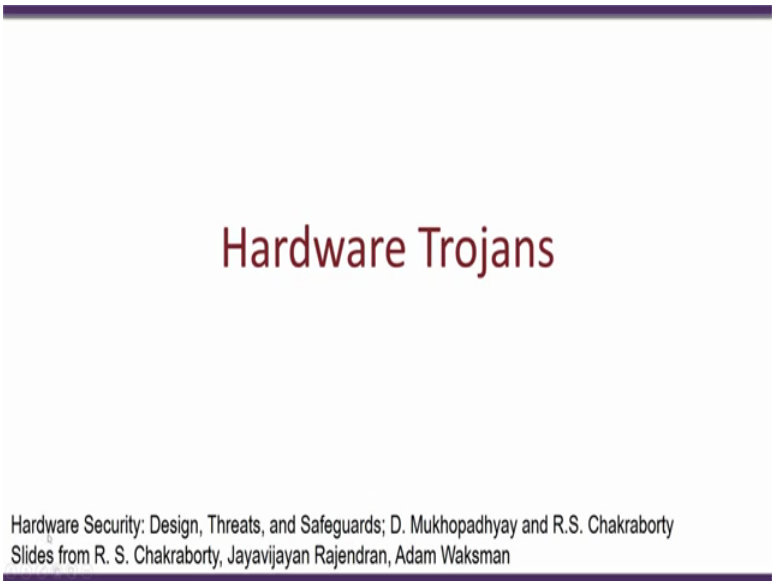
Information Security - 5 - Secure Systems Engineering
Professor Chester Rebeiro
Indian Institute of Technology, Madras
Hardware Trojans

Hello and welcome to this video lecture in the course for secure systems engineering. In the previous lectures in this particular course we had looked at malware and we had talked about a lot of techniques by which malware could affect the system and also how these malware could actually be prevented, but all of these discussions were related to malware which affects the software of the computer system.

(Refer Slide Time: 1:05)

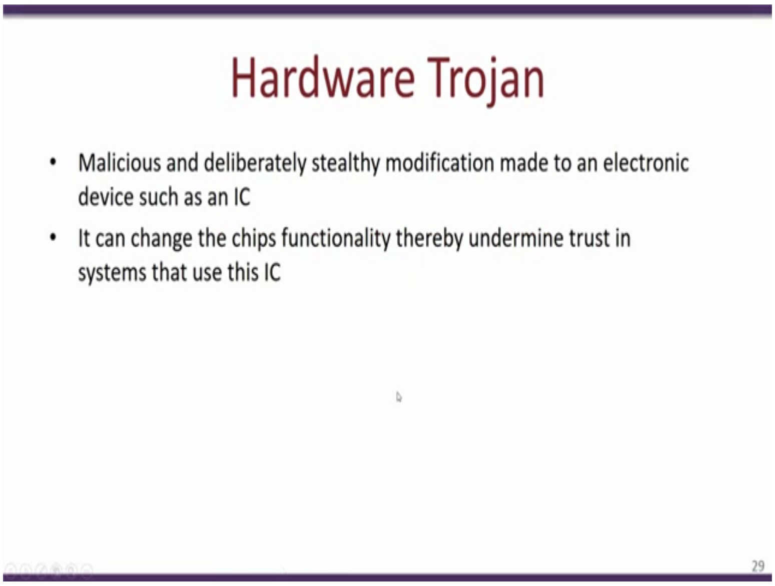


Hardware Trojans

Hardware Security: Design, Threats, and Safeguards; D. Mukhopadhyay and R.S. Chakraborty
Slides from R. S. Chakraborty, Jayavijayan Rajendran, Adam Waksman

In today's video lecture we would talk about malware which is present in the hardware. So these are popularly known as hardware Trojans and this would be the topic of discussion in this particular video. A lot of what I am going to be talking about is present in this book hardware security, design threats and safeguards by Professor Debdeep Mukhopadhyay and Rajat Subhra Chakravarthy from IIT Kharagpur. Also I have used slides from Doctor Adam Waksman and Jayavijayan Rajendran from Texas A & M.

