

Advanced IOT Applications
Dr. T V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

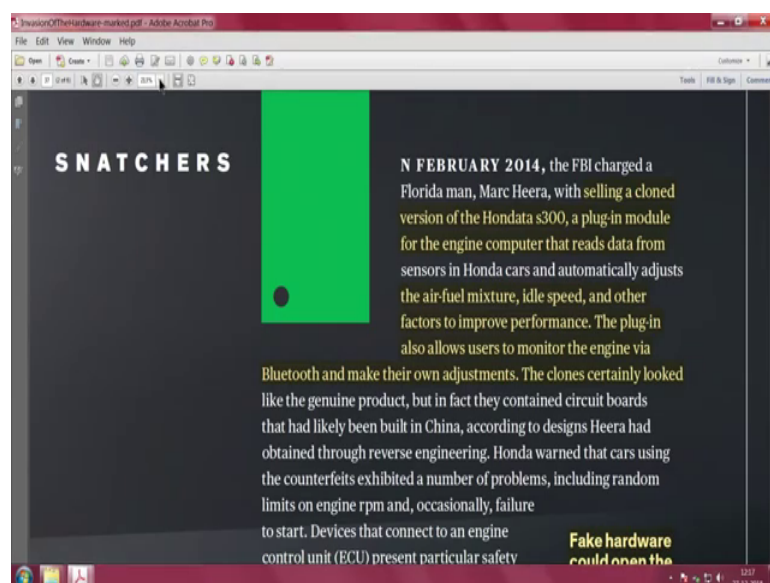
Lecture - 29
Device Security Part – 2

So, you can see if you have to understand this area of counter feiting, we definitely have a module on the automotive part in this right. So, we have a large module on automotive this one IOT. And, we did discuss about air, fuel, mixture, ratio and all that which essentially improves the performance of the engine fuel efficiency and so on and so forth. What happens if you have a counterfeit of that ok?

Suppose you have a throttle body which is a counterfeit of the original one and or throttle body along with the ECU. Because, ECU is the one that ultimately you know giving command to the throttle body to do a few things that is opening the no energizing the motor to open the butterfly valve and all that.

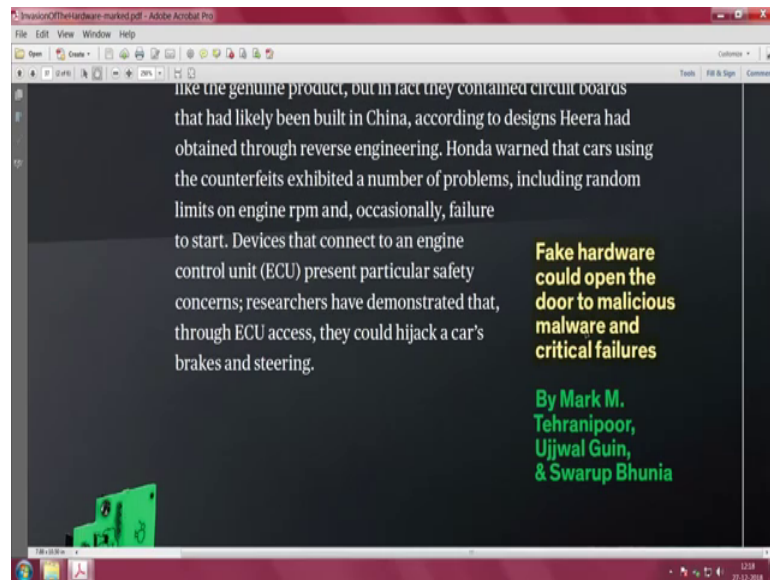
So, ECU if I clone, I will have my own system and I can you know sort of essentially take control of the complete car right. So, such vulnerabilities have been identified way back in 2014 on very famous you know in well branded vehicles.

(Refer Slide Time: 01:53)



I will point you to an article here which appeared in the triple e spectrum, you can see that if I can perhaps even enhance it new more it may be useful. So, let me see I will go 400 percent no that was too much. So, maybe 200 percent or maybe 250 percent let me try. So, that yes I think this is good right.

(Refer Slide Time: 02:27)



So, you can download this article anytime which appeared. So, so here is the issue if you change the air fuel mixture right; this is on Honda car this happened in February 2014. And, they were selling a cloned version of Honda S 300 this is a plugin module for engine computer essentially and ECU, that reads data from sensors and it is modifying air fuel mixture idling speed other factors all that it can have access to moment you have a cloned hardware right

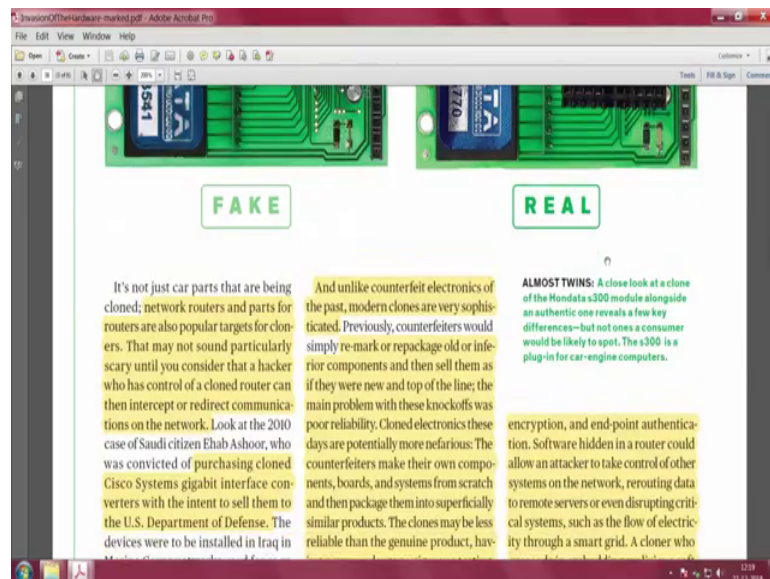
So, this is very close to a genuine product and so, can be leading to very dangerous situations. And, it could open door to malicious hardware and critical failures of these complete system.

(Refer Slide Time: 03:23)



So, one may have to you put it to a little lower. So, that you get a full picture.

(Refer Slide Time: 03:30)



So, you see here is picture of one is fake the other is real hardly you can make out if I put it on screen here, you do not know which is fake and which is real right. So, quite state forward is just not only about automotive parts that one have to be worried about, it is also about the IOT devices the data that they generate, and the network to which they inject they basically communicate to networks. And, the there are network routers which

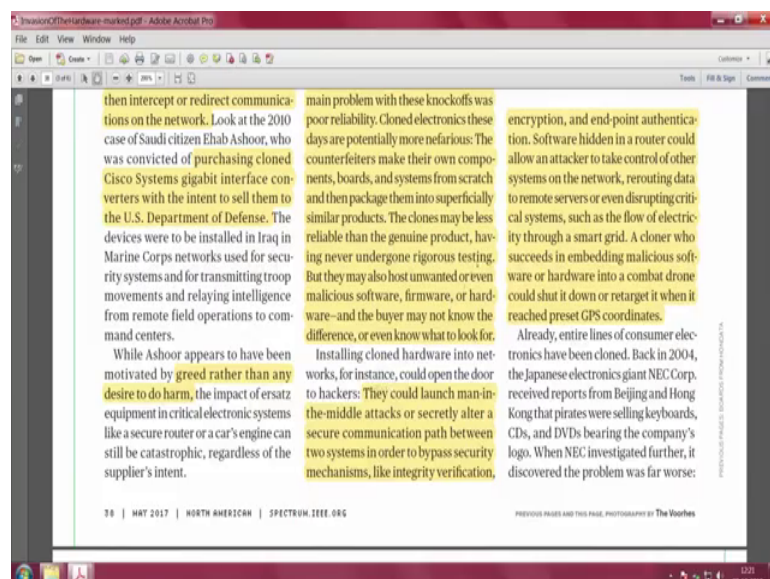
are there as you know that when we talk about IOT devices they all have end IPV 6 addresses right.

And, they are sensing something and are perhaps they have some actuation also with their connected to and they are giving out data. And, then there are these routers of different which are essentially connecting several subnets and these routers are critical to get the data to the right aggregating point where commands have to be sure.

So, if the router gets compromised again data from the sensors is going in a different direction is going to is being directed to different spot. So, it is also about network routers, which essentially are can be popular targets for cloning. And, see the point really is that if you want to clone, it should make sense to the person who is trying to clone. If, we invest a very small amount of time and has a huge impact on cloning it, it is worth cloning. But, if you are to do a lot of sophistication to clone it I think most people will just give it up right. You want to do all that sophistication in cloning you know reading layer by layer of chip and all that, only if it is going to give you billions and billions of dollars much more than your high investments.

So, that is the point return on investment in time in money is what cloners also will examine very carefully and only then they will trend. So, it is not for fun or for anything that people want to do, but really they want to disrupt the proper working of the system because there is some gain in the in the in the process. So, this is another thing.

(Refer Slide Time: 06:03)



So, you really do not know whether you are buying a cloned hardware or an unclone or a genuine one. This is also with respect to very famous networking companies like Cisco systems where gigabit converters, cloned once we were trying to be sold to defense departments. So, it is more sometimes there are also people who may want to do it for greed rather than to do any harm just because they can make a fast buck by selling of a cloned hardware at, which they buy at some throw way price. They are basically middle people they are not the ones who actually clone they are people who buy and they want to sell this knowing fully well that is a cloned hardware that is point.

So, those people are only after it is only grid. So, the whole echo system of cloning and counterfeiting is quite large different players doing different things. So, this particular cisco systems gigabit interface, which was sold to us department of defense is was more from greed rather than to do any harm particularly by this person.

