山东大学
SHANDONG UNIVERSITY

网络与大数据安全

3 – Access Control and Authentication

李琨
Email: kunli@sdu.edu.cn

学无止境 气有浩然

目录

CONTENTS

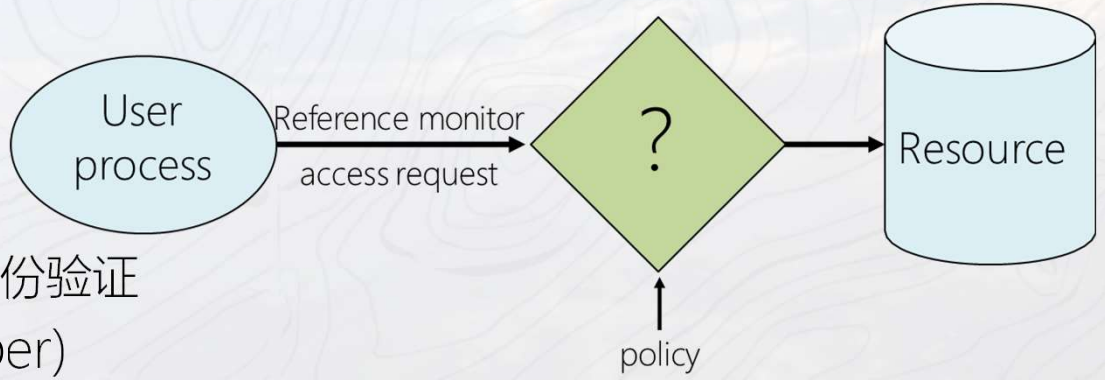
访问控制

什么是访问控制（Access control）

访问控制是**网络信息系统的基本安全机制**。访问控制是指对资源对象的访问者授权、控制的方法及运行机制。

- 访问者又称为主体，可以是用户、进程、应用程序等；
- 资源对象又称为客体，即被访问的对象，可以是文件、应用服务、数据等；
- 授权是访问者可以对资源对象进行访问的方式，如文件的读、写、删除、追加或电子邮件服务的接收。

- 假设条件
  - 系统知道用户是谁
    - 通过姓名和密码、其他凭证进行身份验证
  - 访问请求通过“把关人”(gatekeeper)
    - 例：引用监控器
    - 系统不得允许绕过监控器的访问



访问控制的场景

引用监控器（reference monitor）：系统中所有主体对客体的访问都通过引用监控器作为中介，由引用监控器根据安全访问控制策略来进行授权访问，所有访问记录也都由引用监控器生成审计日志。

引用监控器是一个访问控制概念，而不是实际的物理概念，因此通常被称为“引用监控器概念”或“抽象机”。

与访问控制有关的功能

- Authentication 身份识别：验证用户或其他系统实体的凭据是否有效。
- Authorization 授权：授予系统实体访问系统资源的权利或许可。
  - This function determines who is trusted for a given purpose.
- Audit 审计：对系统记录和活动的独立审核。
  - in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy and procedures.

访问控制矩阵

- 场景：一个实体可以使另一个实体访问某些资源
- 通常使用的访问控制矩阵
  - 一个维度包括可尝试对资源进行数据访问的已识别主体
  - 另一个维度列出了可能被访问的对象
- 矩阵中的每个条目指示特定主体（subjects）对特定对象（objects）的访问权限

Objects

Subjects

	File 1	File 2	File 3	…	File n
User 1	read	write	-	-	read
User 2	write	write	write	-	-
User 3	-	-	-	read	read
…					
User m	read	write	read	write	read

访问控制矩阵

- 场景：一个实体可以使另一个实体访问某些资源
- 通常使用的访问控制矩阵
  - 一个维度包括可尝试对资源进行数据访问的已识别主体
  - 另一个维度列出了可能被访问的对象
- 矩阵中的每个条目指示特定主体（subjects）对特定对象（objects）的访问权限

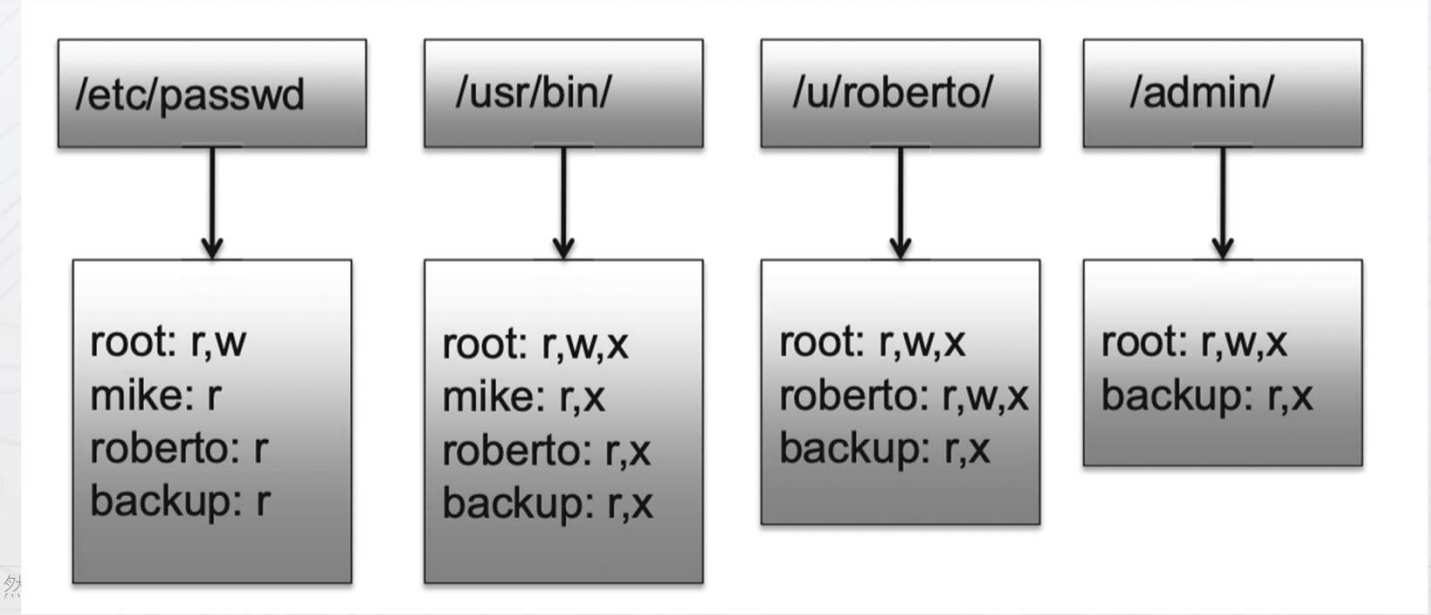
	/etc/passwd	/usr/bin/	/u/roberto/	/admin/
root	read, write	read, write, exec	read, write, exec	read, write, exec
mike	read	read, exec		
roberto	read	read, exec	read, write, exec	
backup	read	read, exec	read, exec	read, exec
. . .	. . .	. . .	. . .	. . .

访问控制矩阵的两种特殊形式

- 访问控制列表（ACL, Access control list）——访问控制矩阵的列

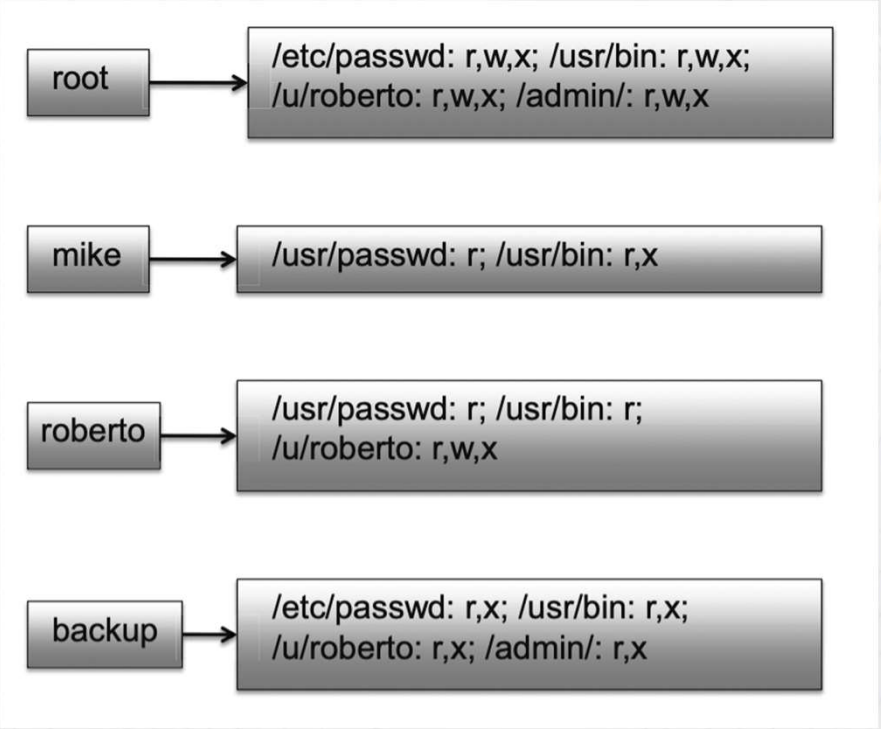
  它为每个对象O定义了一个列表 L，称为O的访问控制列表，该列表列举了对O有访问权限的所有主体，并为每个主体S定义了S对对象O的访问权限。

  - 权限与客体关联
  - 在客体上附加一个主体明细表的方法来表示访问控制矩阵



访问控制矩阵的两种特殊形式

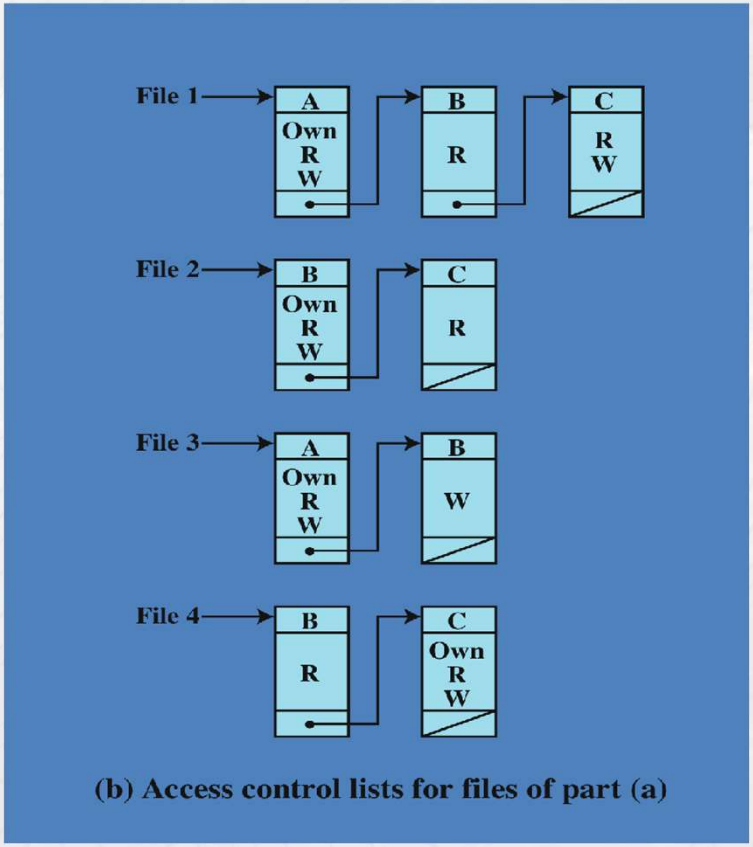
- 能力表（Capability List）——访问控制矩阵的行
  它为每个主体S定义了S拥有非空访问控制权限的对象列表，以及每个对象的具体权限。
  - 权限与主体关联
  - 为每个用户维护一个表，表示主体可以访问的客体及权限



访问控制矩阵的两种特殊形式

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix



(b) Access control lists for files of part (a)