(Refer Slide Time: 1:33)



Hardware Trojan

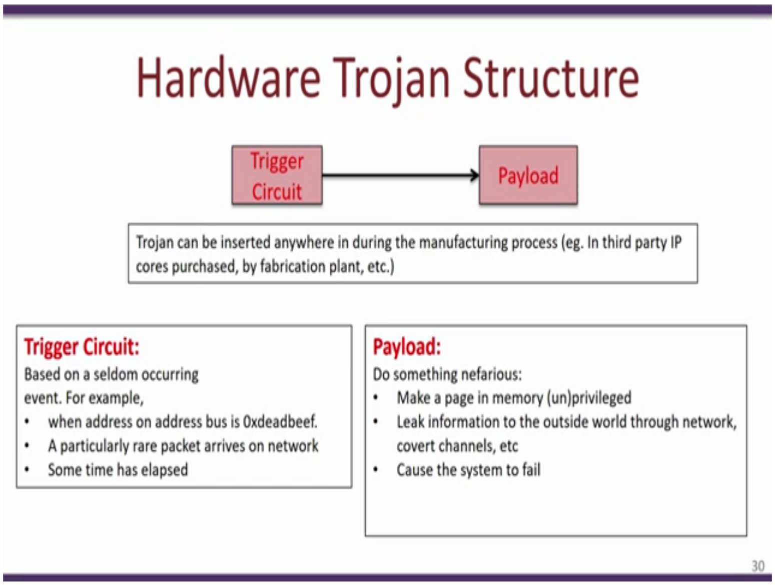
- Malicious and deliberately stealthy modification made to an electronic device such as an IC
- It can change the chips functionality thereby undermine trust in systems that use this IC

29

So essentially what our hardware Trojans? So hardware Trojans are essentially Trojans which are inserted right at the hardware, so you could for example have a processor and small additional hardware module present in the processor which is inserted at during the design time would act as a hardware Trojan. So these hardware Trojans are since they are present right at the hardware in the system, they cannot be detected by techniques such as the antivirus solutions which are present or the various other techniques that we discussed (all of the) like the confinement techniques, the OKWS, the trusted execution environment, all of these things will not work and will not be able to prevent Trojans which are inserted right at the hardware.

So these hardware Trojans are typically passive hardware components present in your processor or any other device and these hardware Trojans can potentially alter the functionality of the hardware device. So hardware Trojans have now become quite popular in the last decade or so and it is extremely difficult to actually find ways to detect the presence of hardware Trojans present in a particular device.

(Refer Slide Time: 3:06)



Hardware Trojan Structure

Trigger Circuit
Payload

Trojan can be inserted anywhere in during the manufacturing process (eg. In third party IP cores purchased, by fabrication plant, etc.)

Trigger Circuit:
Based on a seldom occurring event. For example,

- when address on address bus is 0xdeadbeef.
- A particularly rare packet arrives on network
- Some time has elapsed

Payload:
Do something nefarious:

- Make a page in memory (un)privileged
- Leak information to the outside world through network, covert channels, etc
- Cause the system to fail

30

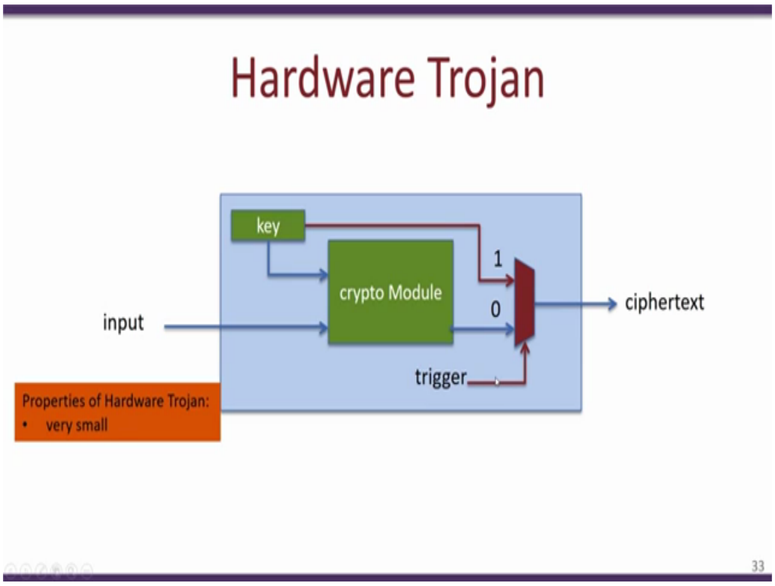
So what exactly constitutes a hardware Trojan? So very much like the Trojans present in software, a hardware Trojan comprises of two components, one is a trigger circuit and the other is a payload, the difference between a hardware Trojan compared to the Trojans which are present in the software is that the hardware Trojan is inserted in the device itself or right at the hardware, in most cases completely independent of all the software layers that are running on this particular processor.