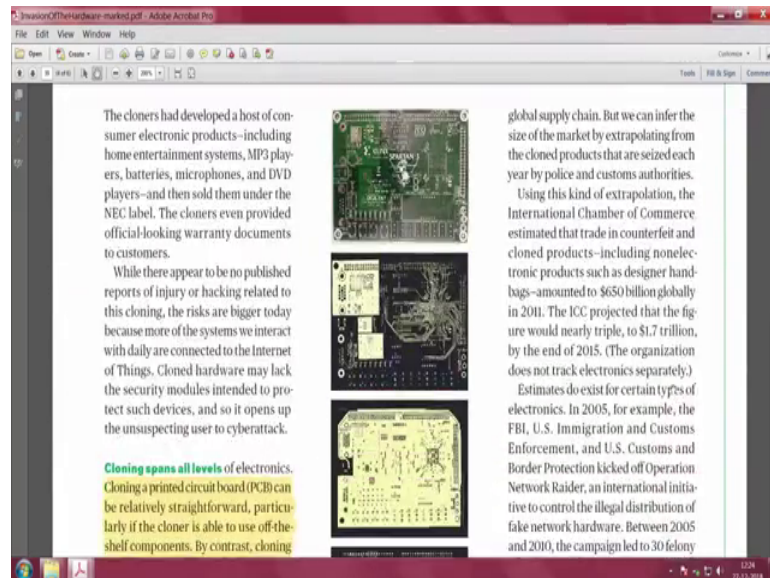
So, unlike counterfeit electronics of the past is the (Refer Time: 07:13) today's attacks are much more much more sophisticated. So, this is really the point of this further things that you are talk about. Those include you know you are essentially looking at just not going beyond you know re marking and repackaging of inferior components, you go into to manufacture of their own components, you manufacture components, you manufacturer boards, you manufacture systems from scratch, and then package them into superficially similar product.

So, it is just not simple game of just making something, but in fact, going into much more sophistication. So, this clone may be less reliable than the genuine product and they under go some rigorous because they do not undergo any rigorous testing, but they are all, but they may also host unwanted or even malicious software firmware and all that.

So, buyer who knows nothing about what is actually happening simply takes it and does not know, whether it is really a clone product or something else. Now, once you install the cloned hardware you can do a lot of interesting things right. You can do man in the middle attacks and you can secretly establish you know secure communication between 2 systems, bypass security mechanism, integrity verification, all of that you can do and you can you know you can bypass all of them. And so, the software that is hidden in a router could allow any attacker to take control of other systems on the network re rooting as I mentioned.

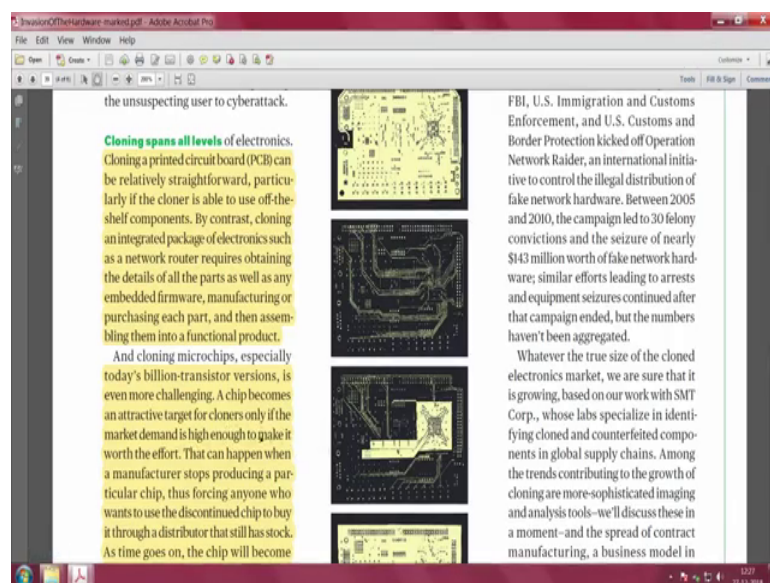
So, many things once you get access to the ones you have your own hardware installed there which looks exactly like the original one and sold it off to an unsuspecting customer several things happen. So, a cloner who succeeds in embedding malicious software or hardware into a combat drone for instance can shut down or re-target it when it reaches its GPS locations.

(Refer Slide Time: 09:25)



All kinds of issues can happen ok.

(Refer Slide Time: 09:27)



Cloning is not just limited to very low levels; it can be at you can do it at different levels. This article further says you can do cloning a PCB which is pretty straight forward and essentially it is an integrated package of electronics and obtaining details of embedded firmware manufacturing and all that. So, it essentially means that while if you look at lot of things that are available in open source right. You have schematics in open source you have Gerber files, of PCB is in open source you can just take them by your own components put them together and hardware cloning is pretty straight forward particularly open source has given you that ability to replicate hardware. Even very sophisticated systems are out there in open source. For example, if you want a open source radio platform, that schematic is available components are available. And, they give you some location where you can get the boot software and you can install it and make your own system and all that.

All in good faith all in time terms of trying to support open frees of open source software and getting things free cheap and to for the betterment of humanity. But, there is just 1 or 2 percent even lesser percent of people who actually take this into account and then they turn it around and say, let me do something bad by all that is available freely out on the internet this is really the issue here.

So, you do not only look at PCB, but you can also go to the chip level ok. So, you can also clone chip suppose there is an IP on a chip and chip that does lot of wonders ah. For instance in today's world I can imagine that if you are designing a for an automotive application you are designing a radar chip radar on chip. So, the whole chip is very and that is working at a very high frequency right. So, you are looking at the ism band in the millimeter wave frequencies. Particularly automotive you talk about I think it is in the 60 to 77 mega gigahertz range.

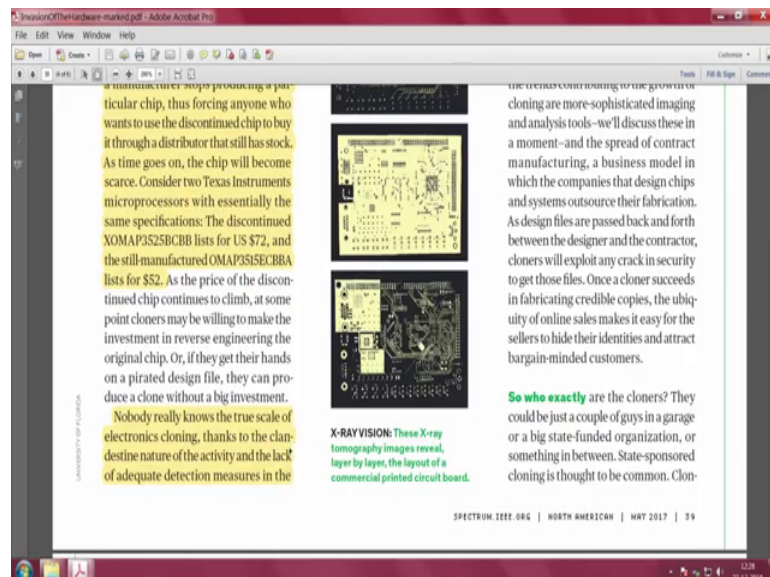
So, that frequency at that frequency you have a nice electronic chip and if you are able to clone it, you can make billions and billions, because every car today every vehicle today will have radar onboard right, because you are looking at safety measures and so on. So, it is worth it if you want to try looking at those high value and billions of components which are made.

So, people by want to look at those issues. So, today you are looking a billion transistor versions chip become attractive for cloner's right. And, that can happen when a

manufacturer stop producing a particular chip and thus forcing everyone who wants to use the discount chip by through some distribute some lower version of the chip is available. And, then there is a case here which says about as time goes on the chip will becomes cares ok. Because, the manufacturer stop producing it and there is a discounted chip which is available at 72 dolar and still manufactured one is at 52 dollars ok.

So, essentially as the price of the discounted chip continuous to client this, the cloners may be willing to make the investment in reverse engineering and then start producing that because it is in high demand. So, they may want to continue producing it.

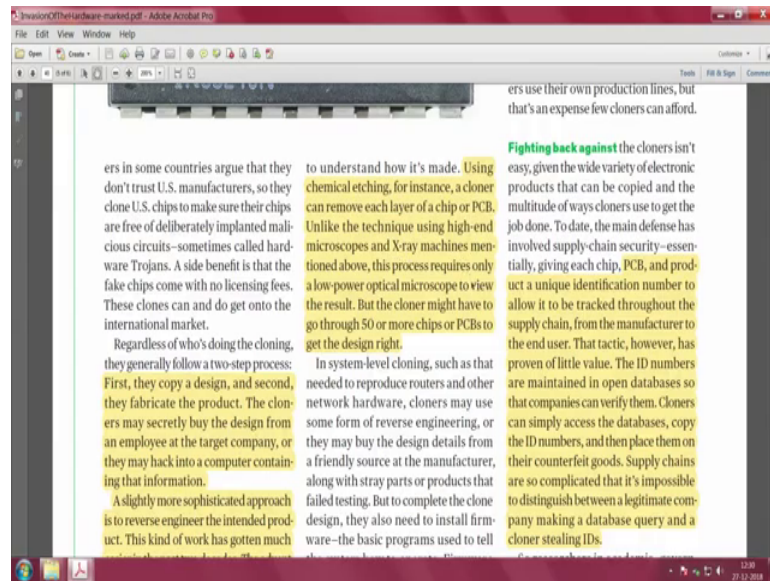
(Refer Slide Time: 13:20)



So, all kinds of issues can happen from a market perspective. Nobody really knows the true scale of electronics cloning, because these are all done by less than 1 percent arrived not when 1 percent. I think it is a very small 0.1 0.2 percent of people who really look at the you know this kind of clandestine nature of activity.