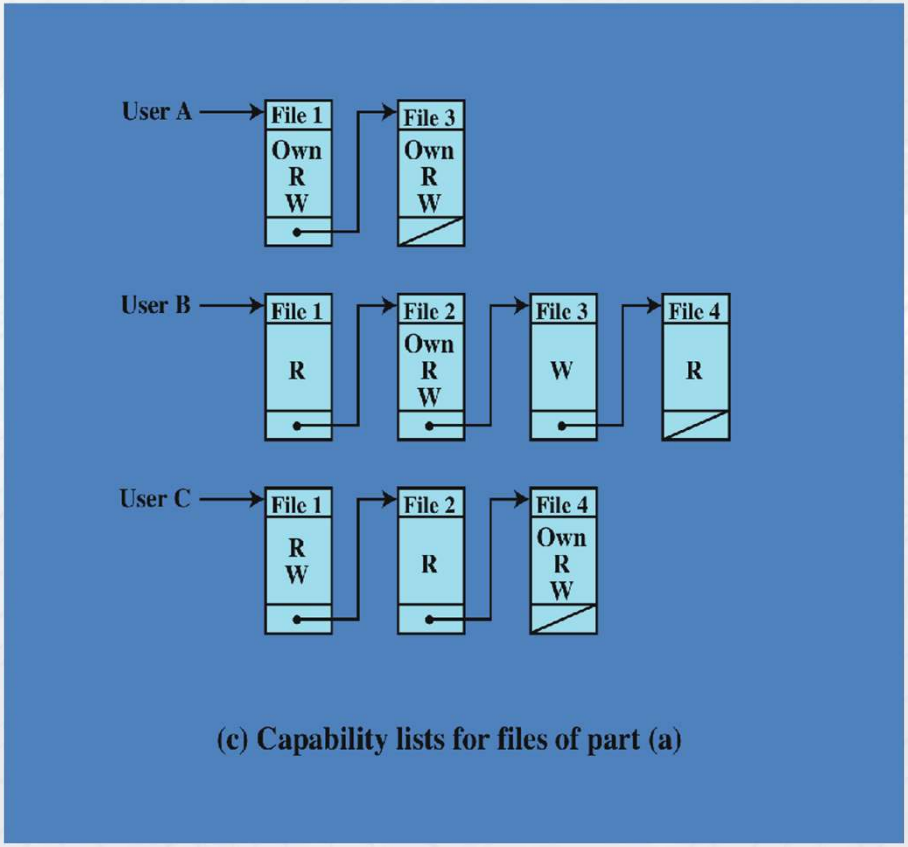
ACL 列出每个对象的用户及其允许的访问权限。列表元素可包括单个用户和用户组。

访问控制矩阵的两种特殊形式

OBJECTS

SUBJECTS	File 1	File 2	File 3	File 4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

(a) Access matrix



(c) Capability lists for files of part (a)

能力表规定了特定用户的授权对象和操作。
操作系统可代表用户持有所有能力表，并使用户无法访问这些表。
必须保护/保证能力表的完整性。

访问控制矩阵的两种特殊形式

- 访问控制列表
  - 将列表与每个对象关联
  - 根据列表检查用户/组
  - 依赖于身份验证：需要了解用户
- 能力表
  - 是不可伪造的表
    - 随机位序列，或由操作系统管理
    - 可从一个进程传递到另一个进程
  - 引用监控器检查能力表
    - 无需知道用户/进程的标识

访问控制机制

- 自主访问控制模型 Discretionary Access Control DAC
  - 客体的所有者按照自己的安全策略授予系统中的其他用户对其的访问权
- 强制访问控制模型 Mandatory Access Control MAC
  - 系统根据主体和客体的安全属性，以强制方式控制主体对客体的访问
- 基于角色的访问控制模型 Role-Based Access Control role-BAC
  - 根据完成某些职责任务所需要的访问权限来进行授权、管理，由系统管理员负责管理系统的角色集合和访问权限集合
- 基于属性的访问控制 Attribute-Based Access Control attribute-BAC
  - 根据主体属性、客体属性、环境条件、访问策略对主体的请求操作进行授权许可/拒绝

访问控制机制——以图书馆借书为例

- 自主访问控制模型 DAC
  - 客体的所有者按照自己的安全策略授予系统中的其他用户对其的访问权
  - 访问控制的规则维护完全下发到了所有者手上，管理员在理论上不需要对访问控制规则进行维护。
  - DAC 具备很高的灵活性，维护成本也很低。
  - 增加整体访问控制监管的难度，安全性完全取决于所有者的个人安全意识。

访问控制机制——以图书馆借书为例

- 强制访问控制模型 MAC
  - 系统根据主体和客体的安全属性，以强制方式控制主体对客体的访问
  - 例：机密性不能低读、高写；完整性不能高读、低写。
  - 是安全性最高的访问控制策略。但它对实施的要求也很高，需要对系统中的所有数据都进行标记。

访问控制机制——以图书馆借书为例

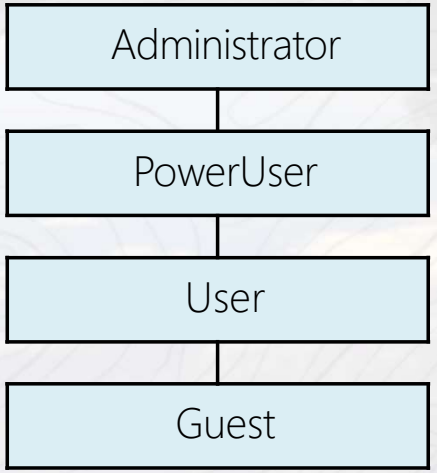
- 基于角色的访问控制模型 role-BAC
  - 根据完成某些职责任务所需要的访问权限来进行授权、管理，由系统管理员负责管理系统的角色集合和访问权限集合
  - 适合在管理员集中管理的时候进行使用。在这种情况下，所有的权限都由管理员进行分配和变更，role-BAC能降低管理员的工作难度，提高工作效率。
  - 是防止权限泛滥，实现最小特权原则的经典解决方案。

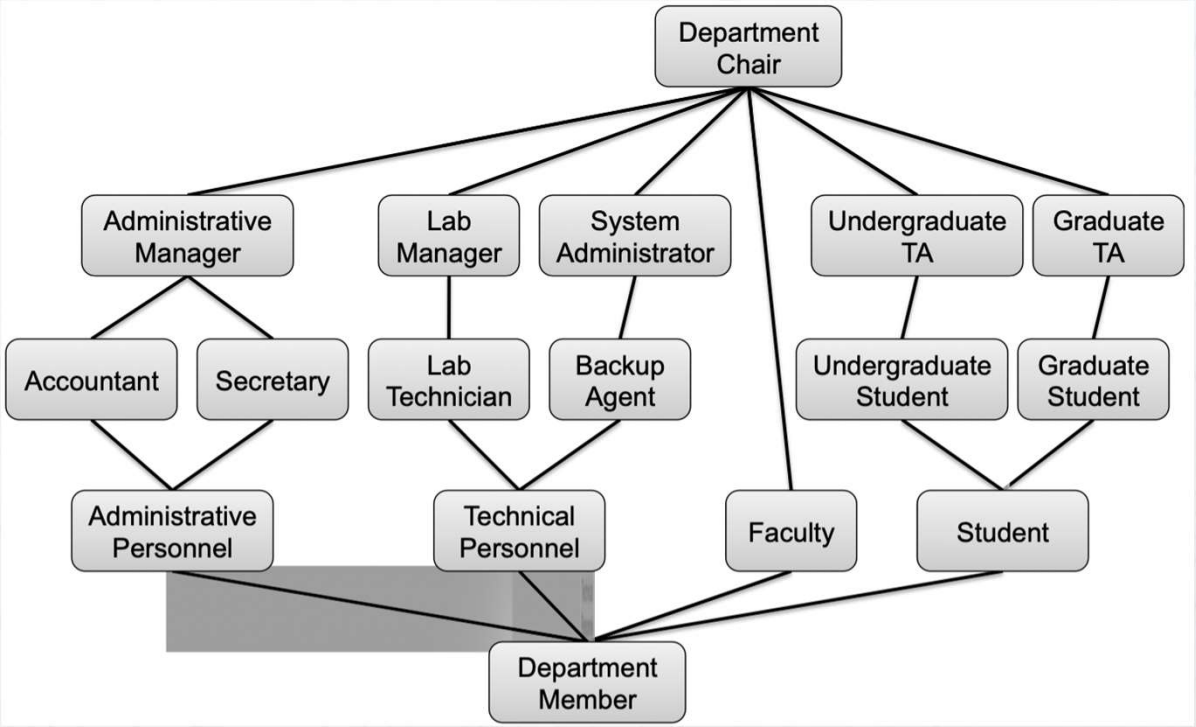
访问控制机制——以图书馆借书为例

- 基于属性的访问控制 attribute-BAC
  - 根据主体属性、客体属性、环境条件、访问策略对主体的请求操作进行授权许可/拒绝
  - 是针对请求本身制定的访问控制策略。
  - 需要定义是“默认通过”还是“默认拒绝”，为了保障更高的可用性，应用会采取“默认通过”的策略。
  - 适合在复杂场景下提供访问控制保护
  - 例：防火墙根据定义的规则（IP，协议类型、设备型号等），来判定是否允许主体访问。

访问控制机制——基于角色的访问控制 Role-based Access Control

- 角色（role）也称为组（group）
  - 分级：Administrator, PowerUser, User, Guest
  - 为角色分配权限，每个用户都有对应的角色





访问控制机制——基于角色的访问控制 Role-based Access Control

- 遵循以下三个基本的安全原则:
  - 最小特权（Least Privilege)
  - 责任分离（Separation of Duty)
  - 数据抽象（Data Abstract)

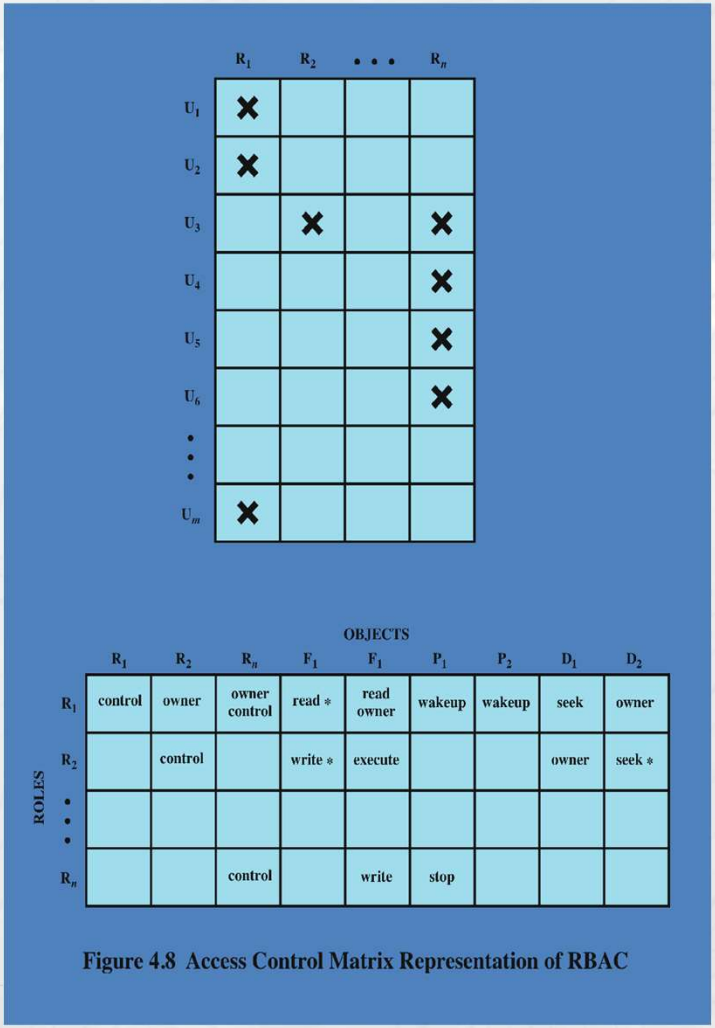


Figure 4.8 Access Control Matrix Representation of RBAC

访问控制机制

访问控制	特点	关注对象	适用场景	案例
DAC	自主控制	关注客体的权限列表	由用户自主控制权限	Linux
MAC	基于标签	关注主体、客体、请求的标签	能够对全部数据打上标签	政府系统
Role-BAC	基于角色	关注主体的权限列表	管理员进行集中权限管控	公司内部系统
Attribute-BAC	基于属性	关注主体、客体、请求的属性	无法清晰定义角色的复杂场景	网络请求

