Introduction to Information Security
—— Authentication and Access Control

Dr. Tianlei HU

Associate Professor

College of Computer Science, Zhejiang Univ.

htl@zju.edu.cn

Outline

- Authentication
- Authorization

Authentication

Who are you?

Concepts & Realize of Authentication

- Authentication—— you are who you say you are:
  - Who are you?
  - How to prove that you are the one you declared to others or computer systems?
  - Any system that needs access control must solve this problem first.
- How to authenticate?
  - What do you have? ID card, passport, key, smart card, USB card, mobile phone
  - What do you know? Password, birthday, ID, answers to presetting questions
  - where are you? IP address
  - what are you?
    - Biometrics characteristics: fingerprint, palm print, iris, facial contours, DNA, etc.
    - Behavioral characteristics: handwriting, vocal print, stroke, walking, etc.
- We will mainly discuss Authentication services like below:
  - Authentication Technology: Password and Biometric Authentication
  - Authentication Protocol: Challenge-Response Authentication and KERBEROS

Password-Based Authentication

Most widely used:

The system keeps a file of Authenticated users and passwords.

The user has a secret password.

The system checks it to authenticate the user.

- Vulnerable to eavesdropping when the password is communicated from user to system

How is the password stored?

- The password file is difficult to keep secret

How easy is it to guess the password?

- Easy-to-remember passwords are easy to guess

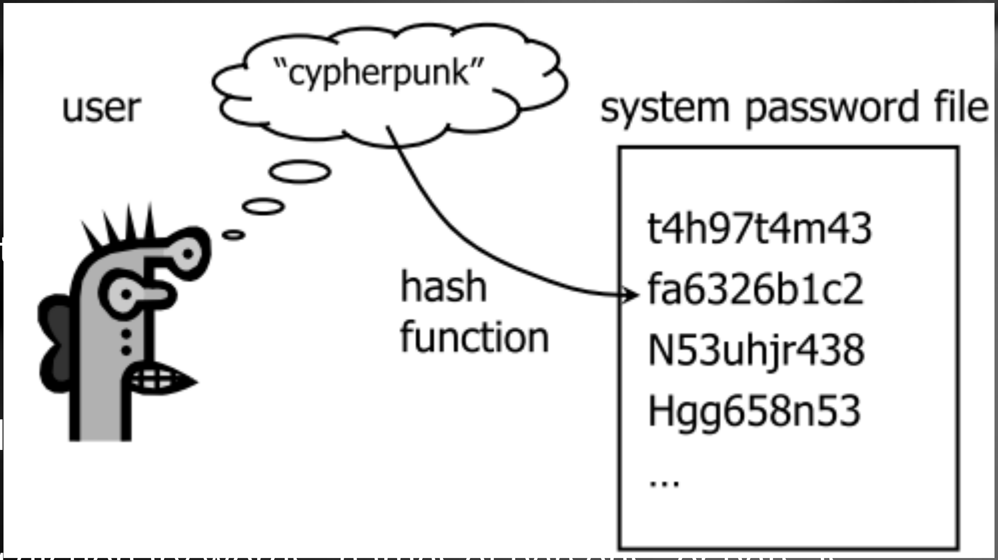
How does the system check the password?

Password-Based Authentication

- **Factors that affect the safety of a password system:**
  - Limit of frequency/times the attacker input password
  - Storage of password （plaintext/Hash）
  - Contents transferred on the Internet （plaintext/hash, etc.）
  - Process of modifying password （How to authenticate the user when modifying password? ）
  - Lifecycle of password （How often to modify password?）
  - **Other factors:**
    - Don’t show password on the screen （instead, show ’*’）
    - Define the length and format of the password
    - Ask for reentering password for a while
    - Forbidden the logging if many times wrong in a while

Implementation of Password —— Early

- Prehistoric Password: Plaintext password
  - 2011 CSDN password leak——The shame of Chinese programmer
- Early Stage Password: Password hashing
  - Don't' store passwd, but store H(passwd)
  - When verifying password, compare H(p') with H(p) stored in the system
    - Don't store real passwords!
  - H(.) —— One-Way Hash Function
- Early Unix Password:
  - Use DES as One-Way Hash Function:
    - Encrypt a NUL, and cut the password t
    - Artificial reduction: run DES 25 times!
  - Existing Problems:
    - 52 characters, 10 numbers, 32 symbol
      - $94^8 \approx 6 * 10^{15}$ possible passwords
    - However, people are used to choosing dictionary words, names of persons, or pets as passwords, which have only 1 million…
    - Unix system originally stored users' names and passwords in file /etc/passwd, which all can read!



Traditional Dictionary Attack

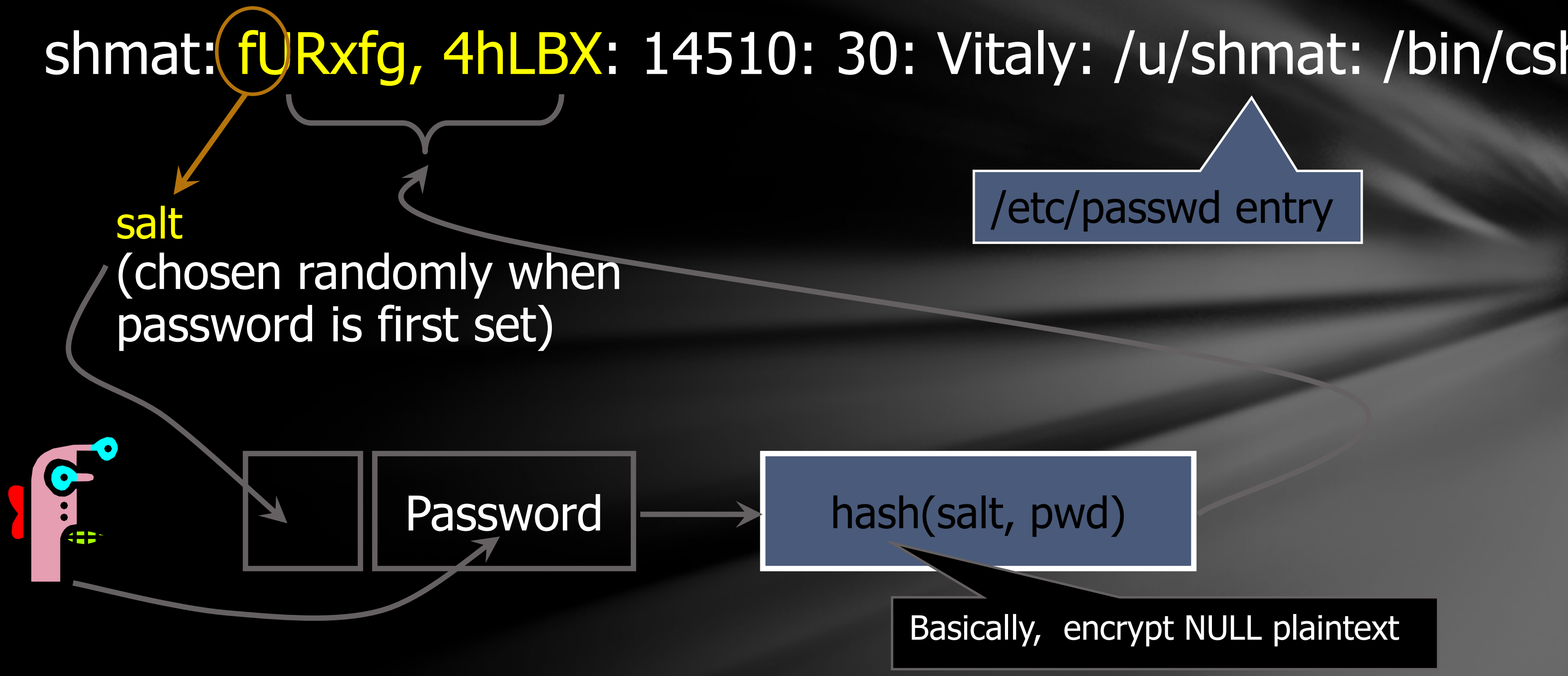
Password file /etc/passwd is world-readable

- It contains user IDs and group IDs which are used by many system programs

Dictionary attack is possible because many passwords come from a small dictionary

- An attacker can compute H(word) for every word in the dictionary and see if the result is in the password file
- With a 1, 000, 000-word dictionary and assuming 10 guesses per second, a brute-force online attack takes 50, 000 seconds (14 hours) on average
    - This is very conservative. The offline attack is much faster!

Upgrading Phase 1: Salting


shmat: fURxfg, 4hLBX: 14510: 30: Vitaly: /u/shmat: /bin/cs
salt
(chosen randomly when password is first set)
/etc/passwd entry
Password
hash(salt, pwd)
Basically, encrypt NULL plaintext

- Users with the same password have <u>different</u> entries in the password file
- Dictionary attack is still possible!

Advantages of Salting

Without salt, an attacker can pre-compute hashes of all dictionary words once for all password entries

- Same hash function on all UNIX machines
- Same passwords hash to the same values; one table of hash values can be used for all password files.

With salt, the attacker must compute hashes of all dictionary words once for each password entry.

- With 12-bit random salt, the same password can hash to 2^{12} different hash values.
- The attacker must try all dictionary words for each salt value in the password file.

Upgrading Phase 2: Shadow Passwords

shmat: x: 14510: 30: Vitaly: /u/shmat: /bin/csh

/etc/passwd entry

Hashed password is not
stored in a world-readable file

- Store hashed passwords in the /etc/shadow file, which is only readable by the system administrator (root)
- Add expiration dates for passwords

Password - Ten Common Mistakes

- Leaving passwords blank or unchanged from the default value.
- Using the letters p-a-s-s-w-o-r-d as the password.
- Using a favorite movie star’s name as the password.
- Using your girlfriend/boyfriend’s name as the password.
- Using the same password for everything.
- Writing passwords on post-it notes.
- Pasting a list of passwords under the keyboard.
- Storing all passwords in an Excel spreadsheet on a PDA or inserting passwords into a Rolodex.
- Writing all passwords in a personal diary/notebook.
- Giving the password to someone who claims to be the system administrator.