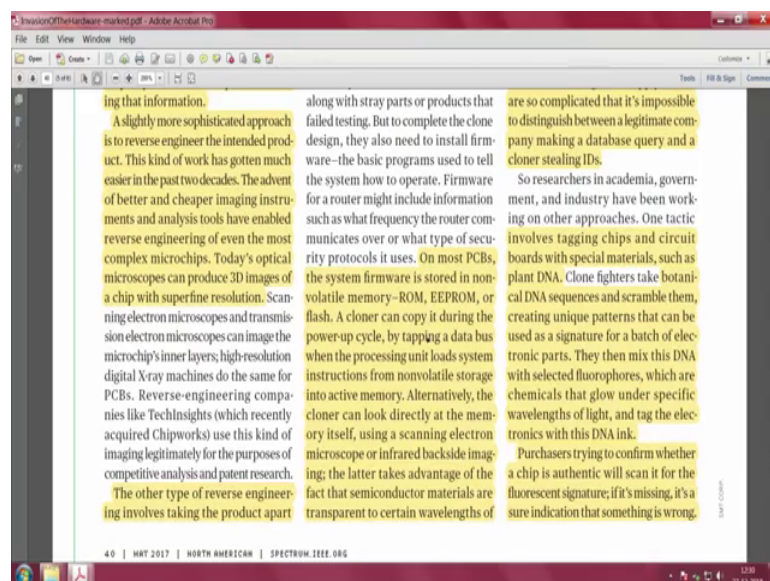
So, that is what this paper is actually saying, now let us move soon. So, I am sure you are getting a feel of where the cloners are coming from.

(Refer Slide Time: 14:05)



And you will also see that they do all kinds of things the copy a design and fabricate the product that is simple to do, they may do sophisticated things like they may do imaging of the chip, they do they use imaging instruments and analysis tools and so on. They use optical microscopes and they can produce 3 D images and they may do chemical etching which by which they can do layer by layer and so on. So, you can use techniques like high and microscopes and X-ray machines and go and actually get to you know get a picture of the chip that is of interest alright.

(Refer Slide Time: 14:50)



So, as I mentioned chemical etching and all that is another possibility. So, the cloner can copy it during so on most so, let me just read the sentence on most PCBs the system software is stored in non-volatile memory. You can the cloner can simply copy the software by tapping into the data bus and when the processing unit load system instructions from the non-volatile storage interactive memory. Or the cloner can directly look into the memories see these very sophisticated techniques, you can read off the memory locations, and you can get to the key ok. You can use microscope or infrared backside imaging the large the infra the infrared backside imaging has an advantage that semiconductor materials are transparent to certain wavelengths right.

(Refer Slide Time: 15:49)



And, under powerful microscope the closer can actually see the stored ones and zeros and reconstruct the code. So, you can see that. So, much of sophistication is applied if it really makes financial sense to the cloner to get to that point, how do you fight back? ok. So, that is really the issue. So, one thing that you can do is you know it is not easy that is for sure ah.

So, you can say fighting back against the cloners is not easy given that the wide variety of electronic products that are there. The main defense has been supply chain security ok. Essentially you do something giving each chip PCB product unique some ID you put and you start tracking that ID, but the problem with this id based system is and see all this things that is each a PCB or product identification ID and all

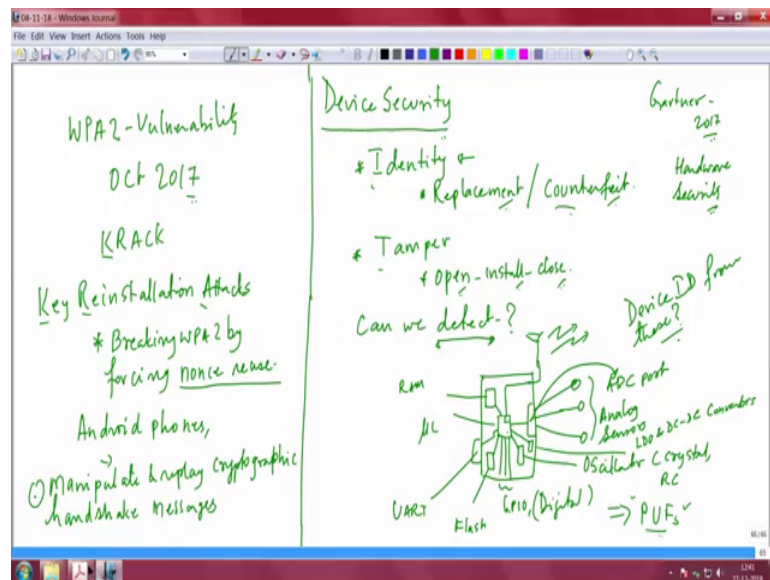
that is part of the supply chain and it is in some database right, it is stored in some database cloner has very easy access to the database.

So, you can simply go to the database and pull out this information and start using it on whatever he has cloned. So, cloners can simply access databases copy ID numbers and then place them on their counterfeit goods. So, it is very simple this is not this is what is currently happening, it is not going to work really ok.

Now, here is a nice solution to this problem pretty expensive solution though, but I would say it is a novel taught that how do you get all get out of this counterfeit issue. So, you can it is just an introduction because we do not have, I do not have any lab setups to show you, what the solution is, but am also excited to say that this is what you should be doing if you want to move forward with the 2 problems which I mention to you related to device hardware device part which essentially is pans the device identification I mention to you.

So, if we open that back so that.

(Refer Slide Time: 18:09)

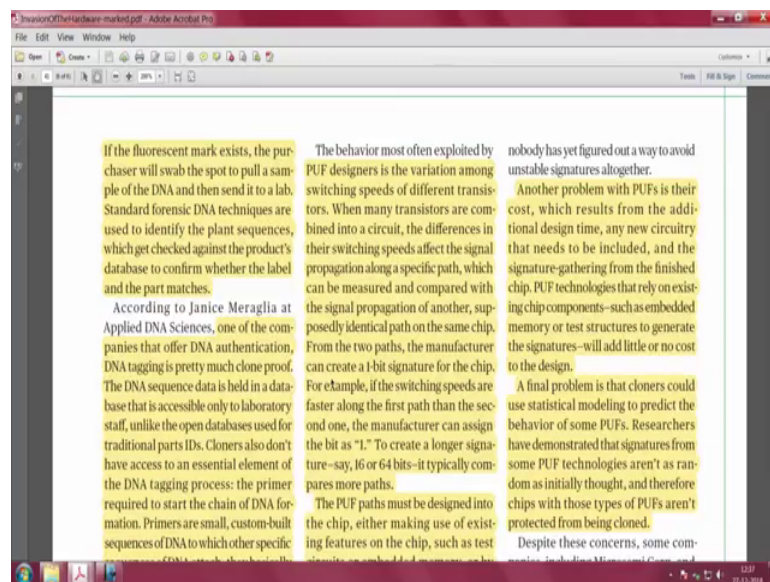


So, the 2 problems I mention to you which are of interest to us is related to identity and of course, tamper open install close as we said. And, you look at all these things get back nice possible what can you do if you have such a very rich amount of hardware sitting on a small embedded device. So, let us move on to see what you can do? What this article is

saying is that, you can tag chips and circuit boards with special materials. So, materials engineering has become a taken fore-front essentially you look at plant DNA.

So, I would urge you look at plant DNA and then you start fighting the cloners by take clone fighters take botanical DNA sequences and scramble them, creating a unique pattern that can be used as a signature for a batch of electronic parts. They then mix this DNA with selected fluorophores, which are chemicals that glow under specific wavelengths of light. And, tag the electronics with the with this DNA ink. Now, purchases trying to confirm whether the chip is authentic will scan it for the fluorescent signature, if it is missing it is a sure indication that something is wrong.

(Refer Slide Time: 19:39)



If the fluorescent mark exists the purchaser that will swab this part to pull out pull a sample of the DNA. And, then send it to a lab standard forensic DNA techniques are used to identify the plant sequences which is checked which get checked again the products database to confirm whether the label and part.

So, important so, important that you got to do it at multiple levels in order to know that yes this is what I got and this is indeed the genuine part that I have. Imagine important parts ok, landing gear for instance of a plane. If, it gets you get a cloned landing gear you are you are in serious trouble right. The aircraft manufacturers is in serious trouble in the; if he uses these kind most often they are manufactured locally they are manufactured within the company, but critical parts may also sometimes be manufactured not

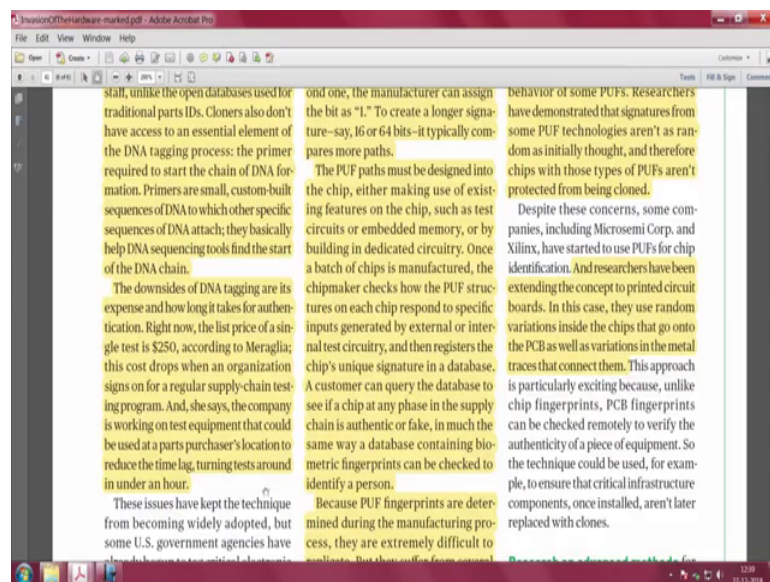
necessarily in the same factory. There could be an integration place where there is an assembly of all the parts. And, the part itself could be manufactured by the company in a different location.

During transit something can happen ok, when it is brought to the place of assembly something can get swapped. So, these issues will have to be born in mind and it is possible that you will have to somehow solve this.

So, this DNA methods are appearing to be very very attractive. So, let us move on because it is indeed something that I want you to urge and look up as we go along to solve this major problem.

So, in summary if the fluorescent mark exists the purchaser will swab the spot to pull a sample of the DNA. And, then send it to a lab then you do some DNA techniques forensic DNA techniques, you identify the plant sequences you can check it against the products database. And confirm whether the label and part matches this is how you would exactly say that it is genuine.

(Refer Slide Time: 21:55)



One of the companies that it is not far off right, so, this is already there in practice. DNA authentication DNA tagging is pretty much clone proof that is what they say because the DNA sequence data is held in a database, that is accessible only to laboratory staff unlike the open database is used for traditional parts ids. Still I would say this is a you know

sort of a risky affair, if this database which is held in the lab get compromised then you are again in little bit of a trouble., But, what can you do this the best that you can think of today, cloners also do not have access to essential element of DNA tagging process, the primer required to start the chain of DNA formations.

