

**Storage Systems**  
**Dr. K. Gopinath**  
**Department of Computer Science and Engineering**  
**Indian Institute of Science, Bangalore**

**Storage Reliability, Performance, Security**  
**Lecture – 29**  
**Storage Security**

Welcome again to the NPTEL course on storage systems. In the previous class, we looked at certain concepts regarding performance of the storage systems. So, another aspect of storage is security; we will briefly look at it to get some idea about some kind of issues and security.


(Refer Slide Time: 00:43)

**What does Secure Storage mean?**

If storage has to be a bedrock, have to ensure that it is

- Highly available
  - Resilient to failures
  - Resilient to DoS/DDoS Attacks
- Protected from intruders
  - Prevent malicious tampering
  - Controlled access; avoid leakage of information
  - Prevent replay of stale information
- To the extent feasible, use formally verified ("correct") storage protocols
  - No trapdoors either from a systems, protocol or cryptographic perspective: NFS root; X11 auth, ...

**May need tamper-proof archival storage (legal&c reasons)**



Now, one thing you should notice is that storage is critical aspect of a system without storage we are like person has no memory; you cannot boot a system, etcetera. So, it is basically critical piece of data without which nothing is possible. So, if storage is to be a bedrock on which rest of it all stands, we have to make sure it is highly available what does it mean highly available in spite of various problems; it is still be accessible; for example, resilient to failures nowadays there is this what is called distributed denial of service attacks or denial of service attacks somebody is trying to prevent you from doing any work; they not luckily stealing your data or destroying your data or basically corrupting your data, but they prevent preventing access to your data. This is a really big

problem and so, people and actually is now its becoming attacks between nation states also certain countries are deciding to figure out how to get back on other countries without any blood this is basically do it through computers right.

So, that also is going on. So, you also if your storage you might have to prevent it from malicious tampering and this can happen inside and outside you can have people inside your organization and do it and people outside also can do it and people generally say people inside are the more difficult enemy basically most studies say that 70 percent of all tampering happens because of inside people 30 percent among them and you also want to control access you want to avoid leakage of information certain things; I sensitive for example, you know that there is somebody building in the airport right and it is leaked to the right people, they can buy the land near the airport and make building right.

So, some information is critical; if you are if you take your job of governance very seriously you should avoid giving inputs by which certain un-script participants get advantage. So, same story and this can be in the form of speech or it can be in the form of files and. So, it is up to you to figure out how to make sure these files are not in the wrong hands it had be really stolen at a particular time not before that. So, that is one other thing also is what is called prevent replay of stale information sometimes they are asking for some information and somebody maliciously giving you old information. So, that you are have a wrong information and do the wrong things. So, this also can be quiet dangerous; that means that if some data has changed in spite of multiple copies anywhere you have to ensure that the wrong information wrong information is not sent to you.

There are some situations that are absolutely critical there are some situations that are not critical the good example is if you are watching cricket if somebody gives you a score which is delayed by 2 minutes that is not the current score right, does not matter that much it sort of a soft system it does not really matter, but for some people it absolutely critical it has to be accurate information. So, that is why it is important for you to use correct protocols we are not got into protocols right now, but generally protocols are messages sent across 2 entities storage devices or storage consumers or producers and we have to make sure that these protocols are also safe or correct.

So, this turns out to be non trivial problem and sometimes it turns out that you need what is called tamper proof archival storage how do you show that some storage has not been tampered with a good example is if you have in a mobile phone we have what is it called we have a SIM card and it stores some information and it has to be tampered proof basically because you have the device in your hands and if you are a very clever guy you can if you can see enough of what is there inside, we can probably change it. The idea is that if you can change it, then it not good for monitoring or tracking certain things which is the government or people might want to look into or some probability reasons I do not want to get into why somebody wants to monitoring, but sometimes it is necessary.

So, if somebody is because the gadgets in your hands you can be physicists who knows how to send leisure pulses which can modify the bit who knows what sophistication we have; nobody can tell you; which person has got what is instigation or it could be a somebody probably the criminal band and he is basically have the whole lab which does not re-bond this and there are people who can take a chip and take out one layer at a time and then see what the bits are inside people can do all these things because basically you are talking about few microns we can get microscopes and whatever similar kind of devices which can look at microns nothing very deep.

So, you can look at it. So, what is important is sometimes you need what is called tamper proof you can you are a sophisticated guy you can peel one layer at a time and figure out a bits out, but the important thing is that in a minute you remove it right somebody should be able to know that you removed it is like the way in our country for example, or many other countries when you send a post they have to put a lac; they put some what is called lac right, you steal seal with a lac oh right; l a c and somebody is opened; it comes out, it is a bit fragile stuff the minute, you open it, you can see somebody has taken it off it is not that that seal is not there right therefore, you can figure out what is being tampered somebody in principle can do it.


We need some such things even in the case of storage systems typically you can do it by what you can do is you can have someone is called if you use computer cryptographic models you can have hollow called hashes and if you do certain things, then if you change some things the hashes do not; no longer match and therefore, you know something has happened. So, your tamper proof can be at multiple levels it can be

physical levels or it could be at the logical or bit levels or at the cryptographic level this can handles ok.

