Jae Duk Seo – CPS 633 – Textbook questions solve!

# Chapter 1 – Introduction

# Chapter 2 - Toolbox: Authentication, Access Control, and Cryptography

**1. Describe each of the following four kinds of access control mechanisms in**

**terms of   
(a) ease of determining authorized access during execution,   
(b) ease of adding access for a new subject,   
(c) ease of deleting access by a subject, and   
(d) ease of creating a new object to which all subjects by default have access.**

**• per-subject access control list (that is, one list for each subject tells**

**all the objects to which that subject has access)**

**• per-object access control list (that is, one list for each object tells all**

**the subjects who have access to that object)**

**• access control matrix**

**• capability**

**Access Control List = per-object access control list  
Capability Lists = per-subject access control list  
Access Control Matrix = The whole matrix (LARGE) that contains all of the information regarding both in the view of the subject and the object.   
Capability = this is the system where the user is given a token in which determines what kind of authority they have. This Token is managed by the OS so this means that it is encrypted. (Token mechanism)**

* Per subject list contains the information for one user, regarding what object they can access. Naturally this is easy to determine authorized access during execution. Since all of the object that the user access in this list will have the associated authority to that particular user. This is cannot add a new subject, since one list is connected to one user respect, adding another user will mean creating a new list entirely.

**2. Suppose a per-subject access control list is used. Deleting an object in such a**

**system is inconvenient because all changes must be made to the control lists of**

**all subjects who did have access to the object. Suggest an alternative, less costly**

**means of handling deletion.**

* One alternative would be to use the per – object list. This list is a list where it contains all of the user who have access to this object. Deleting an object is super easy, since we only can delete that list for one respected object.

**3. File access control relates largely to the secrecy (the action of keeping something secret or the state of being kept secret) dimension of security. What**

**is the relationship between an access control matrix and the integrity of the**

**objects to which access is being controlled?**

* The access control matrix – is a matrix that is created in the point of both the subject and the object. The subject is row (vertical) and the object is the column (horizontal). This is a LARGE matrix in which it contains all of the information regarding all of the user! The relationship between the ACM and the integrity of a certain object is only specified users can modify the object with the associated permission. (Read/Write/Execute – Linux)

**4. One feature of a capability-based protection system is the ability of one**

**process to transfer a copy of a capability to another process. Describe a situation**

**in which one process should be able to transfer a capability to another.**

* When we are creating another admin user for the system. The original admin user must give full access to the system to the newly created user. In situations like this process of transferring copy of a capability would be helpful.   
  For the process example – imagine where one process is using a printer. And when another A wants to use printer – The os raises a printer Token to that process. This is a transfer of capability.

**5. Suggest an efficient scheme for maintaining a per-user protection scheme.**

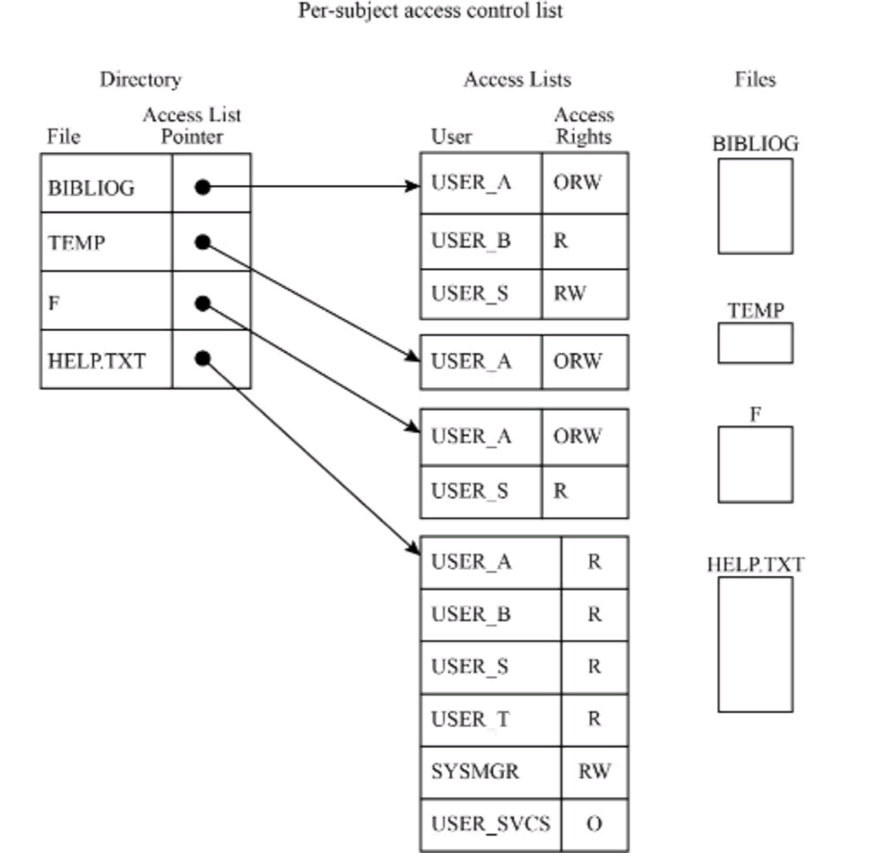
**That is, the system maintains one directory per user, and that directory lists all**

