PKI技術及其安全應用簡介

中華電信股份有限公司
電信研究所
資通安全研究室
王文正 博士

內容

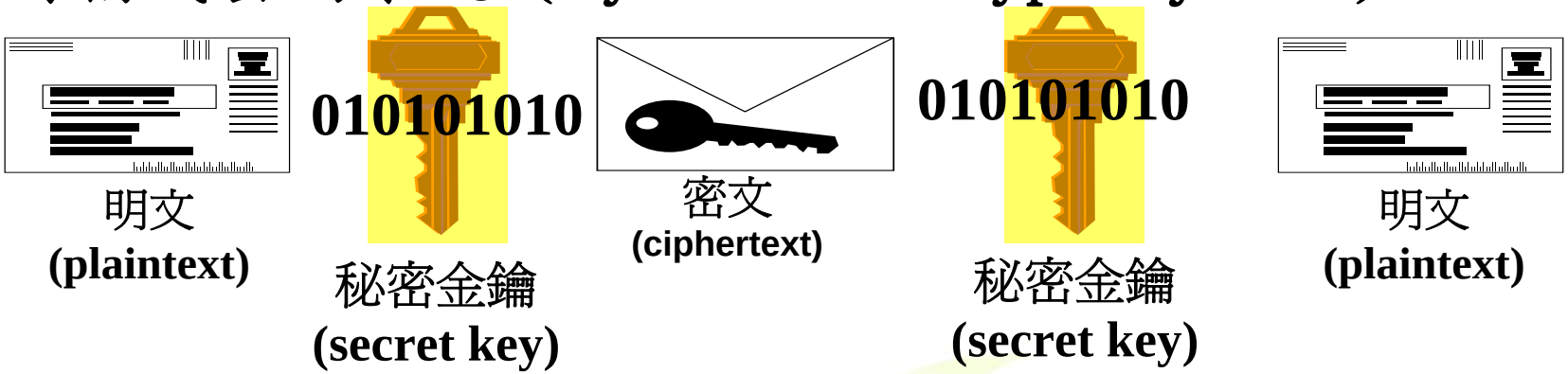
- 密碼學簡介（cryptography）
  - secret-key cryptosystem
  - public-key cryptosystem
- Man-in-the-middle攻擊
- 公開金鑰基礎建設（PKI）
- 我國PKI架構及應用
- 對抗Man-in-the-middle攻擊的對策
  - 如何正確驗證憑證
- Replay攻擊及其對策
- 總結