(Refer Slide Time: 07:40)

**At What points within the system is security needed?**

- Security at FS, block, device levels
  - Also at std network security issues if storage is networked
- Standard security issues
  - Integrity
  - Secrecy
  - Availability (DoS attacks)
- New security issues:
  - Flash wear (DoS)
- Viruses often spread thru storage devices (floppy, USB, ...)
- Security for Metadata (small amounts) vs Data (large amounts)
  - Public Key encryption OK for metadata but not for data
  - Stream ciphers with symmetric encryption for data

 **Aggregation attacks**  
When lots of data, new patterns or secrets can be deduced

Again if you look at security in storage with certain minimum multiple levels and I am just listing few of them at a file system level at the block level at the device level file system level means the file system takes care of encryption and decryption. For example, that is the possibility or it may not use encryption decryption, it just make sure that you cannot look at my stuff vice versa; how will that possible by using what is called access controls it is not using cryptography; cryptography is the mechanism. So, the question is whether use that mechanism or some other mechanisms if you use cryptography then use encryption decryption that is one way you make the bits available, but the person cannot read it or you can do what is called physical separation; I ensure that whatever happens you cannot look into my stuff by having for example, access controls right.

I lock certain things and keep it aside it is not encryption decryption, it is just physical in some sense equivalent of physically you are separating it out. So, that also you can do at in the block level; for example, if you have a red device somebody makes you will provide a block level interface and he can assume the responsibility for encryption and decryption or the device level; for example, many disk vendors nowadays give you whole disk encryption you can encrypt the disk and it is decrypted at as faster as regular disk can basically because they have chips which can do what is called on its called

online decryption or forgotten the term it is called I am not able remember the term, but basically it means that I as you are writing it can encrypt and as you are reading it can decrypt at the same speed you do not lose any speed in this.

So, in addition to this things if you are building very large storage systems you are going to use the networks to connect up multiple pieces of storage or to keep replicas you want to keep them right. So, if the storage network you have to all the every single aspect of network storage issues will also be here just like the number of layers in a network are going to be also present in your storage stack because you are going to have stack and power the network same here also typical security issues are integrity secrecy availability integrity means unable to assure to modify something secrecy means somebody should not be able to see it if it is not appropriate for somebody to other than authorized parties to see it availability is figure out a way in which in spite of people trying to preventive from accessing information still be able to survive dos attacks ok.

Now, the other interesting issues also coming up with new devices for example, flash wear what is the issue here you I am trying to destroy some information I am unable to let say get access to normal roots what I will do is I will find a way in which you cannot use a device somehow how can you do it I just keep writing preferably to the same location; I especially burn the flash data equivalent of that one essentially keep making sure that the device still start running out of space this has to happened also in other context; for example, in the nineteen sixties you had something called core memory the core memory was basically magnetic kind of things and it will have 2 state; 0 and 1 and it goes to some current present it flips in one direction and on other direction.

Now somebody can let a small type of loop basically you branch to the same memory location and keep on executing it; that means, the current is going through that coil all the time and finally, that particular coil element gets burnt out similarly in flash also in spite of your wear leveling you can somehow ensure that you are essentially basically you want to force the device to be unavailable; how do you make it unavailable normally in flash kind of devices SSD kind of devices you have some spare space. So, that you handle this worn out things the idea is to keep writing in such a way just like the way you are doing in a core memory right you just keep on looping at across the single word right you just go only on keep touching one particular area; here also you can keep on

touching in spite of wear leveling you try to get make sure that there is so many rights are done.

So, that your spare capacity is completely gone and people finally have to write up the device; somebody has to take it offline copy it or essentially is dosic act. So, that also is possible. So, there are now solutions even for this that even if you try to write it they see the kind of pattern that being written and they try to get account of measures for it there are some people are working under certain kind of problems.

Other aspect of security on storage is you will find that this storage medium is often a good medium for spreading viruses because we use that as a way to exchange information. So, viruses often travel through this mechanism you must be quiet well aware of this problems and what is the problem with this it turns out that there are no pool proof we have checking it if something is have virus we can have lot of false positives right and so theoretically; it is impossible that is no way to prove that something is a virus or something is not without false positives or false negatives not possible other aspect of security in storage is that you can as I mentioned before we have this notion of metadata and data also metadata is small data is large.

So, often times you might have heard something called public key encryption system there are 2 types symmetric and asymmetric keys versus basically you have systems with 2 keys public key encryption systems, but this public key encryption systems are extremely difficult for large amount of data we can only use it for small amount of data. So, the solutions we depend on public key encryption models for data will not work absolutely not work because there is too large because essentially this involves public key encryption requires exponentiation; exponentiation in that takes time. So, typically for example, if you can do I got the word over-wire line speeds that what I was trying to say at this was wire line speeds typically you want to you can do this symmetric encryption at what is called wire line speeds; that means, as it is coming in you can do that is a in same speed whereas, public key is not possible.

For example, if you often get about few millisecond delays with public key encryption. So, luckily it is not much bigger than the disk delays. So, you can hide it basically this or it is in a noise, but still you cannot really depend on it is just not a reasonable thing for large amount of data. So, that is why you need what is called stream ciphers with

symmetric encryption data. So, this is happening at bar line speeds. So, our system actually has got both metadata and data; that means, you have to often times do both sometimes you might want to public key encryption for metadata or you will do stream ciphers for symmetric encryption for data what or sometimes you just use these symmetric encryption for everything; we will if we look at DVD; DVD use a some type of encryption mechanism we will; let us look at what they do there.

Other aspect of security in storage is that when you collect lot of data you aggregate lots of tested information and sometimes new patterns arise or new secrets leak out from the aggregated information what is if information kept separate some patterns may not able at some secrets or may not complete and not useful, but once it put, then that it turns out that you can start seeing something useful coming out of it and. So, this is certainly a problem with network storage because by definition storage actually aggregated for a information right. So, is a aggregation attacks becomes possibly large amount of storage.


So, as I mentioned before there are 2 types basically we can do a access control or you can do cryptography we can do that is physical separation mechanisms or cryptographic mechanisms various kinds of mechanisms possible.