So, primers are small custom built sequences DNA to which other specific sequence of DNA attach they basically help DNA sequencing tools find the start of the DNA chain. So, this tagging process is not available with cloners. So, therefore, cloners to do something with DNA tagging is not going to be easy, but there are problems with DNA tagging as well it is very expensive, it is something around the range of 200 and 50 dollars, but they expect that it will come down and see that the latest equipment that could at a parts purchases location can be there right there.

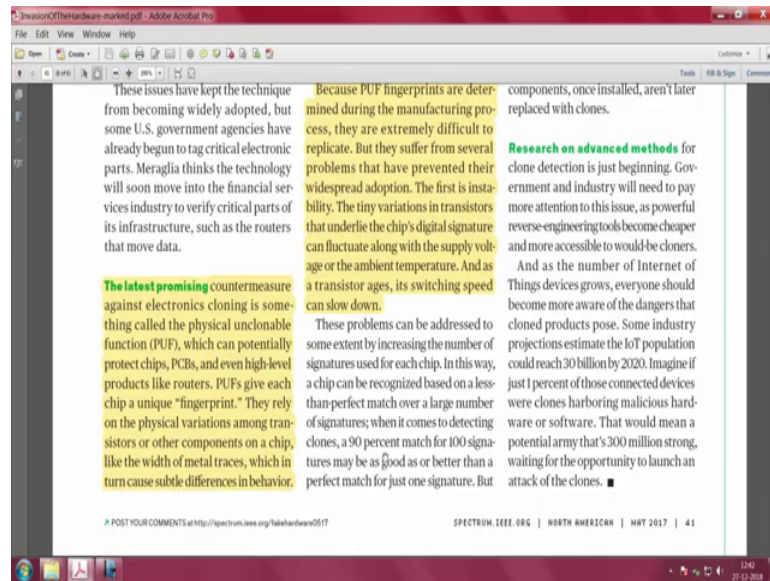
So, you do not have to take it to labs and do all the testing and all that. So, these issues are still there, but I think it is making good progress.

This is one part of the story. The other part of the story is if you do not have DNA matching and all that sophistication built in is there anything, which can trigger from this picture is the question ok, from this picture what can be triggered? It turns out that you can do a lot, already if you have a system like this and generate what is known as a physically unclonable function. It is called PUF and PUFs are well known.

They are not taken off, but they are a cheap good replacement for solving this problem of device identity at least. For sure you if you do any replacement of any subsystem of this you should be able to do an identity tampering that is another thing you can do it differently, but directly PUF may not you know try to tell you anything with respect to tamper, because that is essentially opening a product, installing something and closing if the installation is a hardware installation, then PUF can directly pitch in right hardware installation.

But, it is a software installation what you can only do this you must be able to detect that the system was being opened. And, then you could perhaps you know only flag that part. So, I would say identity is something for sure; you will be able to avoid. And therefore, counterfeiting is something that can come very strongly into this product. So, let us see what is article father says about physically unclonable functions. Now, so, we looked at that alright.

(Refer Slide Time: 25:47)



So, the latest promising countermeasure against electronic cloning is something called the PUF. Now, what is essentially means is potentially PUF physically unclonable function, it can protect chips, it can protect the PCB, even high level products like routers and all that, essentially PUF will give each chip a unique fingerprint. So, it is essentially about fingerprinting and they relay on the physical variation among transistor or other component on a chip. Like the width of metal traces which in turn cause subtle differences in behavior ok.

So, what this means is it is very profound statement to make, that if you buy a chip silicon chip. The process variation from chip to chip is so high, that you should be able to pull out some sort of a signature, some sort of fingerprinting from that chip. This is in a sense what the physically unclonable function is trying to say. That is I should be able to identify uniquely between 2 controllers, that this controller is different from that.

Although they look identical they manufacture in the same assembly line, in the same manufacturing process, because of the process variation you can have you can have different signatures, you can generate signatures, you can generate a fingerprint of the chip, just because of process variation in the process manufacturing process variation ok.

So, let me read it again they relay on the physical variations among transistors or other components on a chip, like the width of metal traces which in turn causes subtle differences in behavior, small differences are there in behavior and this is essentially

what you may want to use. So, PUF designers will look for this variation right. And, it could be in terms of switching speed of different transistors manifesting itself in switching speed right.

So, when many transistors are combined into a circuit, because a chip compresses of billions of transistors like, the differences in their switching speed affect the signal propagation along a specific path which can be measured and compared with the signal propagation of another supposedly identical path on the same chip right. So, this is the key. From the 2 parts the manufacturer can create a one bit signature for the chip ok. Now, he gives an example if the switching speeds are faster along first path than the second one the manufacturer can assign the bit as 1.

So, then you create a longer signature say 16 or 64 bits, it typically compares more parts. So, essentially you are looking at this sentence, when many transistors are combined into a circuit the differences in the switching speed affect the signal propagation along a specific path, which can be measured and compared with the signal propagation of another. Supposedly identical path on the same chip, supposedly the identical path, you make a comparison you will get someone bit signature, you compare with many parts, then you will get you can get a longer signature 16 or 64 bits.

So, essentially the this whole module on security is related to PUFs and I took this long to arrive at this point ok. So, please concentrate on this hot topic and try to look up as much literature as possible on the area of physically unclonable functions, which are essentially generated due to process variations, which manifest itself in terms of what just read about a signal parts timing related, delays that are there between different parts, all that is happening because of the process variation which is indeed the cause of the problem.

Now PUFs parts was be designed into the chip either making use of existing features and you must have a way of measuring it right. So, you must have test circuits or embedded memory or you should build some embedded circuitry which make makes a measurement of this you know path delays right.

So, you must you need circuitry. So, you essentially should make one more chip, which again can have it is own variation, but it has the ability to measure this variations in the existing chip and then. Either you make it chip or you make an additional circuitry and

you have to embed that circuitry, that is the penalty you pay, you build something, you want make it secure, you want to extract something in terms of hardware signature from the actual device because you want to uniquely identify it then this circuitry will be required. That is perhaps a problem, you may say that oh how will I put that additional circuitry, but that indeed has a lot of value in making in to get into to PUF.

Right once a batch of chips is manufactured the chipmakers checks how the PUF structures on each chip respond to specific inputs are generated by external or internal test circuitry. And, then registers the chips unique signature in a database. Manufacturer can also do this test and put it into a data base. Now, a customer can query the database to see if a chip at any phase in the supply chain is authentic or fake, in much the same way a database containing biometric fingerprints can be checked alright.

So, this is still taking help from the manufacturer, because PUF fingerprints are determined during manufacturing process there extremely difficult to replicate, but the question really is why would the manufacturer want to maintain such a database. For every chip that he or she manufacturers in the assembly line it is not going to be a scalable solution right, it is not going to work.

So, I think whoever is trying to use those chips into their systems, they will have to take up the responsibility of adding that additional circuitry externally perhaps. And, then make a measurement themselves and create a private database of all the you know this modules electronic embedded module that they are likely to use in their installation. So, that I think would be better approach, than perhaps asking the manufacture because you going to manufacture billions to cut cause and to put down price of each components it is not going to work, if you have to ask the manufacturer to generate this alright.

So, because PUF fingerprints are determined during manufacturing, they all difficult and all that. The first is in is you know that the tiny variations in transistors that underlie the chips digital signature can fluctuate along with voltage or the ambient temperature. And as the transistor ages it is switching speed can slow down. See, now problems are coming slowly from manufacturing there is some way by which you can perhaps the manufacturer can generate put it into a database right.

But, that is not going to be the final one because what happens is the end use end application of that chip could be in high temperature. End application of the chip could be in some other harsh condition very low temperatures for instance.

Now, will the signature be the same across variations in temperature that is one aspect will the signature remain what it is for all possible voltages that are applied to the chip, because chips essentially if you take controllers they can work from 1.8 volts to 3.6 volts right or even lower also sometimes. So, what is the nature of the signature when the voltages is varied will this delay, you know increase or decrease or. So, will that signature actually vary at all.

So, how does the process essentially we are asking, how does the process variation affect get affected by change in voltages and change in variation in voltages and variation in temperatures. This is the reason perhaps that manufacturers may not want to really keep a database, because it is going to be very fluidic. Also as the time progresses age progresses this can also change.

So, that is another problem of switching speed getting slower as the system goes down. So, that is a problem with PUFs. There is another problem which I already mentioned to you that manufacturers have to include a circuitry or even another chip perhaps.

So, adding this additional circuitry is another problem, which essential says PUF technologies that rely on existing chip components such as embedded memory or test structures to generate the signatures will add little or low cost to the design right. A final problem is that cloners could use some form of modeling. So, it is not that once you generate let us a unique signature across temperature variations, across voltage variation, and store it in your private database, and putting that complicated circuitry to do a measurement, all that or even embed that circuitry on the embedded system itself. Put that circuitry on the chip all that is fine.

But, then you can apply hackers cloners and you know they go any distance right. We also said they do chemical etching to see layer by layer what is happening, looking at microscopes to look at bits in the memory, all that we have seen in this article. So, they can go any length any distance to clone the hardware.

So, the cloner can also use I would say reasonably sophisticated statistical techniques to generate back the behavior of these PUFs. So, you can see behavior of PUFs through some statistical modeling is another possibility. And, researchers have demonstrated that signatures from some PUFs technologies are not as random as initially thought.

So, really possible that several signatures that appeared at the time when the chip was manufactured under room conditions, under certain voltage conditions, just vanished moment temperature changed and where the moment the voltages change. So, you may not have too much too many invariant kind of signatures that you can extract from.

So, that is a one major issue with PUF, but never the less it is still attractive for you to try ok. But, what people researchers have tried is ok, you have some problem with respect to the chip itself and the signature the PUF signature that you can extract from the cheap, but what if this chip is sitting on a PCB and you extend the variability of whatever is happening within the chip to the board level, here is what people have said? Researchers have been extending the concept of concept to PCB.

