Cryptanalytic attack 分析明文、密文或 key
Brute-Force 暴力一波
DES key too short(56)→暴力
3DES key 變長但 data 一樣 64、加密太慢→沒有效率
1.ECB 很多 block 用同一個 key
2.CBC ECB+前後 block chain(XOR)起來
3.Stream cipher:一個 byte 流進來和 random key 加密 xor
→Faster/ far less code /適合用在網路
Message authenticate 內容不要改，來源 authentic
Q：Symmetric key NOT suitable for data authentication?
No need to be reversible / waste of processor resource
廣播給所有人，都要解密太麻煩了→**產生 tag**
1. MAC(Message authenticate code)加密
2. One-way Hash 不用 key
3. Hash+encrypt 先 hash 在 symmetric/public encrypt
4. Hash **w/o encrypt** 對稱的 Key 夾 msg 一起 hash
N 是 hash 完的長度
A. one way(preimage)H(x)=h 2^n
B. Second preimage(weak collision)H(y)=H(x)given x 2^n
C. Collision(strong)H(x)=H(y) 2^(n/2)
SHA(secure hash alg) authenticate/digital signature/pswd
Public key：easy compute/public key 不能猜到 msg private key
Digital signature: hash and encrypt(可以 CA 自己做)
public key 加密(its certificate=key+ID)
Symmetric key exchange(Hellman)需要 authentication
Digital envelopes:
Msg 用 random symmetric key(用 public key 加密)加密

E-Authenticate
Registration authority 註冊 claimant Relying party 登入(已註冊)
Credential service provider 憑證 Verifier 驗證
Credential 護照 Token 身分證 ID

Offline dictionary attack 離線字典攻擊
1. 離線拿到系統 hash 密碼檔
2. 破解 hash value
3. 得到常見 passwords
Specific account attack 特定帳號開猜密碼
Popular pswd attack 試常見的密碼
Pswd quessing against 1 user 了解他開猜
Workstation hijack 工作站不自動登出，滿危險的
Exploiting user mistakes 很智障給密碼
Password sniffing 攔截一波
用 salt+pswd 一起 hash 很難
Q：For a salt of length *b*bits, the number of possible passwords is increased by a factor of 2^*b*
UNIX 用 DES x25/ MD5 有 salt+inner loop
Reactive pswd checking 週期性跑 cracker 發現密碼不好就跟你說
Proactive 註冊的時候確認密馬好不好(Bloom filter)

Memory card(ex 磁條)	Smart token
Need special reader Token loss User 太多不夠用 (缺點)	有 Authentication protocol Static Dynamic Challeng-response

Biometric Auth
Verify 他知道你是 waterso 就去找你在的地方，看指紋 OK 不 OK/
Identify(不知道你是誰，掃 database)
Remote user Auth 想傳會被攔：challenge-response
Security issue
Replay 登入上一個想燈的人(隨機產生驗證碼)
Trojan horse 讓你以為是官方，你送他密碼他超爽
Dos 一直登入

Ping flood: ICMP/network bandwidth 缺點:來源會被 clearly identified/自己會有 reflection Randomly spoof source: backcatter traffic
SYN spoofing: TCP victim 一直收到 ACK 導致 victim 不能連線 SYN flood 目標是數量取勝(network bandwidth) **However, the flooding attacks are limited by a single system!!** **防 SYN spoof:** cookie/selective drop/table size(大) and timeout period(小)
DDoS botnet zombie
Application-based SIP(session):INVITE HTTP: Slowloris(blank line)/ HTTP flood 下載很大的
Reflector and Amplifier Attacks Q: Why Normal server systems?大多常看到的 server 1. 容易上手 2. 難找到攻擊者 **Reflection**: UDP/TCP **Amplification**(broadcast):控制一堆 zombie 弄一堆 request 一堆 requests 被廣播網路中，裡面的 server 們收到就 response (**TCP 不行廣播，UDP 可以**) **防 flood**: block spoof/ ensure path back/limit 一些 rate