(Refer Slide Time: 16:47)

**Systems security**

- Systems with basic access control since timesharing systems began ('60)
  - Multics, (Unix) *rwX* at file level
  - MAC vs DAC
    - SELinux model
- Cryptography used widely but...
  - "If you think cryptography is the solution to your problem, you don't know what your problem is." Roger Needham
  - Key mgmt critical
- Complex world-wide information systems, netw/storage subsystems, etc require much more sophisticated models
  - anonymous users/services, delegation, trust mgmt, scalability
  - need to have an integrated model of all authentication/ authorization models: *rwX*, *setuid*, PAM, SELinux, cryptofs, X11 auth, NFS, ssh, httpd, IPsec, firewalls, iSCSI, ...
  - highly available access control: eg: clusters, SANs

**Info Flow Models**  
Need proof that info flow respects some security policies

 NPTEL

So, what initially started out in the beginning was that we started with access control mechanisms I think all of your familiar with this notion of RWX at a file level, it tells

you the user owner can read write or execute from files whether the if you are member of particular group what group owners can do with that file and if anybody are outside this user or group what they can do there are 3 things occur right also this has this is what is called a discretionary access control mechanisms; that means, that if you get a permission to do something you can in turn give it to somebody else.

Sometimes this is not a good model with some very valuable information is there I give it to you, but I also want to control what you do with it and that is what is called mandatory access control that is that in spite of my giving you some piece of information I still want to completely control what you do with that information typical file systems use discretionary access control for example, EXT 3, X 2; all those things have typically only DAC discretionary access control that is I give you some file and you can do what you want to do with you can give it to your friend or anybody I do not have any control about what will be that permission and there are near models in SELinux in this model in available in Linux; there you can do mandatory access control we will briefly talk about it later ok.

So, we have access control mechanisms you also use you can also use cryptography I think we discussed it earlier it turns out key management becomes very critical here without key management cryptography is a very difficult thing there is something called identity base encryption which is a new research area that people have working on that might solve this problem, but it is still not clear. So, again one thing you have to keep clear in mind is cryptography is a mechanism whereas, security policy that are higher level you decide who whom should share what that is a policy decision whereas, cryptography is a mechanism its one way to do something. So, the policy can be same, but it can be done in many different ways and cryptography is one mechanism.

So, often times people confuse cryptography with secrecy and other things that is why that is an interesting comment often if you think that somehow you put this masala called cryptography into it is also problem right; that means, if you probably do not know what you are talking about that is interesting quote if you think cryptography is the solution to your problem then you do not know; what the problem is; why is that because you are not really figured out how to do the key management and. So, many things are not specified just because it is a cryptography; I am going to solve its not solved and will see



in the case of DVDs; why the DVD system finally, you know it is a completely broken system the DVD encryption system.

Again, if you take if you want to think in a certain larger sense world wide information system network storage systems etcetera you require much more sophisticated models because in this kind of models in this RWX order right we had users right, but some other I want to add what is called anonymous users the Unix allow anonymous users they does not have the notions seriously ok, you also want to have the notion of delegation I have for example, many busy people are secretaries right I want to do this I do not have the time to do it I delegated to somebody can I express this notion delegation trust management how many levels at which I can delegate it and I want to delegate only to some only 3-4 people I cannot go beyond that what it can be smaller trust is scalable that I think some of you might have come across a situations where Google or Facebook seems to be very unresponsive because it might be getting lot of authentication and somehow those servers are there is some faults somewhere and volt this overload somewhere and it is not going through.

So, we have to worry about scalability also and all those things are critical because if you notice in our country also we have something called people are pushing something called unique identifier unique identity identifiers UIDs, right and there also if they are saying that if they want to do for lot of transactions banking transactions so; that means, that you have to now keep track of all this authentication etcetera for everybody as they are doing banking transactions many of them if they want to be a real time system, it turns out to be a really difficult preposition because you have to have a available system which will work in spite of all the load that you can put on it is not a trivial problem that is why you need highly available access control.

You also need information flow models which is basically what I was talking about earlier if you have this mandatory access control kind of models I want to be able to say get things in order where things are going where the information is going can I control over it a good example of this is suppose I have a virus scanner I am doing virus scanning on my machine and this virus scanning program is I got from somewhere it could be open source software or it could be for company, but I cannot figure out what its capabilities are I am just naïve user and now the it is for virus scanning it has to touch

a every single file right; that means, that virus scanner can read every file in principle it has to be able to otherwise there is no way to you can find the device user.

Now, what I am not clear is can it be sending some of my data outside how will I know it is not doing it and notice that you can say positively that I will cut off a network then it cannot do any bad step, but you notice that virus scanners are as good as how updated the database is you buy a virus scanner and a new attack comes they will somebody will update that database of network viruses or not; that means, that your virus scanner should be able to update itself.


If it is not able to update itself its basically becomes workless over a period of time; that means, that by definition your virus scanner should have update ability to talk to network, but now I have a problem. Now the minute I give a network access that guy could actually be sending it wherever it wants right; that means, that I have to do something about making sure that the information I have in my hands in the data of if I have in hands; somehow it has to able to say that it can only do virus scanning, but not send it out of the network, but it should able to get updates more side.

So, something on that you need to construct systems which have all this kind of capabilities and for this there is a whole area called information flow models that people are looking at and even if you come up with this flow models how do you know that whatever policies you have actually is support a security policy you have in mind this also another issue that also.