密碼學簡介
Cryptography

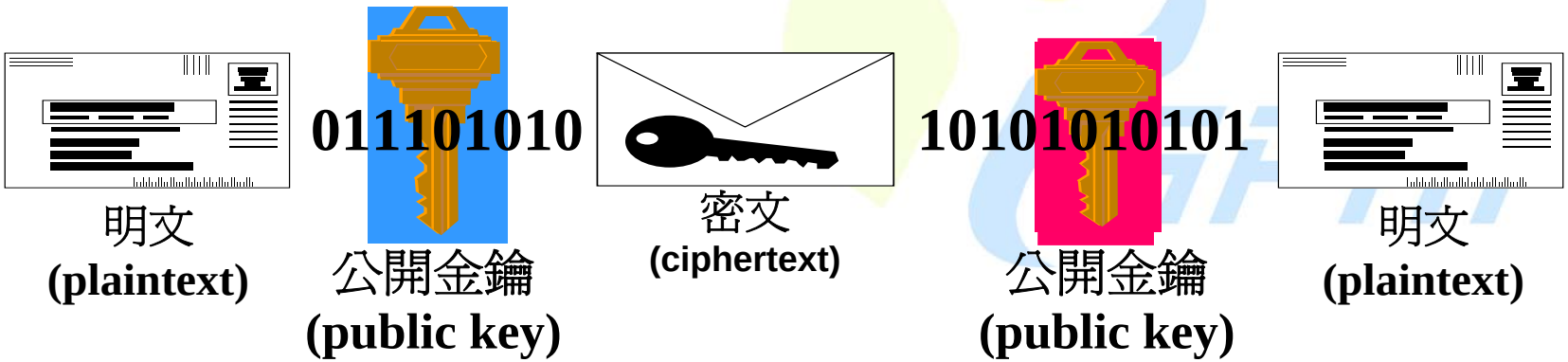
網路通訊的資訊安全問題

密碼系統（Cryptosystem）

□ 對稱式密碼系統（symmetric cryptosystem）



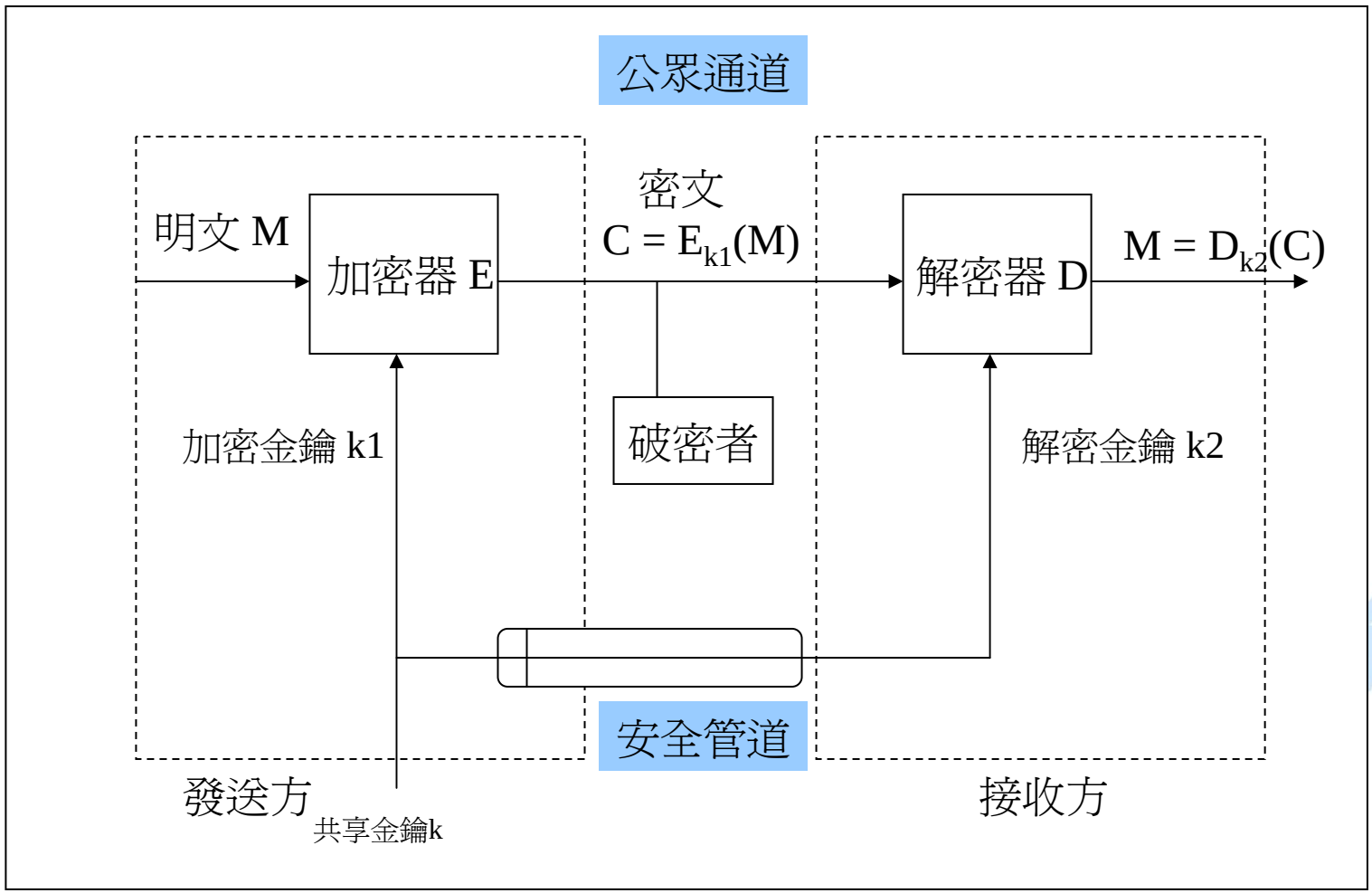
□ 非對稱式密碼系統（asymmetric cryptosystem）



註：密碼學上所謂金鑰（Key）指的是一組經由特殊方法產生的數字

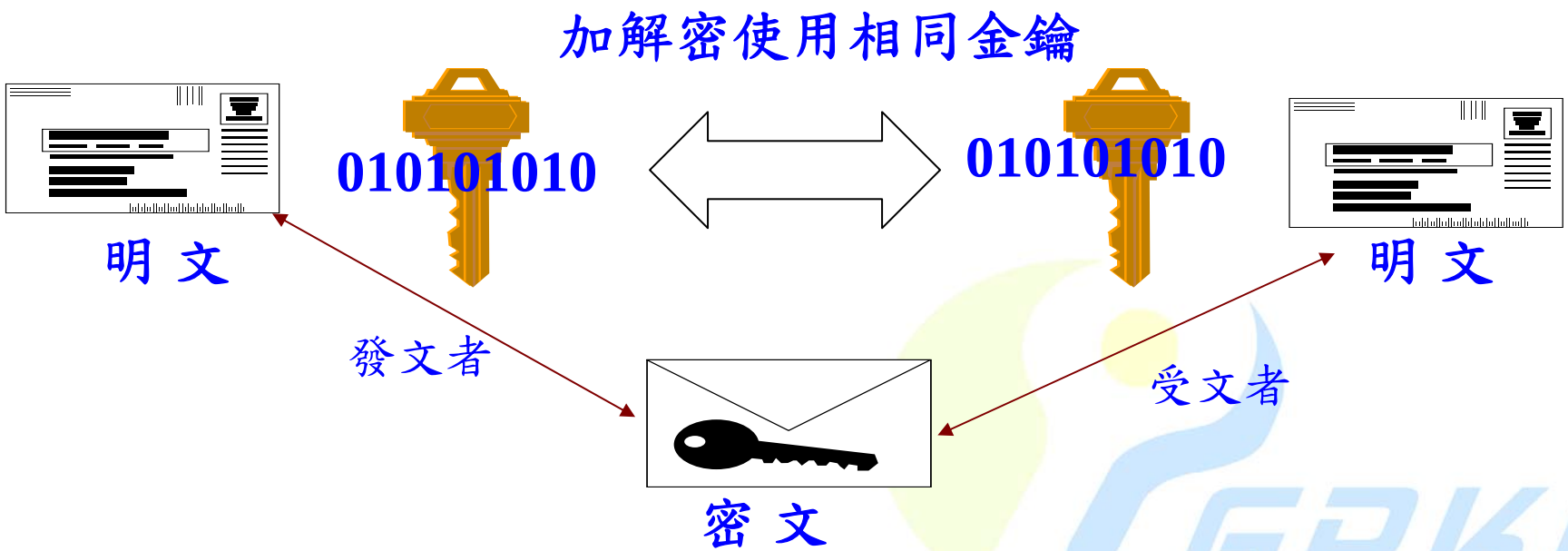
對稱式密碼系統

(秘密金鑰密碼系統，secret-key cryptosystem)