In this case they use random variations inside the chips, that go on to the PCB as well as variation in the metal traces that connect them. So, if you can extend that it becomes a very exciting possibility, which is unlike chip fingerprints which manufacturers may not want to do. PCB fingerprints can be checked remotely to verify the authenticity of a piece of equipment.

So, this technique could be used for example, to ensure that critical infrastructure components once installed are not later replaced with clones. So, this is really what could be the way out, when you go away from chip based fingerprinting to a system level fingerprinting where PCB fingerprints can be used.

And, also companies like micro semi Xilinx they have started using PUFs for chip identification. Although they have been using it this would be a much more practical manner to look at it from a PCB fingerprint perspective.

Now, research on advanced methods so, government and industry will have to do a lot more and this last sentence is absolutely critical. You see that the number of devices are growing everyone sees the dangers of cloned products, estimates are that IOT population could reach 30 billion by 20 just imagine that 1 percent of this connected devices, where

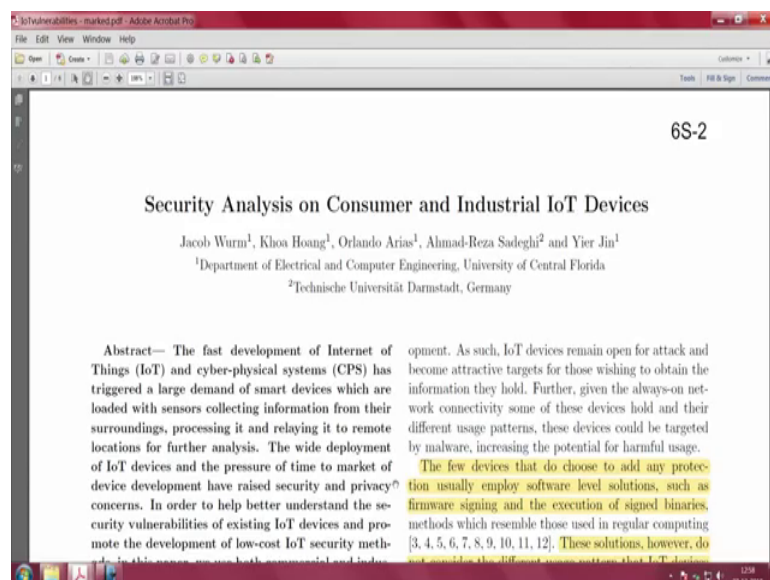
clones harboring malicious hardware or software. That would be in potential army, that 300 millions strong waiting for the opportunity to launch an attack of the clones ok.

So, so, it is really a exciting area at the same time you may want to consider security from the PUF perspective. And, we can see that this article is not very old it is appeared in 2017. And, I urge you to read this article and contemplate on the invention of hardware, as possible way to extract signatures which goes beyond the well known CPS techniques.

Now, let us move on to take one use case of how a sort of a gateway device was hacked into because certain things were open and how the step by step process with which one could get access to if you leave an embedded system open to certain vulnerabilities. Again, I am telling you this is not a CPS vulnerability we are not looking at that we are looking at devices, gateway devices that you can physically see.

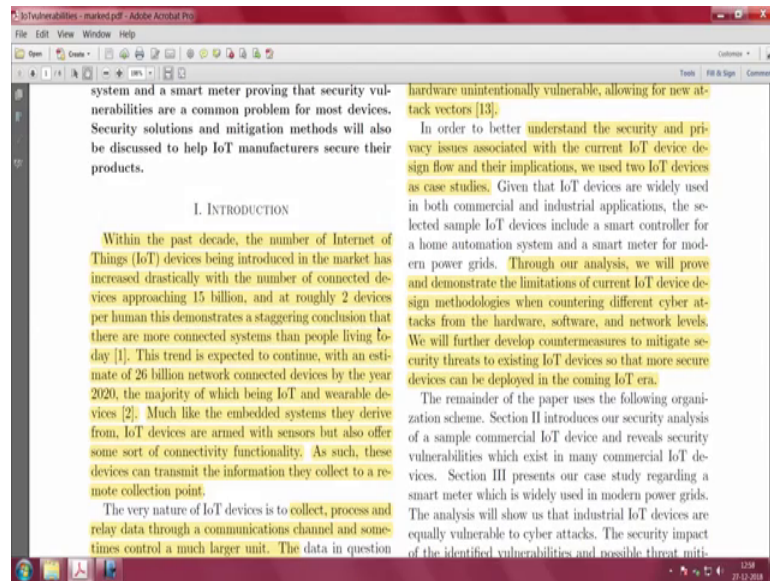
And, you may want to do some sort of reversing on the reverse engineering on the existing not so, much from a product reverse engineering, but access to the device ok. And, then essentially tampering the device that is the second part which is essentially doing a open and then you do as replace and then you close something back right.

(Refer Slide Time: 41:54)



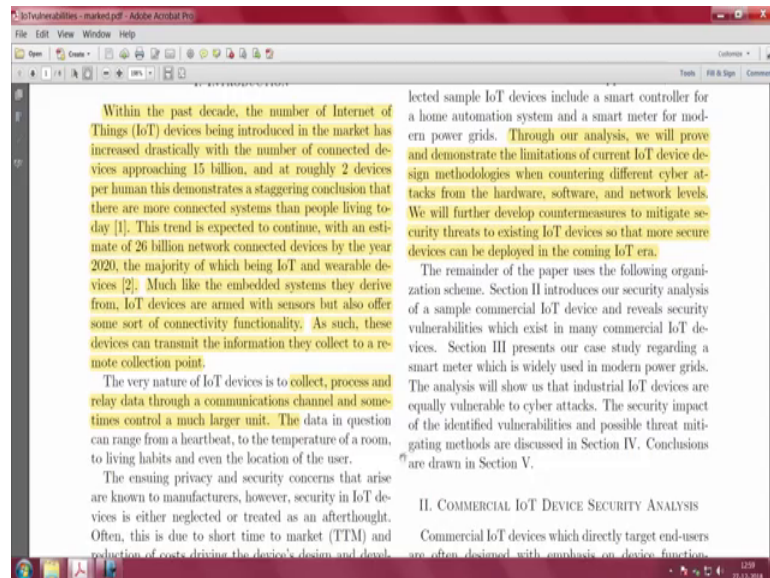
Let me take your attention to this paper which essentially is talking about security analysis on consumer and industrial IOT devices, this is also recently published article.

(Refer Slide Time: 42:06)



And this initial part is all about the IOT devices being introduced to the market you will roughly you have 2 devices per human, that demonstrates staggering conclusion. That there are more connected systems than people living today so, it is going to be a large 25 30 billion devices which are going to appear right.

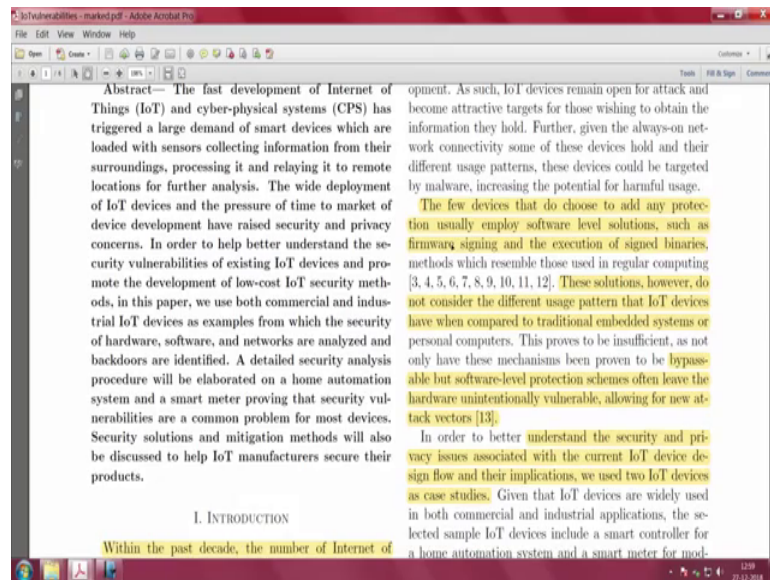
(Refer Slide Time: 42:35)



So primarily this work of the IOT device is to collect process and relay data through communication channel and sometimes control a much larger unit. So, will see what exactly this security analysis of this device is all about, you can do it as I mentioned the

software way right. If, you do it the software way, you want to bring in certain protection, you can do it using a firmware signing for instance, you can do it through execution of and the execution of signed binaries, and these are methods which are very popular you can see this so, much of literature around that part.

(Refer Slide Time: 43:08)



These solutions do not considered the different usage pattern that IOT devices have when compared to traditional embedded systems. So, that is really a problem and if you have this software solutions you actually do a bypass you can do bypass, but software level protections often leave the hardware unintentionally venerable. So, again the axis is about is all about hardware and it is vulnerability that you allow new attack vectors to coming in and vandalize tamper the device.

So, let us see so, so let us see step by step how a product a gateway equivalent product, when I say gateway I mean slightly bigger device right. I will not talking about small embedded sensing devices, but slightly bigger product and how that can be exploited and tampered with.

(Refer Slide Time: 44:29)

home automation system is selected as a case study in this paper.

A. High Level Overview

The Haier SmartCare is a smart device designed to control and read information from various sensors placed throughout a user's home which include a smoke detector, a water leakage sensor, a sensor to check whether doors are open or closed, and a remote power switch. These sensors are connected through the ZigBee protocol. The primary function of this device is to allow the user to better monitor their homes when they are away and to get alerts based on sensor information.

systems such as Linux and Android. Upon analyzing the data sheet for the processor, we were able to locate traces for UART on the device. The SmartCare PCB is shown in Figure 2.