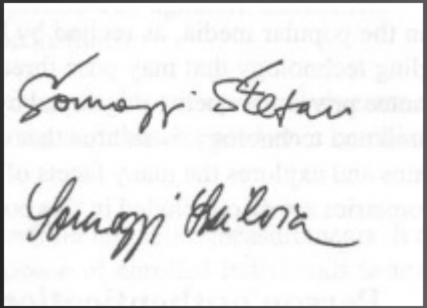
Password Guessing and Protection

- Password Guessing:
  - Try the default password, short password(1~3 characters), and all words in the electronic dictionary.
  - Try the user's information, for example, phone number, ID, house number, etc.
  - Listen to the keyboard or the link of communication with Trojan.
  - The social engineering approach.
- Password choosing and Protection:
  - Password safety evaluation and check: character + number, or random password
  - Trojan protection, using a safe logging tool.
  - Different sites with different passwords: one for inessential service, one for privacy related, and one for money-related.
  - Don't believe in any third-party software(recording passwords in that software).
    - If you record passwords in that software, once connected to the network, your passwords "can and are very likely to" be known by the company that distributes the software.
    - Judge by yourself which company and software you can believe. (such as Microsoft, Apple, open-source software, etc.)
  - Don't log into important services in any unreliable place.
    - Most QQ numbers are stolen in the cybercafé.
    - Password can be stolen by attacking and manipulating the DNS.

Biometric/Behaviometric Authentication

- Biological Examples
  - Fingerprint, Iris, Retina, Face, & Hand Recognition
- Behavioral Examples
  - Handwriting, Gait, Typing Rhythm, Mouse Gesture Recognition

- Advantages:
  - It can't be stolen, lost, or forgotten.
- Disadvantages:
  - Cost of equipment, installation, and maintain
  - Accuracy of comparison algorithms
    - Allow access to unauthorized
    - Forbidden the access of authorized
  - Privacy…
    - If your fingerprints are faked, what can you do?



Error Rate of Biometric Authentication

- False Accept Rate vs. False Reject Rate
  - Fraud / False Accept: accept a malicious user
  - Insult / False Reject: deny a normal user

- Increasing the threshold value of accept will increase the false accept rate and reduce the false reject rate.
  - If the false accept rate equals to false reject rate, it is called the **equal error rate**.
  - Different applications and situations have different requirements for false accept rate and false reject rate.

Other Biometrics (1)

Face recognition (by a computer algorithm)

- Error rates up to 20%, given reasonable variations in lighting, viewpoint, and expression

Fingerprints

- The traditional method for identification
- 1911: first US conviction on fingerprint evidence
- The U.K. traditionally requires a 16-point match
  - Fraud rate < 0.001%, Insult rate < 0.1%
- Fingerprint damage impairs recognition

Other Biometrics (2)

Iris scanning

- Irises are very random but stable throughout life
  - Difference between the two eyes of the same individual
- 256-byte iris code based on concentric rings between the pupil and the outside of the iris
- Equal error rate better than < 0.0001%
- Best biometric mechanism currently known

Hand geometry

- The U. S. Immigration and Naturalization Service (INS) uses RSI hand geometry scanners to allow over 60, 000 frequent travelers to bypass immigration lines (through the INSPASS program)

Voice, ear shape, vein pattern, face temperature

Other Biometrics (3)

Recent advances in emerging biometrics

Recently, biometrics based on brain (electroencephalogram) and heart (electrocardiogram) signals have emerged. Researchers have shown that people have distinct brain and heart patterns specific to each individual.

Another example is finger vein recognition, using pattern-recognition techniques based on images of human vascular patterns.

The advantage of such 'futuristic' technology is that it is more fraud-resistant than conventional biometrics like fingerprints. However, such technology is generally more cumbersome and still needs higher accuracy and better reproducibility.

This new generation of biometrical systems is called **biometrics of intent,** and it aims to scan *intent*. The technology will analyze physiological features such as eye movement, body temperature, breathing, etc., and predict dangerous behavior or hostile intent before it materializes into action.

Biometric Considerations

Universality	How commonly biometric is found
Uniqueness	How well biometric distinguishes between others
Permanence	How well biometric resists aging
Collectability	How easy biometric is to acquire
Performance	Accuracy, speed, and robustness of system capturing biometric
Acceptability	Degree of approval by the public for use
Circumvention	How hard it is to fool authentication system

Will passwords disappear soon? Doubtful

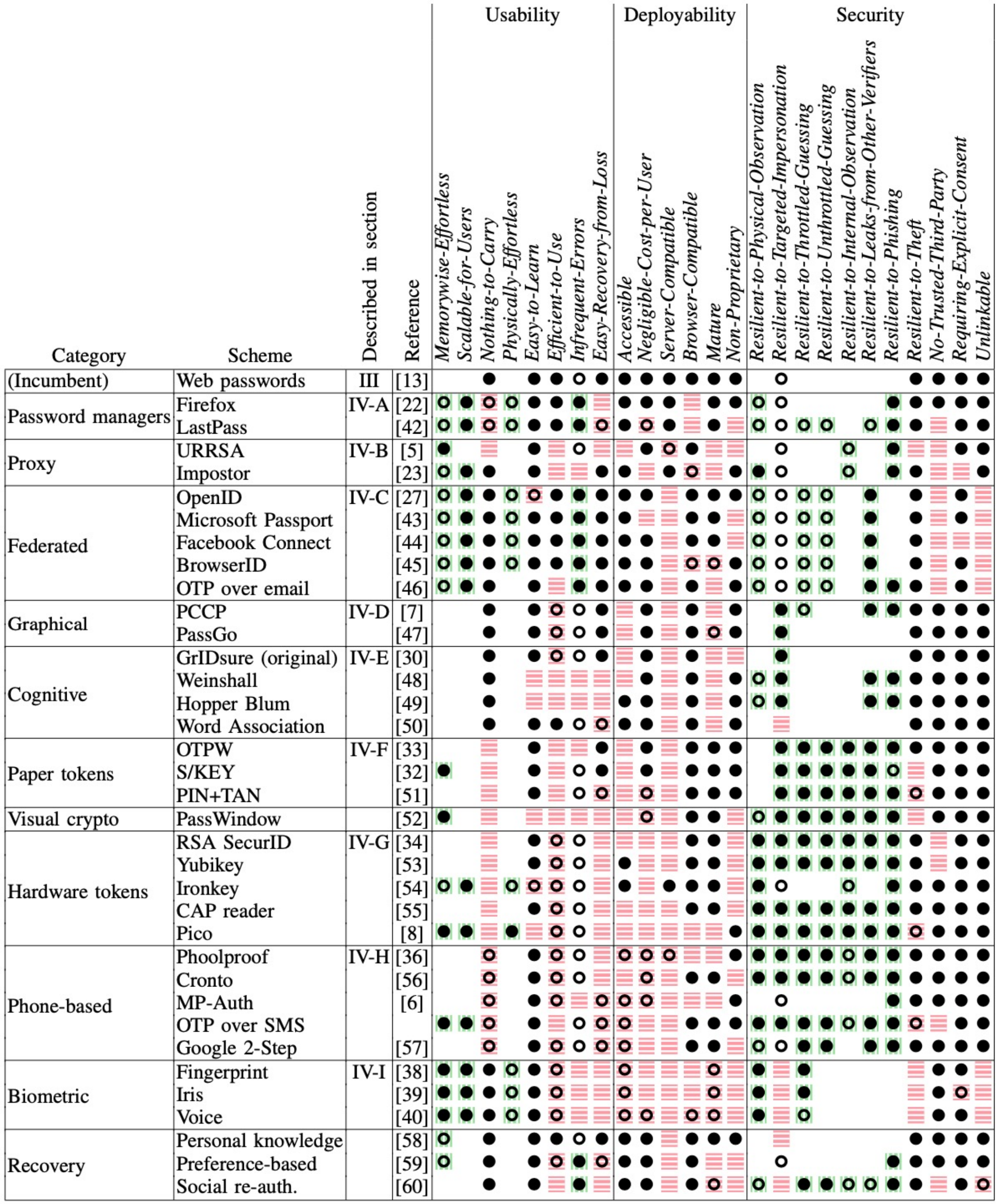
“Hey, have you seen [insert thing here]? It's going to kill passwords! No, it's not.”

- “Despite its many flaws, the one thing that the humble password has going for it over technically superior alternatives is that everyone understands how to use it. Everyone.”
- “If your [password-replacement] product is so awesome, have you considered why no one uses it? ” [Hunt]

“We evaluate two decades of proposals to replace text passwords… Our comprehensive approach leads to key insights about the difficulty of replacing passwords… no known scheme come[s] close to providing all desired benefits: none even retains the full set of benefits that legacy passwords already provide… many academic proposals have failed to gain traction because researchers rarely consider a sufficiently wide range of real-world constraints.”

They’ll be replaced someday, but it’s not as easy as you might think.