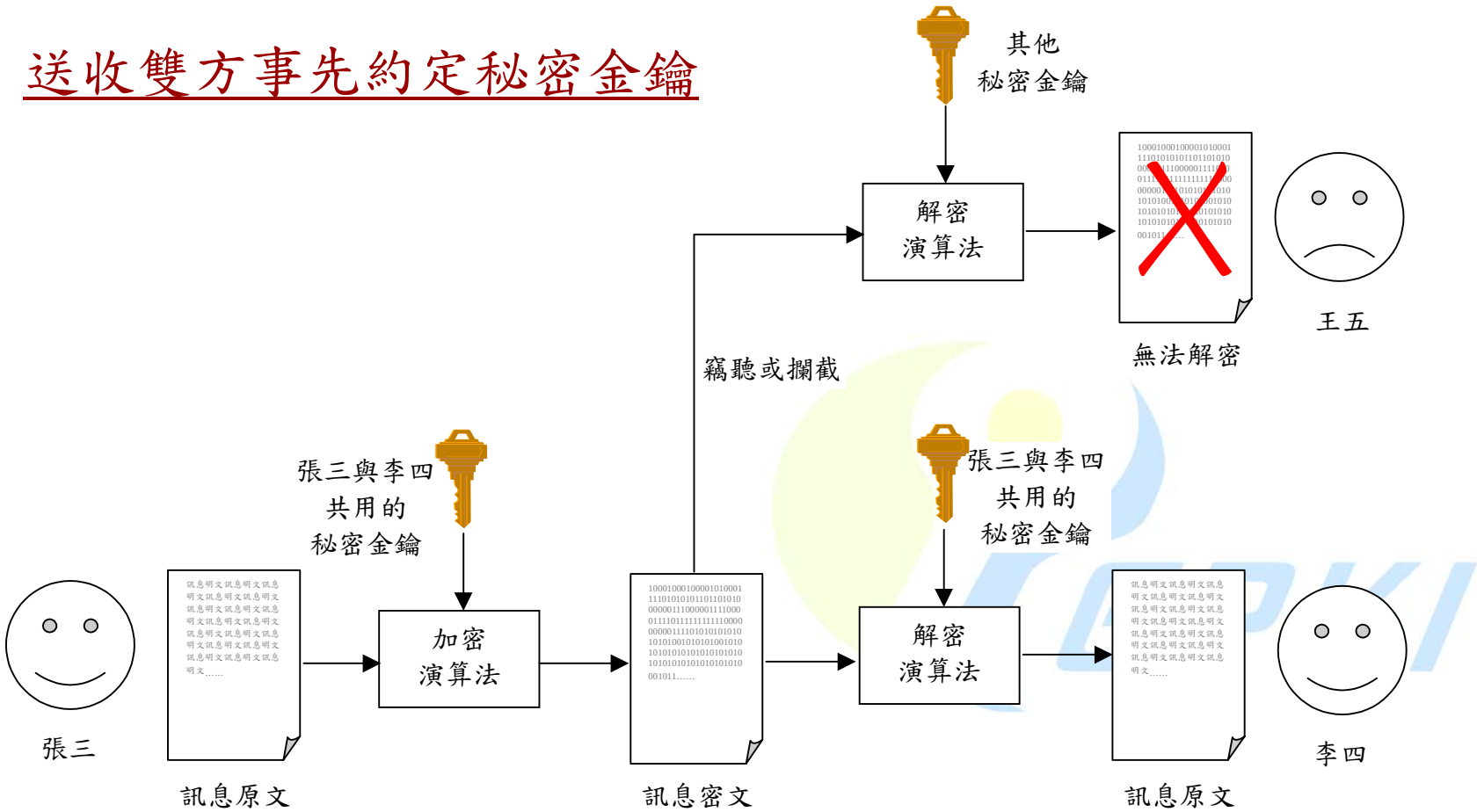
通常k1 = k2=k或是k1=f(k) , k2=g(k)

對稱式加密系統
(秘密金鑰密碼系統)



對稱式密碼系統之特性

送收雙方事先約定秘密金鑰



對稱式密碼系統之優缺點

- 優點：
  - 加解密運算速度快
  - 金鑰長度較短
  - 歷史悠久（經得起考驗）
  - 金鑰產製較為簡單
- 缺點：
  - 金鑰散佈問題(如何建立傳送金鑰的安全管道)
  - 金鑰數目太大(金鑰管理問題)
  - 無法達到不可否認性(因為共享祕鑰)

非對稱式密碼系統
(公開金鑰密碼系統，public-key cryptosystem)