These hardware Trojans are typically passive they may just comprise of a few set of gates which typically (are not) do not operate and do not actually modify the output of the device and therefore they are very difficult to detect. A typical hardware Trojan would wait for a specific trigger to occur and once that trigger comes then a specific payload gets executed. Now this payload essentially could modify the functionality of the device, the hardware or the trigger could be of several forms as we will see during these lectures.

For instance a trigger could be a specific input given by a user or which is obtained from a network or a specific address which comes on the address bus such as the case over here where the address bus has 0xdeadbeef, so these specific trigger inputs is what actually causes the hardware Trojan to wake up and then execute the payload like this. So the payload very similar to that of the software Trojans would do something malicious.

For example it could leak some secret information to the outside world say through the network interface or through cover channels, it could make a page in memory which is say in the kernel space this page could be modified to become accessible from the user space and therefore a user process would be able to actually read protected data present in the system or if you have something like say a trusted execution environment a hardware Trojan (would be) could possibly disable this entire trusted execution or compromised the entire trusted execution such that a normal untrusted application would be able to observe or modify the execution of the trusted environment or alternatively the payload could do anything maliciously for example it could also cause the entire system to fail.

(Refer Slide Time: 6:06)



Hardware Trojan

Properties of Hardware Trojan:

- very small

33

So let us take a simple example of a hardware Trojan. So the example we would take is a crypto device, so you can consider this as integrated circuit that is an IC and within this IC we have a crypto module, a secret key which is stored in within this particular IC. So what this device does is that it takes an input, uses the key and feeds these two into the crypto module, does an encryption and then produces a ciphertext.

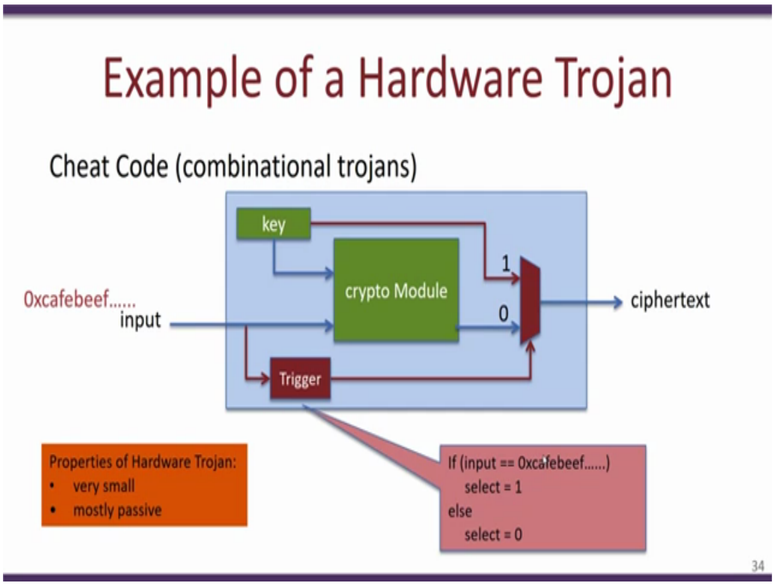
So what we see is that as long as the key is kept secret and well concealed within the device the cipher text does not contain any information about the corresponding input. Now what could happen is during the design or manufacture of this particular device, developer could insert a hardware Trojan. So we will just take a small example of how a specific hardware Trojan can be inserted and what the functionality of this hardware Trojan could be.