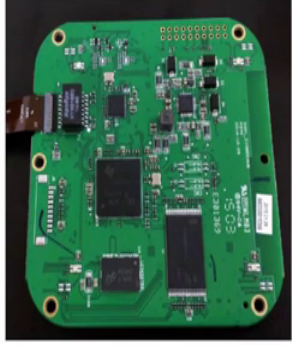


Fig. 2. SmartCare Hardware Platform

So, the device that we are looking at is this one. So, I will show you a picture this is that higher smart care device. So, you can see this is the size of the device. And, it is design it is like a gateway it takes data from several sensors ok. And, you talking about smoke is talking about water leakage, and it is like a hub which essentially gets data from several system including power related that is energy consumption in the home.

So, idea is if you have such hub device at home you will get alerts based on different information.

(Refer Slide Time: 45:04)

6S-2

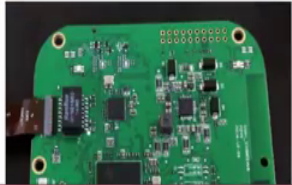
ality. Security features are often added in an ad-hoc manner where remote attacks are treated as the main threats. Therefore, commercial IoT devices often suffer from hardware-level vulnerabilities [14] which may be remotely exploited. In order to demonstrate these security vulnerabilities and help designers/consumers better understand the design backdoors, the Haier SmartCare home automation system is selected as a case study in this paper.

A. High Level Overview

The Haier SmartCare is a smart device designed to control and read information from various sensors placed throughout a user's home which include a smoke detector, a water leakage sensor, a sensor to check whether doors are open or closed, and a remote power switch. These sensors are connected through the ZigBee protocol. The primary function of this device is to allow the user to better monitor their homes when they are away and to get alerts based on sensor information.

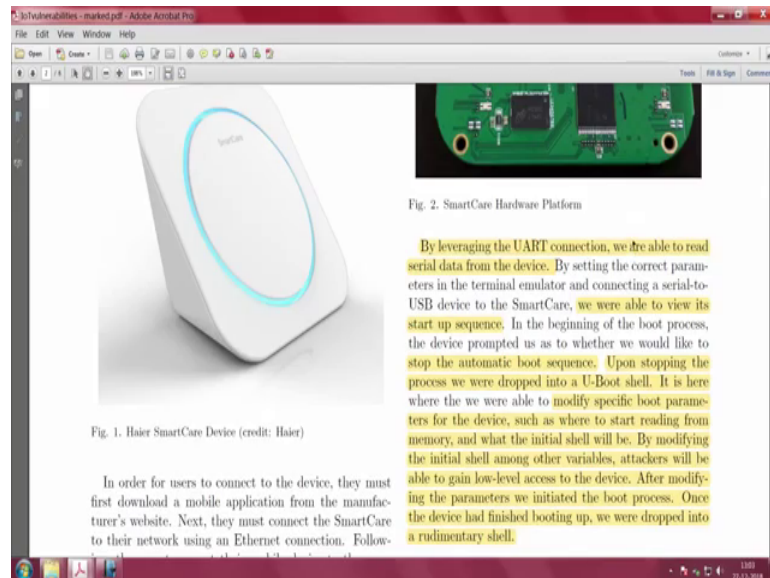
B. Hardware Analysis

The first step in our vulnerability analysis was to analyze the components on the SmartCare's hardware platform. The main processing unit is a TI AM3352BZCZ60, which is a part of TI's Sitara line of processors. The processor contains an ARM Cortex A8 with NEON extensions. The processor also supports the use of operating systems such as Linux and Android. Upon analyzing the data sheet for the processor, we were able to locate traces for UART on the device. The SmartCare PCB is shown in Figure 2.



Now, the question is how does one hack into the system. You can see this is the hardware it runs linux ok. And, it is based on arm cortex a 8 and it is it is a neon extension processor and all that and it has the UART interface ok. So, upon analyzing the data sheet for the processor by looking at the see processor you know that it has a UART interface.

(Refer Slide Time: 45:38)



The screenshot shows a PDF document with two figures. Figure 1 is a photograph of a white, rectangular SmartCare device with a glowing blue ring around its top edge. Figure 2 is a photograph of the device's internal green printed circuit board (PCB) with various electronic components. The text in the PDF describes the device's connectivity and the process of accessing its boot sequence via a UART interface.

Fig. 1. Haier SmartCare Device (credit: Haier)

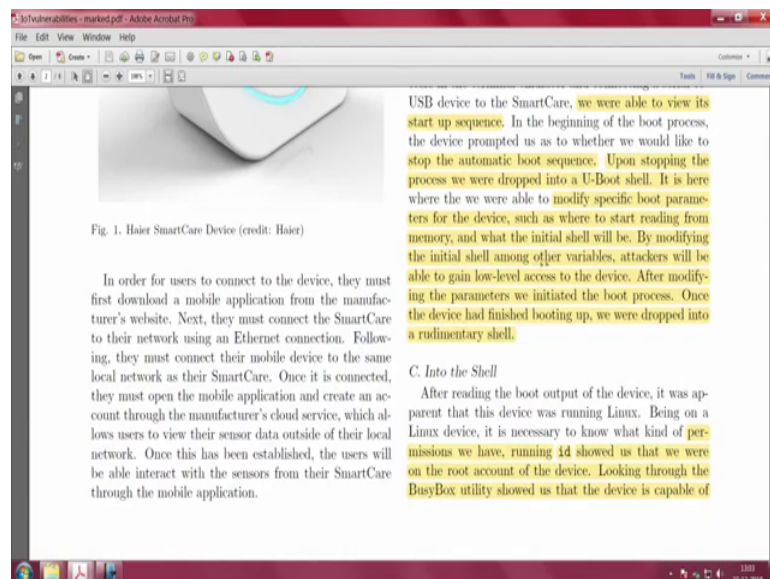
In order for users to connect to the device, they must first download a mobile application from the manufacturer's website. Next, they must connect the SmartCare to their network using an Ethernet connection. Following this, they must connect their mobile device to the same local network as their SmartCare. Once it is connected, they must open the mobile application and create an account through the manufacturer's cloud service, which allows users to view their sensor data outside of their local network. Once this has been established, the users will be able to interact with the sensors from their SmartCare through the mobile application.

Fig. 2. SmartCare Hardware Platform

By leveraging the UART connection, we are able to read serial data from the device. By setting the correct parameters in the terminal emulator and connecting a serial-to-USB device to the SmartCare, we were able to view its start up sequence. In the beginning of the boot process, the device prompted us as to whether we would like to stop the automatic boot sequence. Upon stopping the process we were dropped into a U-Boot shell. It is here where we were able to modify specific boot parameters for the device, such as where to start reading from memory, and what the initial shell will be. By modifying the initial shell among other variables, attackers will be able to gain low-level access to the device. After modifying the parameters we initiated the boot process. Once the device had finished booting up, we were dropped into a rudimentary shell.

And, then you know that your first thing is let us see if I can hack into the use that UART interface and do a few things there alright.

(Refer Slide Time: 45:48)



This screenshot shows a different page from the PDF document. It features Figure 1 (the SmartCare device) and a detailed paragraph about the boot process. A section titled 'C. Into the Shell' describes the steps taken to access the device's shell and the permissions obtained.

Fig. 1. Haier SmartCare Device (credit: Haier)

In order for users to connect to the device, they must first download a mobile application from the manufacturer's website. Next, they must connect the SmartCare to their network using an Ethernet connection. Following this, they must connect their mobile device to the same local network as their SmartCare. Once it is connected, they must open the mobile application and create an account through the manufacturer's cloud service, which allows users to view their sensor data outside of their local network. Once this has been established, the users will be able to interact with the sensors from their SmartCare through the mobile application.

USB device to the SmartCare, we were able to view its start up sequence. In the beginning of the boot process, the device prompted us as to whether we would like to stop the automatic boot sequence. Upon stopping the process we were dropped into a U-Boot shell. It is here where we were able to modify specific boot parameters for the device, such as where to start reading from memory, and what the initial shell will be. By modifying the initial shell among other variables, attackers will be able to gain low-level access to the device. After modifying the parameters we initiated the boot process. Once the device had finished booting up, we were dropped into a rudimentary shell.

C. Into the Shell

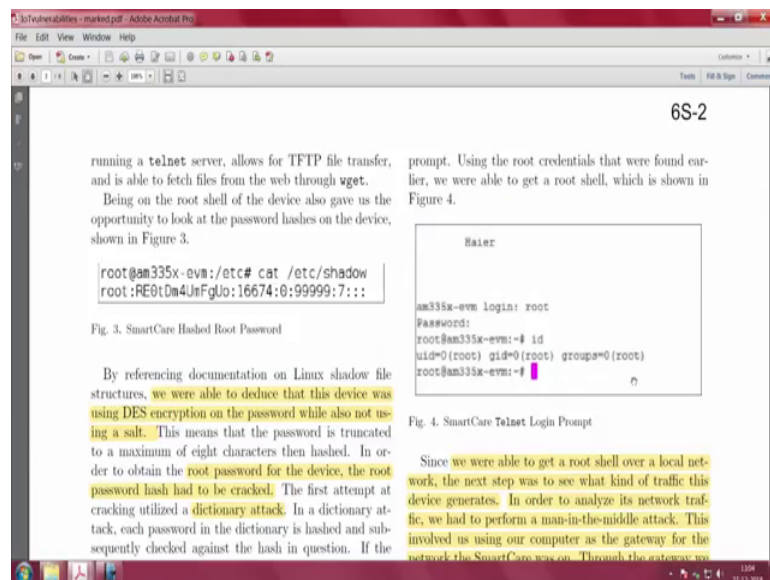
After reading the boot output of the device, it was apparent that this device was running Linux. Being on a Linux device, it is necessary to know what kind of permissions we have, running `id` showed us that we were on the root account of the device. Looking through the BusyBox utility showed us that the device is capable of

So, that is the first step. In order for the users to connect to the device they must first download some mobile app that is all from an applications perspective ok. Next, they must connect the smart care to the network using Ethernet connection and they must follow some procedure and all that right, but for the person who wants to you know tamper with it, he is leveraging he or she is leveraging UART connection ok, to read the serial data alright. Once you connect that you are able to go to the u boot shell ok, we are dropped into the u boot shell.