操作系统中的访问控制机制——Windows系统

Windows访问控制模型有两个基本部分：

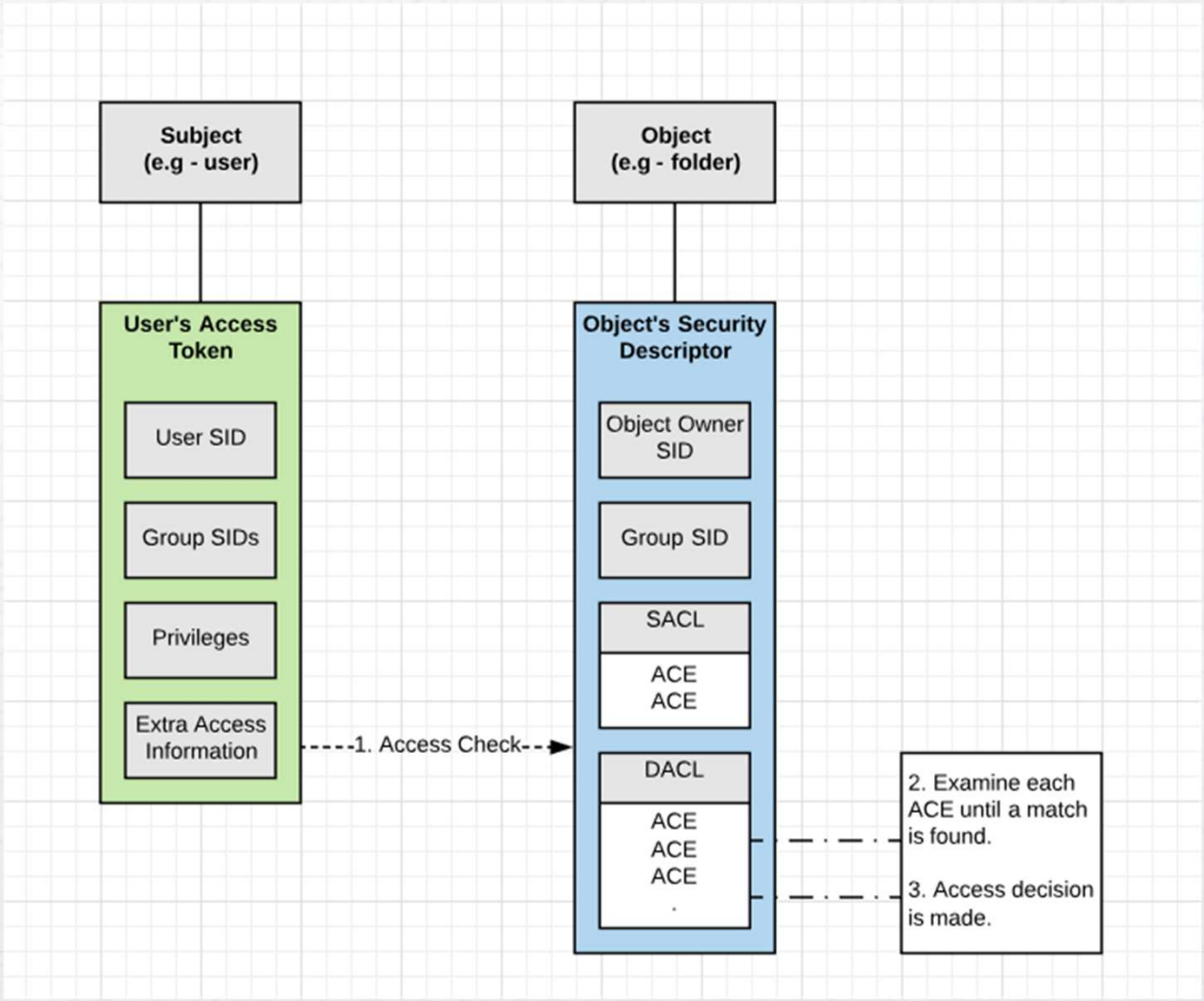
- 访问令牌（Access tokens）：包含了有关已登录用户的信息（与特定的windows账户关联）
- 安全描述符（Security descriptors）：包含了保护安全对象的安全信息（与被访问对象关联）

Windows用户登录到系统时，WinLogon进程为用户创建访问令牌，包含用户帐户、用户所属组、用户及其所属组的权限列表。

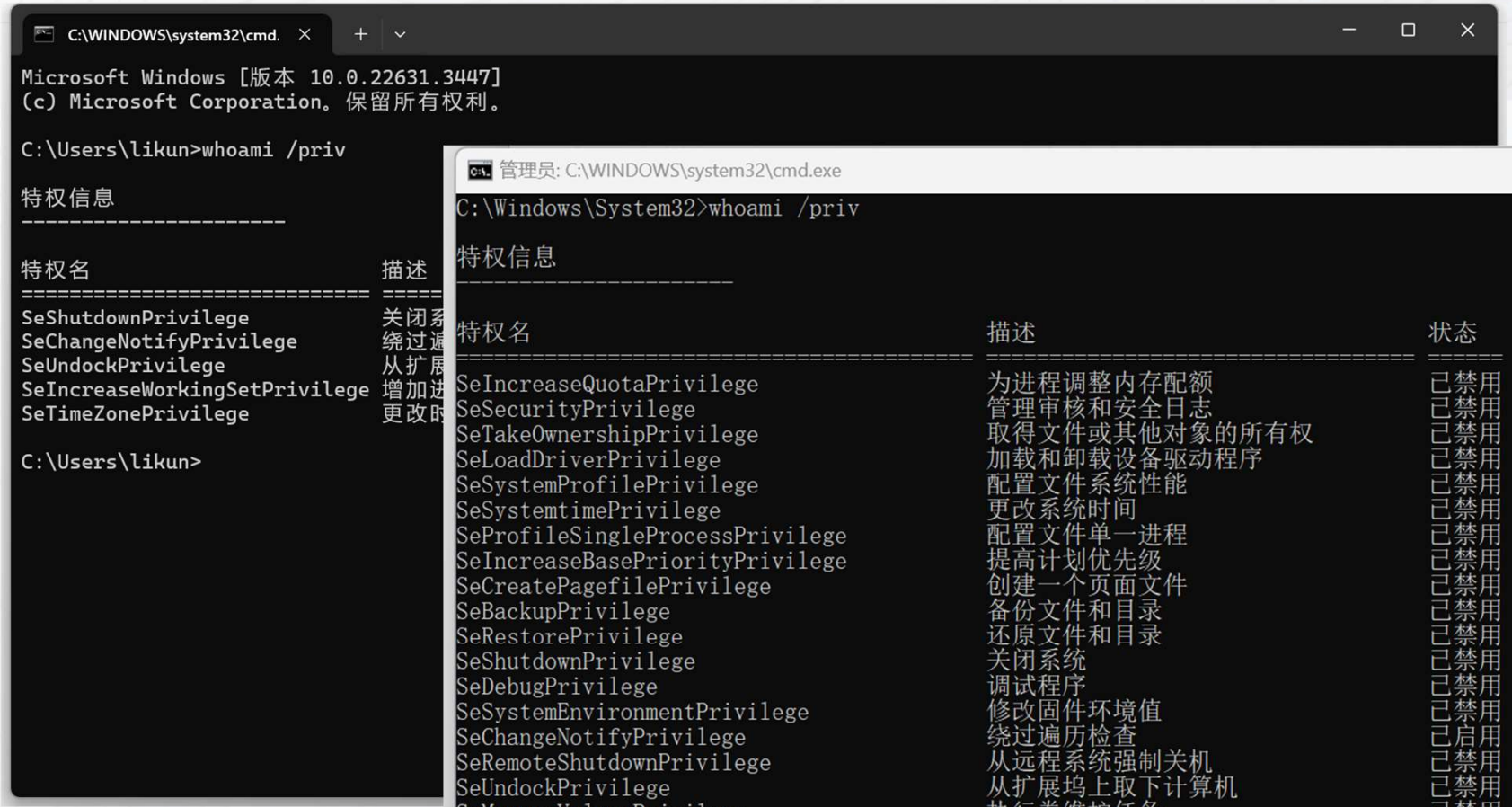
当进程尝试访问安全对象或执行需要特权的系统管理任务时，系统将使用此令牌来标识关联的用户是否拥有相应的权限。创建安全对象后，系统会为其分配安全描述符，指出对象的所有者，并且还可以包含DACL和SACL。

用户及所属组的安全标识符 SID	作为用户的身份标识
文件等客体的自主访问控制列表 DACL	标明谁有权访问
系统访问控制列表 SACL	标明哪些主体的访问需要被记录

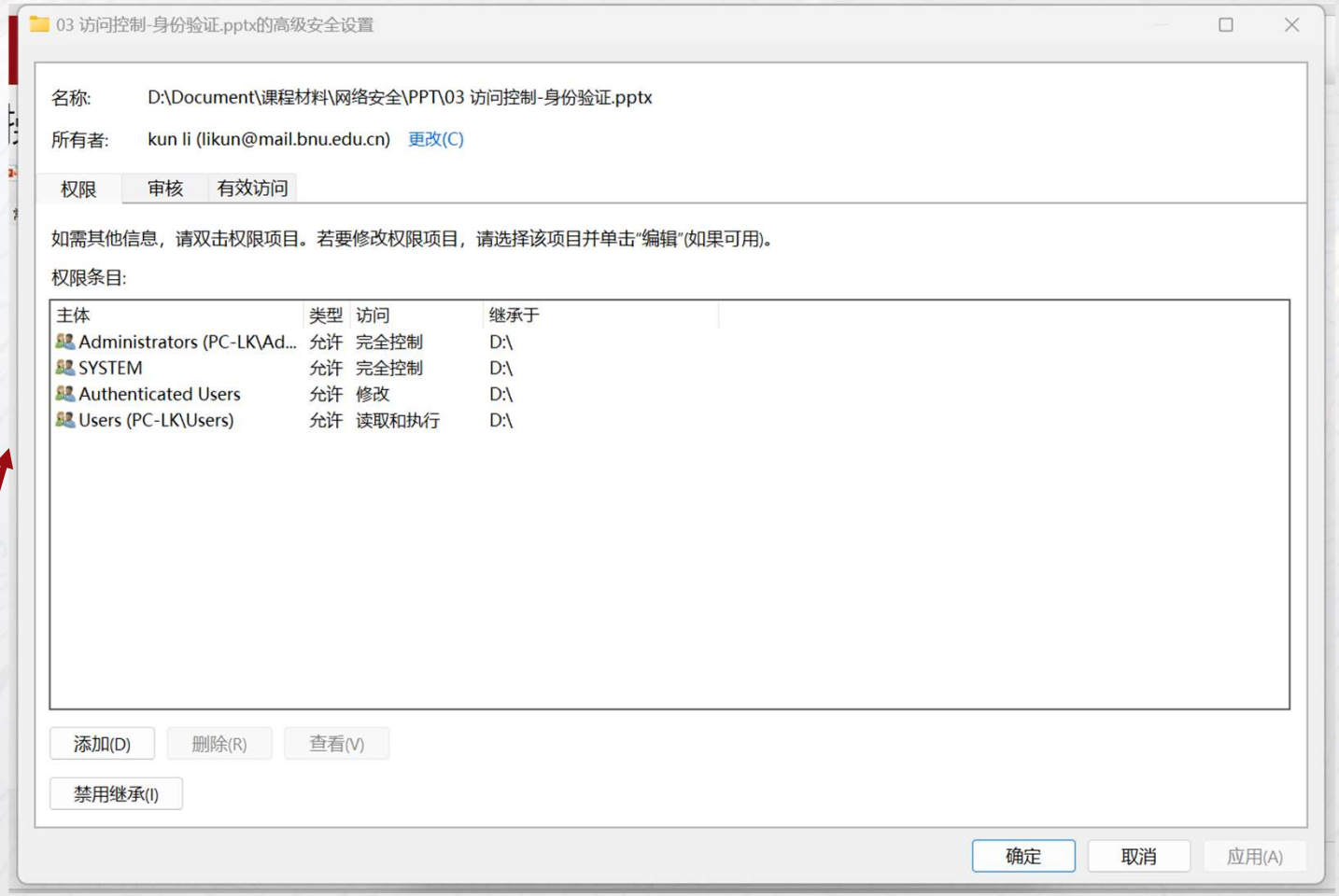
操作系统中的访问控制机制——Windows系统



操作系统中的访问控制机制——Windows系统



操作系统中的访问控制机制——Windows系统



操作系统中的访问控制机制——Windows系统

- a 表示存档状态。
- r 表示只读文件。
- h 表示隐藏文件。
- s 表示系统文件。
- dir –Force可查看隐藏文件


Any Questions?