So we see that a hardware Trojan like this can be inserted which takes a trigger which acts as a select line to a multiplexer and then we have a MUX. So what the multiplexer does is that it either passes on the cipher text which is computed by the crypto module whenever the trigger has a value of 0 or alternatively if the trigger has a value of 1 then the key gets transmitted to the output. Thus, we see that the secret data stored secretly or concealed in the device is visible at the output.

So in this particular example what we see is that we have this trigger and the payload over here and this payload is actually causing sensitive or secret data to be exposed to outside the device. Now we look at two properties of a typical hardware Trojan, the first thing is that the hardware Trojan is extremely small, notice that you could have a very large crypto module and the device could be extremely complex but what we notice is that the additional modifications due to the hardware Trojan is extremely small all that is required is just a few multiplexers in this particular case and a trigger signal.

The objective of making the hardware Trojan extremely small is the fact that it is very difficult to actually detect such a hardware Trojan while for example viewing the code or just looking at the device it is very difficult given the size and the number of transistors and gates present in the device, this additional hardware present due to the hardware Trojan may easily go unnoticeable.

(Refer Slide Time: 9:18)



Example of a Hardware Trojan

Cheat Code (combinational trojans)

Properties of Hardware Trojan:
- very small
- mostly passive

```
If (input == 0xcafebeef......)
    select = 1
else
    select = 0
```

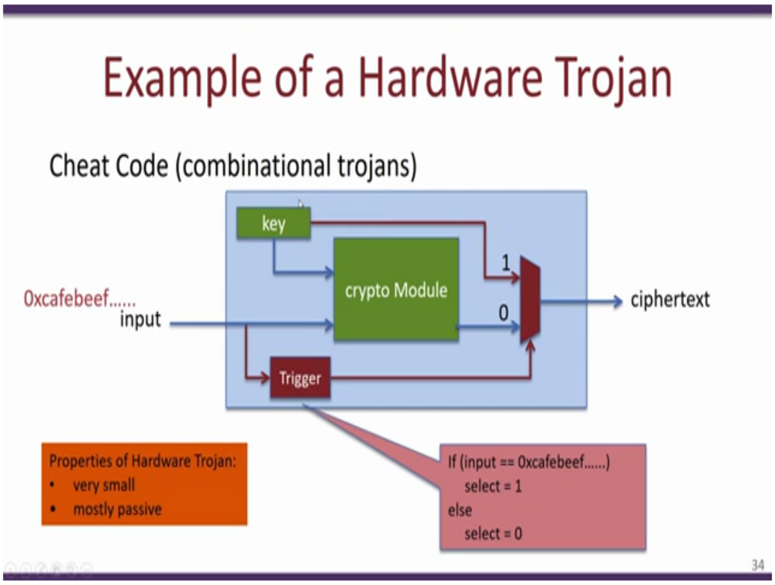
Second and an important property of a hardware Trojan is that the hardware Trojan is mostly passive. So for example over here we have shown a combinational Trojan this combinational Trojan uses a cheat code to trigger. So in this particular trigger what happens is that the input is tapped and goes into this trigger circuit and this particular trigger waits for precisely one specific input to appear, when this input appears it provides an output of 1.

So let us say for example the input line is of 128 bits and therefore the possible number of inputs is 2 power 128. Now since the trigger is waiting precisely for one cheat code in this case 0xcafebeef to actually activate the payload. Therefore, in most of these cases this hardware Trojan is going to be passive, it is only when this pacific input appears to the device would the trigger be activated and the payload be used to leak out the secret key.

Now the entire design methodology of a hardware Trojan is that this cheat code cafe beef is only known to the attacker but not to any other person who is designing or using this particular IC. Thus, during testing or during various tests or during normal operation of this device the probability that a user actually finds an input cafe beef is extremely small. For example in our case where we assume that the input is of 128 bits, the probability that a valid user or a designer during the testing of this device actually finds such a trigger cheat code is 1 by 2 to power of 128, which is an extremely small probability and therefore highly unlikely that such trigger cheat codes would not appear during the regular testing or usage of the device.