Source: “Here's Why [Insert Thing Here] Is Not a Password Killer” by Troy Hunt, 2018-11-05, https: //www.troyhunt.com/heres-why-insert-thing-here-is-not-a-password-killer/
“The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes” by Joseph Bonneau et al, 2012 IEEE Symposium on Security and Privacy,

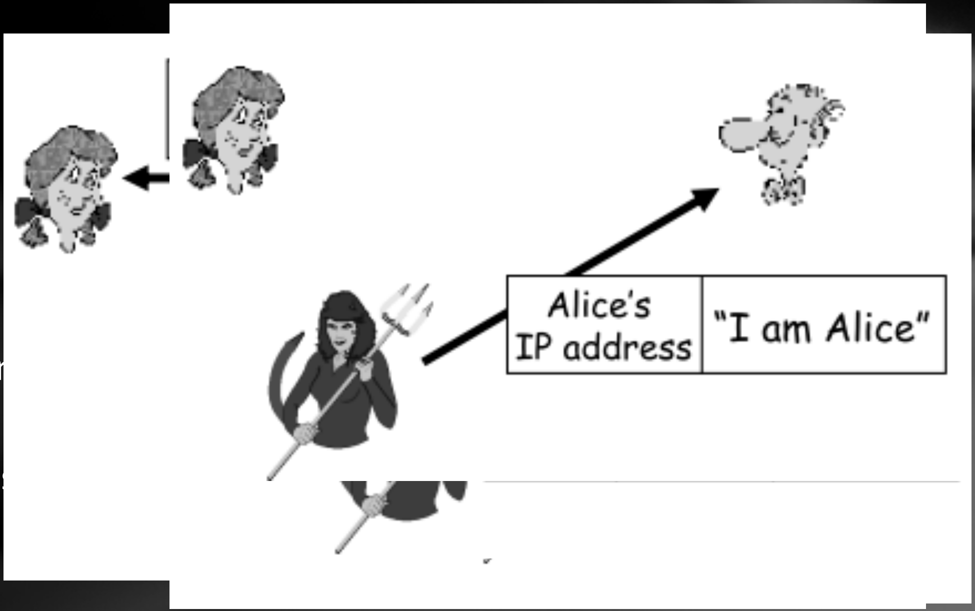


				Usability								Deployability						Security										
Category	Scheme	Described in section	Reference	*Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*
(Incumbent)	Web passwords	III	[13]			●		●	●	○	●	●	●	●	●	●	●		○						●	●	●	●
Password managers	Firefox	IV-A	[22]	○🟩	●🟩	○🟥	○🟩	●	●	●🟩	🟥	●	●	●	🟥	●	●	○🟩	○					●🟩	●	●	●	●
	LastPass		[42]	○🟩	●🟩	○🟥	○🟩	●	●	●🟩	○🟥	●	○🟥	●	🟥	●	🟥	○🟩	○	○🟩	○🟩		○🟩	●🟩	●	🟥	●	●
Proxy	URRSA	IV-B	[5]	●🟩		🟥		●	🟥	○	🟥	🟥	●	○🟥	●	🟥	🟥		○			○🟩		●🟩	🟥	🟥	●	●
	Impostor		[23]	○🟩	●🟩	●		●	🟥	🟥	●	●	🟥	●	○🟥	🟥	●	●🟩	○			○🟩		●🟩	●	🟥	🟥	●
Federated	OpenID	IV-C	[27]	○🟩	●🟩	●	○🟩	○🟥	●	●🟩	●	●	●	🟥	●	●	●	○🟩	○	○🟩	○🟩		●🟩		●	🟥	●	🟥
	Microsoft Passport		[43]	○🟩	●🟩	●	○🟩	●	●	●🟩	●	●	🟥	🟥	●	●	🟥	○🟩	○	○🟩	○🟩		●🟩		●	🟥	●	🟥
	Facebook Connect		[44]	○🟩	●🟩	●	○🟩	●	●	●🟩	●	●	●	🟥	●	●	🟥	○🟩	○	○🟩	○🟩		●🟩		●	🟥	🟥	🟥
	BrowserID		[45]	○🟩	●🟩	●	○🟩	●	●	●🟩	●	●	●	🟥	○🟥	○🟥	●	○🟩	○	○🟩	○🟩		●🟩		●	🟥	●	🟥
	OTP over email		[46]	○🟩	●🟩	●		●	🟥	●🟩	●	●	●	🟥	●	🟥	●	○🟩	○	○🟩	○🟩		●🟩	●🟩	●	🟥	●	🟥
Graphical	PCCP	IV-D	[7]			●		●	○🟥	○	●	🟥	●	🟥	●	🟥	●		●🟩	○🟩			●🟩	●🟩	●	●	●	●
	PassGo		[47]			●		●	○🟥	○	●	🟥	●	🟥	●	○🟥	●		●🟩						●	●	●	●
Cognitive	GrIDsure (original)	IV-E	[30]			●		●	○🟥	○	●	🟥	●	🟥	●	🟥	🟥		●🟩						●	●	●	●
	Weinshall		[48]			●		🟥	🟥	🟥	🟥	🟥	●	🟥	●	🟥	●	○🟩	●🟩				●🟩	●🟩	●	●	●	●
	Hopper Blum		[49]			●		🟥	🟥	🟥	🟥	●	●	🟥	●	🟥	●	○🟩	●🟩				●🟩	●🟩	●	●	●	●
	Word Association		[50]			●		●	●	○	○🟥	●	●	🟥	●	🟥	●		🟥						●	●	●	●
Paper tokens	OTPW	IV-F	[33]			🟥		●	🟥	🟥	●	🟥	●	🟥	●	●	●		●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●	●	●	●
	S/KEY		[32]	●🟩		🟥		●	🟥	○	●	🟥	●	🟥	●	●	●		●🟩	●🟩	●🟩	●🟩	●🟩	○🟩	🟥	●	●	●
	PIN+TAN		[51]			🟥		●	🟥	○	○🟥	🟥	○🟥	🟥	●	●	●		●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	○🟥	●	●	●
Visual crypto	PassWindow		[52]	●🟩		🟥		🟥	🟥	🟥	🟥	🟥	○🟥	🟥	●	●	🟥	○🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	🟥	●	●	●
Hardware tokens	RSA SecurID	IV-G	[34]			🟥		●	○🟥	○	🟥	🟥	🟥	🟥	●	●	🟥	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●	🟥	●	●
	Yubikey		[53]			🟥		●	○🟥	○	🟥	●	🟥	🟥	●	●	🟥	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●	🟥	●	●
	Ironkey		[54]	○🟩	●🟩	🟥	○🟩	○🟥	○🟥	○	🟥	●	🟥	●	●	●	🟥	●🟩	○			○🟩		●🟩	●	●	●	●
	CAP reader		[55]			🟥		●	○🟥	○	🟥	🟥	🟥	🟥	●	●	🟥	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●	●	●	●
	Pico		[8]	●🟩	●🟩	🟥	●🟩	🟥	○🟥	○	🟥	🟥	🟥	🟥	🟥	🟥	●	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	●🟩	○🟥	●	●	●
Phone-based	Phoolproof	IV-H	[36]			○🟥		●	○🟥	○	🟥	○🟥	○🟥	○🟥	🟥	🟥	●	●🟩	●🟩	●🟩	●🟩	○🟩	●🟩	●🟩	●	●	●	●
	Cronto		[56]			○🟥		●	○🟥	○	🟥	🟥	○🟥	🟥	●	●	🟥	●🟩	●🟩	●🟩	●🟩	○🟩	●🟩	●🟩	●	●	●	●
	MP-Auth		[6]			○🟥		●	○🟥	🟥	○🟥	○🟥	○🟥	🟥	🟥	🟥	●		○					●🟩	●	●	●	●
	OTP over SMS			●🟩	●🟩	○🟥		●	🟥	○	○🟥	○🟥	🟥	🟥	●	●	●	●🟩	●🟩	●🟩	●🟩	○🟩	●🟩	●🟩	○🟥	🟥	●	●
	Google 2-Step		[57]			○🟥		●	○🟥	○	○🟥	○🟥	🟥	🟥	●	●	🟥	○🟩	○	●🟩	●🟩		●🟩	●🟩	●	●	●	●
Biometric	Fingerprint	IV-I	[38]	●🟩	●🟩	●	○🟩	●	○🟥	🟥	🟥	○🟥	🟥	🟥	🟥	○🟥	🟥	●🟩	🟥	●🟩					🟥	●	●	🟥
	Iris		[39]	●🟩	●🟩	●	○🟩	●	○🟥	🟥	🟥	○🟥	🟥	🟥	🟥	○🟥	🟥	●🟩	🟥	●🟩					🟥	●	○🟥	🟥
	Voice		[40]	●🟩	●🟩	●	○🟩	●	○🟥	🟥	🟥	○🟥	○🟥	🟥	○🟥	○🟥	🟥	●🟩	🟥	○🟩					🟥	●	●	🟥
Recovery	Personal knowledge		[58]	○🟩		●		●	●	○	●	●	●	🟥	●	●	●		🟥						●	●	●	●
	Preference-based		[59]	○🟩		●		●	○🟥	●🟩	○🟥	●	●	🟥	●	🟥	🟥		○					●🟩	●	●	●	●
	Social re-auth.		[60]			●		●	🟥	●🟩	🟥	●	●	🟥	●	○🟥	🟥	○🟩	🟥	●🟩	●🟩	○🟩	○🟩	●🟩	●	🟥	●	○🟥

●= offers the benefit; ○= almost offers the benefit; *no circle* = does not offer the benefit.
🟩= better than passwords; 🟥= worse than passwords; *no background pattern* = no change.