身份识别

什么是身份识别（Authentication）

- 身份识别技术是信息安全的一项关键技术，身份识别包括用户向系统出示自己的身份证明和系统查核用户的身份证明两个过程，它们是判明和确定通信双方真实身份的重要环节
- 有效的身份识别是信息安全的保障，在信息的访问或使用中，必须有严格的身份验证保证信息及信息系统的安全，以保障授权用户的权利。
- 过程：
  - 识别：向安全系统提供识别码，是系统应用的基本组成部分和主要防御线
  - 验证：呈现或生成验证信息，证实实体与标识符之间的绑定关系，是访问控制和用户责任制的基础

身份识别与密钥分发

- 在实际应用中，身份识别跟密钥分发紧密联系在一起。身份识别可以分为双向鉴别和单向鉴别。
  - 双向鉴别是双方要互相向对方证明自己的身份，一般适用于通信双方同时在线的情况；
  - 单向鉴别是只要一方向对方证明自己的身份，如登录邮件服务器，只需用户向服务器证明自己是授权用户即可。

身份识别中的攻击

1. 拒绝服务denial-of-service：试图通过大量的身份验证尝试来禁用用户身份验证服务。
2. 窃听eavesdropping：对手试图通过某种涉及用户和对手物理距离的攻击来获取密码
3. 主机攻击：针对主机上存储密码、令牌密码或生物识别模板的用户文件的攻击
4. 重放攻击：对手重复之前捕获的用户响应
5. 客户端攻击：对手试图在无法访问远程主机或中间通信路径的情况下实现用户身份验证
6. 木马：一种应用程序或物理设备伪装成真实的应用程序或设备，目的是获取用户密码、通过码或生物识别信息

身份识别中的攻击——密码破解

密码长度很重要

普渡大学的一项研究在 54 台机器上进行，代表大约 13,000 个用户帐户。大约 3% 的密码是三个字符。或更少。

Length	Number	Fraction of Total
1	55	.004
2	87	.006
3	212	.02
4	449	.03
5	1260	.09
6	3035	.22
7	2917	.21
8	5772	.42
Total	13787	1.0

身份识别中的攻击——密码破解

密码复杂性很重要
许多人在被允许选择自己的密码时，会选择一个可猜测的密码。比如他们自己的名字，他们的街道名称，一个常用的字典词，等等。

一般猜测策略：

1. 尝试用户的姓名、姓名缩写、账户名和其他相关个人信息
2. 尝试使用各种字典中的单词（可在网上查阅）
3. 尝试步骤 2 中单词的各种排列组合。
   - 第一个字母大写或控制字符、整个单词大写、单词颠倒、将字母 "o " 改为数字 "0 "等等。
4. 对步骤 2 中的单词尝试步骤 3 中未考虑的各种大写排列组合。

一个密码字典的例子：https://github.com/wwl012345/PasswordDic

身份识别中的攻击——密码破解

通过猜测策略，来自 13,797 个样本集的四分之一密码账户被猜到了

Type of Password	Search Size	Number of Matches	Percentage of Passwords Matched	Cost/Benefit Ratio[a]
User/account name	130	368	2.7%	2.830
Character sequences	866	22	0.2%	0.025
Numbers	427	9	0.1%	0.021
Chinese	392	56	0.4%	0.143
Place names	628	82	0.6%	0.131
Common names	2239	548	4.0%	0.245
Female names	4280	161	1.2%	0.038
Male names	2866	140	1.0%	0.049
Uncommon names	4955	130	0.9%	0.026
Myths and legends	1246	66	0.5%	0.053
Shakespearean	473	11	0.1%	0.023
Sports terms	238	32	0.2%	0.134
Science fiction	691	59	0.4%	0.085
Movies and actors	99	12	0.1%	0.121
Cartoons	92	9	0.1%	0.098
Famous people	290	55	0.4%	0.190
Phrases and patterns	933	253	1.8%	0.271
Surnames	33	9	0.1%	0.273
Biology	58	1	0.0%	0.017
System dictionary	19683	1027	7.4%	0.052
Machine names	9018	132	1.0%	0.015
Mnemonics	14	2	0.0%	0.143
King James bible	7525	83	0.6%	0.011
Miscellaneous words	3212	54	0.4%	0.017
Yiddish words	56	0	0.0%	0.000
Asteroids	2407	19	0.1%	0.007
TOTAL	62727	3340	24.2%	0.053

身份识别中的攻击——设计密码的思路

选择密码的一个好方法是使用短语中每个单词的第一个字母。

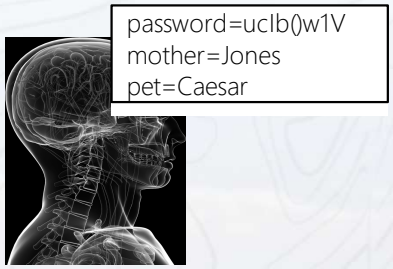
但是，不要选择像“ An apple a day keeps the doctor away ”---> (Aaadktda)这样众所周知的短语。而是用更不为人知的短语，例如：

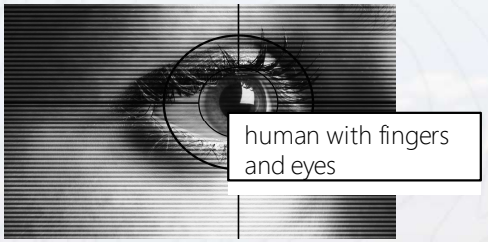
" My dog's first name is Rex "---> (MdfniR)
" My sister Peg is 24 years old "---> (MsPi24yo)。

研究表明，用户一般都能记住这样的密码，但他们不容易受到基于常用密码的密码猜测攻击。

身份识别的三个依据

- 角色所持有的(like a smart card or a radio key fob storing secret keys),
- 角色所知道的 (like a password),
- 角色固有特征(like a human with a fingerprint).





身份识别的四种方法

- 角色所持有的：钥匙、智能卡
- 角色所知道的：密码、预设问题的答案
- 角色所固有的：指纹、视网膜、人脸
- 角色所做的事：语音识别、手写字迹、打字节奏

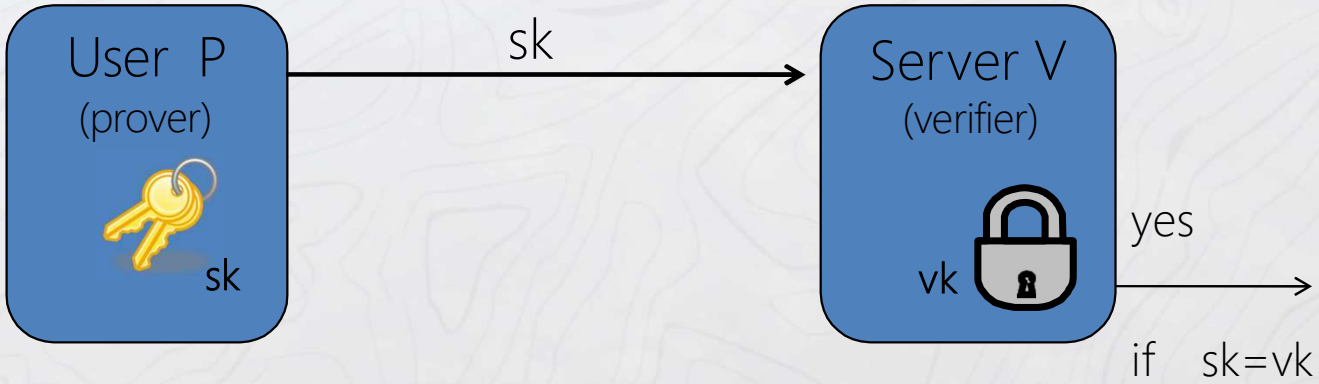
使用密码的身份识别

- 用户提供账户名与密码，系统将密码与指定登录名存储的密码进行比较
- 账户名（ID）的意义：
  - 确定用户是否有权访问系统
  - 确定用户的权限
  - 用于自行决定的访问控制（DAC）

Alice	pw_{alice}
Bob	pw_{bob}

基础密码协议

- 有限的密钥集 PWD
- 算法 G (KeyGen):
  - 选择随机密钥 pw in PWD.
  - 输出 sk = vk = pw.
- 存在问题：明文存储不安全



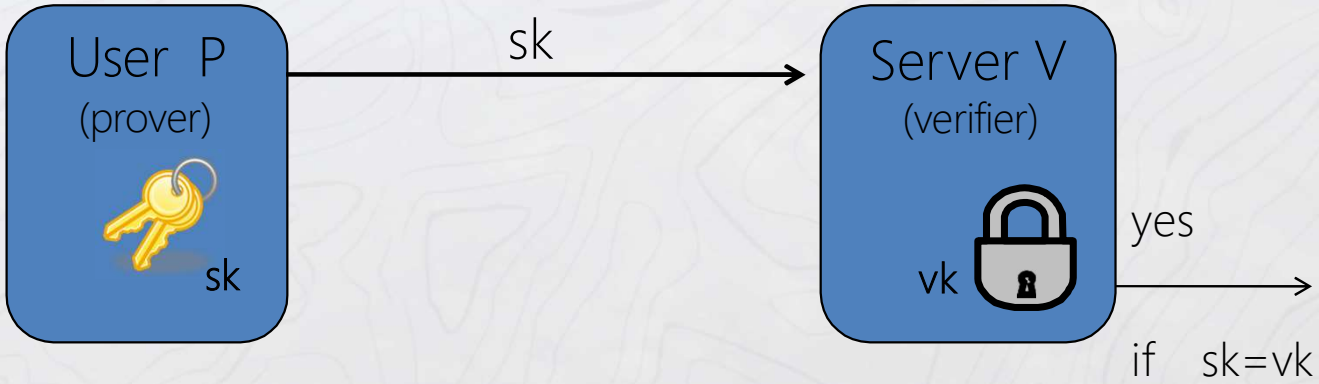
改进的密码协议 version 1

- 对密码用哈希算法并存储哈希值
- 存在问题：
  - 弱密码不安全
  - 字典攻击、彩虹表攻击

Algorithm	Speed/sec
DES	2 383 000
MD5	4 905 000
LanMan	12 114 000

一个密码破解工具的例子
https://www.openwall.com/john/

Alice	$H(pw_A)$
Bob	$H(pw_B)$



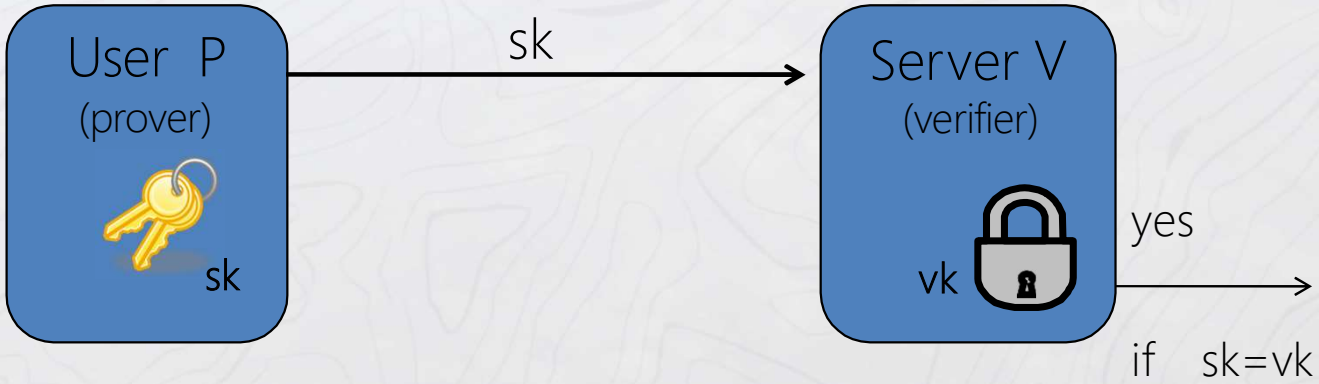
改进的密码协议 version 2

- 对密码先加盐，并存储每个用户的盐，再对加了盐的密码用哈希算法并存储哈希值
- 当用户登录时：
  - 用户提供ID和密码
  - 操作系统OS使用ID索引到密码文件，并检索文本盐和哈希密码
  - salt和用户提供的密码用作哈希的输入

id	S	h
Alice	S_A	$H(pw_A, S_A)$
Bob	S_B	$H(pw_B, S_B)$

优势：

- 防止在密码文件中看到重复的密码
- 增加了字典攻击的难度



改进的密码协议 version 3

- 慢哈希：一种计算过程非常慢的哈希算法 bcrypt(SHA512(password), salt, cost)

优势：

- 安全性高：由于Bcrypt采用了salt和cost两种机制，它可以有效地防止彩虹表攻击和暴力破解攻击，从而保证密码的安全性。
- 灵活性强：Bcrypt算法可以根据实际情况进行调整，可以设置不同的cost值和salt值，从而满足不同的安全需求。
- 易于使用：Bcrypt算法已经被广泛应用于各种编程语言和操作系统中，使用起来非常方便。

劣势：

- 运算速度较慢：由于Bcrypt算法需要进行多次哈希运算，所以它的运算速度比其他密码哈希函数要慢一些，从而可能影响系统的性能。
- 不可逆：Bcrypt算法是一种单向哈希函数，不能够将哈希值转换回原始密码。这也就意味着，一旦密码被哈希后，就无法再次获取明文密码。