However, since the attacker knows about what the cheat code is, whenever required the attacker could actually feed this pacific cheat code to the input of the device and then activate the hardware Trojan because by the cheat code would force the select line to 1 and thereby passing on the secret key to the output of the device. So you may have also noticed that this trigger where which waits for a specific input to appear is just one example of a trigger there may be many other ways by which such triggers can be actually designed.

(Refer Slide Time: 12:30)



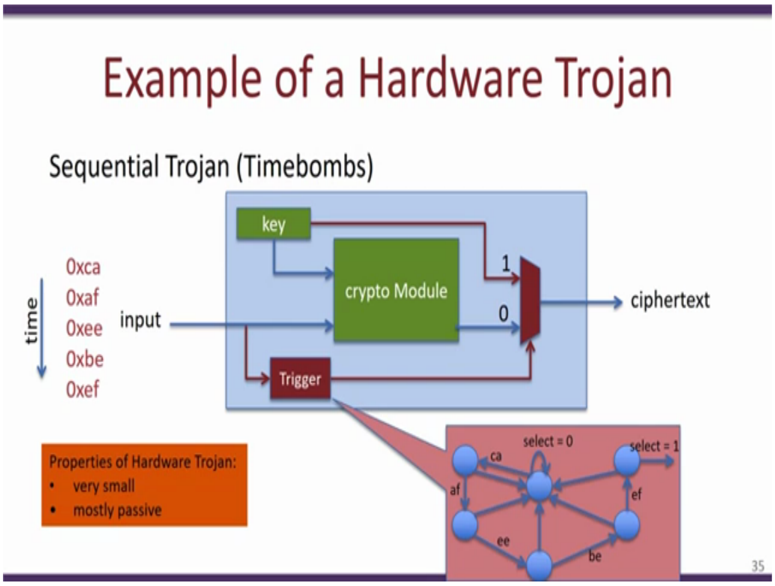
Example of a Hardware Trojan

Cheat Code (combinational trojans)

Properties of Hardware Trojan:
- very small
- mostly passive

```
If (input == 0xcafebeef......)
   select = 1
else
   select = 0
```

34



Example of a Hardware Trojan

Sequential Trojan (Timebombs)

Properties of Hardware Trojan:

- very small
- mostly passive

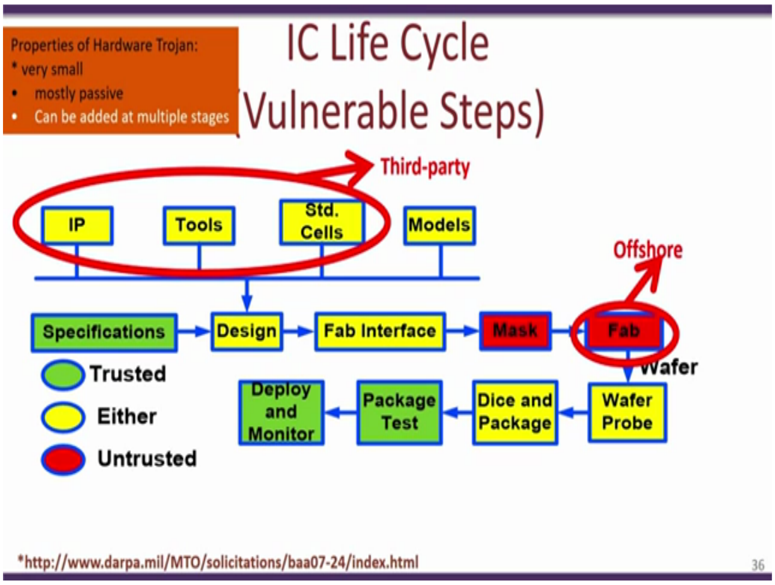
Another very popular way of designing the triggers which is even more difficult to detect is something known as time bombs or sequential Trojans. So what we have over here is a state machine, in this sequential Trojan the trigger is waiting for a specific sequence of inputs to appear. Now this sequence for example ca is an input which is followed by af and after some time there is an ee, then after some time there is a be and then an ef would finally activate the Trojan and force the secret key to be visible at the output.