So, you see the first step is your able to view it is startup sequence and then you know that it is going to U-Boot shell. It is here that you modify the boot parameters and you modify the initial shell among other variables attackers will modify gain low level access to the device. After, modifying the parameters you again initiate the boot process, once the device is finished booting up you are dropped into a rudimentary shell.

So, you make some changes to U-Boot shell make the initial changes you get into this rudimentary shell. Once, you have this rudimentary shell you look at permissions, the running id that he showed to us and looking through the busy box utility.

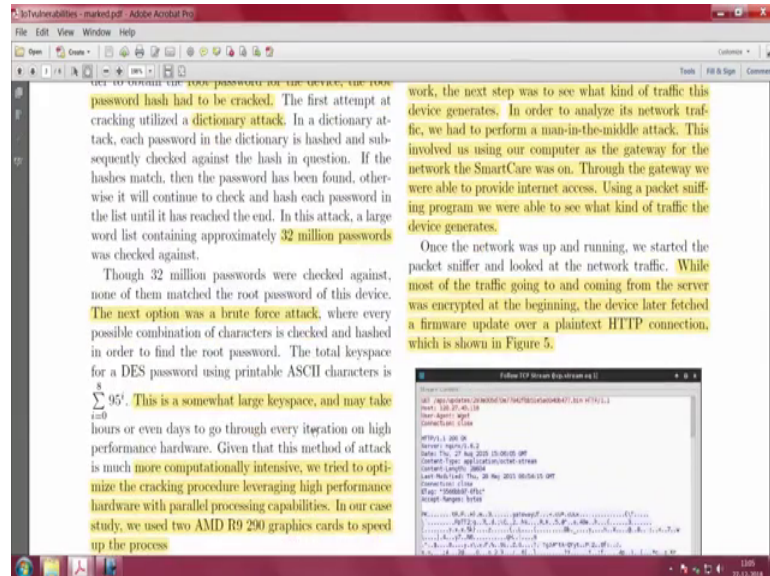
(Refer Slide Time: 47:10)



So, that the device is capable of running some telnet and all that and referencing the document you will see that we are able to reduce that, the device is running des encryption ok. On the password while also not using not using a sort of you know it is basically is using a des encryption.

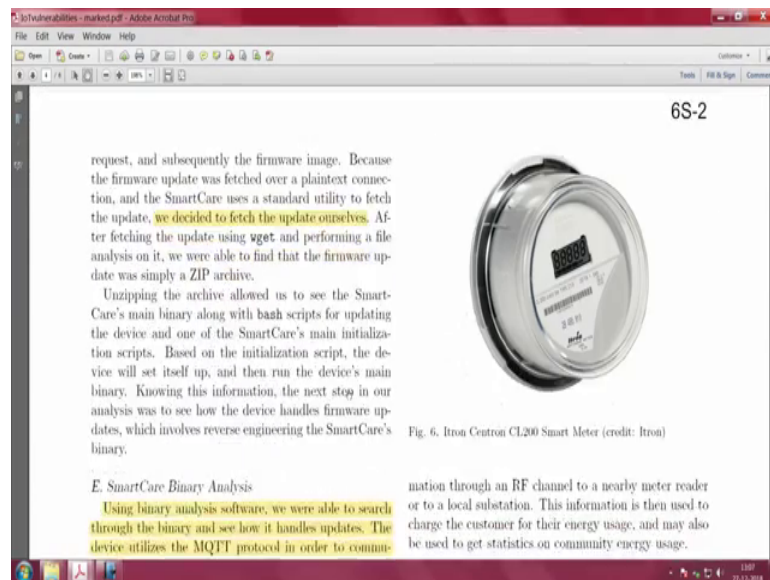
So, now the question is this means that the password is truncated to maximum 8 characters.

(Refer Slide Time: 47:38)



You need to obtain the root password for the device and the root password hash you have to crack into it. So, what you do if you want the root password, you do a dictionary attack. So, you will have 32 million passwords you run through 32 million passwords and checked. And, you see if you can get to the password or you have another option, you can do a brute force attack as well and all though this is a large key space and may take hours to get it into that.

(Refer Slide Time: 49:32)




request, and subsequently the firmware image. Because the firmware update was fetched over a plaintext connection, and the SmartCare uses a standard utility to fetch the update, we decided to fetch the update ourselves. After fetching the update using `wget`, and performing a file analysis on it, we were able to find that the firmware update was simply a ZIP archive.

Unzipping the archive allowed us to see the SmartCare's main binary along with `bash` scripts for updating the device and one of the SmartCare's main initialization scripts. Based on the initialization script, the device will set itself up, and then run the device's main binary. Knowing this information, the next step in our analysis was to see how the device handles firmware updates, which involves reverse engineering the SmartCare's binary.

E. SmartCare Binary Analysis

Using binary analysis software, we were able to search through the binary and see how it handles updates. The device utilizes the MQTT protocol in order to communicate through an RF channel to a nearby meter reader or to a local substation. This information is then used to charge the customer for their energy usage, and may also be used to get statistics on community energy usage.

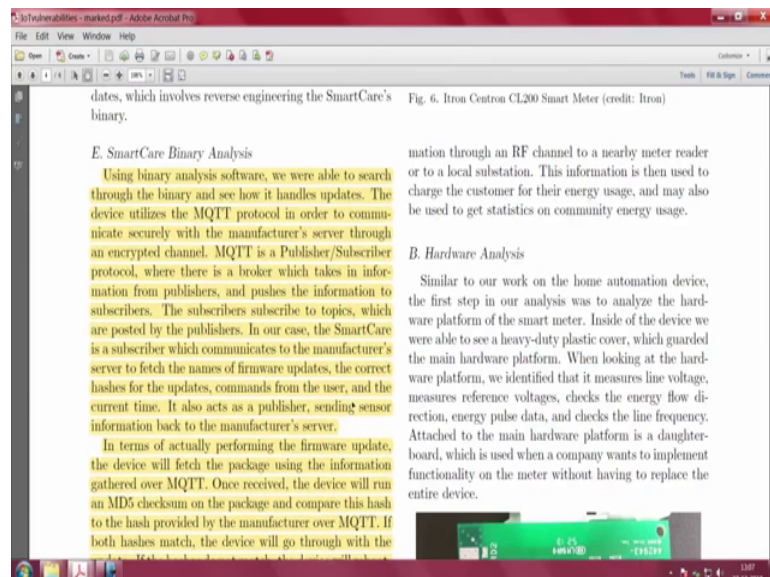


6S-2

Fig. 6. Iron Centron CL200 Smart Meter (credit: Itron)

So, any firmware update uses http and it uses simple plain text ah. So, if you know that much why not go and get the update ourselves right, you can since you all know that. So, you can get the update yourself and put it there. So, this is one form of attack that you are bypassed the complete system and you are able to get it yourself.

(Refer Slide Time: 49:54)



dates, which involves reverse engineering the SmartCare's binary.

E. SmartCare Binary Analysis

Using binary analysis software, we were able to search through the binary and see how it handles updates. The device utilizes the MQTT protocol in order to communicate securely with the manufacturer's server through an encrypted channel. MQTT is a Publisher/Subscriber protocol, where there is a broker which takes in information from publishers, and pushes the information to subscribers. The subscribers subscribe to topics, which are posted by the publishers. In our case, the SmartCare is a subscriber which communicates to the manufacturer's server to fetch the names of firmware updates, the correct hashes for the updates, commands from the user, and the current time. It also acts as a publisher, sending sensor information back to the manufacturer's server.

In terms of actually performing the firmware update, the device will fetch the package using the information gathered over MQTT. Once received, the device will run an MD5 checksum on the package and compare this hash to the hash provided by the manufacturer over MQTT. If both hashes match, the device will go through with the update.

B. Hardware Analysis

Similar to our work on the home automation device, the first step in our analysis was to analyze the hardware platform of the smart meter. Inside of the device we were able to see a heavy-duty plastic cover, which guarded the main hardware platform. When looking at the hardware platform, we identified that it measures line voltage, measures reference voltages, checks the energy flow direction, energy pulse data, and checks the line frequency. Attached to the main hardware platform is a daughterboard, which is used when a company wants to implement functionality on the meter without having to replace the entire device.


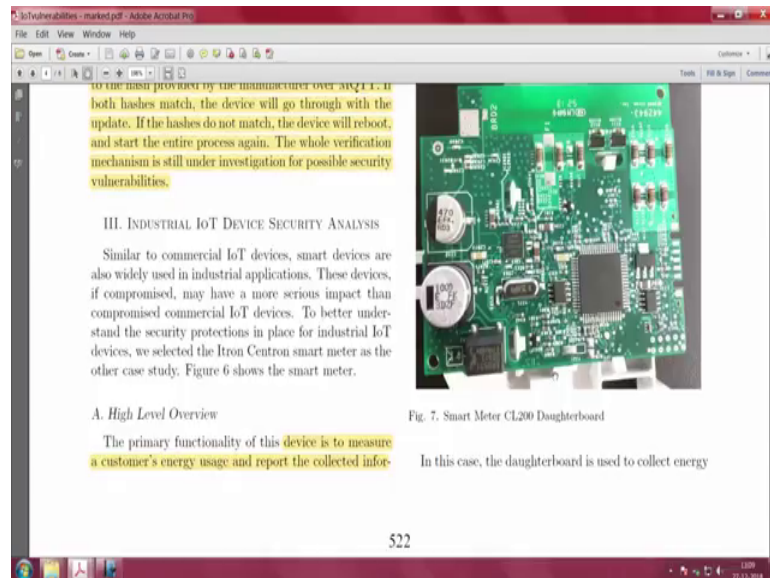


Fig. 6. Iron Centron CL200 Smart Meter (credit: Itron)

Then, you come to know from binary analysis that it uses MQTT, which uses publish subscribe there are brokers, there are publishers, and there are subscribers, perhaps the nodes the sensor nodes are publishing the data and this gateway device is subscribing

towards more like a broker and come also the subscribers. So, it has access to all the data right.

(Refer Slide Time: 50:18)



The screenshot shows a PDF document with the following content:

to the manufacturer over a secure channel. If both hashes match, the device will go through with the update. If the hashes do not match, the device will reboot, and start the entire process again. The whole verification mechanism is still under investigation for possible security vulnerabilities.