(Refer Slide Time: 24:00)

### CD/DVD/Blu Ray

- CD: no protection or ad hoc
- DVD: CSS (content scrambling system)
  - Every DVD player equipped with a small set of player keys (per DVD player manufacturer)
  - Every disk has a disk key data block organized as:
    - 5 bytes hash of decrypted disk key (H)
    - disk key encrypted with player key 1 (dk1), player key 2 (dk2)... player key 409 (dk409)
  - When presented with a new disc, a player will attempt to decrypt contents with set of keys it possesses
    - Suppose a player has a valid key for slot 100, it will calculate
      - $Kd? = \text{decrypt}(dk100, Kp100)$
    - To verify that Kd is correct, check following; otherwise, next player key
      - $H == \text{hash}(Kd?)$
  - Problem! By trying all  $2^{40}$  possible Kd, disk key can be deduced without knowing any valid player key.
  - To decrypt contents, an additional key tk (title key) decrypted with valid Kd (Kt)
  - Each sector of data files optionally encrypted by a key derived from Kt by XOR of specified bytes from the unencrypted first 128 bytes of the 2048 bytes sector
- Uses a stream cipher (LFSR).
- However, due to flaws,  $2^{40}$  checks reduced to  $2^{16} \Rightarrow$  450MHz Pentium needs <1 min



So, just make a thing bit more clear or more realistic, let us take a look at one type of simple system storage systems all of us are familiar with CDs, DVD probably not blue ray, but you must have heard of it right. So, CDs when the designed when was it designed it designed in the late 1970s, music CDs came out first in 1970s; late 1970s before that; tapes were there and tapes did not have any protection and neither get series because computers were very rarely used in the 1970s; early late 1970s and PC was also not born it is not a common available gadget PC.

So, this thing had no protection; nowadays, if there is protection on the CD; it is typically some password based on system that is some trivial system that is there which will ask you to login something; that means, that it executing some piece of code; that means, it is somebody has mounted that CD from program has taken control, it has complete control has that CD and the program itself is asking some question you have to be login, etcetera or you have to give it to some string some string which you cannot easily manufactured it is basically for example, if we are installing some typical Microsoft software in the olden days, right; it will ask you for a big string right you copy that string and then that guy will do some internal computation hash it do whatever and say that this looks like legitimate thing because hash gave me some value which is what I am looking for therefore, that is so, it had those kind of mechanisms.

When comes to DVDs, they did something slightly more they wanted to be do something more systematic, but I am told actually that this DVD was probability was a sub attached with him this security for the system somebody was in the committee which did not want serious cryptography or there is also some other claim that in those days when DVDs security was decided United States had a very restrictive policy on security or gadgets they because they wanted to break into any system the US government or the CIA; they have a specific requirement that should be able to break into any system in those days. So, the keys which are used for encryption decryption had to be less than or equal to 40 bits that is it is widely believed that US government had passed enough machines in those days; if you have a key equal to 40 bits, they can essentially use brook force attacks to figure out what is it.

So, DVD was designed in those days; that is why we will see that your key cannot be bigger than 40 bits. So, let us look what these guys did ok.

So, now there are what is the system with DVD, we have a DVD players and we have DVDs right, DVD player is manufactured by some DVD player manufacturer not the media itself I am talking about a DVD player it could be Sony, Panasonic, etcetera, right. So, every DVD player equipped with small set of player keys possibly one or more per DVD player manufacturer. So, Sony will have 3 or 4 keys some other some Panasonic will have some of few keys, right. So, every DVD player that we buy will have some set of player keys depending on who the manufacturers is in addition every disk these are different from the DVD player manufacturer, it could be anybody else it could be for example, Hollywood or Hollywood or whatever variety every disk has a disk key data block organized follows what do they have there is a decryption key for that particular DVD that particular DVD that movie or whatever and what they do is they have a hash of it there are hashing algorithms they use they are not going to do with details.

So, it has got a 5 byte hash of that disk key that is stored and these things stored in a particular place on the DVD and there are some special restrictions on the players the DVD players they have to follow certain protocol to read; it is not widely known it is sort of what is called security by obscurity because this DVD players are manufactured by some bunch of people among them also they know the security how to read it same thing about disk other things the disk key is encrypted with various player keys. So, there are. So, many manufacturers of DVD players for every manufacturer there going to have it

encrypted and that will be stored in these places so, but what is the issue here every disk has got a hash and you have the disk key what is required to that DVD the movie right that is encrypted with the various player keys there are various manufacturers for every manufacturer there is a corresponding encryption of that particular disk that you required to read that movie to see that movie.

So, these are the things. So, we have data for some reason it is 409; 409. So, I suspect because of the limitations of getting hidden to a one sector probably the size of this. So, for they also felt that probably, it cannot be more than 409 manufacturer of DVD players in the world wide like whatever.

So, what do we have on a disk you have the hash as well as the encrypted version of that disk key for various players for various manufacturers you might say. So, when you present a new disk, then the player knows what which one it is whether it is key 1 or key 10 or key 100 or whatever it is; it has a set of them a few of them 5, 6 and whatever. So, what it does is it will attempt to decrypt contents the set of keys it possesses it tries decrypt. So, suppose for example, the player knows that it has a valid key for because of truly there are so many slots 1 to 409. So, it is valid key for slot 100 for example; so, it will basically decrypt it use that it will try to use the player keys and will decrypt in slot 100 because there are encrypted here.

So, what comes a possible key which can be used to decrypt the whole DVD contents, but does not know because it is because there are so many players in; so, many DVDs you do not know which one is which if it to verify that is correct you hash that same key that you got and if it is same as the what you already have because there is hash of it already has of the key required to repeat the DVDs already present here. So, what we are doing is you are basically taking the disk key that is encrypted you decrypt with a player keys and then decrypt and see if you get that actual disk key and to ensure that is same as what you expected we hash it again see it if it is same as what you have earlier if that is the case then you know that the key is perfectly we can use it then you can decrypt it ok.

Now, the only problem here is that we notice this equation the H is now and you just have to you can generate all this K D reverser, there is no requirement of any player key here. So, you generate what is if it is a 40 bit, key you have generate 2 to the power of 40

how much is 2 to the power of 40, it is about 4 gigabyte possibility for 32 bit multiplied by 256; how much is that.

