

So what this means is that if I provide the same challenge to the same device with different conditions like change of time, change of temperature or after a long time or so on, the response is very much similar.

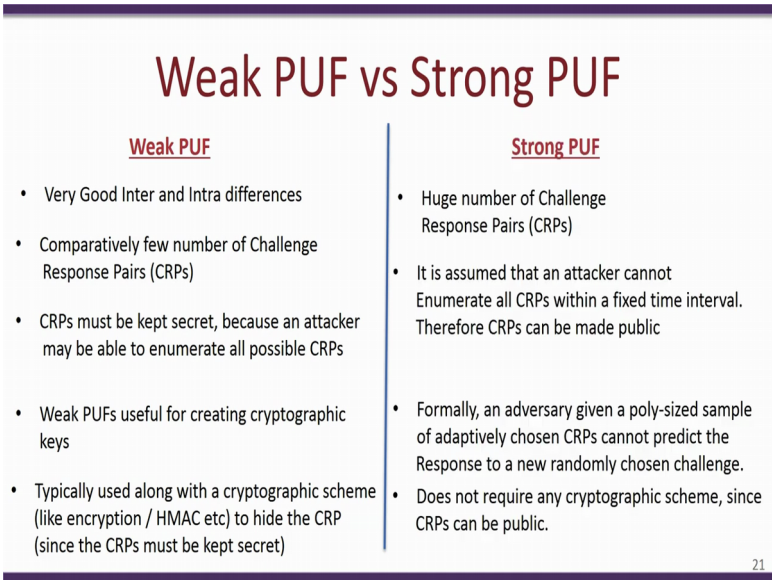
(Refer Slide Time: 20:13)



So what we have seen; 2 pubs, the ring oscillator PUF and arbiter PUF and what we will see now is we will try to compare 2 of them and see what are the characteristics of the two. So the 1st thing to note is that in ring oscillator PUF we have these 2 multiplexers here and N bit challenge, so they are essentially N bit challenge is choosing a pair of these 2 ring oscillators and therefore the number of challenges possible is N chose 2. On the other hand, the arbiter PUF we essentially set using the select line to choose a challenge and if there are N such switches arbiter switches which are present then the number of challenges possible is 2 power N. Therefore the number of possible challenges for this for the arbiter PUF is 2 power N.

So based on the number of challenges we categorize these PUFs as a weak PUF because it has a very small set of challenges, in fact the challenges linearly dependent on the number of ring oscillators or the strong PUF in case of arbiter because the number of challenges that are possible are exponentially related to the number of arbiter switches that are present.

(Refer Slide Time: 21:31)



So these are the properties of weak PUFs, 1st of all it has been noticed that weak PUFs have very good inter and intra differences. There are comparatively few challenge response patterns as we have seen in the ring oscillator and because of this reason the CRPs or the Challenge Response Pairs have to be kept secret and cannot be exposed to the attacker because the number of such CRPs are very less. So essentially weak PUFs are used for creating cryptographic keys, they are used together with other encryption schemes like they are typically used they are typically not used by itself but typically combined with other cryptographic schemes like encryption or HMAC and so on in order to hide the CRP.

So the problem with the weak PUF is that because the number of different challenges are limited, the attacker may be able to enumerate all of these challenges and for each device the attacker could be able to create a database of all challenge response pairs and therefore without even having the device the attacker would be able to provide a response from his database for any given challenge therefore, weak PUFs have limited applications. Now strong PUFs as we have seen has huge number of challenge response pairs, in fact we have seen that there is exponential number of challenge response pairs based on the number of arbiter switches present in the arbiter PUF.

And it is also assumed that because of this the attacker will not be able to capture all the challenge response pairs or attacker will not be able to build a database of all these challenge response pairs therefore, the used challenge response pairs can be made public and typically these strong PUF will not require cryptographic scheme because there is nothing secret which

needs to be stored. So with this civil stop this lecture, in the next lecture which is a third part of PUF, we will look at how these PUFs could be used to have an authentication without the need for any cryptography or secret keys. We will also see how there are several weaknesses which can occur due to this authentication scheme and also ways to actually mitigate these potential problems, thank you.