**the objects to which the user is allowed access. Your design should address the**

**needs of a system with 1000 users, of whom no more than 20 are active at any**

**time. Each user has an average of 200 permitted objects; there are 50,000 total**

**objects in the system.**

* Per user protection scheme – this sounds like – per subject access control list. (where one row is made for specific user that contains all of the object in which that user have access to, and the level of that authority.)  
    
  For this system to be efficient I believe that we need a list full of overlapping objects that multiple subject can access. This means that many subjects are going to use multiple overlapping object – so the row will not over populate with different object names and the associated authority to that object.   
    
  The image below describes the situation well.    
  One row contains all of the object and that contains the pointer to the per object access control list – in which contains all of the info regarding the user who is accessing the object at any given moment of time!

**6. Calculate the timing of password-guessing attacks:**

**(a) If passwords are three uppercase alphabetic characters long, how much**

**time would it take to determine a particular password, assuming that testing**

**an individual password requires 5 seconds? How much time if testing**

**requires 0.001 seconds?**

**(b) Argue for a particular amount of time as the starting point for “secure.”**

**That is, suppose an attacker plans to use a brute-force attack to determine a**

**password. For what value of x (the total amount of time to try as many**

**passwords as necessary) would the attacker find this attack prohibitively**

**long?**

**(c) If the cutoff between “insecure” and “secure” were x amount of time,**

**how long would a secure password have to be? State and justify your**

**assumptions regarding the character set from which the password is**

**selected and the amount of time required to test a single password.**



**7. Design a protocol by which two mutually suspicious parties can authenticate**

**each other. Your protocol should be usable the first time these parties try to**

**authenticate each other.**

* Challenge and respond module will not work since, both parties must agree on a function before the first transaction.   
    
  LOL do not fucking know the answer so I will assume that the client and the server can communicate with one another at the first contact, this means that creating a user name and a password is allowed! – if this assumption is allowed then a traditional password protocol can make the authentication for both parties.   
    
  LMFAO – we can use the protocol called Kerberos – this is the default protocol in the windows based system. The Kerberos protocol is one of the capability system. This means that it generates a ticket for each user – and this ticket basically authenticates the user, for finite period of time!   
    
  How it works – when the client makes a request to the server the server   
  1) Client submits an authentication pattern to the server   
  2) Server extracts a pattern from the submitted pattern from each user.   
  3) The server then generates a ticket based of that pattern and returns it back to the user!  
    
  By the Kerberos protocol above it is possible for even a first time user to authenticate to the server.

**8. List three reasons people might be reluctant to use biometrics for**

**authentication. Can you think of ways to counter those objections?**

* One of the main reason can be exposer of personal information. This can be a problem for some people since they will care about their personal information.   
    
  To counter this flaw – the system must use a system such as Kerberos authentication – which the pattern part of the authentication will be the features of the biometric info. And promise the client that the personal info will be discarded right after the client have done his job.

**9. False positive and false negative rates can be adjusted, and they are often**

**complementary: Lowering one raises the other. List two situations in which false**

**negatives are significantly more serious than false positives**.

* False positive – when the system gives access to user who is not the proper person to give authentication.
* False negative – when the system DO NOT give access to the proper person who have the right to access an object.   
    