Confidentiality:
Data confidentiality: 我的資訊不能被知道
Privacy:你的資料被別人存
Integrity:
Data integrity: 可以改的人才可以改
System integrity: 系統能正常運作
Availibilty:可以用的人可以用
Authenticity:可信賴(訊息來源要正當)
Accountablity:責任性(你的動作可被追蹤)
Model for security: Hardware/Software/Data/Communication facilities
Vulnerability: weakness of system resources
Corrupted: loss of integrity
Leaky: loss of confidentiality
Unavailable or very slow: loss of availability
Attack: a threat that is carried out (threat action)
Passive: 沒改系統(純拿資訊 ex.竊聽)
Active: 有改(replay, masquerade, DoS)
Inside: by an authorized user
Outside: by an unauthorized user

Unauthorized Disclosure【Confidentiality】 Exposure 資訊直接暴露 Interception 攔截資訊 Inference 推斷:被猜到資訊 Intrusion 侵入系統拿資訊	Deception【Integrity】 Masquerade 假裝自己是官方 Falsification 用錯誤資訊欺騙官方 Repudiation 不承認自己欺騙
Disruption【availability or system integrity】 Incapacitation 癱瘓系統 Corruption 亂改系統 Obstruction 阻塞傳送	Usurpation 竄改【system integrity】 Misappropriation 侵吞別人系統資料 DDoS Misuse 讓別人執行後不安全

Fundamental Security Design Principles
都用同一套系統性方法很好，但是別人也知道
Economy 設計簡單/Fail-safe 想你要甚麼不是你要甚麼/mediation 檢查/Open design 公開才會進步/separate of previlege 規則不要太複雜/least common
Psychological acceptability 不能安全到影響 user/isolation 隔離/encapsulation 用 oop 壓縮/modualarity/layering/least astonish
Attack surface 對誰攻擊 Network/software/human
Shallow layering+Large attack surface→high risk
Attack tree 規劃攻擊路徑

Authtication 驗證有效 / Authorize 授權/ Audit 審核
Access control

DAC(discretionary) 想幹嘛就幹嘛 →Subject 可以改 protection state →查 access matrix 控制 →一個 user 一個 row(protection domain)
MAC(Mandatory) OS 強制檢查 access right
RBAC(Role-based) 不同腳色不同權力 RBAC0: minimum functionality RBAC1: RBAC0 + role hierarchies RBAC2: RBAC0 + **constrains** **Mutually exclusive** Role 之間都不要有重疊的 access **Cardinality** 限制 role 裡面面 users 的最大大數量量 **Prerequisite** 有前提的 role RBAC3: RBAC0 + RBAC1 + RBAC2
ABAC(Attribute-base) 用屬性當條件 Distinguishable 屬性要被定義好 Strength: flexibility and expressive power Drawback: the performance impact 無限制數量的屬性 fine-grained(大燕麥片是無限的)
傳統 UNIX file access control 用 inode No scalability: unwieldly and difficult to manage 現代 UNIX:用 ACL

Access matrix 常常太稀疏(sparse)
ACL(Access control list)對 File 來說清楚，對 user 爛
Capability ticket(ACL 反過來)
Q: Have greater security problem than ACLs??
Tickets may be dispersed around the system
→OS holds all tickets on behalf of users
→An unforgeable token in the capability
Authentication table

MIME mail format				
S/MIME 有加密就是安全(sign only msg content)				
Plain text	**Bob's Private k**	**1time session key**	**Alice's Public key**	Radix64 encode
	Digital signature(SHA)	3DES	encrypt	

Enveloped data→Encrypted content(RSA) and associated keys/**Signed data**→Encoded[message + signed digest]/**Clear-signed data**→Cleartext message + encoded signed digest/**Signed and**

Q: 傳的時候會被擋，怎麼辦? 1. Encrypt 2. signal hiding 3. detection 4. Authentication protocol

Mobile Device Security Strategy

Device security

Supply mobile devices for employee use and pre-configure those devices or bring-your-own-device (BYOD) policy