So, you need to check about 4 into 256 that is about 1000 giga; this one trillion members of this things that in those days was felt to be very huge; they decided it is not a problem or it was intentionally weak end. So, whatever is we do not know what. So, it turns out there is some additional things that I do and you do not have to actually check 2 to the power 40, there has some other let us say things wrong with the system since the 2 the power of 40, it turns out you need only 2 to the power of 16.

So, because it requires 2 to the power of 16 possibilities it turns out you can easily check it in those days even 450 mega hertz less than a minute, but how does it be that is how it was decrypted. So, this is basically the one key you want to actually see the contents of the DVD there is something called title key we have to decrypt it with that K D; you got and then they have some additional methods of making it even straightly making it difficult for you, but all of this failed finally, what that it was each sector of data files optionally encrypted by a key derived from this title key as really had disk key first and then you get a title key and then you take the title key and Xor with some bytes from the unencrypted first 128 bytes of each 2 to the power of 2 K bytes, etcetera.

There are 2 K sectors in the DVD it take the 128 bits use that as a input along with the KT; the key you got use that as a way to feed it towards what is called stream cipher and this stream cipher as symmetric key therefore, it will give you; you encrypt, we can use the same stream or if we decrypt also you can get the same stream you get it back this can be done fairly fast chose simple encrypt techniques that they can do all this without any trouble, but the basic problem was that in this part of it they made big mistakes, I do not have the time to go into it.


They make some difference basically instead of 2 to the power of 40 it turned out to be 2 to the power of 16 or something like that that is only number of possibility that you have to check and you will see that this kind of mistake also was done in Wi-Fi, I think you might have heard that in an when a 2002 or 3 for example, there was some type of security which was totally broken and anybody can actually access to your system very easily in those days and later now you have a slightly better system now. So, so this DVD thing just failed flopped in a serious way and you might have heard of something

called DCSs some hackers who figured out the system, they are realistic that is why you can read any DVD now it is not difficult ok.

(Refer Slide Time: 35:08)

### AACS (Advanced Access Control System)

- Blu-Ray
- Fixed some of the problems of CCS but broken here also due to another attack
  - In spite of many layers of encryption, keys needed to obtain unencrypted content stream that is available somewhere in memory for playback
  - Write a simple device driver to scan kernel memory for keys and check!
- Called "Trusted client" problem
- Need "trusted computing platform" that only lets validated sw to run (not, the dd above!)
- But PC is not such a platform

 With "Trusted Boot" PC. May be possible.

However: "Against the average user, anything works. Against the skilled attacker, nothing works." B. Schneier

Now, based on that particular problem this blue ray people tried to avoid this problem and they called it advanced access control system that it turns out this problem was broken for a slight different reason which some of you must already know by now basically whatever you do; however, level; however, many levels of encryption I do the keys needed to obtain and unencrypted content is available somewhere in memory for playback.

Finally, you are using some computer system; even it is an embedded system its memory somewhere; that means, that you write a simple device driver it scans the memory kernel memory and then tries it out. So, you make sure that your embedded system is small enough its only what let us say not one terabyte of memory make it sure it is got the most let us say let us say 4 mega bytes in a memory 16 megabytes in a memory make it 250 megabytes on a memory and everything you check every single I do not know whatever 40 bit or 64 bit whatever check it and see if you can decrypt it.

So, this is much smaller than the previous system because here you wanted 2 to the power of 40 one trillion possibility here if you take reasonable amount of memory and try to on the embedded system nowadays you can open embedded system and put your

probes or oscilloscopes in designing you can do all those things just needed determined guy ok and look at it and see and then see all the things and figure out which part of it somewhere set in a memory is the actual key that is being used for encrypting or decrypting essentially you have to write it device driver and why it is what is the problem here is basically because it is your stuff is running on your device and your device you can always open it up if you are clever enough put a probes and figure it out there is people who are good at this game. So, we are done all this ok.

So, essentially the only way to solve this problem is by using what is called a trusted computing platform what is that essentially a trusted computing platform will only run trusted software means those things that I may certify it to be safe for example, it cannot be a virus I do not want virus to be executing that because virus can be anything arbitrary right. So, a trusted computing platform will avoid allowing any software that does not match certain criteria example being hashed for example, if you are running a compiler it might have hashed will check whether that executable you have has a hash which is same as what is to be allowed if it is allowable then it is allowable otherwise do not do it. So, any random virus will not it hash will not match.

So, it will throw it out of course, this is very difficult to design and make it work there are lot of kind of attacks just like your simple device driver attack right there you just can scan the memory it turns out even if you build a trusted computing platform there is some place that is a bus somewhere you know computer architecture there is some bus etcetera you can tap the bus and you can do various things.

So, it is a very difficult problem and the best you can do is you can do all this stuff against the average user all this things are quiet fine the minute you go against the skilled attacker the guy can essentially get whatever he wants essentially it has just kind of amount of time. So, essentially it turns out that the PC is not a trusted computing platform and nowadays there is some effort moving in the direction there is something called trusted boot you might have heard about windows 8 is coming with trusted boot.