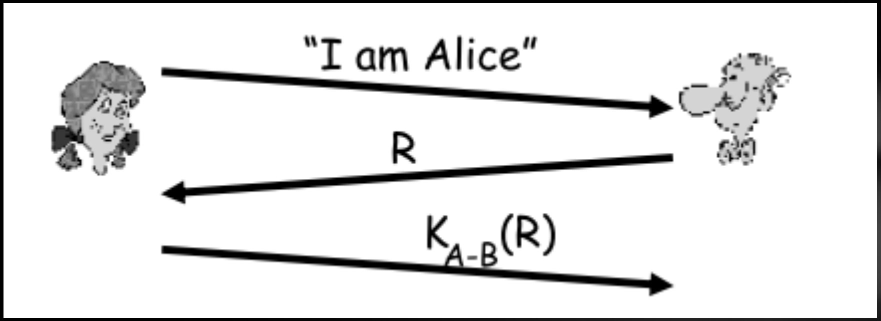
Network Authentication

- Password authentication and biometric authentication happen locally, but on the Internet, how to prove "you are who you say you are" to others?
  - "You can't know if the one you are talking to on the Internet is either a human or a dog."
- Target:
  - Bob hopes Alice to prove herself
  - **Protocol v1.0: Alice said: "I am Alice!"**
    - On the Internet, Bob can't "see" Alice;
    - Marvin declares to Bob, " I am Alice!"
  - **Protocol v2.0:**
    - Alice packets "I am Alice!" with the IP addr and sends it to Bob.
    - Marvin can fake an IP packet with content to cheat Bob!
  - **Protocol v3.0:**
    - Alice packets "I am Alice" with the IP address and her password, then sends the packet to Bob.
    - Replay Attack: Marvin records Alice's message and replays it!
  - **Protocol v4.0:**
    - Alice packets "I am Alice" with the IP address and her encrypted password, then sends the packet to Bob.
    - Replay Attack: It still works!!



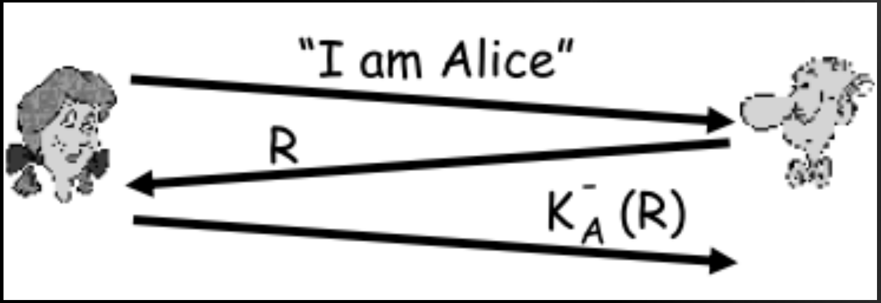
Challenge Response Authentication

- Target: avoid replay attack!
  - Challenge Code: Ra number R will be used once in a lifetime.
- **Protocol v5.0: to prove it's not a replay, Bob sends a random challenge code R to Alice, and Alice must answer the R encrypted with a shared secret key!**



Alice is alive, and only Alice know the secret key, it can't be Alice, it won't wrong!

- In Protocol v5.0, if Marvin knows the shared secret key, Marvin can still fake a response. How to solve this problem? Public key!
- **Protocol v6.0: use challenge and public key!**



Bob calculates:
$K_A^+(K_A^-(R)) = R$
He knows that only Alice knows secret key K_A^-:
$K_A^+(K_A^-(R)) = R$

KERBEROS

Problem to solve: In a distributed situation, customers want to get service on the server. The server can limit authorized users' visits and distinguish the required service.

- Against three threats:
  - Fake as another user to access the server
  - Change the workstation's network address and send a request to the statio
  - Listen to the transmitted message and attack the server by replaying the a
- KERBEROS's Goal:
  - **Safety**: you can't get enough information to fake the other user.
  - **Reliability**: can't get Kerberos' service means can't get the required service reliability is required.
  - **Transparency**: Users don't have to know the existence of an authenticatio
  - **Scalability**: the system can support a large number of users and servers.
- Realize of KERBEROS:
  - Provide an authentication server to identify the server and user.
  - Base on normal encryption, without using public key encryption.
  - Two versions: 4 and 5

KERBEROS v4

Concepts:

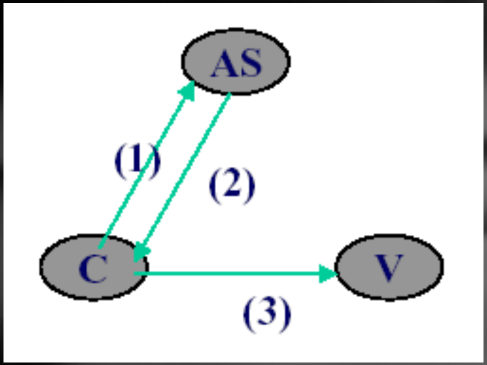
- C: client
- AS: Authentication Server(stores user’s password)
- TGS: Ticket Granting Server
- V: Application Server
- IDc: User’s Identifier on C
- IDv: Identifier of V
- Pc: Password on C
- ADc: Network Address of C
- Kv: Shared key between AS and V

KERBEROS v4 —— The Simplest Authentication Protocol

- Bring in the Authentication Server(AS), which knows and stores all the users' passwords in a central database. AS shares a secret key with every server.

(1)	C → AS:	$ID_C \| \| P_C \| \| ID_V$
(2)	AS → C:	Ticket
(3)	C → V:	$ID_C \| \| Ticket$

Ticket = $E(K_V, [ID_C \| \| AD_C \| \| ID_V])$



- Existing Problems:
  - Request the user continually input password
    - Requiring the same server at different times will need a new ticket.
    - Requiring a different server will need a new ticket.
  - Password is transferred in plaintext, which can be stolen.
  - Once stole the ticket, the attacker can fake C to make a replay attack.

KERBEROS v4 —— Improved Protocol

- Add a Ticket Granting Server(TGS)

Get ticket when log in (once per logging):

(1) C → AS: $ID_C \| ID_{tgs}$

(2) AS → C: $E(K_C, Ticket_{tgs})$

Get service-ticket requiring a service (once per service):

(3) C → TGS: $ID_C \| ID_V \| Ticket_{tgs}$

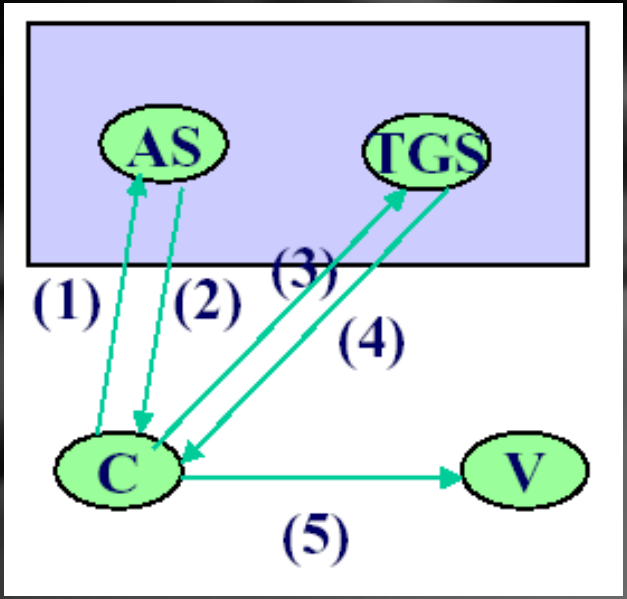
(4) TGS → C: $Ticket_V$

Get service (once per session):

(5) C → V: $ID_C \| Ticket_V$

$Ticket_{tgs} = E(K_{tgs}, ID_C \| AD_C \| ID_{tgs} \| TS1 \| Lifetime1])$

$Ticket_V = E(K_V, [ID_C \| AD_C \| ID_V \| TS2 \| Lifetime2])$



- Improvement:
  - Don't transfer plaintext on the Internet
  - Reduce the frequency of password inputting
  - Reduce the threats of replay attacks by setting lifetime

KERBEROS v4 —— Improved Protocol

- Add a Ticket Granting Server(TGS)

Get ticket when log in (once per logging):

(1) C → AS: $ID_C || ID_{tgs}$

(2) AS → C: $E(K_C, Ticket_{tgs})$

Get service-ticket requiring a service (once per service):

(3) C → TGS: $ID_C || ID_V || Ticket_{tgs}$

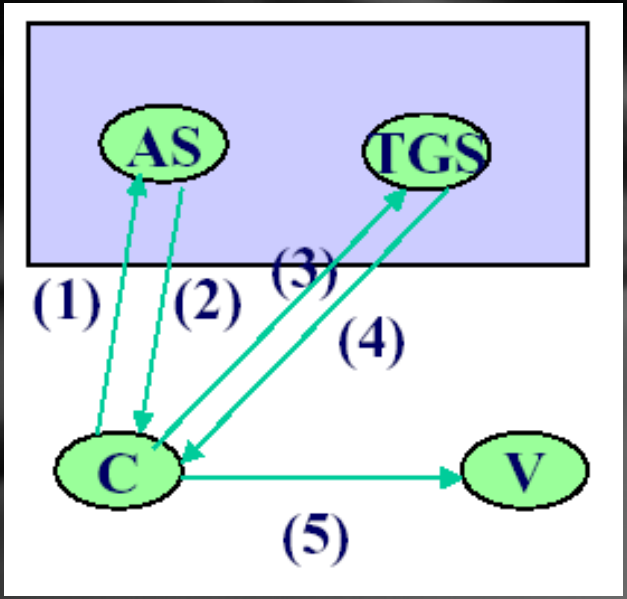
(4) TGS → C: $Ticket_V$

Get service (once per session):

(5) C → V: $ID_C || Ticket_V$

$Ticket_{tgs} = E(K_{tgs}, ID_C || AD_C || ID_{tgs} || TS1 || Lifetime1])$

$Ticket_V = E(K_V, [ID_C || AD_C || ID_V || TS2 || Lifetime2])$



- Issues:
  - Lifetime setup: short - increasing the frequency of password inputting; long – increasing the risk of replay attack
  - How to prove that the ticket user is the ticket applier?
  - How does V prove itself to a client?