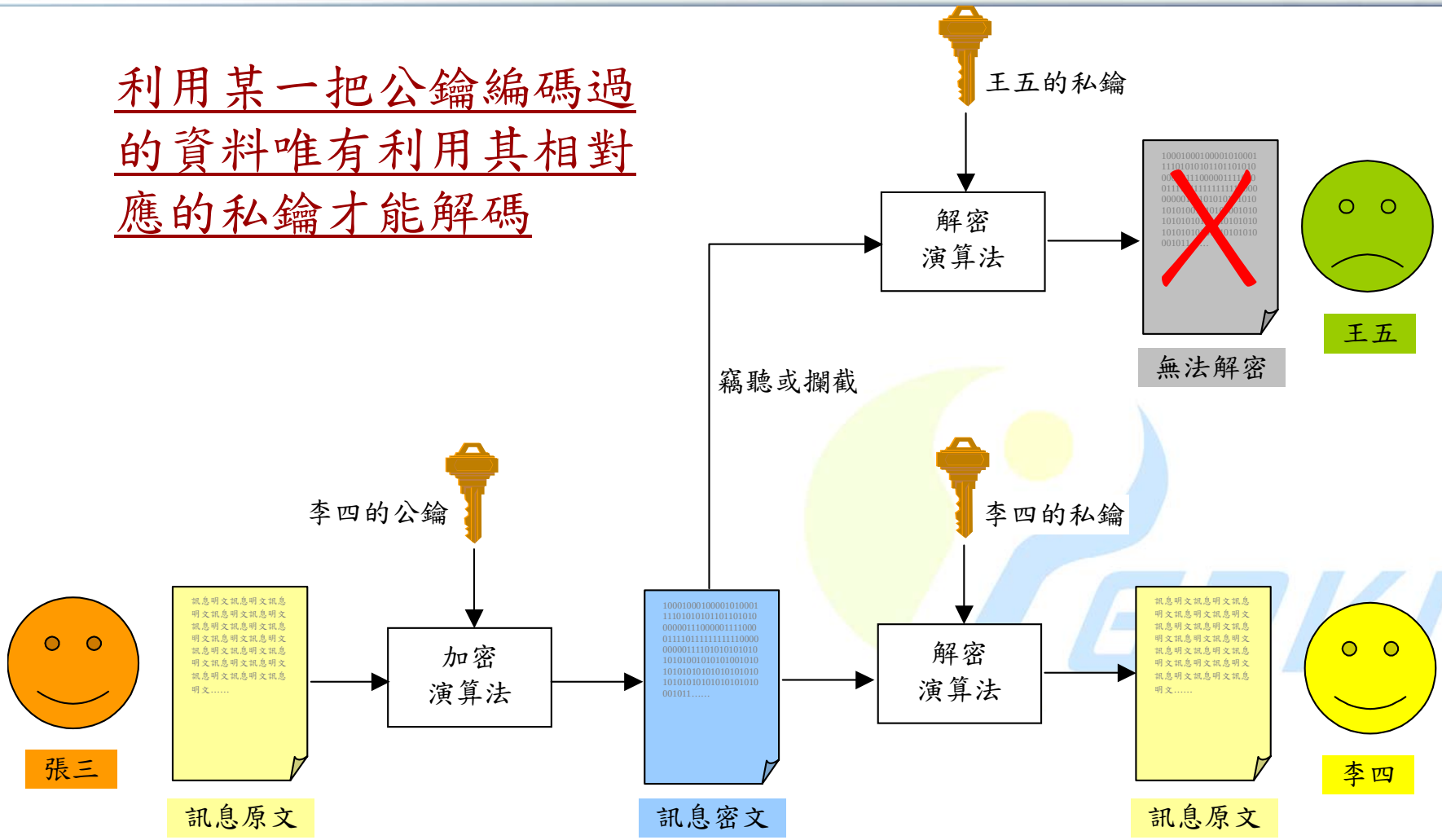
- 1976 Diffie 和 Hellman 發表一篇名為”New Directions in Cryptography”的論文
  - 提出公開金鑰密碼學的概念
  - 同時發表一個利用公開金鑰密碼學的概念所設計的加解密演算法
    - Diffie-Hellman Key Exchange Algorithm
- 公開金鑰密碼系統的優點
  - 保護私鑰機密
  - 簡化金鑰分配及管理的問題
  - 能達到不可否認性(數位簽章)
- 公開金鑰密碼系統的缺點
  - 加解密的運算較複雜且速度較慢
  - 金鑰長度較長
  - 金鑰產製較為困難

公開金鑰密碼系統之特性

- 公開金鑰密碼系統所使用的金鑰分為公鑰與私鑰兩種，公鑰對外公開，私鑰私密保存；
- 每一對公鑰與私鑰都是唯一成對的，任何兩對金鑰對不會共用同一把公鑰或私鑰；（唯一對應特性）
- 利用某一把公鑰編碼過的資料唯有利用與其成對的私鑰才能解碼；（機密特性）
- 利用某一把私鑰編碼過的資料唯有利用與其成對的公鑰才能解碼；（簽章特性）
- 公鑰與私鑰雖然具有數學上的對應關係，但其產生方法是不可逆的，即在實務上無法由公鑰推算得到其相對應的私鑰。（安全特性）