So, because there windows platform has been widely attacked through viruses and what not malware part there are so many 20; 20 years plus. So, finally, windows 8 is going to have what is called trusted boot. So, that viruses cannot easily get in; that means, that only those software only those executable where the particular hash right will be

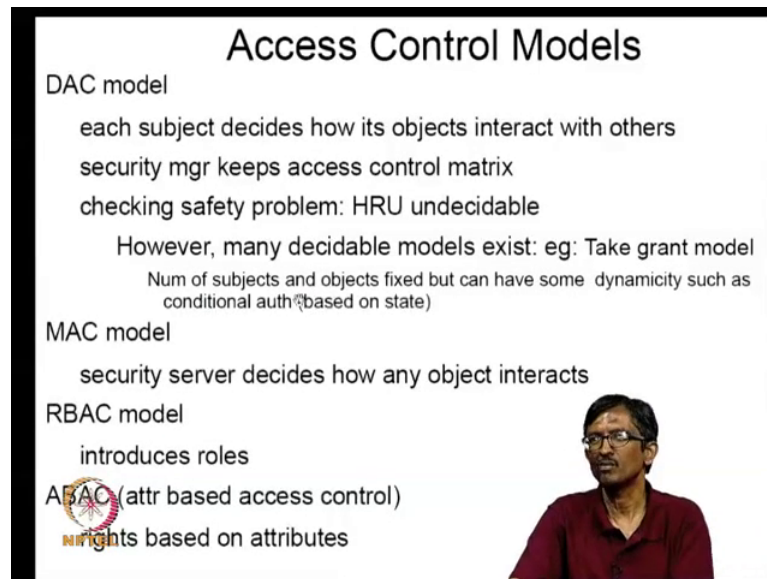


allowed; that means, that there is a chip inside which is as tamper proof storage, we discussed already tampered proof storage basically that kind of storage which is sitting on its chip even if somebody is crapes it off and tries to look into it will destroy the information that is how it is designed is called tamper proof storage. So, somebody will have tamper proof storage at a chip level and it will have some of the hashes, it will be allowed and these hashes can be inserted are taken out by some specific mechanisms.

So, if you have that kind of platform, then you can essentially avoid this kind of problems, but currently there are very platforms which have this property windows 8 and newer generation devices might have this property and typically mobiles also if you want to do what is called mobile commerce right you want to because its finally, again your mobile is in your hands for a clever guy and the bank or organization is keeping information about your transaction in the mobile; let us say then for a clever guy you can go and change that thing right; that means, again you need to have what is called tamper proof storage whatever protocols are to be there. So, that when you modify something somebody is alerted that yes the guy has something wrong information is lost at least I can say that somebody is back ok.

So, there are various attempts to get a trusted computing platform, but they have not yet they are not at widespread now, but probably the next year also it will start happening because right now windows 8 is coming I do not know how successful it will be, but that one has got some trusted boot aspects. So, that might have a trusted computing platform in which case it will be, but generally when all this systems are being designed they could not provide this trusted computing platform therefore, I am also has been hacked it also basically by using this device driver hack you can basically figure out what is going on.

(Refer Slide Time: 41:32)



**Access Control Models**

**DAC model**  
each subject decides how its objects interact with others  
security mgr keeps access control matrix  
checking safety problem: HRU undecidable  
However, many decidable models exist: eg: Take grant model  
Num of subjects and objects fixed but can have some dynamicity such as conditional auth (based on state)

**MAC model**  
security server decides how any object interacts

**RBAC model**  
introduces roles

**ABAC (attr based access control)**  
rights based on attributes

I think I will just you already mentioned a bit particularly just take a look at some access control models in a discretionary access control models each subject decides how its object interact with others security manager keeps access control matrix, it turns out there are theoretical problems, here also it turns out even in discretionary access model checking what is called safety problem; what is I give certain privileges does it is it possible for some other person to get more privileges than what is allowed what is called the confinement problem it turns out even this is not decidable.

So, I think I want to there is also called mandatory access control model the security server in every instance will decide how any objects interacts there is also something called role base access control model; that means, that just like in Unix you can be part of a role you can be a system administrator right. So, the thing is here you are not going by a user functions you are going by roles depending on role you are allowed certain things this is quiet common in administration; if you are a vice principle; you get certain you can look at certain files can sign certain things if you are not vice principle you just cannot do it which is basically role based access.

So, there is also a something called attribute based access control the rights are based on attributes; for example, if somebody might say the person who has a let us say who has a in a bank; for example, they might say that if you have more than certain amount of money bank with them they get certain rights it is an attribute you have more than you

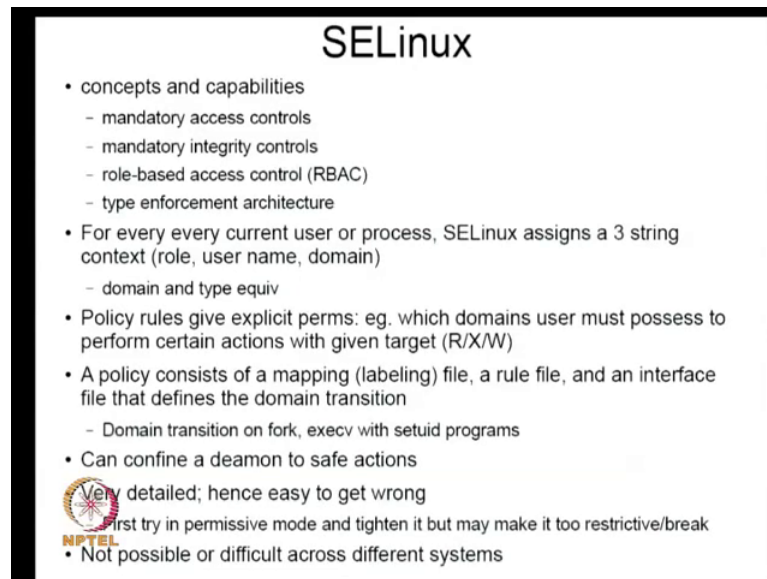
deposited more than 75 lakh rupees he became a privileged customer and you get. So, it is in a basically based on attributes its not based on user id it is not based on roles etcetera just purely on some attributes it can be any complex attribute or for example, you might say instructor can say the guy who has scored consistently let us say S grade can take off without informing where as the guy who has having problems with the course if he has to take a leave has to check with me before that also is possible things of you basically have you get rights based on attributes that also possible.