操作系统中的密码方案——UNIX

生成新密码：

1. 密码加盐
2. 慢哈希运算
3. 建立ID-Salt-Hash的存储表

密码验证：

1. 存储表中检索输入的ID
2. 输入的密码加检索到的对应盐
3. 慢哈希运算
4. 对比运算结果与存储表中检索到的对应哈希值是否相同

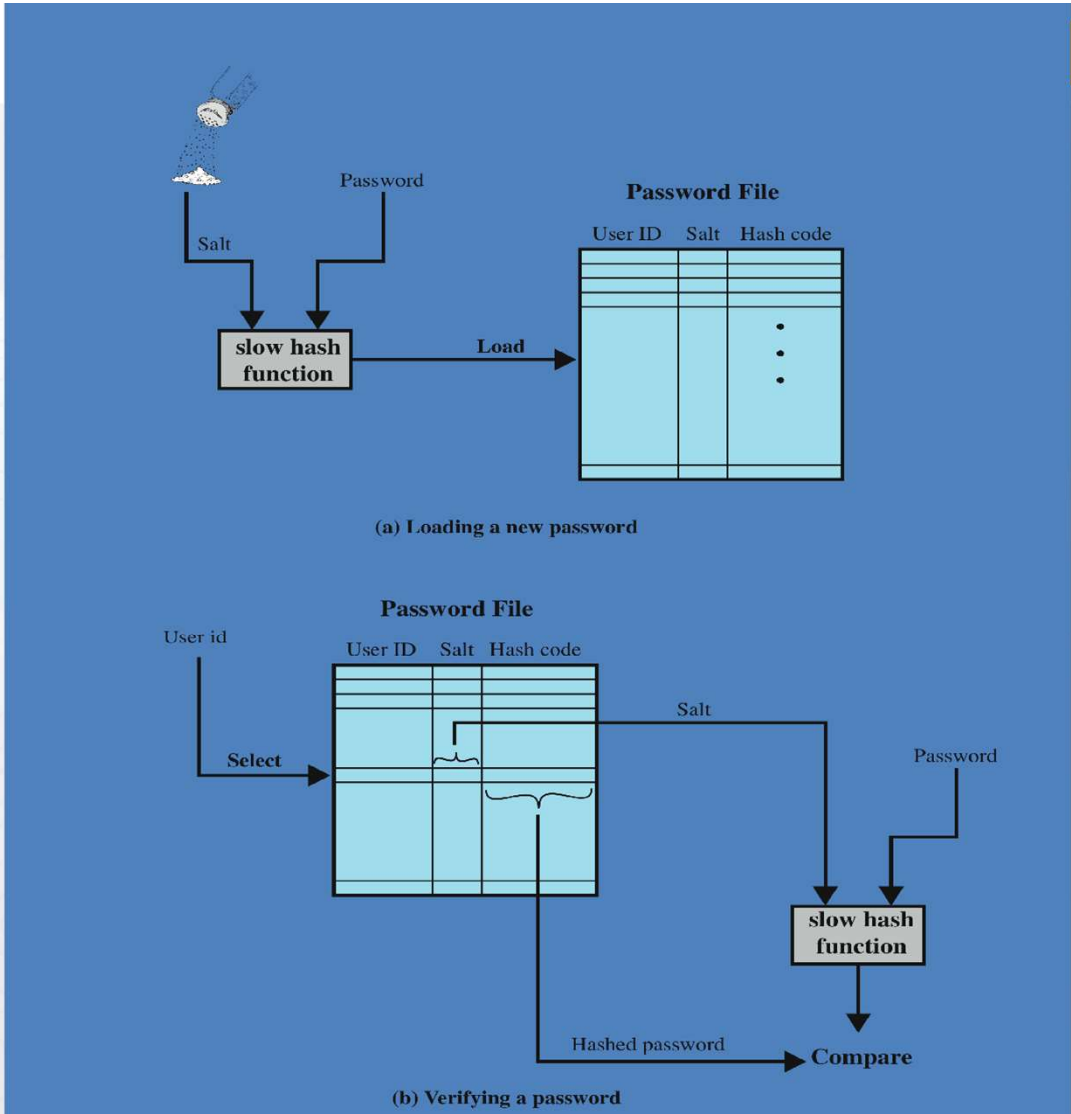
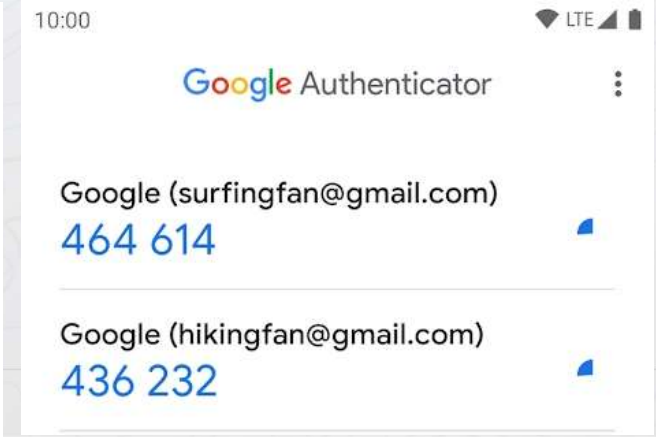


Figure 3.1 UNIX Password Scheme

一次性密码的身份识别

- 一次性密码 (one-time-password, OTP) 是一种用于验证用户身份的身份验证方法。 它被用作确保在线帐户和交易安全的附加安全措施。
- 场景案例：登录账号、找回密码，更改密码和转账操作
- 常用到的方式有
  - 手机短信+短信验证码；
  - 邮件+邮件验证码；
  - 认证器软件+验证码，比如Microsoft Authenticator App，Google Authenticator App等等；
  - 硬件+验证码：比如网银的电子密码器。



一次性密码的身份识别

一次性密码生成过程：

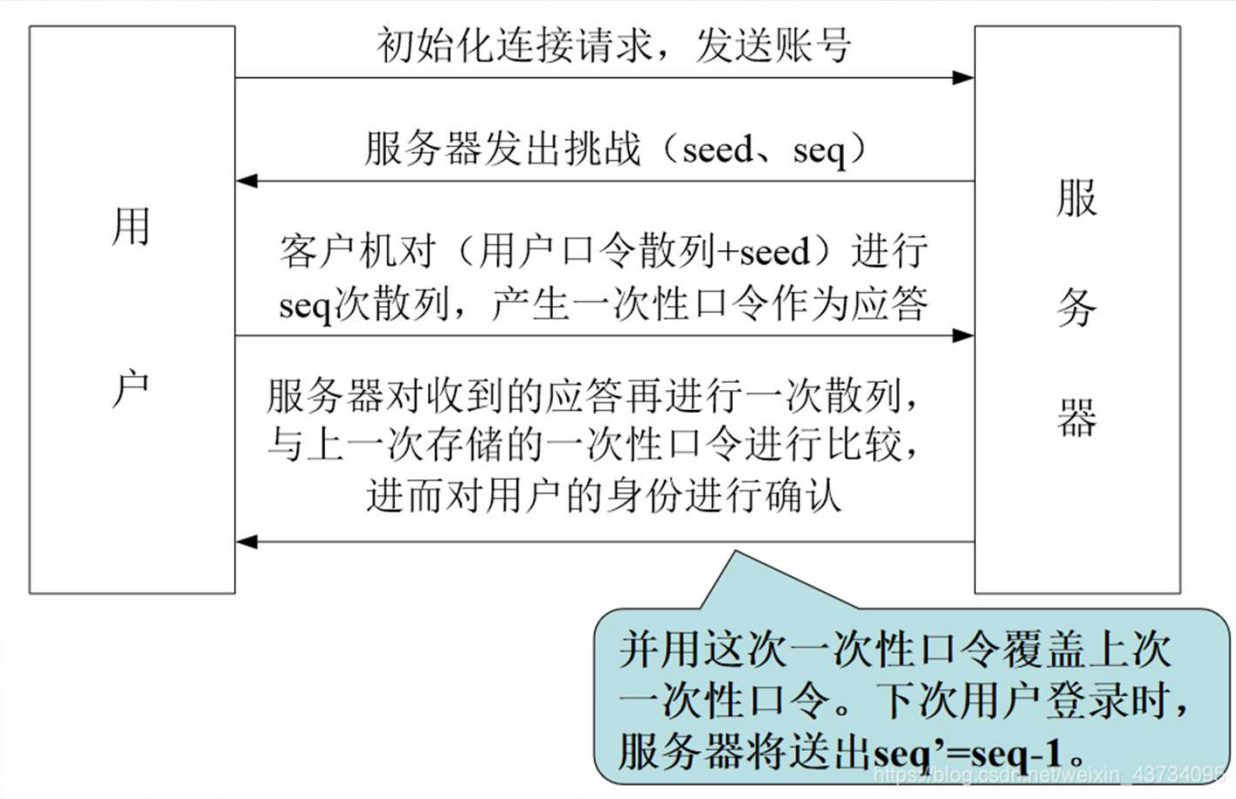
1. 密钥生成：首先，生成一个密钥（Key），这是一个随机生成的字符串，通常为16字节长。密钥用于加密后续生成的随机数。
2. 随机数生成：使用密钥加密一个随机数（Random Number），这个随机数通常为32字节长。加密算法可以是AES、SHA-256或其他强加密算法。加密后的随机数称为加密随机数（Encrypted Random Number）。
3. OTP生成：将加密随机数分割成若干段，每段通常为4字节长。每段加密随机数转换为十六进制字符串，并拼接在一起形成OTP。
4. 校验码生成：从加密随机数的最后几位生成校验码。校验码用于验证OTP的正确性。
5. OTP发送：将OTP和校验码一起发送给用户。用户需要输入OTP和校验码以完成身份验证。

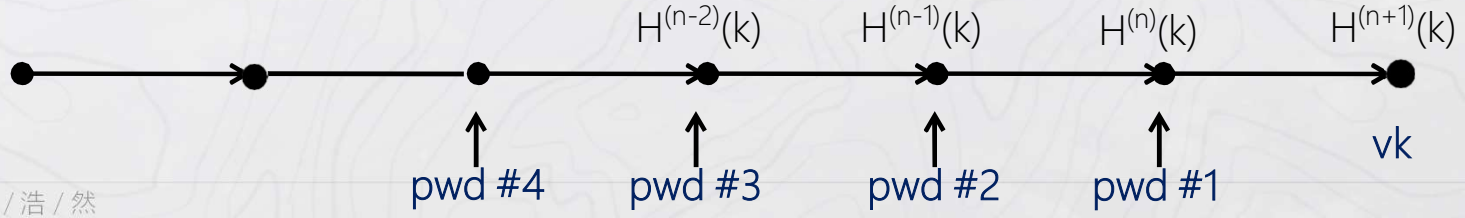
一次性密码的身份识别——基于时间的一次性密码算法TOTP

利用当前时间作为算法的输入之一。时间被分成固定长度的时间段，通常是30秒或60秒。这意味着生成的密码在此时间段内有效，过后将生成一个新的密码。

一次性密码的身份识别——S/KEY

基于MD4 和MD5的一次性口令生成方案。他可以对访问者的身份与设备进行综合验证。

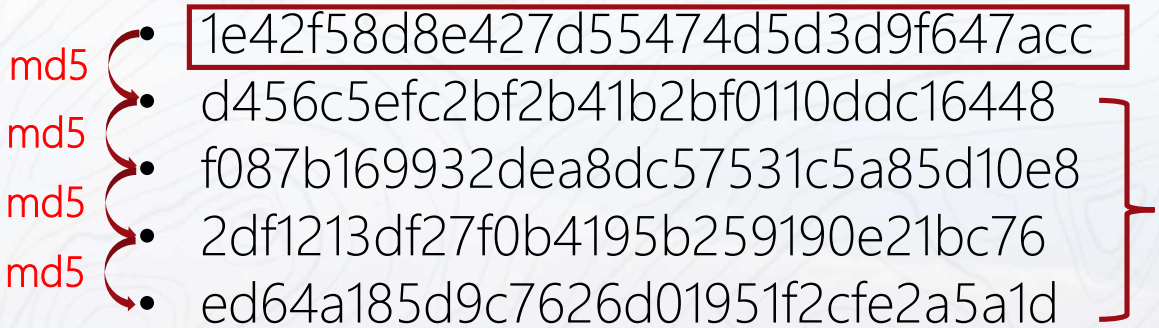




一次性密码的身份识别——S/KEY

例：用户输入账号2201110729，加盐和种子等随机数生成口令，服务器的seq=4

1. 生成五条口令：



2. 用户第一次登录时：用户输入第二条口令[d456c5efc2bf2b41b2bf0110ddc16448]，服务器对存储口令[1e42f58d8e427d55474d5d3d9f647acc]哈希一次，与用户输入进行比对，比对通过后，用户可登录。seq-1变为3，服务器用哈希后的口令覆盖存储口令。

3. 用户第二次登录时：用户输入第三条口令[f087b169932dea8dc57531c5a85d10e8]，服务器仍然是对存储口令哈希、比对、覆盖，seq-1变为2。