  In a urgent situation false negative are a hassle to deal with. Since the user must access some object in a hurry. Those times it is better to give the permission rather than hiding it.   
    
  1) Stock market information access – the people who make decision for stock prices, need the info fast as possible – better to give them access.   
    
  2) A person have to access the ATM to get money out of it urgently, but if the system treats his authentication as false negative the situation will not look good for the person.

**10. In a typical office, biometric authentication might be used to control access to**

**employees and registered visitors only. We know the system will have some false**

**negatives, some employees falsely denied access, so we need a human override,**

**someone who can examine the employee and allow access in spite of the failed**

**authentication. Thus, we need a human guard at the door to handle problems, as well**

**as the authentication device; without biometrics we would have had just the guard.**

**Consequently, we have the same number of personnel with or without biometrics,**

**plus we have the added cost to acquire and maintain the biometrics system. Explain**

**the security advantage in this situation that justifies the extra expense.**

* In the situation above, since the accuracy of authorized person access the building or the object increases. This is a reason why it is better to pay more for both of the security guard and the biometric system.
* Yup the accuracy will increase – since biometric system is more robust to human error and the human can take care of the situation when the biometric system fails.

**11. Outline the design of an authentication scheme that “learns.” The authentication**

**scheme would start with certain primitive information about a user, such as name and**

**password. As the use of the computing system continued, the authentication system**

**would gather such information as commonly used programming languages; dates,**

**times, and lengths of computing sessions; and use of distinctive resources. The**

**authentication challenges would become more individualized as the system learned**

**more information about the user.**

**• Your design should include a list of many pieces of information**

**about a user that the system could collect. It is permissible for the**

**system to ask an authenticated user for certain additional information,**

**such as a favorite book, to use in subsequent challenges.**

**• Your design should also consider the problem of presenting and**

**validating these challenges: Does the would-be user answer a truefalse**

**or a multiple-choice question? Does the system interpret natural**

**language prose?**



**12. How are passwords stored on your personal computer?**

**-** For the system like linux, the password for users are stored as a hashed value. With some of salt values added on to the original password!

**16. Defeating authentication follows the method–opportunity–motive paradigm**

**described in Chapter 1. Discuss how these three factors apply to an attack on**

**authentication.**

* Method is the knowhow, on how to actually hack the system. Opportunity is when the perform the knowhow to hack the system. And finally, motivation is the start point of everything. Without the motivation – the hacker will not even start the attack using their knowhow, even if a good opportunity comes by.   
    
    
  **Authentication is the step before access is granted to some sensitive resource. Thus,**
* **the attractive resource provides motive for wanting to defeat authentication. Method**
* **entails skills and knowhow: Passwords are of some finite length from a finite**
* **alphabet, so in theory all passwords can be enumerated (although the process takes**
* **a long time). For technology, used with biometrics and tokens, design specifications**
* **and usage manuals are often widely available, so the attacker can obtain details with**
* **which to attack. Finally, opportunity translates into time and physical access, which**
* **may be the controlling factors in an authentication attack**

**17. Suggest a source of some very long unpredictable numbers. Your source must be**

**something that both the sender and receiver can readily access but that is not obvious**

**to outsiders and not transmitted directly from sender to receiver.**

* Man in the middle attack! – For example when a client who is using a web application, submits a form, this attack can happen. The attacker might want to perform a buffer overflow attack to the server. So they will sniff the original input from the form, and the resubmit a very long numeric value to cause harm to the program.
* Shared books, messages from broadcasts, content of (static) web pages.

**18. What are the risks of having the United States government select a cryptosystem**

**for widespread commercial use (both inside and outside the United States). How**

**could users from outside the United States overcome some or all of these risks?**

* Do we really need to do this question? Possible answer – Fucking NSA can be hacked or something. I really have no clue.