So internally this trigger would have a state machine like this where typically the select line would be having a value of 0 and based on the various input sequence which would trigger this particular Trojan the state machine would move through several states and eventually reach a state where the select line becomes 1 activating the Trojan. So you see that this sequential Trojan may be more difficult to actually detect compared to the cheat code or the combinational trigger that we discussed.

The reason being that these sequential values could either appear consequently or could be spread over several clock cycles and therefore the probability that that such a sequential trigger is actually detected is much more smaller. Now these two properties of a hardware Trojan that being very small and also mostly passive makes hardware Trojans extremely difficult to detect.

The detection of hardware Trojan is a very hot topic of research especially in the last decade or so and there have been several interesting papers and techniques that have been discussed and presented in order to detect such hardware Trojans. In this video lecture and perhaps the next as well we would discuss two such techniques, one is known as fanci and the other one is a technique by which we could design a circuit or rather design a particular hardware unit where inserting a Trojan would be considerably more difficult.

(Refer Slide Time: 15:01)



Properties of Hardware Trojan:
- very small
- mostly passive
- Can be added at multiple stages

IC Life Cycle (Vulnerable Steps)

Third-party

IP

Tools

Std. Cells

Models

Offshore

Specifications

Design

Fab Interface

Mask

Fab

Wafer

Wafer Probe

Dice and Package

Package Test

Deploy and Monitor

🟢 Trusted

🟡 Either

🔴 Untrusted

*http://www.darpa.mil/MTO/solicitations/baa07-24/index.html

36

Another aspect which complicates the detection of a hardware Trojan is the fact that a Trojan can be inserted in the hardware at multiple places during the design cycle. So what we see in this particular slide is a typical way IC is designed and manufactured. So typically if a company wants to actually design and manufacture a new IC it would start off with the specifications for that IC.

So for example you could say you could think of a new communication controller or let us say a new cryptographic accelerator or so on. So this particular company would start off with actually writing the specifications for that particular hardware IC. So then once the specifications are written it goes to a design phase and during this design phase there will be a lot of coders that are used.

In addition to these codes a lot of third party IPs are also integrated into the design, a lot of models and standard cell libraries are used and all of these are integrated using several EDA tools. Now then we have a process of fabrication it goes to the fabrication unit there is a masking, then there is a dice and packaging, then there is a packaged test and finally a deployment and monitor.

So what we see is that a typical IC design and manufacture has several of these steps the company that is designing and manufacturing this specific IC is not the only one which is involved and there are a lot of third party tools and units which are also involved during the design and manufacture of a particular IC. So typically the IP cores are bought from a third party, the tools used for a designing of that particular IC is also brought from third parties as well as these standard cells.

Similarly most companies use an offshore fabrication unit to actually design and develop these ICs. Now out of all of these various steps and organizations involved during the design and manufacture of an IC the only steps which are completely trusted is this step where the specifications are written the package testing that is done after the chip is fabricated and the deployment and monitoring of the chip.

So besides these three steps in the entire process Trojans can be inserted in any of the other steps. For example a programmer who is an adversary can insert a Trojan during the design phase or IP cores that are brought from third parties may also have in them certain Trojans which are not easily detectable. Similarly tools and libraries like standard cells may force Trojans to be inserted into the device.

In addition to the design phase once the design is actually sent out for manufacture the offshore manufacturing companies may also insert Trojans in the device. Thus, due to the fact that hardware Trojans can be actually inserted in many different stages during the development and manufacture of the IC it becomes extensively difficult to detect such Trojans or for that matter actually it is very expensive to actually design ICs where Trojans are not present.

So in the next video lecture we will take one example of a tool which is known as fanci and we will show how fanci is able to potentially detect locations within a design of hardware where hardware Trojans may potentially be inserted, thank you.