KERBEROS v4 —— Final protocol

Get the ticket when log in (once per logging):

(1) C → AS: IDC || IDtgs ||TS1

(2) AS → C: E(Kc, [Kc, tgs||IDtgs||TS2||Lifetime2||Tickettgs])

Tickettgs＝E(Ktgs, [Kc, tgs||IDC||ADC||IDtgs||TS2||Lifetime2])

Get service-ticket requiring a service (once per service):

(3) C → TGS: IDV||Tickettgs ||AuthenticatorC

(4) TGS → C: E(Kc, tgs, [Kc, v||IDV||TS4||TicketV])

AuthenticatorC ＝E(Kc, tgs, [IDC||ADC||TS3])

TicketV＝E(Kv, [Kc, v||IDC||ADC||IDV||TS4||Lifetime4])

Get service (once per session):

(5) C → V: TicketV ||AuthenticatorC

(6) V → C: E(Kc, v, [TS5+1])

AuthenticatorC ＝E(Kc, v, [IDC||ADC||TS5])

KERBEROS Analysis

- The correctness of Kerberos was proved.
- Kerberos model depends on a safe time service
- Trusted third-party:
  - Distribution and management of shared keys become easier.
  - Kerberos server reduces the burden of the application server.
  - Control and protection of safety-related data make the attack difficult.
- Ticket
  - AS's authentication result and session key can be safely delivered to the application server.
  - The ticket can be reused in a lifetime, reducing the cost and making it more convenient.
- Ticket Granting Service
  - Reduce the frequency of using passwords which can protect the password.
  - Reduce the burden of AS and improve the efficiency of the identification system.
- Timestamp
  - Prevent the replay attack of Ticket and Authenticator.
- Take the session key as evidence of authentication.

Difference between V4 and V5

- Dependency of encryption system.
  - V4 relies on the DES, while V5 allows the selection of other encryption algorithms.
- Dependency of Internet protocol.
  - V4 is effective only for IP protocol, while V5 allows using other network addresses.
- Dependency of telegraph sequence.
- The lifetime of the ticket.
  - The longest lifetime of V4 is 1280 minutes, while V5 doesn't have a lifetime limit.
- Transmit of authentication.
  - The V5 allows the transmission of authentication, while the V4 doesn't allow it.
- Authentication between different fields.
  - Decreases the exchange of secret keys.

Analysis of KERBEROS' Weakness

- Time Dependence
  - It is always difficult to realize time synchronization.
  - The attacker can mislead the system time and make a replay attack
- Guessing Attack
  - It's easy to be attacked for a weak password.
  - The protocol doesn't provide further protection for passwords.
- The problem with Key Storage
  - Password and session keys can't be stored in the typical computer system.

Authorization

What can you do?

Concepts and Realize of Authorization

- Authorization and Access Control are almost the same.
- Authorization —— you are permitted to do what you are trying to do
  - **Authentication** provides basic access control;
  - **Authentication** provides the basic function of verifying user identity;
  - **Authorization** is needed to do deeper control.
- **Three elements of access control:**
  - **Subject**: An entity that can access objects, such as user or application process, etc
  - **Object**: The object being accessed, such as files, programs, data, etc
  - **Privilege**: The permission of the subject to use the object, such as read, write, delete, execute, authorize, etc
- **Secure Access Control:**
  - Three main functions: Authorization, Revoke, Checker
  - Two stages: Make Policy, Execute Policy

Secure Access Control Model

DAC, Discretionary Access Control	自主访问控制
MAC, Mandatory Access Control	强制访问控制
RBAC, Role-Based Access Control	基于角色的访问控制

Discretionary Access Control

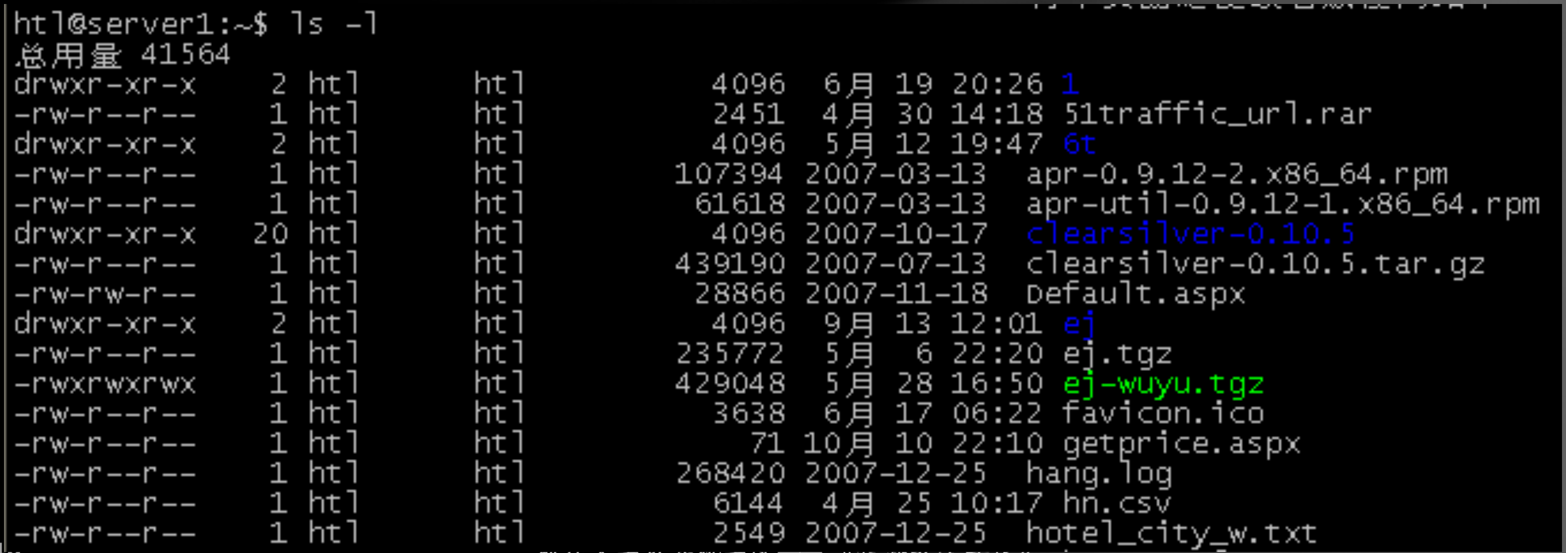
- **Feature:**
  - Making decisions according to the subject's identity and access authorization
  - **Discretionary** means that the subject with certain privileges can automatically grant the subs

	Object x	Object y	Object z
Subject A	R, W, Own	R, W	R, W
Subject B	R	/	/
Subject C	R	R, W, Own	/

- **Shortcomin**
  - The access per nformation, and access control in som
    - User A may p r B so that User B originally has no access permission and can access O.
- **Access control matrix:**
  - Access control list: Each **object** is related to a list of **subjects** permitted to access it.
  - Capability List: Each **subject** is related to a list of **objects** it can access.

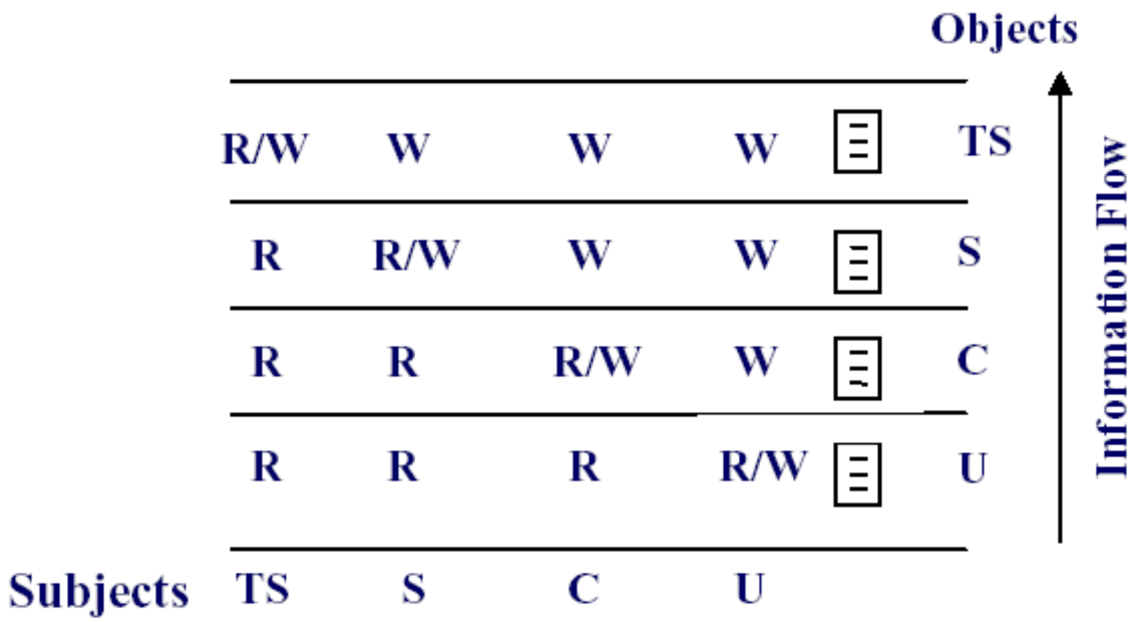
Example of Discretionary Access Control —— Unix OS