**19. If the useful life of DES was about 20 years (1977–1999), how long do you**

**predict the useful life of AES will be? Justify your answer.**

* **Because of Demorgan law the computers are getting faster and faster – in this way. It would only take about 5 years for the AES to be cracked since the distributed computing system wil play another part for cracking the code.**

**20. Humans are said to be the weakest link in any security system. Give an example**

**for each of the following:**

**(a) a situation in which human failure could lead to a compromise of**

**encrypted data**

**(b) a situation in which human failure could lead to a compromise of**

**identification and authentication**

**(c) a situation in which human failure could lead to a compromise of access**

**control**

* **A) Social engineering – when a person who has the password tells another person about the password that could be one situation where an encrypted data can be compromise.  
  Or the person who have access to rainbow table forgets to log off from the database, and anyone can access that rainbow table.   
  B) When someone write a user name and password on a note pad.  
  C) In a capability list situation – one user might give access to a file to another user – who does not have the right to read a particular file.**

**21. Why do cryptologists recommend changing the encryption key from time to**

**time? Is it the same reason security experts recommend changing a password from**

**time to time? How can one determine how frequently to change keys or passwords?**

* **Since different key encrypt the given data differently. Yes, it is a similar reason why they recommend to change the password from time to time. One can know when you have changed a key, when they give an input of which they already know the encrypted value is known. When the final result is different from what they already know/have then they can conclude that the key have been changed.**

Periodically changing the Reporting Services encryption key is a security best practice. A recommended time to change the key is immediately following a major version upgrade of Reporting Services. Changing the key after an upgrade minimizes additional service interruption caused by changing the Reporting Services encryption key outside of the upgrade cycle.

Classic cryptographic protocols based on user-chosen keys allow an attacker to mount password-guessing attacks. A combination of asymmetric (public-key) and symmetric (secret-key) cryptography that allow two parties sharing a common password to exchange confidential and authenticated information over an insecure network is introduced. In particular, a protocol relying on the counter-intuitive motion of using a secret key to encrypt a public key is presented. Such protocols are secure against active attacks, and have the property that the password is protected against offline dictionary attacks.

When you change your password every few months, it limits how long a stolen password is useful to a stealthy attacker—how long he/she has access to your account. If someone steals your password and you don't know about it, the attacker could eavesdrop for an unlimited time and glean all sorts of information about you or do other damage.

Therefore, for decades, many security guidelines have recommended frequent password changes, usually between 30 and 180 days. Windows Server has a default of 42 days.

**22. Explain why hash collisions occur. That is, why must there always be two**

**different plaintexts that have the same hash value?**

* **Weak hash collision is when the attacker have the m’ value and that m’ when hashed have the same result as h(m) – Strong is when the attacker does not have anything – but sill h(m) = h(m’).  
    
  This situation occurs because of limitation of the hash function. The alphabet in the English language is limited, when using a simple substitution encryption the probability of alphabet being distributed is finite. So the collision will happen.**
* Hashing turns something (usually a key or a password) into a (usually fixed length string of characters. So for example, your hashing algorithm might always produce a string that is 8 bytes long. So as with encryption you transform something intelligible into something unintelligible. One might call the product of the hashing the hash codes of the hash sums.
* One difference is that hashing two different messages might produce the same hash values. So you cannot *decrypt* something that you have hashed. But even though you can't decrypt a hashed value, it is (or it should) in general hard to make two messages that have the same hash value. Another difference is that hashing doesn't require a key.
* The security property of a cryptographic hash function is called *collision resistance*. Collision resistance means, that an attacker given access to the hash function should be unable to produce two messages m0, m1 such that H(m0)=H(m1). (This also implies a few other properties often defined separately, such as first and second preimage resistance.)
* **How can an attacker exploit this?**
* Lets see how an attacker will exploit the collision vulnerability in the hashing functions. An attacker typically begins by constructing two messages with the same hash value where one message appears legal. For example when an attacker, ABC, discovers that the message “I,XYZ, agree to pay ABC $5000.00 on 15/03/2016.” has the same hash as “I,XYZ, agree to pay ABC $50000.00 on 19/09/2016.”,then he can try to get the victim,XYZ, to digitally sign the first message. The attacker, ABC, can then claim that XYZ actually signed the second message, and ‘prove’ his claim by showing that signature matches the second message.
* Note, however, that hash functions **do** have collisions. A large set is mapped onto a smaller set, so collisions are bound to happen. However, an attacker should not be able to **find** such collisions. (Which is also why we actually need hash function families to define security)

**23. What property of a hash function means that collisions are not a security**

**problem? That is, why can an attacker not capitalize on collisions and change the**

**underlying plaintext to another form whose value collides with the hash value of the**

**original plaintext?**

# Chap 3 - Programs and Programming