So, it turns out most file systems use DAC and if you go beyond DAC you became information you are talking about information systems larger and larger data is aggregated and you start having larger systems in place typically it is often times role based access control and sometimes MAC. MAC is tough to implement because it requires that there is a check in spite of wherever the information is flowing somebody is checking it all the time and typically its used in military guys military guys use it a lot the military guys worry about what happens for example, somebody has a switch to detonate a bomb right you and if you assign the privilege to somebody else you want to make sure that that guy does not give to somebody else right these are all very delicate things you want to keep complete control what happens.

So, typically MAC is for highly organized systems where something very critical and where you cannot trust everybody there is very great and danger can happen if you allow discretionary access to things this kind of things. So, most file systems are in the DAC model there are some file systems can give MAC and with Linux with ACLinux you can get MAC and role based access control its already with the Unix a bit, Linux a bit because you have this notion of groups and attribute based control is not that common its available usually at much higher level subtraction.


The other systems like MAC and fewer things they use also called ACLs access control lists where it is not based on strictly on user or group or everybody else you can actually have much finer grained saying that for this particular, if I want allow this percent that percent that percent for some or something else this percent that percent I can specify exactly who should see what or write what etcetera there are this access ACL models lot more granular way of specifying this things. So, some file systems provide this kind of things.

(Refer Slide Time: 46:06)



## SELinux

- concepts and capabilities
  - mandatory access controls
  - mandatory integrity controls
  - role-based access control (RBAC)
  - type enforcement architecture
- For every every current user or process, SELinux assigns a 3 string context (role, user name, domain)
  - domain and type equiv
- Policy rules give explicit perms: eg. which domains user must possess to perform certain actions with given target (R/X/W)
- A policy consists of a mapping (labeling) file, a rule file, and an interface file that defines the domain transition
  - Domain transition on fork, execv with setuid programs
- Can confine a daemon to safe actions

 Very detailed; hence easy to get wrong  
first try in permissive mode and tighten it but may make it too restrictive/break

- Not possible or difficult across different systems

So, SELinux is one model in Linux, I will briefly mention what all it can do it is a fairly complex topic;so I just mention a few things. So, it has got essentially manses mandatory access controls MAC, it also has got similar to access control integrity controls who can write one who can modify what and this is going to be transitive, I give it to some permission to somebody, I am even if I given a write permission if I give to somebody else I still want to know whether other party also can write, I will check it; I will not let the certain person on a give the person to whom I gave it he does not have the complete control over it. So, basically it also has a role based access control RBAC and type enforcement architecture basically there are you can have what is called at domain and a particular domain only certain things are possible it is something like a type system in a type system.

For example, if it is let us say it is a list for example, right you only have you alone use certain operations let us say we have an abstract data type and you say that I can add an element remove an element right create a list right delete the complete list or some such thing I will not allow arbitrary manipulation because there is a certain semantic property of a list and if I allow arbitrary operations it might break the semantics of the list right. So, only certain interface are provided that is basically an abstract data type and essentially what we they want to do they want to do is something similar to that they want to have a type enforcement architecture.

So, that only certain types of operations are allowed just like an abstract data type. So, what does SELinux have it has got a role username and a domain role with basically essentially a RBAC username of course, it has to do it for mandatory access control, etcetera and then domain basically it tells you what kind of operations you can execute. So, a user for example, or role might have multiple domains 3 domains and he can whatever those domains allow him to do he can do not anything else.

So, for example, there are policy rules that you have to specify which domains user must possess to perform certain actions with given target it could be read execute write etcetera again this things are done at the file system level. So, basically the way this c Linux works is you have a regular file system and you have certain additional capabilities and functions by which you can label the files for example, you might say that a particular domain can access slash bin or certain program can access certain demon for example, you can access slash war slash log something. So, I basically say to specify in your policy rules which domain can access what where. So, and it is going to be checked every single time, if it is going to be not satisfied then it will be the access will not be allowed.

So, essentially you give policy rules you also specify an overall policy which tells you how to label the file rule file and interface file that defines the domain transition. So, let us just look at domain transition; for example, on fork and exec right if you are if you are familiar with Unix a fork can exec right basically what happens I can fork a program with different permissions than what I started with right the child can inherit certain privileges which are the same as the parent, but in certain interesting cases set UID programs right it turns out that you can high higher privileges also basically there are some special programs which only a root or will somebody install in particular place and if you execute it you get privilege escalations you get higher privileges right.

So, basically; so, when you do a domain when you do a fork; for example, you start at the particular domain and you create a new domain right the question is what domain transitions are legitimate you have to specify those things also. So, if you specify those transition then you will it will check the run time whether you are new domains that you have created or the new essentially demons for example, often times you are creating new demons. So, want to check if the new demons have this what has been permitted you check those things only if they are permitted; then you otherwise you set no ok.

So, essentially what you can do is with these kind of policies you can ensure that I made a mistake here I think d e m o n; the demon is can only be allowed to safe actions what are the kinds of demons I am talking about for example, you can be h t t p d is a good example you are running h t t p demon and you want to ensure that it does not go and look at my files and send it somewhere else. So, you can confine it your particular place and therefore, it can avoid instead of this key actions the basic problem with these c Linux is very detailed you have to specify everything if there is a if there is a particular program the effect happens to there to slash time you have to give it to permission direct to temp slash temp also that give every single possible thing that needs to get its job done.

So, normally in our day today life we take certain things for granted right we can do certain things you do not have to specify everything, but in SELinux whether you can walk on the street whether you can go on that street or with this street or you can take a metro everything has to specified every single thing has specified if you miss one of them then cannot take the metros in some particular location it just does not work. So, that kind of problems is there in SELinux that is the reason why most people do what they do is they have what is called a very permissive mode what does it mean you just say that by definition it give you more privileges and necessary and then once you make sure that its sort of working you try to tighten the rules, but there is a problem with this you can either over tighten it or you are still running the program at a time when too many permissive rules are available.