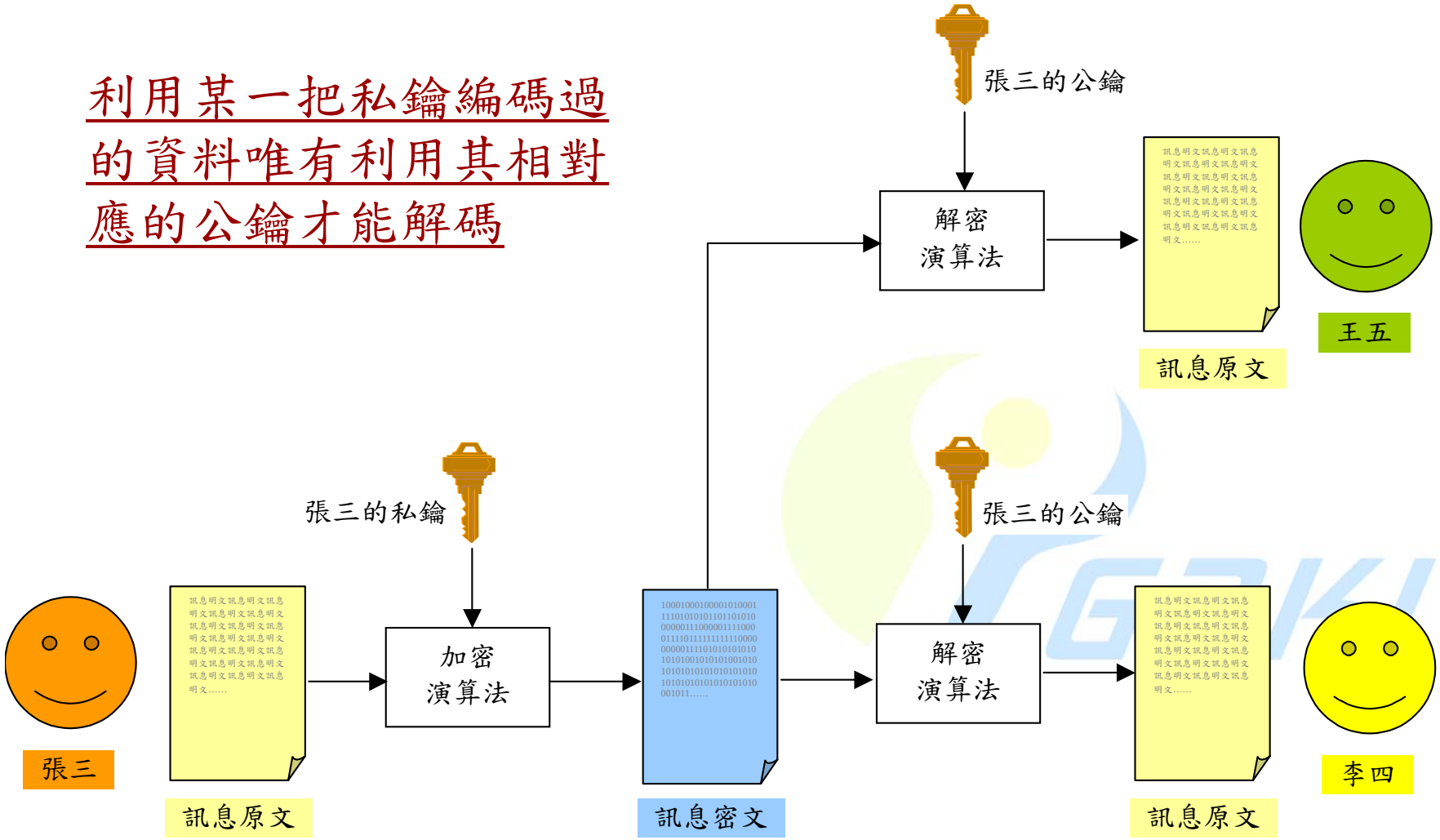
公開金鑰密碼系統之運作—機密特性

利用某一把公鑰編碼過的資料唯有利用其相對應的私鑰才能解碼



公開金鑰密碼系統之運作—簽章特性

利用某一把私鑰編碼過的資料唯有利用其相對應的公鑰才能解碼



著名的對稱與非對稱金鑰密碼演算法

項目/名稱	DES	IDEA	Skipjack	AES	RSA	ElGamal
提出年代	1976	1992	1993	2000	1978	1985
設計者	美國國安局	Lai Massy	美國國家標準技術局(NIST)	Rijmen & Daemen	Rivest & Shamir & Adleman	ElGamal
型式	對稱式	對稱式	對稱式	對稱式	非對稱式	非對稱式
金鑰長度(位元)	56	128	80	128 196 256	金鑰長度可變，現一般採用 1024	金鑰長度可變，為 512 以上
明文區塊長度(位元)	64	64	64	128	需小於金鑰長度	需小於金鑰長度
標準	美國標準	歐州專利	美國標準	美國標準	RSA 公司專利	無
備註	·全世界使用最廣	·PGP 採用	·美國金鑰託管標準 KES 使用	·NIST 於 1997 年公開向全世界甄選 ·新世代的對稱式密碼演算法	·RSA Labs PKCS#1 標準 ·ANSI/ISO/IETF 皆採用 ·使用最廣	·密文擴充一倍 ·機率式加密安全性高

RSA演算法(1/2)

- RSA演算法是由MIT的Ron Rivest、Adi Shamir及Len Adleman在1978年提出的公開金鑰密碼演算法
- 金鑰產製（Key Generation）
  - Select p, q　　p and q both prime
  - Calculate n = p X q
  - Calculate $\phi(n) = (p-1)(q-1)$
  - Select integer e　　$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
  - Determine d　　$d \equiv e^{-1} \bmod \phi(n)$ (i.e., $ed \equiv 1 \bmod \phi(n)$)
  - Public key　　KU = {e, n}
  - Private key　　KR = {d, n}

RSA演算法(2/2)

- 加密運算（Encryption）
  - Plaintext　　$M < n$
  - Ciphertext　　$C = M^e \pmod n$
- 解密運算（Decryption）
  - Ciphertext　　C
  - Plaintext　　$M = C^d \pmod n$
- 安全性（Security）

  如果能夠把n進行質因數分解成 p, q，就能推算出d（private key）

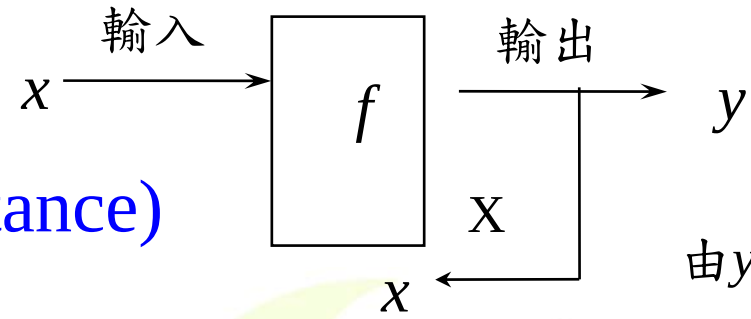
  但是如果n夠大（最好1024 bits以上，約276十進位數），則對n進行質因數分解是一個hard problem