Configuration guidelines for OS and apps (e.g., rooted is not allowed)

Traffic security: based on encryption and authentication/via a VPN

Barrier security: Firewall policies specific to mobile device traffic

Wi Fi Protected Access (WPA)

Distribution • Exchange MPDUs between two BSS

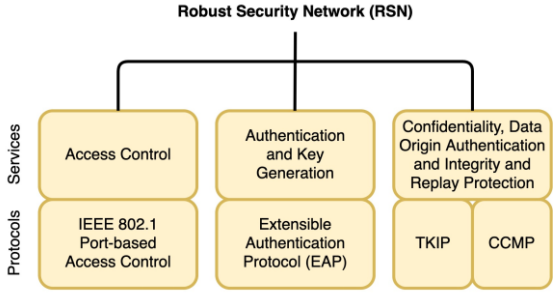
Integration • Data transfer between a Wi-Fi station and an LAN station on an integrated IEEE 802x LAN

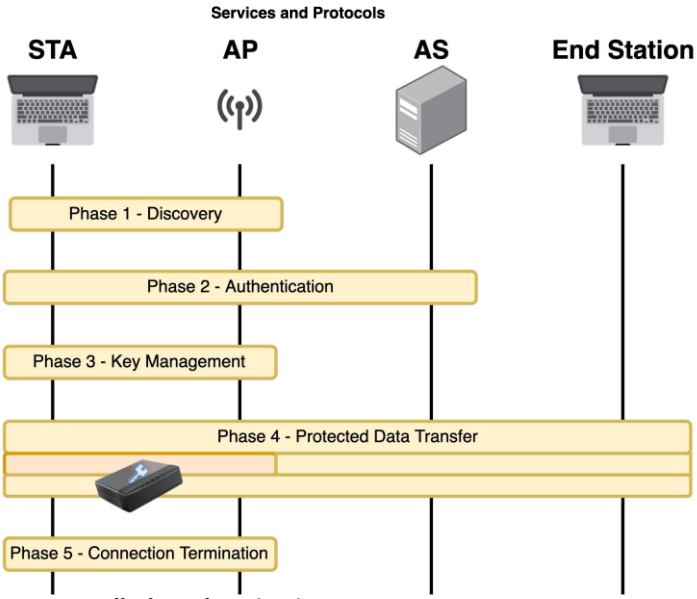
Access point (AP)	Any entity that has station functionality and provides access to the distribution system via the wireless medium for associated station.
Basic service set (BSS)	A set of stations controlled by a single coordination function.
Coordination function	The logical function that determines when a station operating within a BSS is permitted to transmit and may be able to receive PDUs.
Distribution system (DS)	A system used to interconnect a set of BSSs and integrated LANs to create an ESS.
Extended service set (ESS)	A set of one or more interconnected BSSs and integrated LANs that appear as a single BSS to the LLC layer at any station associated with one of these BSSs.
MAC protocol data unit (MPDU)	The unit of data exchanged between two peer MAC entities using the services of the physical layer.
MAC service data unit (MSDU)	Information that is delivered as a unit between MAC users.
Station	Any device that contains an IEEE 802.11 conformant MAC and physical layer.

Transition types, based on mobility:

No transition/BSS transition/ESS transition

Distribution service: association/Reassociation/Disassociation





uncontrolled: authentication server

controlled: DS/other wireless station on this BSS

EAP(extending authentication)

EAP authentication is initiated by the server (authenticator)

Authentication is mutual between the client and authentication server

Protected data transfer

1. Temporal Key Integrity Protocol (TKIP):

→MIC alg, 256 TK, TKIP sequence counter

2. Counter Mode CBC MAC Protocol (CCMP)

→Msg: CBC/MAC data: AES/Same 128 bit AES key for both/A 48 bit packet number: a nonce to prevent replay attacks

802.11i PRF R=HMAC-SHA-1(K, A||B||i)

enveloped data →Nesting of signed and encrypted entities **Why Radix64?** ensure that the data remains intact without modification during transport.
DKIM(DomainKeys Identified Mail): sign at header 在一個網域裡面會有一個 public key(DNS 查詢得到) 大家都可以使用那個 public key 解開任何寄來來的 email 打開之後裡面會有簽章在 email 的 header。 **Why?** An email authentication technique that is transparent to the end user.