4. 像这样一直到第四次登录，用完所有的口令，再重新生成新的口令。

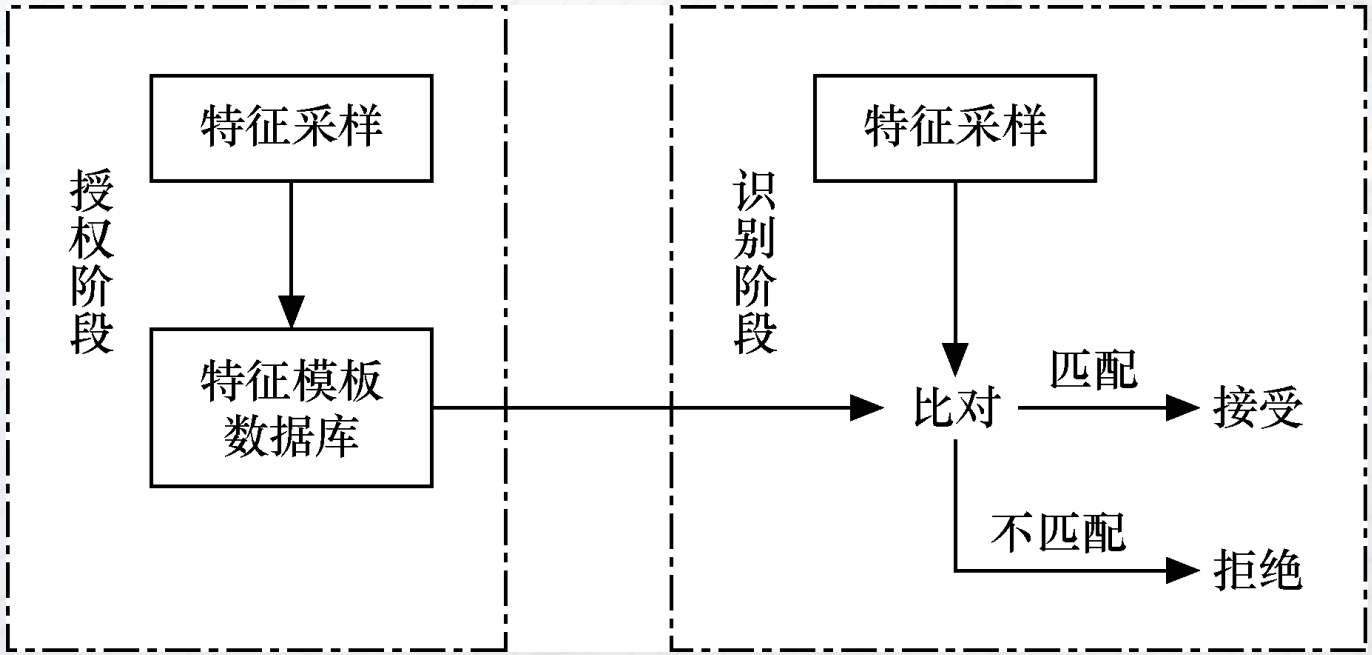
一次性密码的身份识别：S/Key V.S. RSA SecureID

特征	S/Key	RSA SecureID
类型	基于事件的一次性密码系统	既有基于时间也有基于事件的一次性密码系统
算法	基于哈希函数（如MD4, MD5）	使用AES或3DES等加密算法
安全性	较低，使用单向哈希链，安全性取决于哈希函数的抗碰撞性	较高，使用加密算法生成一次性密码，且有物理或软件令牌加密存储密钥
用途	主要用于远程登录系统	广泛用于企业级的两步验证，如网络访问、文件加密等
用户体验	用户需要记住并输入一系列的密码	用户输入PIN码和令牌显示的密码
设备需求	无需专门设备，可以通过软件实现	需要专门的硬件令牌或者安装软件令牌
密码更新	每次验证后使用的密码都从哈希链中移除，需定期更新整个链	密码定期自动更新，无需用户干预
实现成本	较低，主要是软件成本	较高，可能需要购买硬件令牌或软件许可证
适用环境	适用于需要基本登录安全的环境	适用于需要高安全性的环境，如金融服务、政府机构等


Any Questions?

使用生物识别技术的用户身份验证/识别

- 基于物理特征的模式识别
- 与密码和令牌相比在技术上既复杂又昂贵
- 使用的物理特性包括：
  - 面部特征
  - 指纹
  - 掌纹
  - 视网膜图案
  - 虹膜
  - 签名
  - 声音



使用生物识别技术的用户身份验证/识别

- 并不是所有的生物特征都可用来进行身份识别，只有满足以下条件的生理或行为特征才可以用来作为身份识别的依据
  - 普遍性：每个人都应该具有该特征。
  - 唯一性：每个人在该特征上有不同的表现。
  - 稳定性：该特征相对稳定，不会随着年龄等变化。
  - 易采集性：该特征应该容易被测量。
  - 可接受性：人们是否接受以该特征作为身份识别。

生物识别实例

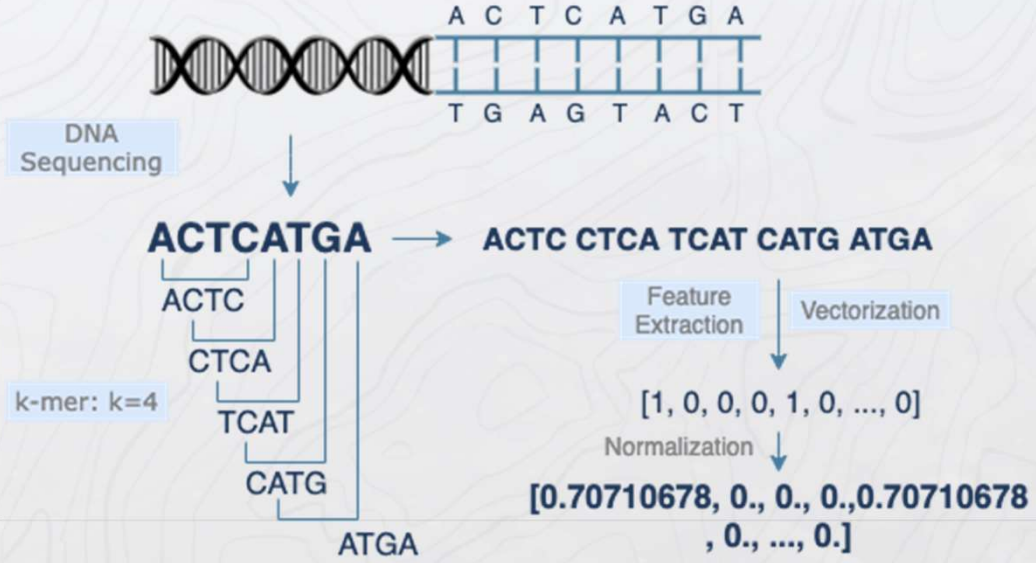
- 指纹
  - 指纹识别是最传统、最成熟的生物鉴定方式。目前，全球范围内都建立有指纹鉴定机构以及罪犯指纹数据库，指纹鉴定已经被官方所接受，成为司法部门有效的身份鉴定手段。
  - 指纹识别处理包括对指纹图像采集、指纹图像处理与特征提取、特征值的比对与匹配等过程。
  - 许多研究表明，指纹识别在所有生物特征识别技术中是对人体最不构成侵犯的一种技术手段。其优点如下
    - 独特性
    - 稳定性
    - 方便性

生物识别实例

- 虹膜
  - 人眼虹膜位于眼角膜之后，水晶体之前，是环形薄膜。其图样具有个人特征，可以提供比指纹更为细致的信息，因此可以作为个人身份识别的重要依据。可以使用一台摄像机在35 ~ 40 cm的距离内采样，然后由软件对所得数据与存储的模板进行比对。每个人的虹膜结构各不相同，并且这种独特的虹膜结构在人的一生中几乎不发生变化。
- 虹膜 v.s. 视网膜 v.s. 眼纹
  - 虹膜识别识别的是眼睛的虹膜部分，即“黑眼仁儿”的纹理。
  - 眼纹识别则是识别眼睛的巩膜部分。即“眼白”的血管排布情况。
  - 视网膜是眼球背部一层非常薄的细胞层，视网膜识别的是视网膜上的血管分布。

生物识别实例

- DNA
  - DNA（脱氧核糖核酸）存在于一切有细胞核的动、植物中，生物的全部遗传信息都存储在DNA分子里。DNA结构中的编码区，即遗传基因或基因序列部分占DNA全长的3%～10%，这部分即遗传密码区。
  - 随着生物技术的发展，尤其是人类基因研究的重大突破，研究人员认为DNA识别技术将是未来生物特征识别技术发展的主流。



身份识别

- 每个要加入授权用户数据库的人都必须先在系统中注册。
  - 类似于为用户分配密码。
  - 系统为用户保留一个名称（ID），或许是一个 PIN 码或密码，以及生物识别值。
- 根据不同的应用，生物识别系统的用户认证包括验证或识别。
  - 在生物识别验证中，用户输入 PIN 码并使用生物识别传感器。
    - 系统提取相应的特征并与模板进行比较。
    - 匹配吗？---->系统验证用户身份。
  - 对于身份识别系统，个人使用生物识别传感器，但不提供其他信息。
    - 系统将提交的模板与存储的模板集进行比较。
    - 匹配吗？---->用户身份被识别。

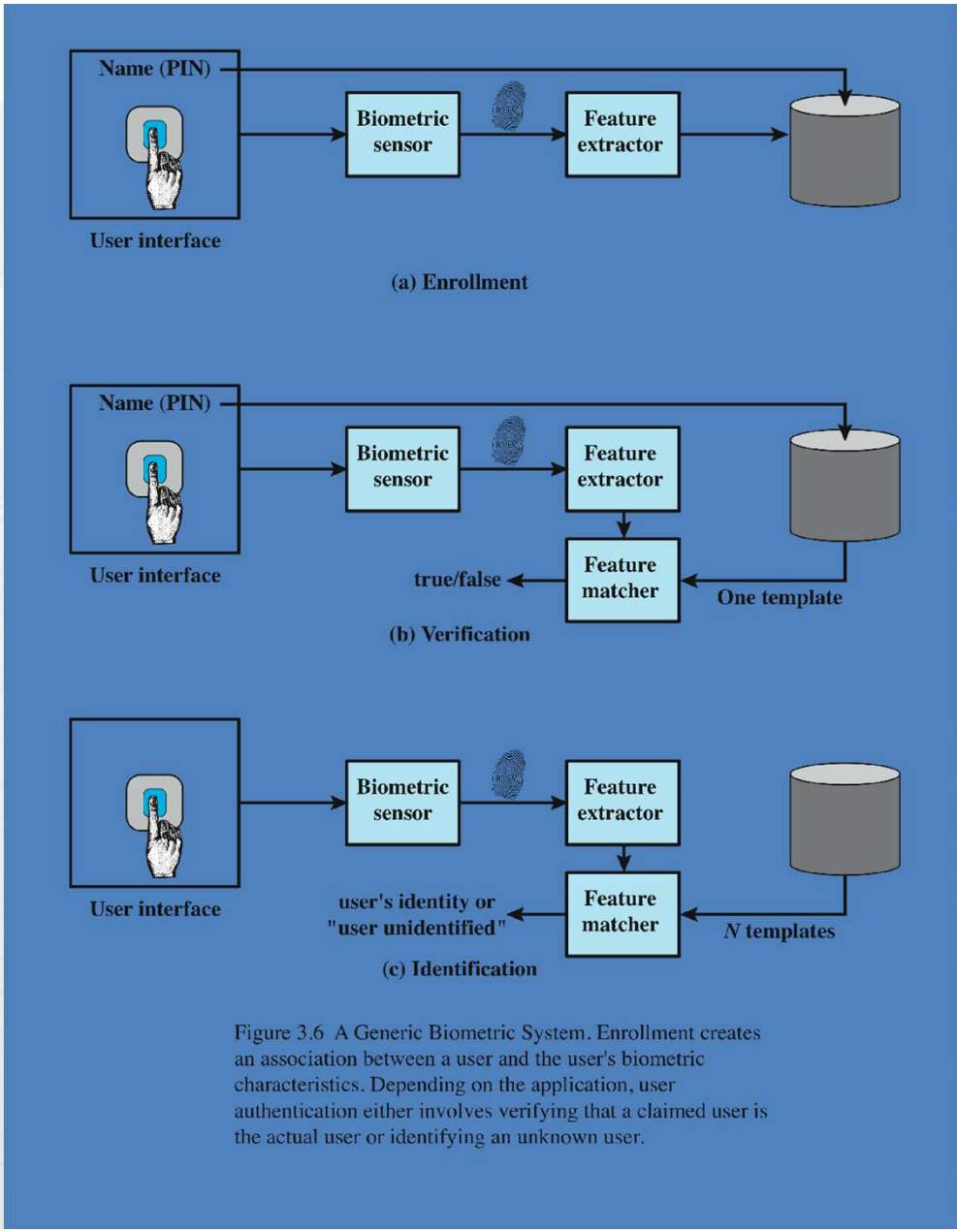


Figure 3.6 A Generic Biometric System. Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