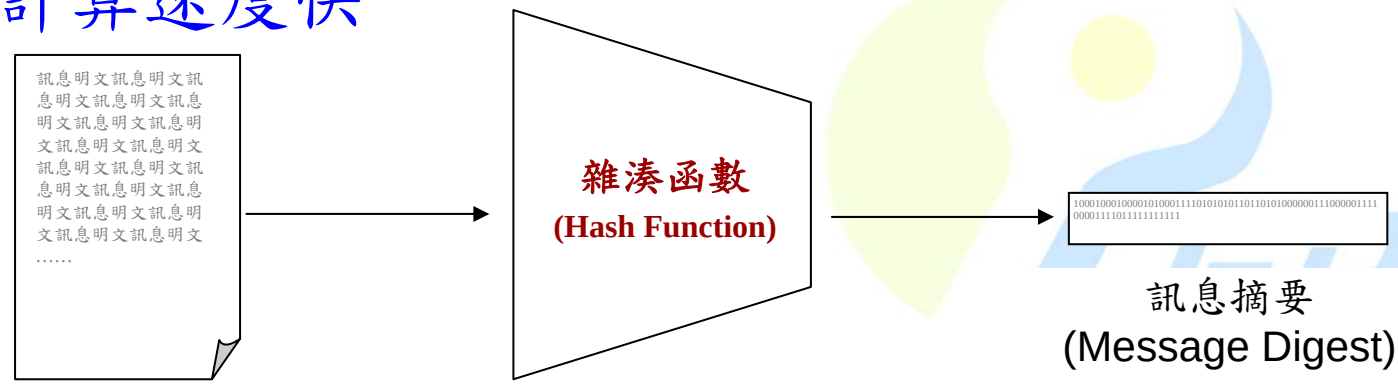
RSA演算法的簡單範例(1/2)

- 金鑰產製（Key Generation）
  - Select two prime numbers, p = 7 and q = 17
  - Calculate n = p X q = 7 X 17 = 119
  - Calculate $\phi(n) = (p-1)(q-1) = 6 \text{ X } 16 = 96$
  - Select integer e such that $\gcd(\phi(n), e) = 1$ and $1 < e < \phi(n)$; in this case, e = 5
  - Determine d such that $d \equiv e^{-1} \bmod \phi(n)$ (i.e., $ed \equiv 1 \bmod \phi(n)$). The correct value is d = 77, because 77 X 5 = 385 = 4 X 96 + 1
  - Public key　　KU = {e, n} = {5, 119}
  - Private key　　KR = {d, n} = {77, 119}

RSA演算法的簡單範例(2/2)

- 加密運算（Encryption）
  - For a plaintext input of M = 19
  - Ciphertext C = M^e (mod n) = 19^5 (mod 119) = 66
    - 19^5 / 119 = 2476099 / 119 = 20807 餘 66
- 解密運算（Decryption）
  - Plaintext C = 66
  - Ciphertext M = C^d (mod n) = 66^{77} (mod 119) = 19
    - 66^{77} / 119 = ($1.27…X10^{140}$) / 119 = $1.06…X10^{138}$ 餘 19





雜湊函數（Hash Function）

- □ 輸入任意大小的訊息，輸出固定大小的訊息摘要 (Message Digest)
- □ 單向(one-way)
- □ 抗碰撞(collision resistance)
- □ 計算速度快