SSL(Secure socket layer)

TLS(Transport layer Security) TCP

TLS connection P2p/1connection→1session
TLS session Client-server/Created by the handshake protocol Define a set of cryptographic security Parameters/ avoid the expensive connection
Record protocol(confidential/msg integrity) Fragment→(compress)→Add MAC→EncryptTLS→record header
handshake protocol(複雜) authenticate each→Negotiate encryption and MAC algorithms →Negotiate cryptographic keys to be used 4 phase:hello→server give key/certificate→client give keychange/certificate→cipher_spec
Cipher spec protocol(a byte with value 1)
Alert protocol: (2bytes) [warning(1), fatal(2)]不會有新的連線 +[甚麼警告]
Heartbeat protocol (phase 1) **週期性確認接收者有沒有活著**

TLS attack: Heartbeat exploit(src:BAE system) Small payload disguised as a big one, so it gets other data.

HTTPS(HTTP over SSL/TLS)

IPSec(network layer): Apply to firewall, router/防繞道

Transparent to user, apps/routing apps:保護 router

Scope: **ESP(encapsulating security payload)** authen/encrypt+key change function

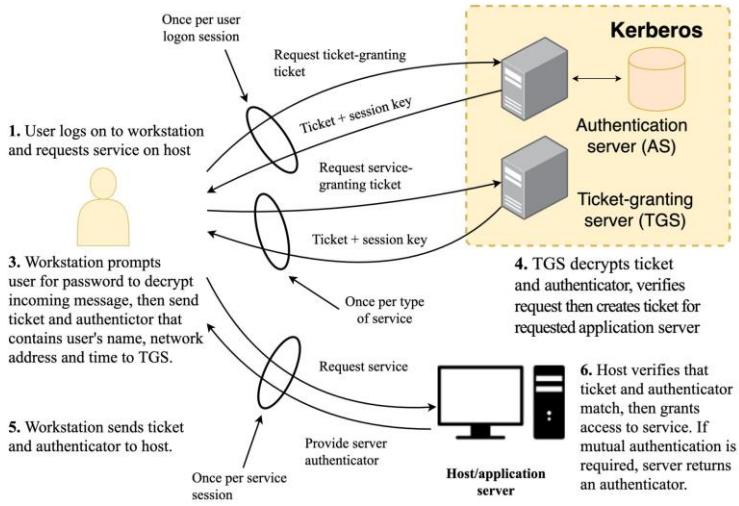
VPN:authen/encrypt **AH(authentication header)**:authen

概念:one-way relationship(sender/receiver)

Two-way secure exchange(2Secure Assotiation)

Transport mode	**Tunnel mode**
保護 IP payload/end2end	Entire IP packet/one or both end/behind firewall may engage in secure communication w/o IPsec

---



Kerberos:internet auth **Why TGS?**

Query the user password for each service→Inconvenient!

Store the password in memory for the duration of the logon session →Security risk!

stolen→timestamp

alteration→encrypt ticket using session key(AS TGS) **spoof**→encrypt ticket using pswd **Replay attack**→authenticator,not usable

Inter-Realm:share a secret key with kerbero server

不會影響 performance/需要一個 dedicated platform(secure)

Multiple realms?? No

X509 : format for public-key certificate: (lightweight)

CA us: 加密解密都用 CA 的 key

CRL(certificate revocation list):填表申請你要 revoke certificate

PKI(public key Instructure)鑰匙圈

X509 的 **trust store** 是一種 PKI: large lists of CAs and public key

CAs in trust store: user 簽或 CA 簽/ hierarchy small, all equally trust

ISSUE: User or CA 有問題，就要處理/不同情況用不同 trust store