学/无/止/境 气/有/浩/然

使用生物识别技术的利弊

- 好处:
  - 难以忘记
  - 个人独特
- 问题:
  - 生物特征识别通常不是秘密的
  - 与密码不同，无法更改
- 主要用作第二因素认证

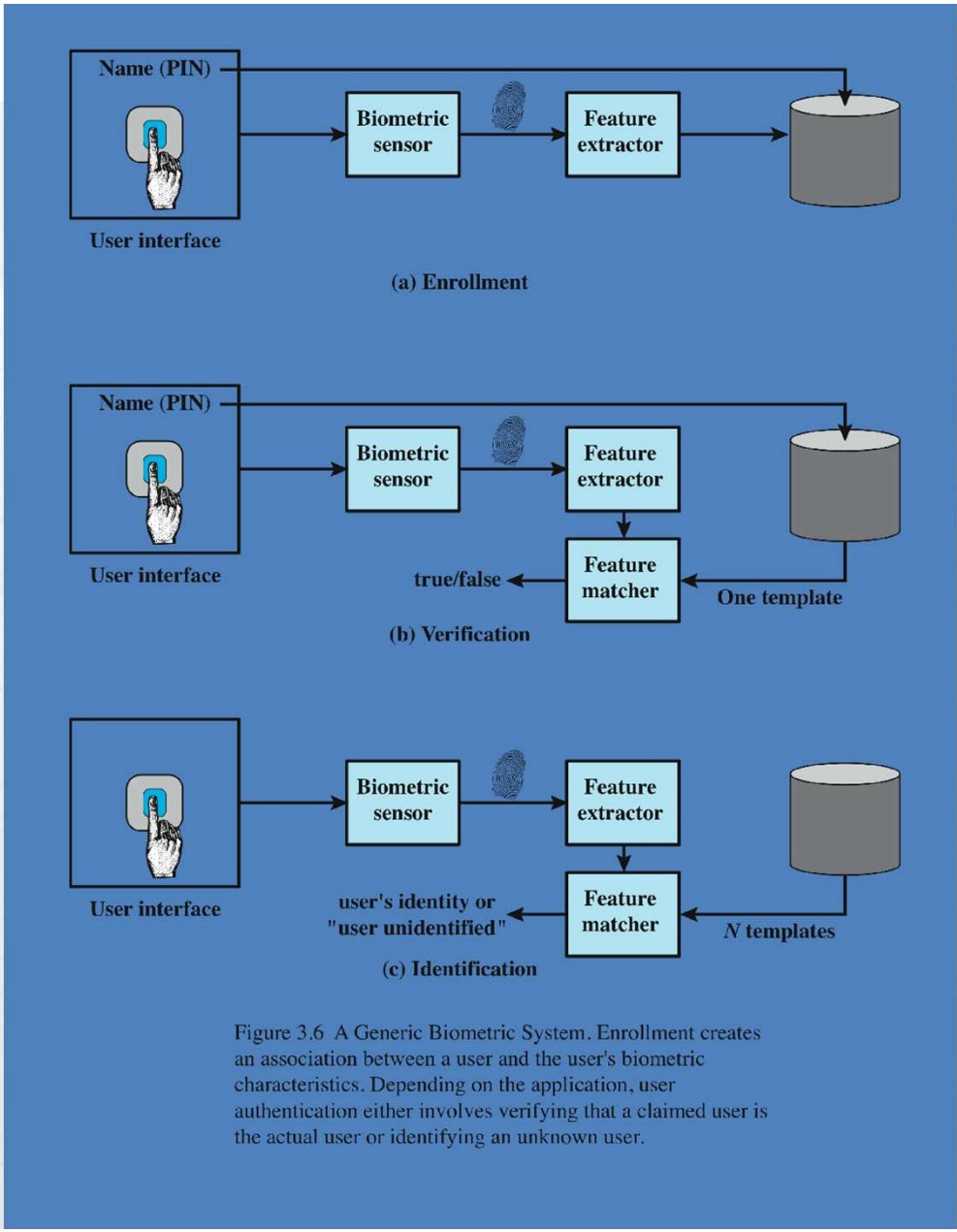
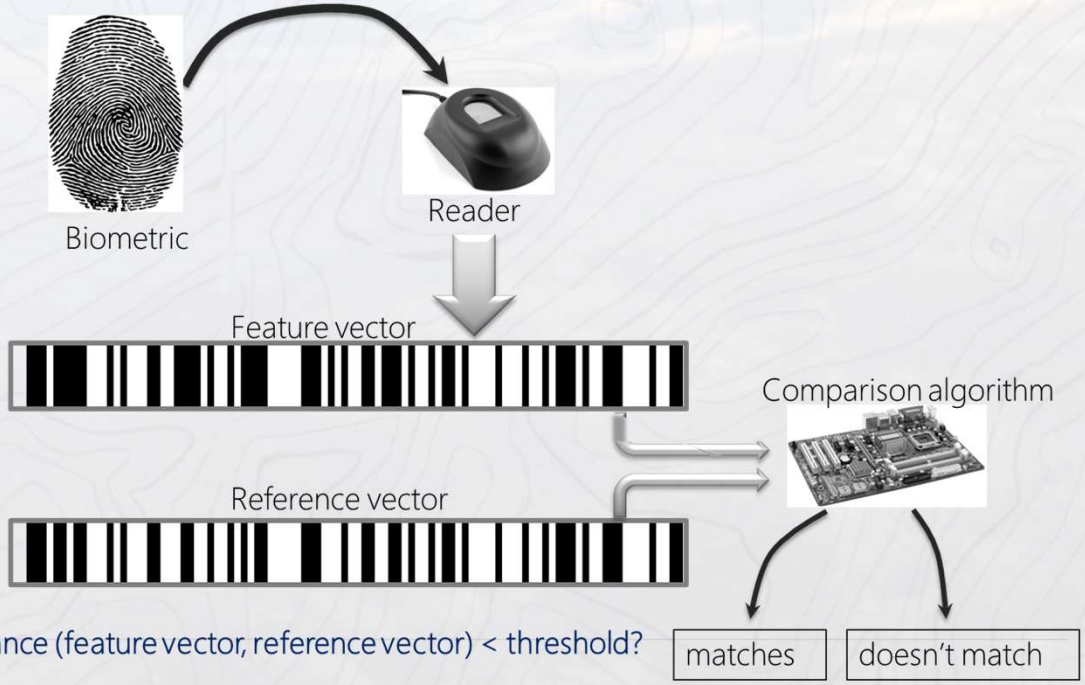


Figure 3.6 A Generic Biometric System. Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

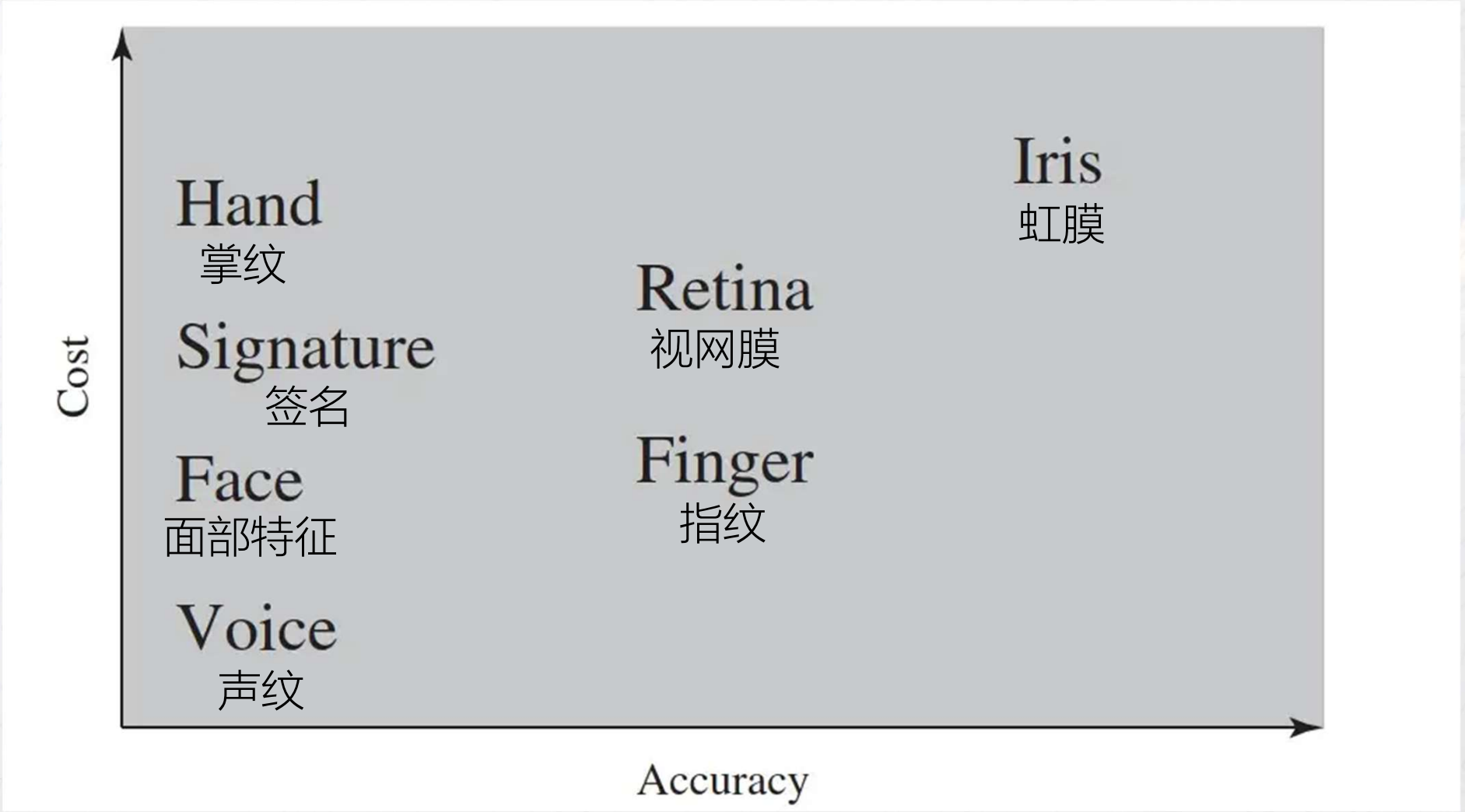
生物识别准确性

- 在任何生物识别方案中，个人的某些物理特征都被映射成数字表示。对于每个人，计算机中都存储有一个数字表示或模板。
- 当要对用户进行身份验证时，系统会将存储的模板与显示的模板进行比较。鉴于物理特征的复杂性，我们不能期望两个模板之间完全匹配。
- 系统会使用一种算法生成一个匹配分数（通常是一个数字），量化输入模板和存储模板之间的相似度。

[-0.23, -0.54, ..., 0.27]



生物识别准确性


Hand
掌纹
Signature
签名
Face
面部特征
Voice
声纹
Retina
视网膜
Finger
指纹
Iris
虹膜
Cost
Accuracy