So, these are very hard problem the reason why it is hard problems because this if you take a program whether it goes to a particular place or not is a what is called undesirable problems right it is equivalent of halting problem right because if I say that it goes to a particular place right; that means, that the intervening codes will not stop will not halt right if I saying that it goes to a particular place; that means, that I am able to say that this particular piece of code does not halt and you know that from theory that the halting problem is unsolvable; that means, that if you give me a program statically I cannot figure out what all the per minimum amount of permission it needs it may be that when I am branching that particular direction.

It requires certain kinds of permissions when it is back in the some other direction it requires some other set of permissions and if I give permissions for all the parts, then it is

essentially what is happening is I am being extremely permissive I am giving everything that that has to be that it principle of touch right is too permissive because. So, it turns out that it is very difficult to give it the right side of the permissions and. So, this has typical problem in any large system ok.

SELinux is not unusual in this that is quiet common in almost all the systems other problem the SELinux is that it is different systems may or may not support c Linux may or may not support the same kind of model now how do you beat across because finally, when you talking about large storage system they might be lot of servers and they might be lot of clients each may or may not have the same model. So, this is a bit of a problem. So, I am just giving some hint about the kind of problems that are there we are not really looked at when storage is realized through multiple nodes multiple distributed level which will be started to talking soon. So, there are lot of protocols involved in this. So, when 2 for example, suppose I want to ensure that there are at least 3 copies of particular file right in spite of failures then there is there has to be a protocol between various entities in the system.

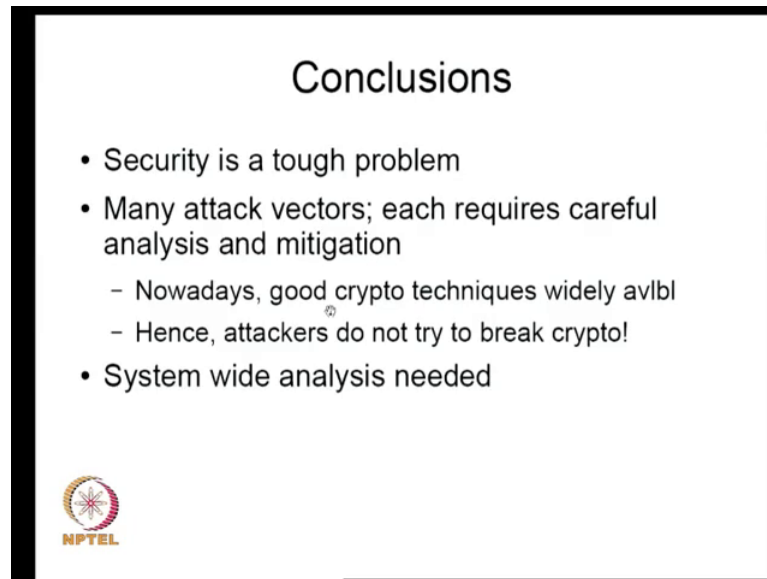
So, that at least 3 copies are there and this protocol will actually go over when it work links and you have to ensure that somehow these links are safe from sub attach safe from let us say malicious tampering etcetera. So, the protocols has to also be secure and. So, there are many many things that have to be worked out. So, that all those things worked together which is slightly non trivial and. So, it is very complex problem; it turns out somewhere some attempts has been made; we will talk about it bit later that something called; come for example, basically using IP sec kind of model we can essentially access storage devices across the internet and basically you will be protected through the IP sec whatever IP sec does for IPV 6 or even IPV 4; the same thing can be used here to protect yourself. So, this or some other things that also you can do.

So, I think I just want to com I just wanted to give a flavor of the kind of kind of permissions that are there in security for storage systems. So, it is a fair intricate problem because the there are many attack vectors as you saw it in the case of in the case of blue ray there was in the case of CSS for DVDs; the encryption mechanism was broken. So, they figured out the way to crack it in the case of blue ray that was fixed a bit properly. So, that route did not take they decided to go and see what is there in memory. So, this is a problem with any system which is trying to give you the device in your hands because

finally, you can do what you want to do with a device once you brought it. So, you can essentially use some brute force or other clever technique by which you can look into it and get out the secrets.


So, and basically what is happening is that cryptography.

(Refer Slide Time: 57:05)



**Conclusions**

- Security is a tough problem
- Many attack vectors; each requires careful analysis and mitigation
  - Nowadays, good crypto techniques widely available
  - Hence, attackers do not try to break crypto!
- System wide analysis needed



Cryptography mechanisms have been widely used and they are reason you could mathematician and people have been worked on it quiet a bit now crypto techniques have quiet good the real problem is because they are so good; nobody wants to crack a system will try to crack the cryptography; they will crack the more easier part of it. So, nowadays; cryptography is no longer an important issue for most of the because the SIM that nobody is trying to crack it through the route once upon a time when cryptography was between people were saying that I will figure out a way of cracking cryptography and data answer out, but now with AES and other kinds of cryptographic routines this extreme difficult to crack cryptography.

So, the standard technique is to bulk not work on the cryptographic error if somebody is doing cryptography there is I will not touch that part I will go and crack it outside of the cryptographic domain and figure out; the way to do it. So, again for doing that unit system about analysis again storage systems solve the same system; same problem, we could be having problems the network level at the device level at the protocol level; you



have to be a comprehensive kind of models for all of these things then you probably find some may be you might find some attack with these 2 both strengthen the system or crack the system you have to do all of it; otherwise, it is not possible I think I just wanted to give a flavor of what is security for storage systems.

I think we will stop here today and probably continue with another short lecture on reliability as another thing that storage systems have to worry about.