- Unix operating systems use access control lists to manage files
  - Divide users into three categories: **Owner** of the file (User, u), User that belongs to the **same group** as the file owner (Group, g), All **other** users (Other, o)
  - Divide the permission into three types : Read(Read, r), Write(Write, w), Execute(Execute, x)
  - Using 9 bits to indicate a file's access control list :
    - 1～3 bit - owner's permission, 4～6 bit - group member's permission, 7～9 bit - others' permission
  - The following is the same expression: RWX, 111, 7； RX, 101, 5； R, 100, 4



```
htl@server1:~$ ls -l
总用量 41564
drwxr-xr-x    2 htl      htl          4096  6月 19 20:26 1
-rw-r--r--    1 htl      htl          2451  4月 30 14:18 51traffic_url.rar
drwxr-xr-x    2 htl      htl          4096  5月 12 19:47 6t
-rw-r--r--    1 htl      htl        107394 2007-03-13  apr-0.9.12-2.x86_64.rpm
-rw-r--r--    1 htl      htl         61618 2007-03-13  apr-util-0.9.12-1.x86_64.rpm
drwxr-xr-x   20 htl      htl          4096 2007-10-17  clearsilver-0.10.5
-rw-r--r--    1 htl      htl        439190 2007-07-13  clearsilver-0.10.5.tar.gz
-rw-rw-r--    1 htl      htl         28866 2007-11-18  Default.aspx
drwxr-xr-x    2 htl      htl          4096  9月 13 12:01 ej
-rw-r--r--    1 htl      htl        235772  5月  6 22:20 ej.tgz
-rwxrwxrwx    1 htl      htl        429048  5月 28 16:50 ej-wuyu.tgz
-rw-r--r--    1 htl      htl          3638  6月 17 06:22 favicon.ico
-rw-r--r--    1 htl      htl            71 10月 10 22:10 getprice.aspx
-rw-r--r--    1 htl      htl        268420 2007-12-25  hang.log
-rw-r--r--    1 htl      htl          6144  4月 25 10:17 hn.csv
-rw-r--r--    1 htl      htl          2549 2007-12-25  hotel_city_w.txt
```

Mandatory Access Control, MAC



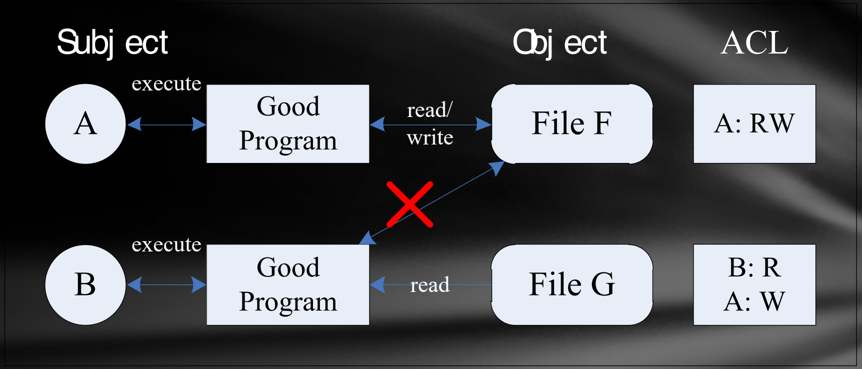
Objects				
TS	R/W	W	W	W
S	R	R/W	W	W
C	R	R	R/W	W
U	R	R	R	R/W
Subjects	TS	S	C	U

密级	英文
绝密TS	Top Secret
秘密S	Secret
机密C	Confidential
无密级U	Unclassified

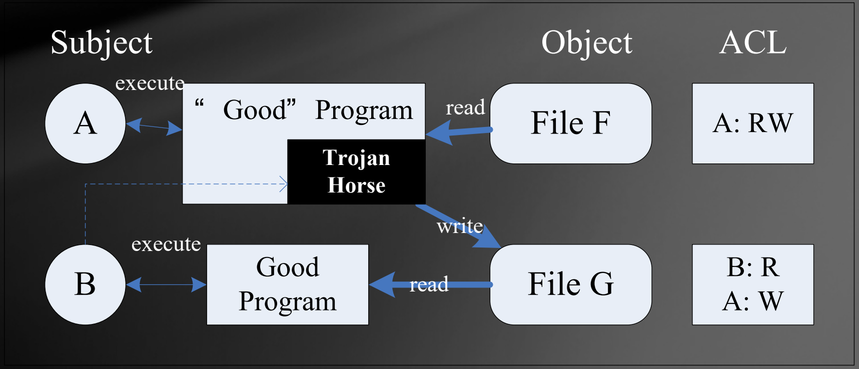
- **Bell-LaPadula Model: Ensure confidentiality**
  - Simple security property (No Read Up): A subject can only read objects of the same or lower security level.
  - * property (No Write Down): A subject can only write a high-level or the same security-level object.
- **Biba Model: Ensure integrity**
  - Contrary to the Bell-LaPadula model: No Read down, No Write up.

Trojan Horse & MAC

- **User A** can read and write **file F**, and write **file G**;
- **User B** can't read **file F**;
- **User B** wants to steal **file F**, what can he do?



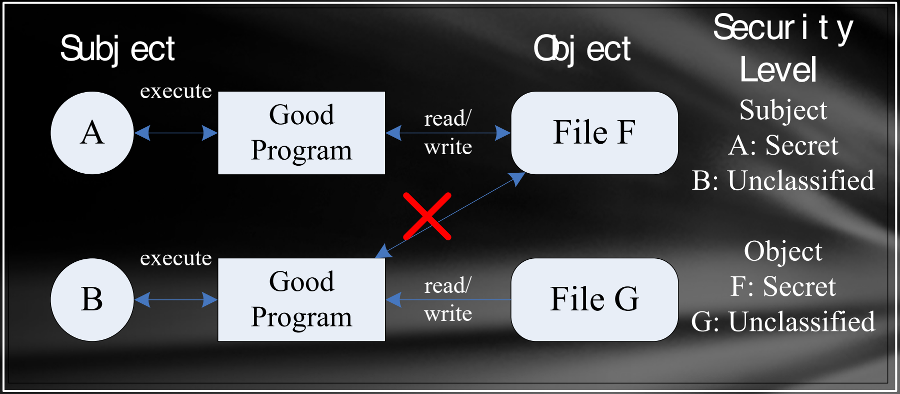
- **User B** installs a Trojan horse in a "Good" program;
- **User A** execute program with the Trojan horse;
- Contents of **file F** are copied to **file G** by the Trojan horse, and **user B** can read **file G**!!!



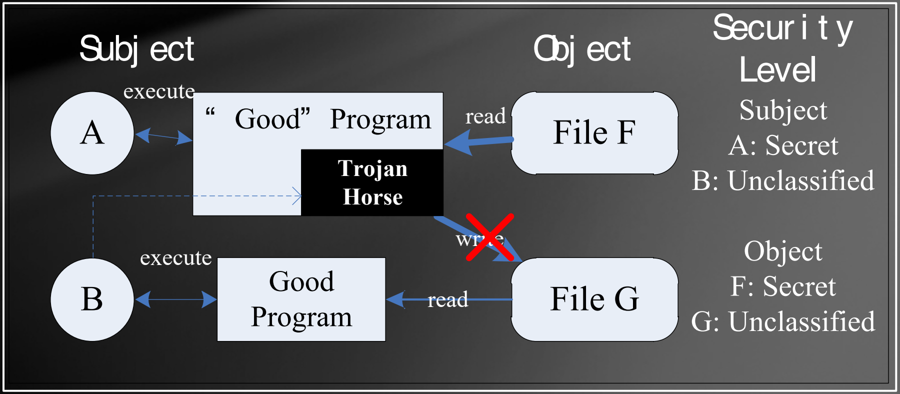
Trojan Horse & MAC

According to the security label:

- **User A** can read and write **file F**; **User A** can read **file G**;
- **User B** can read and write **file G**; **User B** can write **file F**;



- **User B** installs a Trojan Horse in program;
- **User A** executes program with Trojan Horse ;
- Even though the Trojan horse can read **file F**, but it can't write **file G**!!!



Covert Channel & MAC

- **Covert Channel Definition:**
  - By B.W. Lampson: Covert channel is the channel communicating using the system resources without noticing the subject(process).
  - By Wikipedia: A covert channel is a type of computer security attack that creates a capability to transfer information objects between processes that are not supposed to be allowed to communicate by the computer security policy.

Resource Exhaustion Channel

Given a 1024MB pool of dynamically allocated memory

High-Level Process

- bit = 1 $\Rightarrow$ request 1024MB of memory
- bit = 0 $\Rightarrow$ request 0MB of memory

Low-Level Process

- request 1024MB of memory
- if allocated then bit = 0 otherwise bit = 1

Load Sensing Channel

High-Level Process

bit = 1 ⇒ enter a computation-intensive loop

bit = 0 ⇒ go to sleep

Low-Level Process

perform a task with known computational requirements

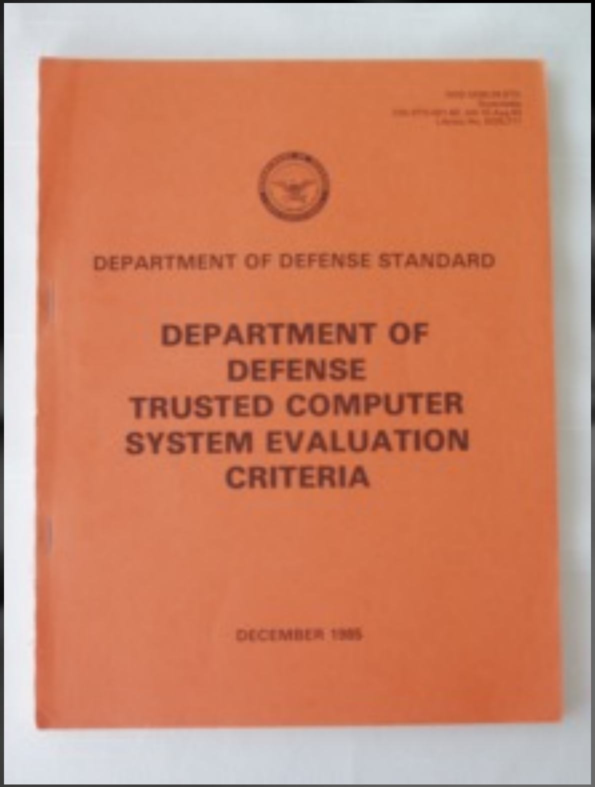
if completed quickly then bit = 0 otherwise bit = 1

Coping with Covert Channels

- After Identification
  - close the channel or slow it down
  - detect attempts to use the channel
  - tolerate its existence
- Covert Channel can’t be eliminated! Ways to deal with Covert Channel:
  - Limit the shared resource(such as disk, CPU, memory, printer, process ID, etc.); for example, the share of a resource only happens between the same security level.
  - Limit the bandwidth, for example, bringing in noise, disturbing, etc.