生物识别准确性的问题所在

如果系统多次测试单个用户，匹配分数 s 将随概率密度函数的变化而变化，通常会形成一条钟形曲线。

- 以指纹为例，结果可能会因传感器噪声、指纹因肿胀、干燥等原因造成的变化、手指位置等因素而不同。

平均而言，任何其他人的匹配分数都会低得多，但同样会呈现钟形概率密度函数。

难点在于，与给定的参考模板相比，两个人（一个是真指纹，一个是假指纹）的匹配分数范围很可能会重叠。

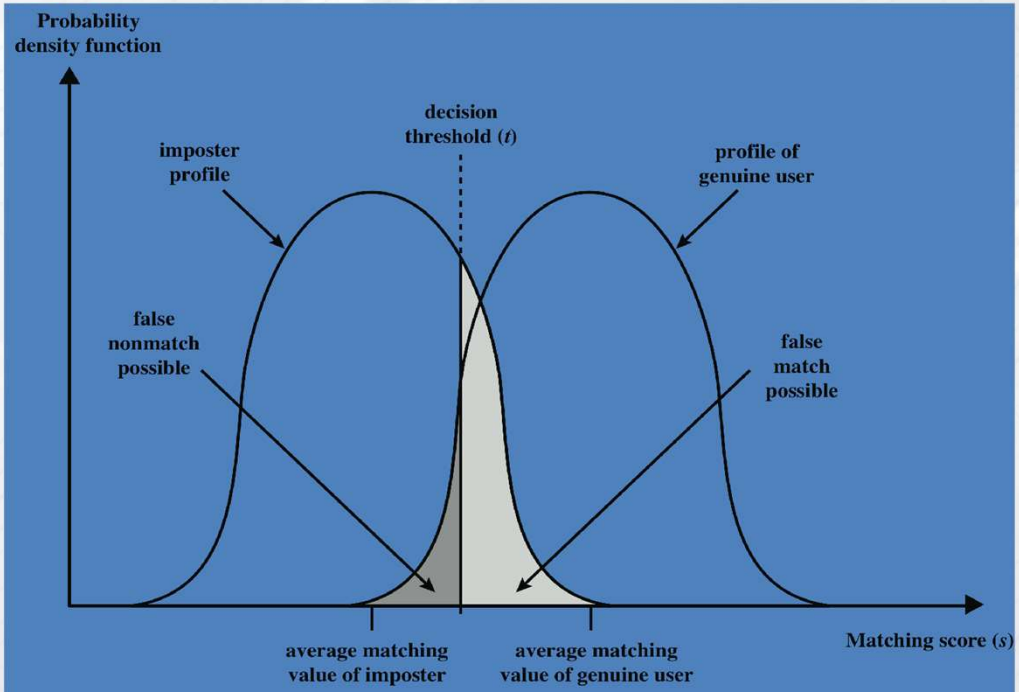
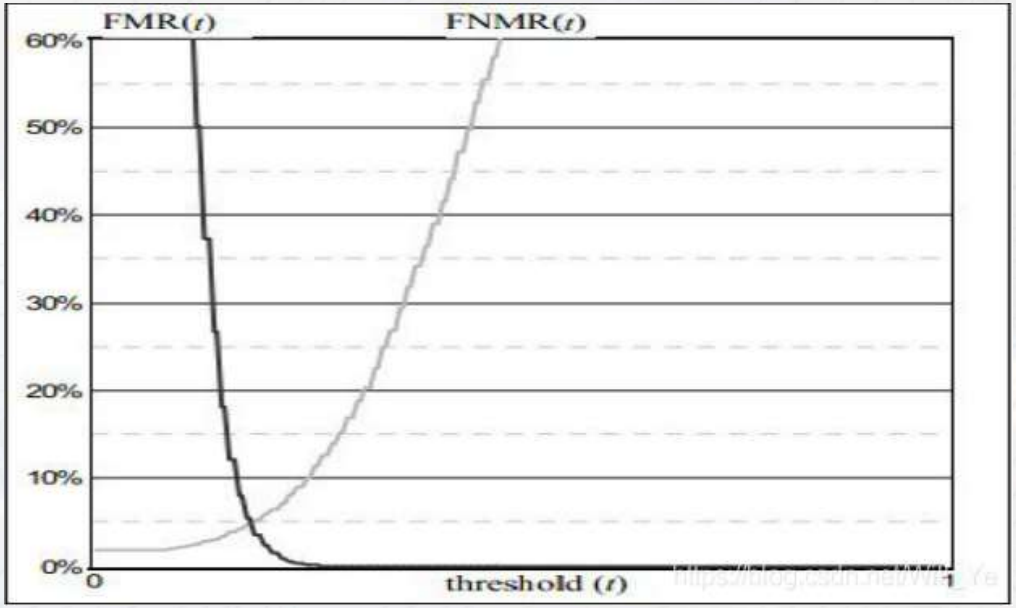


Figure 3.7 Profiles of a Biometric Characteristic of an Imposter and an Authorized Users In this depiction, the comparison between presented feature and a reference feature is reduced to a single numeric value. If the input value (s) is greater than a preassigned threshold (t), a match is declared.

生物识别准确性的阈值调整

虚假匹配率（FMR）也被称为假接受率（False Accept Rate, FAR），表示系统错误地接受非授权用户的概率。例如，如果一个非授权的人试图使用他人的指纹解锁一个设备，而系统错误地允许了这种行为，那么就发生了一次虚假匹配。

虚假非匹配率（FNMR）也被称为假拒绝率（False Reject Rate, FRR），表示系统错误地拒绝授权用户的概率。例如，一个用户使用自己的指纹尝试解锁自己的设备，如果系统错误地识别并拒绝了这次尝试，那么就发生了一次虚假非匹配。



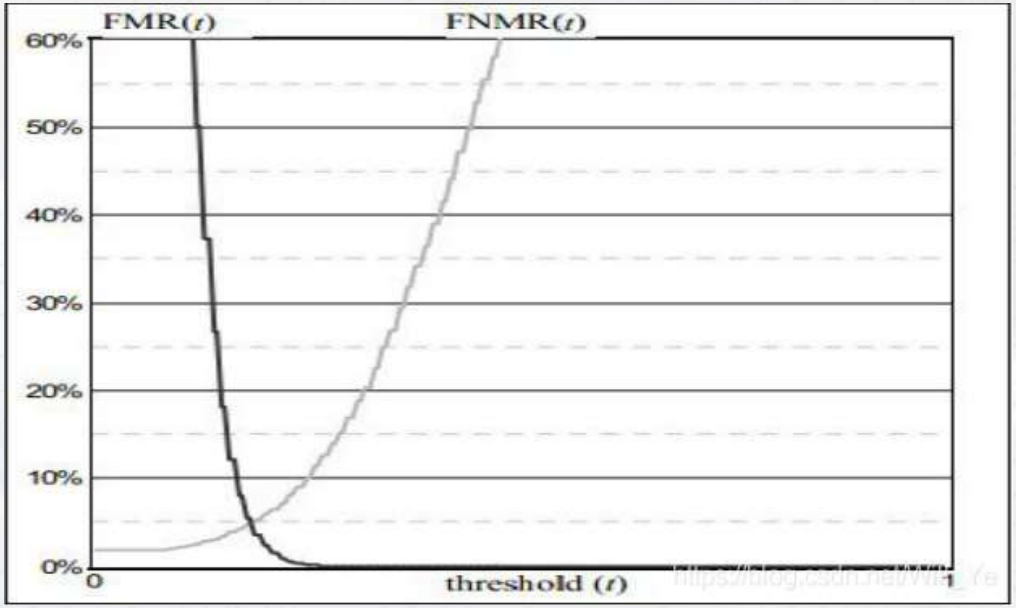
通过向左或向右移动阈值，可以改变概率，但请注意，错误匹配率的降低必然导致错误非匹配率的增加，反之亦然。
对于给定的生物识别方案，我们可以绘制虚假匹配率与虚假非匹配率的关系曲线，即工作特征曲线，以帮助我们确定阈值和权衡。

生物识别准确性的阈值调整

例：公司部署了面部识别系统来控制门禁

更重视防止未授权的进入（增加安全性）：设置一个较高的阈值，系统只有在非常确信是授权用户的情况下才会解锁门
→偶尔一些实际的员工可能会被错误地拒之门外（FNMR增大），但可以最大限度减少非授权人员的误入（FMR减小）。

希望减少员工因系统不识别而等待的情况：降低这个阈值，从而减少虚假非匹配的情况
→可能增加安全风险（即未授权人员有更大的机会被系统错误接受，FMR增大）。



通过向左或向右移动阈值，可以改变概率，但请注意，错误匹配率的降低必然导致错误非匹配率的增加，反之亦然。
对于给定的生物识别方案，我们可以绘制虚假匹配率与虚假非匹配率的关系曲线，即工作特征曲线，以帮助我们确定阈值和权衡。


Any Questions?

远程用户认证

最简单的用户身份验证形式是本地身份验证，即用户尝试访问本地存在的系统，如独立的办公室个人电脑或 ATM 机。更复杂的情况是远程用户身份验证，即通过互联网、网络或通信链路进行身份验证。

- 额外的安全威胁，例如:
  - 窃听、捕获密码、重放攻击

通常依靠某种形式的挑战-响应（challenge-response）协议来应对威胁

挑战-响应协议：每次认证时认证服务器端都给客户端发送一个不同的“挑战”字串，客户端程序收到这个“挑战”字串后，做出相应的“应答”，以此机制而研制的系统。

远程用户认证——通过密码协议进行远程身份验证

Client	Transmission	Host
U，user	U→	
	←{r, h(), f()}	r, random number h(), f() fcns.
P′, pwd. r′, return of r	f(r′, h(P′))→	
	←yes/no	If f(r′, h(P′))= f(r, h(P(U))) then yes else no

h(): hash fcn.
f(): one-way fcn.

Example of a challenge-response protocol

1. 用户向远程主机发送身份信息
2. 主机生成一个随机数（nonce）将nonce 返回给用户
3. 主机存储密码的散列代码
4. 密码散列是函数 f 的参数之一
5. 使用随机数有助于防止对手获取用户的传输信息

使用随机数作为f的参数之一可以防止重放攻击。

远程用户认证——通过令牌（token）协议进行远程身份验证

Client	Transmission	Host
U，user	U→	
	←{r, h(), f()}	r, random number h(), f() fcns.
P′ → W′ pwd. to passcode via token r′, return of r	f(r′, h(W′))→	
	←yes/no	If f(r′, h(W′))= f(r, h(W(U))) then yes else no

h(): hash fcn.
f(): one-way fcn.

Example of a token protocol

1. 用户向远程主机发送身份信息
2. 主机返回随机数和标识符
3. 令牌存储静态密码或生成一次性随机密码
4. 用户输入密码激活
5. 用户和远程主机共享密码

远程用户认证——通过生物识别协议进行远程身份验证（静态）

Client	Transmission	Host
U，user	U→	
	←{r, E()}	r, random number E(), fcns.
B′ → BT′ bio. D′, bio. device r′, return of r	E(r′‖D′‖BT′)→	$E^{-1}E(r'\|\|D'\|\|BT')=(r'\|\|D'\|\|BT')$
	←yes/no	If r′=r and D′=D and BT′ =BT(U) then yes else no

Example of a static biometric protocol

E(): 预设密钥的加密函数

1. 用户向主机发送 ID
2. 主机回应一个随机数和用于加密的标识符
3. 客户端系统控制用户端的生物识别设备
4. 主机对收到的信息进行解密，并与本地存储的值进行比较
5. 主机将收到的设备 ID 与主机数据库中的注册设备列表进行比较，从而提供身份验证

超过预定的阈值

远程用户认证——通过生物识别协议进行远程身份验证（动态）

Client	Transmission	Host
U，user	U→	
	←{r, x, E()}	r, random number x, random sequence challenge E(), fcns.
B′, x′ → BS′(x′) r′, return of r	E(r′, BS′(x′))→	$E^{-1}E(r', BS'(x'))=$ (r′, BS′(x′)) extract B′ from BS′(x′)
	←yes/no	If r′=r and x′=x and B′=B(U) then yes else no

1. 主机提供随机序列和随机数作为挑战
2. 序列挑战是一串数字、字符或单词
3. 然后，客户端的用户必须发声、输入或写入该序列，以动态生成生物识别信号
4. 客户端对生物识别信号和随机数进行加密
5. 主机解密信息并生成比较结果

身份识别中的攻防总结

攻击类型	认证方式	示例	典型防御措施
客户端攻击	密码	猜测、穷举搜索	大熵值；限制尝试次数
	令牌	穷举搜索	大熵值；限制尝试次数；要求物理接触或者物理存在来获取或盗窃身份认证的令牌或设备
	生物特征	误匹配	大熵值；限制尝试次数
主机攻击	密码	明文盗窃、字典/穷举搜索	哈希保护；大熵值；保护密码数据库
	令牌	密码盗窃	与密码相同；一次性密码
	生物特征	模板盗窃	捕获设备认证；挑战-应答协议

身份识别中的攻防总结

攻击类型	认证方式	示例	典型防御措施
窃听、盗窃和复制	密码	“肩窥”（指使用直接的观察技术，如通过某人的肩膀来查看，来获取信息。	用户勤勉保密；管理员迅速撤销泄露的密码；多因素认证
	令牌	盗窃、仿制硬件	多因素认证；抗篡改/明显的令牌
	生物特征	复制（欺骗）	捕获设备上的复制检测和认证
重放攻击	密码	重放被盗密码的响应	挑战-应答协议
	令牌	重放被盗令牌的响应	挑战-应答协议；一次性密码
	生物特征	重放被盗生物特征模板的响应	在捕获设备上进行复制检测和挑战-应答协议的认证
木马	密码、令牌、生物特征	安装恶意客户端或捕获设备	客户端或捕获设备在受信任的安全边界内的认证
拒绝服务	密码、令牌、生物特征	多次失败的认证导致锁定	与令牌一起使用的多因素认证


Any Questions?