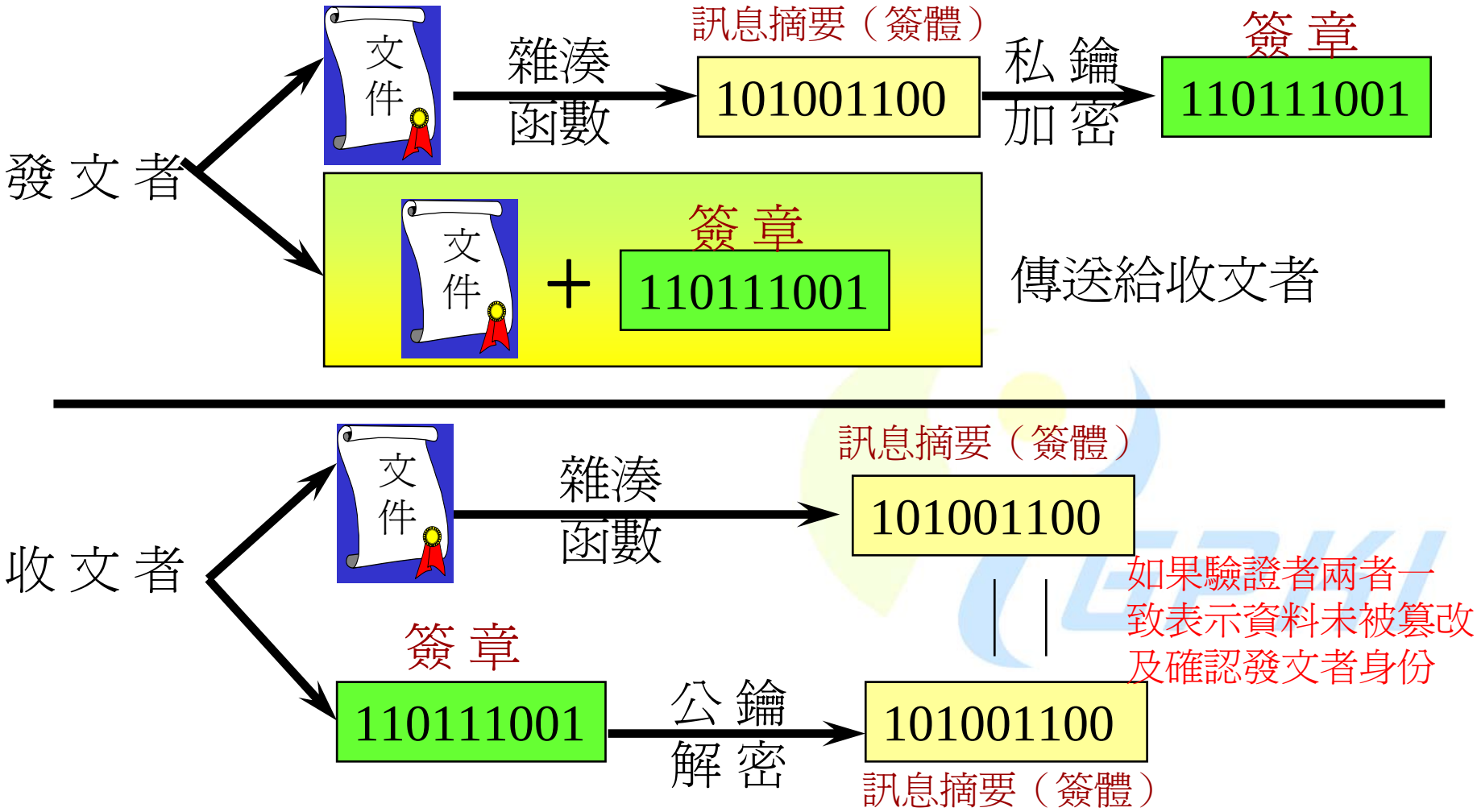
由y無法導出x
->不可逆

著名的雜湊函數演算法

函數	摘要長度	發展者
Message Digest 2 (MD2)	128bits	Ron Rivest
Message Digest 4 (MD4)	128bits	Ron Rivest
Message Digest 5 (MD5)	128bits	Ron Rivest,1992
Secure Hash Algorithm (SHA)及SHA-1	160bits	NIST/NSA,1992

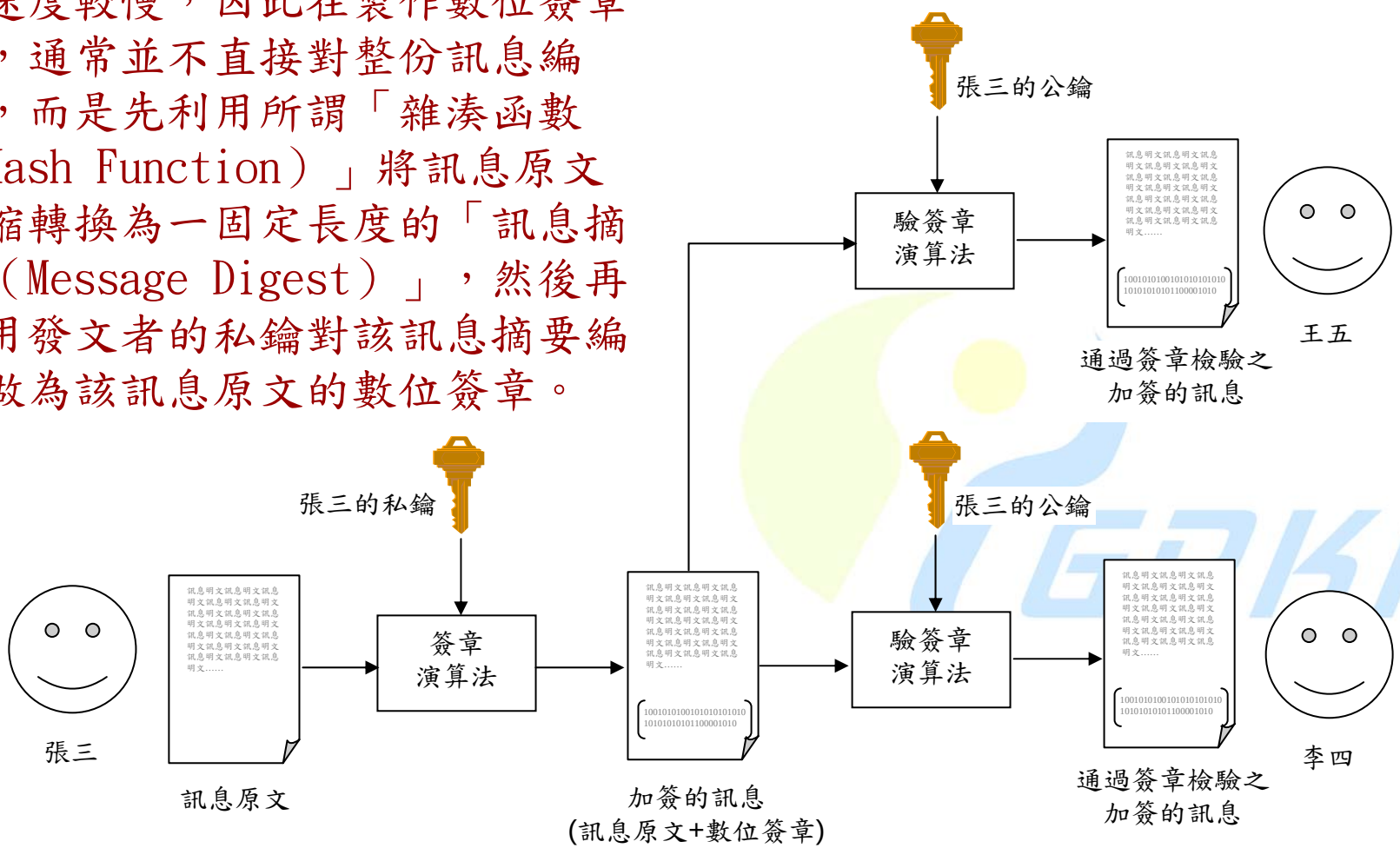
- 另有SHA-256、SHA-384及SHA-512雜湊函數，已由美國NIST正式核定（在2002/08/01公布於NIST FIPS PUB 180-2中），新世代的對稱式密碼演算法
- 2004年8月美國Crypto年會的Rump Session，山東大學王小雲教授宣佈破解MD4、MD5、RIPEMD及HAVAL，並於2005年2月發表論文宣稱可以將找到SHA-1碰撞訊息的複雜度由2^{80}雜湊運算降到在2^{69}雜湊運算。
- NIST宣佈將於2010年淘汰SHA1，全面改用SHA-256、SHA-384及SHA-512。