Standard of MAC —— Orange Book

- American Standard: Orange Book —— TCSEC Standard
  - Trusted Computer System Evaluation Criteria: the first formal standard of computer system security assessment, 1970 proposed by the U.S. Defense Science Board, and in December 1985 announced by the U.S. Department of Defense.
  - TCSEC divides computer security systems into four grades and seven levels.
    - Level D No protection grade, No requirement of security.
    - Level C Grade of self-protection: Implement Discretionary Access Control
    - Level B Mandatory grate of protection: Mandatory access controls must be implemented
    - Level A The highest grade of security
- Chinese national standards: GB 17859-1999 《计算机信息系统安全保护等级划分准则》(Classified criteria for computer information system security)
  - 1: 用户自主保护级 corresponds to TCSEC-C1
  - 2: 系统审计保护级 corresponds to TCSEC-C2
  - 3: 安全标记保护级 corresponds to TCSEC-B1
  - 4: 结构化保护级 corresponds to TCSEC-B2
  - 5: 访问验证保护级 corresponds to TCSEC-A1


DEPARTMENT OF DEFENSE STANDARD

DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA

DECEMBER 1985

Role-based access control, RBAC

- **Weakness of DAC and MAC:**
  - Safety: DAC's security is too weak, which can't meet the requirements;
    MAC's security is too strong, which makes it not flexible enough.
  - The workload of Manage: very large.
- RBAC is the result of the requirements of information security in modern business, mainly to reduce the workload of security administrators
  - It originated from the UNIX system or concept of group in other operating systems and has only ten years of history.
  - Role-based access control assigns each user a set of roles(i.e., authorization groups).
  - Features:
    - separation of duties
    - role hierarchies
    - role activation
    - constraints on user/role membership

Concepts of RBAC

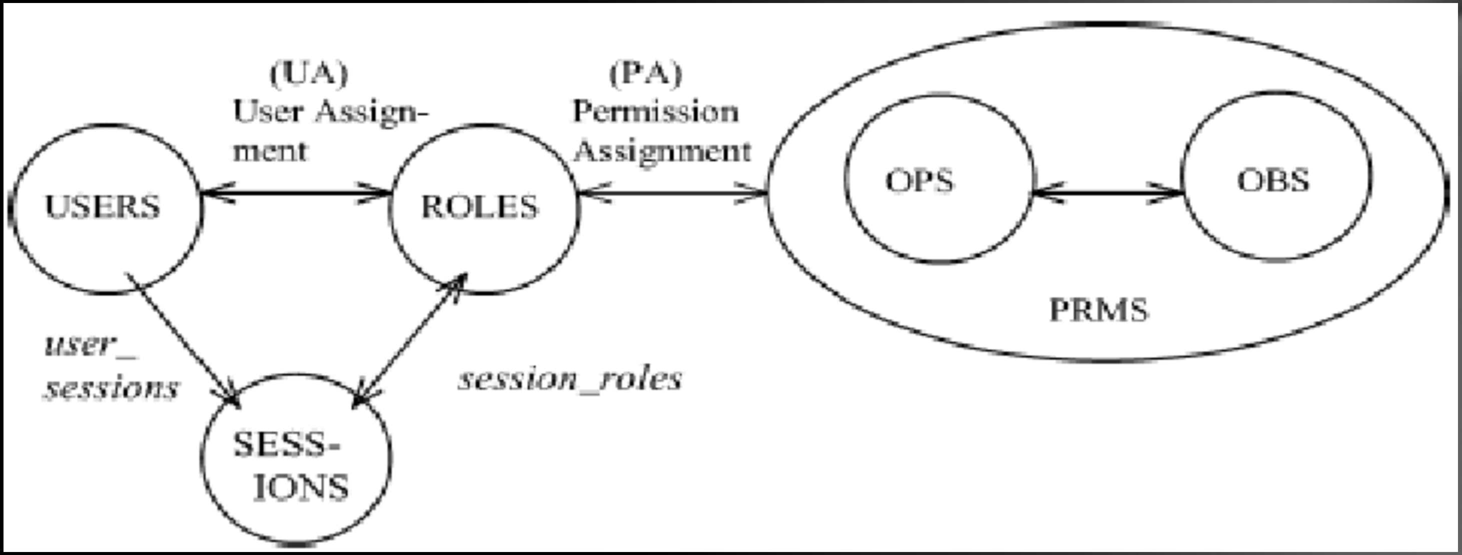
- It consists of five basic data elements :
  - users (USERS)
  - roles (ROLES)
  - objects (OBS)
  - operations (OPS)
  - permissions (PRMS)
- Core Concepts:
  - **Role**:
    - Each role with a group of users is interrelated with the relevant permissions, and users belonging to the role have the right to perform these operations on objects.
    - **Differences between role and group**:
      - group - a set of users;
      - role - a collection of users + a collection of permissions.
  - **Relationship**:
    - Many to many, the user is assigned certain roles, and roles are assigned certain permissions.
  - **Sessions**:
    - The mapping between Users and active role.

Core RBAC Model

Sandhu 1996

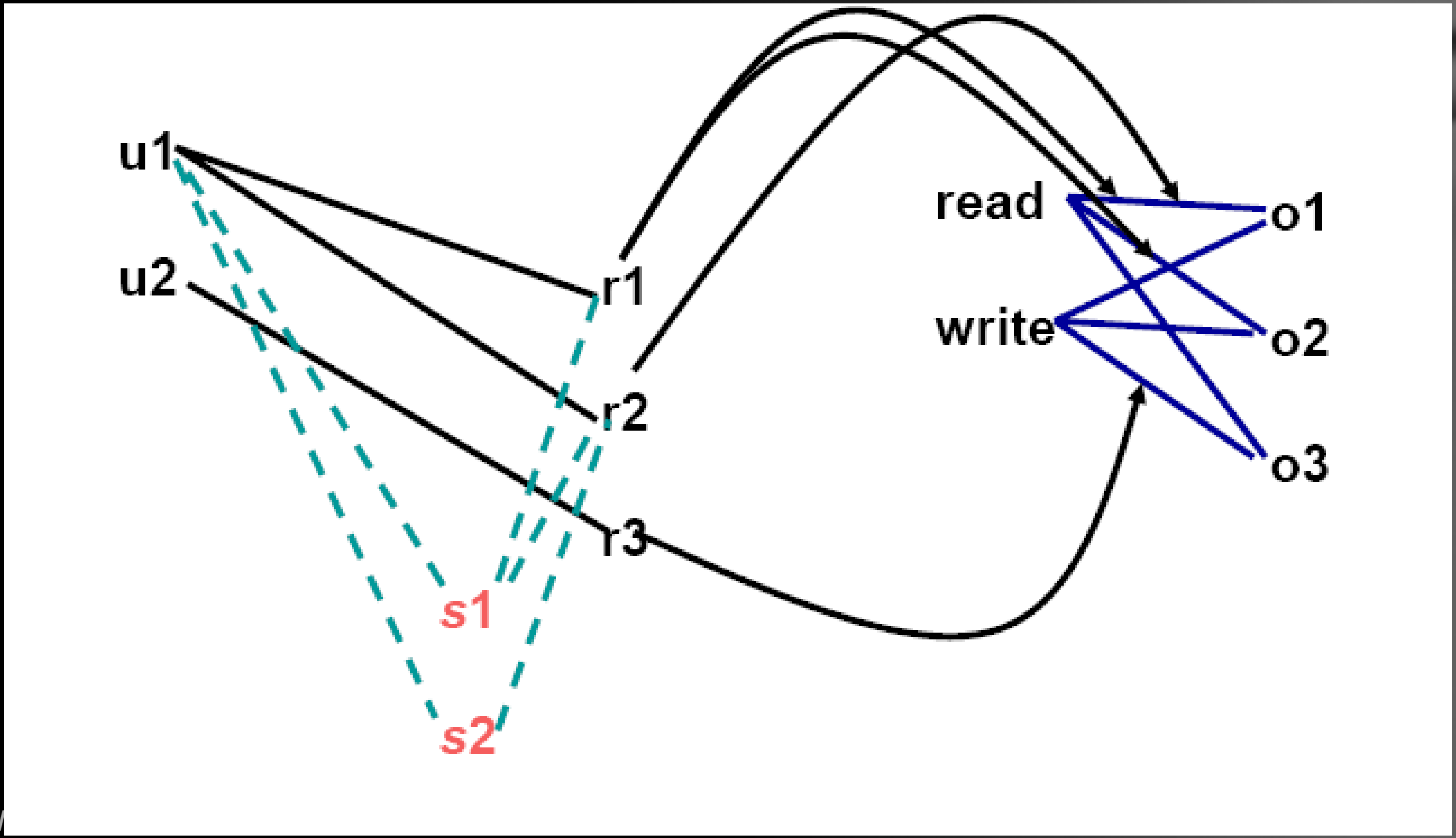
- USERS: Can be people, equipment, process
- PRMS: Implementation of OPS license to protected targets
  - OPS: operations
  - OBS: subjects