III. INDUSTRIAL IoT DEVICE SECURITY ANALYSIS

Similar to commercial IoT devices, smart devices are also widely used in industrial applications. These devices, if compromised, may have a more serious impact than compromised commercial IoT devices. To better understand the security protections in place for industrial IoT devices, we selected the Itron Centron smart meter as the other case study. Figure 6 shows the smart meter.

A. High Level Overview

The primary functionality of this device is to measure a customer's energy usage and report the collected information.

Fig. 7. Smart Meter CL200 Daughterboard

In this case, the daughterboard is used to collect energy

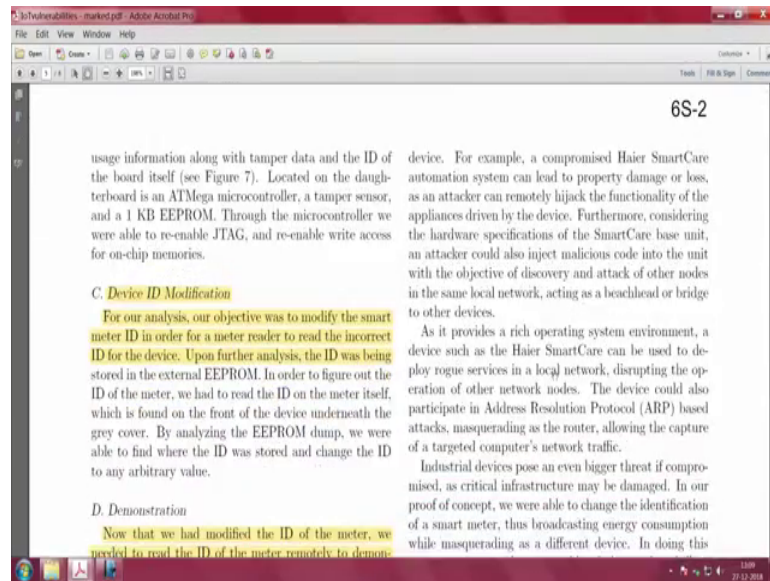
522

So, the since it is a broker, it can also have access, you can also do it all can also act as a publisher right sending sensor information back to the manufacturers server, it can also do that hm.

So, you can now that you can actually do a firmware update, the device will fetch the package using information gathered MQTT and the device will run some MD 5 checksum on the package compared to the hash and all that. If, the hash matches the device will go through with the update. The hashes do not match the device will reboot and start the entire process again the whole verification mechanism is still under investigation for possible security vulnerabilities right.

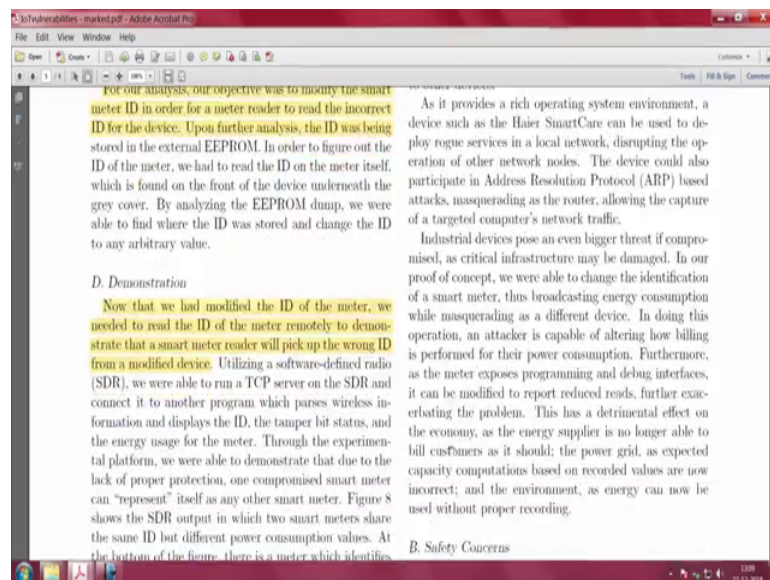
So, this is what you could do further damage. High level you can since this is collecting data from several census which are giving you information on energy data, you can play havoc on the energy data on that part.

(Refer Slide Time: 51:21)



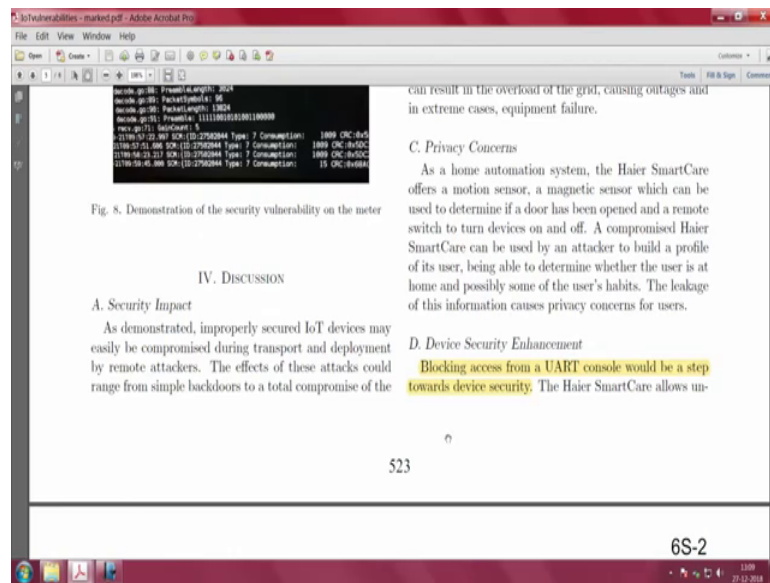
And you can also modify the device ID right and you know where it is being stored you know that it is in e square prom, and you can go and analyze e square prom dump you will be able to get the ID and you can change the ID.

(Refer Slide Time: 51:39)



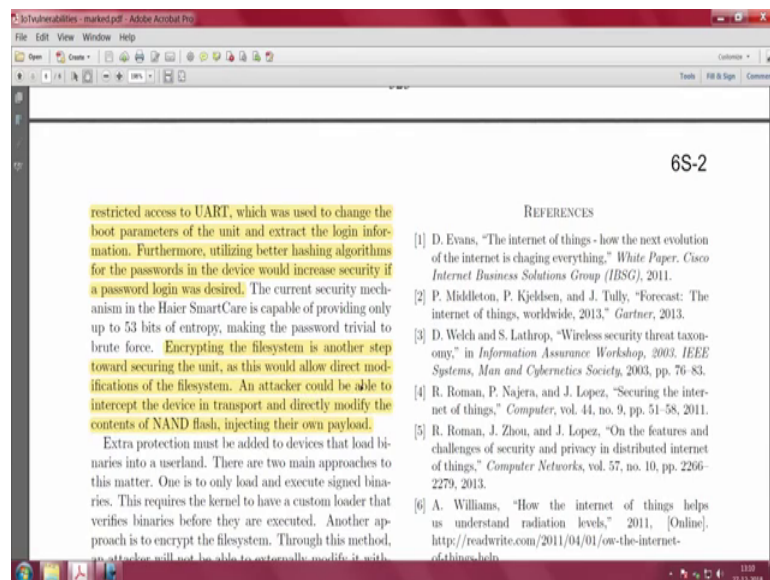
So, all of this means that using this modified ID, you can demonstrate that a smart reader will pick up the wrong ID from a modified device is nothing you can do, because you are modified it and you can manipulate all of that.

(Refer Slide Time: 52:00)



So, what is the way, that is the question really; one is you should actually block UART completely you should never allow. Because the whole root of the problem started because you had UART axes.

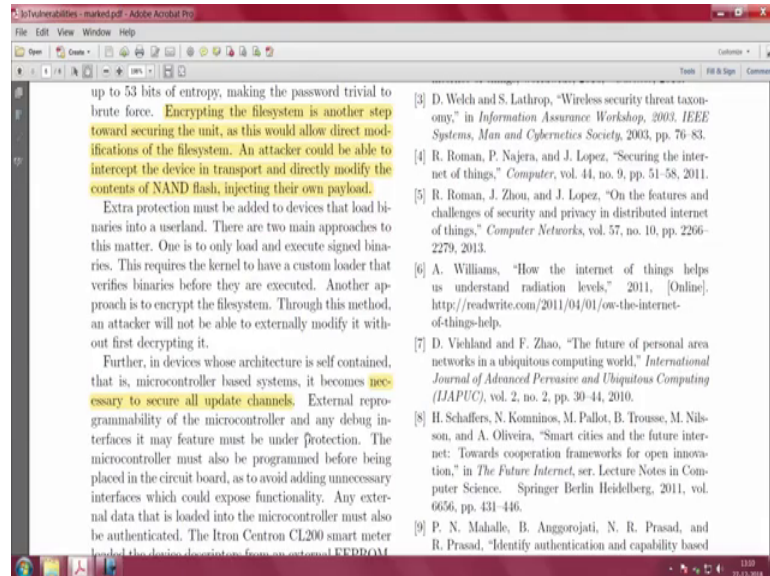
(Refer Slide Time: 52:15)



So, that is a first step towards ensuring that you do not allow UART to be restricted access to UART. And, once you do that all this boot parameter modifications can be avoided, then you can also use better hashing algorithm for passwords, then you can encrypt the file system that is sitting on the embedded device this gateway device. So,

the attacker could be able to intercept the device in transport and directly modify the contents of the nand flash. So, if you could be encrypted that is also good.

(Refer Slide Time: 52:52)



So, all of this means you will have to ensure that the system the device is whose architectural self-contained, that is microcontroller based system it becomes necessary to secure all update channels. Do not do all this plaintext updating and all that. So, you should avoid doing them.

So, you can see that this paper essentially spoke about step by step way by which one can do you know get into the device, you can essentially, you can open, you can install, and you can close. And therefore, you can tamper the device. And, there are simple ways by which you can avoid doing that.

So, in summary all of this means that this area is open and rich in problems and I will explain to you what you can do with all these signatures which are available scattered round an embedded device.