- UA: roles that users can use
- PA: permission assignment relations, Permissions that user can use
- Session_roles: Session active role
- User_sessions: Sets of sessions associated with the user;



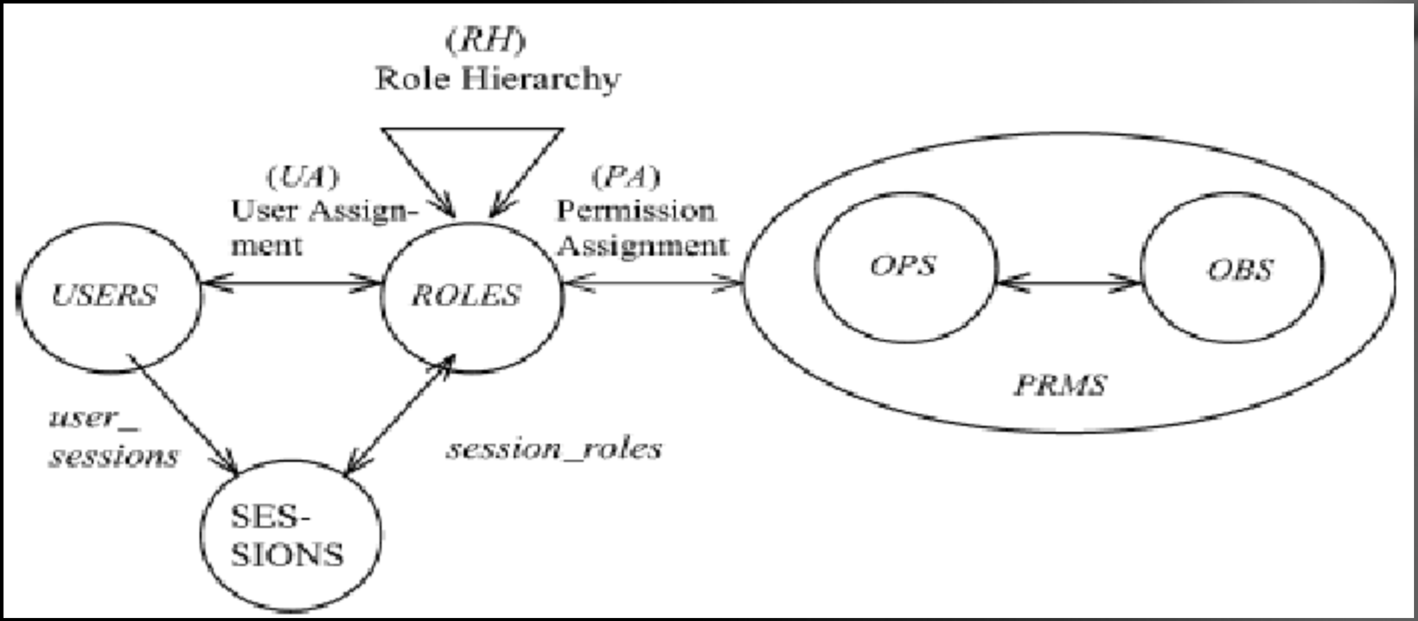
Example of Core RBAC

- Session controlled by the user, allowing dynamic activation/deactivation of roles to achieve the minimum privileges
  - To avoid simultaneous activation of all roles!
- Separating sessions/users can solve problems with multiple accounts caused by the same user.



Hierarchical RBAC Model

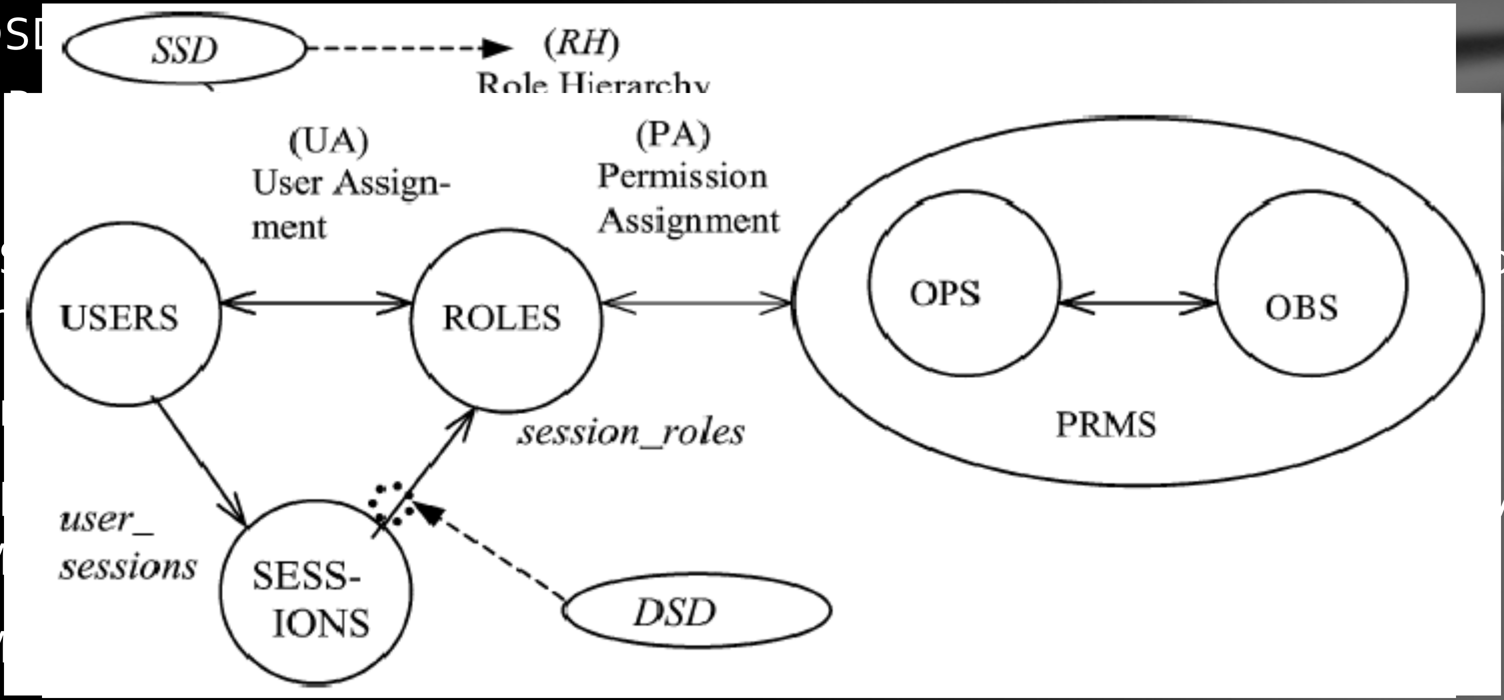
- Role structural hierarchy is the natural way to reflect an organization's authorization and responsibility.
- It defines the inheritance of roles:
    - Role r1 “inherits” role r2, permissions of r2 are also permissions of r1;
    - The role inherits relationship is a partial order relationship: Reflexive, Transitivity, Antisymmetry



Constraint RBAC Model

Increased segregation of duties, to resolve conflicts of interest, to prevent users from beyond the authority

- SSD: Static Separation of Duty Relations
  - SSD defines a constraint relationship of user role assignment; A user can not assign two roles of SSD at the same time.
- DSD
  -
  -
- S ons on th



Exa

- Th Affairs, M
- M
- Separation of duties: a machine administrator, security, and auditors;

The advantages of RBAC

- Easy for authorization management
- Facilitating the classification according to work needs, such as accessing corporate finance permission of the corporate finance department and non-financial sector employees, can be distinguished by the role of financial staff.
- Easy to achieve the least privilege, even if the user was assigned a senior status and can only have the privilege when necessary to prevent accidents.
- Easy for task sharing, different roles perform different tasks
- Easy for hierarchical file management, the document itself can be divided into different roles, such as letters, bills, etc., owned by the different roles of users

- RBAC can be used to realize DAC and MAC!

The Principle of Security Access Control

The model is fixed, but the strategy/rule is flexible. Proper security rules are the key to safety.

- Authorization management—— Decide who is authorized to modify the "Allowed Access."
  - MAC: "Allowed access" based solely on the security level of the subject and object.
    - The system security administrator assigns the subject’s (user, process) security level.
    - The security level of the objects is decided based on the user's security level that created them.
  - DAC: A variety of accepted management methods.
    - **Centralized management**: administrator authorizes and removes authorization to access control.
    - **Hierarchical management**: central administrators distribute the management duty to other centralized control administrators.
    - **Ownership Management**: the owner of an object authorizes unauthorized visitors of the object.
    - **Collaborative management**: access to special resources is not decided by a single user but by the Cooperation authorization of shared users.
    - **Distributed management**: in distributed control, the object owner can authorize the other users the privilege of managing.
  - RBAC: RBAC provides many access controls similar to self-management strategies. However, the delegacy of management permission is an important feature of RBAC, which does not exist in DAC and MAC.

The Principle of Security Access Control

- Minimize rights——Distribute the right originally owned by the super-user to three types of privileged users:
  - **System Administrator** - Responsible for system maintenance, user management, software installation, and other functions；
  - **Security Administrator** - responsible for the security rule configuration, security policy management, and other functions；
  - **Audit Administrator** - responsible for checking the audit records, monitoring system security, and other functions。

Each privileged user performs their duties and restraint mutually. The basic constraint is that, in the eyes of one user class, the permissions the other class users have are the same as ordinary users. **There is no special privilege.**

Review

Authentication:

- Concepts
- Normal Authentication: Password Authentication, Biometric/Behaviometric Authentication
- Network Authentication: Challenge Response Authentication, KERBEROS

Authorization / Access Control:

- Concepts
- DAC, MAC, RBAC
- Principle of Bell-LaPadula
- Covert Channel
- Principle of Secure Access Control