數位簽章（雜湊函數+非對稱密碼演算法）



公開金鑰密碼系統之運作—數位簽章

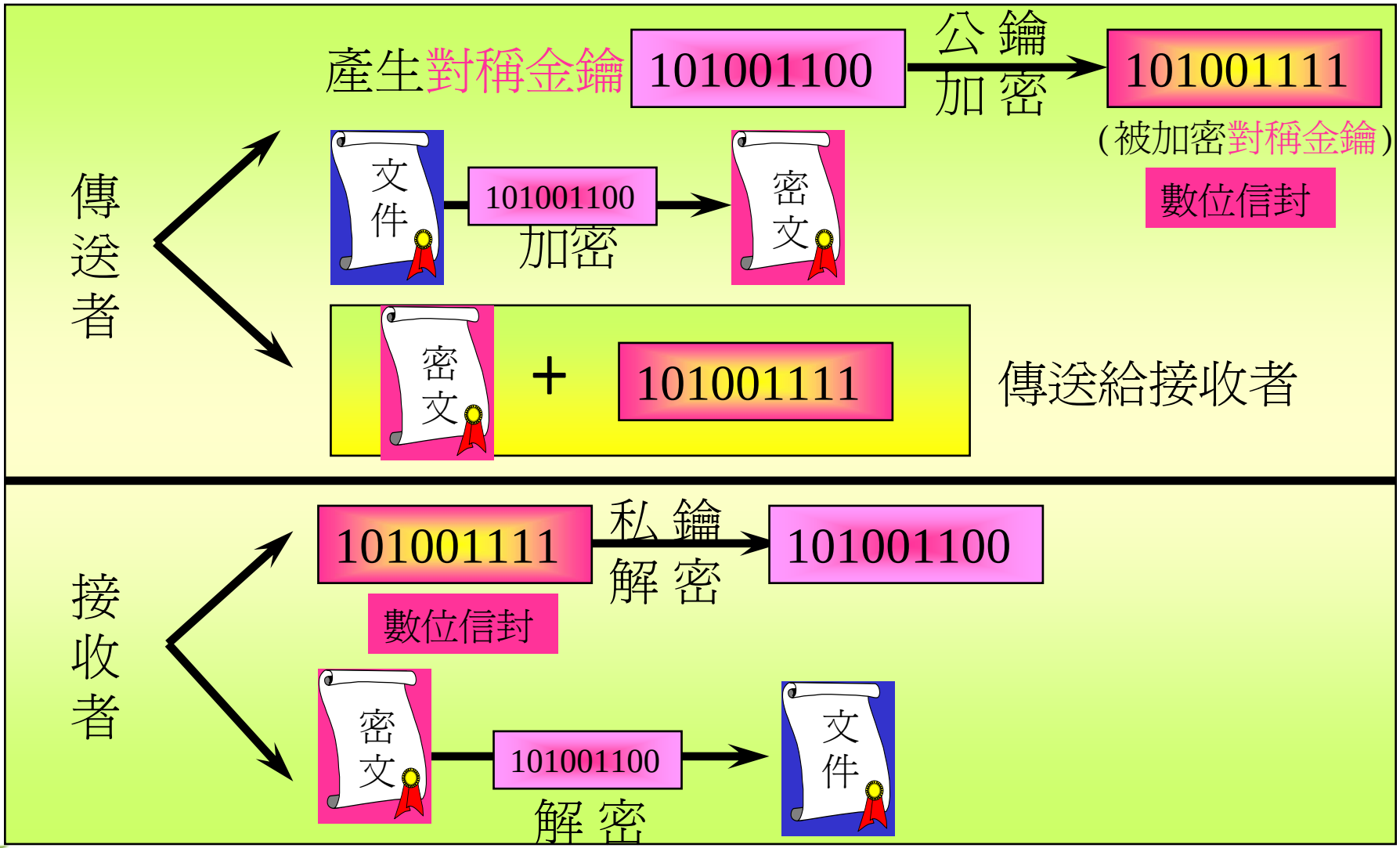
由於公開金鑰密碼系統的編碼與解碼速度較慢，因此在製作數位簽章時，通常並不直接對整份訊息編碼，而是先利用所謂「雜湊函數（Hash Function）」將訊息原文濃縮轉換為一固定長度的「訊息摘要（Message Digest）」，然後再利用發文者的私鑰對該訊息摘要編碼做為該訊息原文的數位簽章。



數位信封

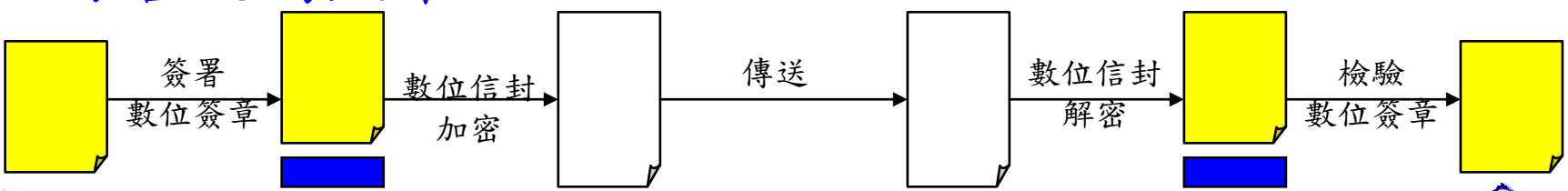
- 公開金鑰密碼系統的缺點是其編碼與解碼速度較慢，所以在實際運用上，通常均以公開金鑰密碼系統搭配秘密金鑰密碼系統來對訊息加解密，以求兼顧方便與效率。
- 一般是先由發文者在每次通訊前先隨機產生一把秘鑰，此把秘鑰特稱為「通訊基碼（Session Key，只用於此次的通訊過程，下次再通訊時，就必須另外再產生一把）」，再利用此通訊基碼對訊息原文加密成為密文（稱為數位信封），然後利用收文者的公鑰將通訊基碼加密，在傳送時訊息密文必須連同加密後之通訊基碼一起送出；收文者收到後，必須先用自己的私鑰將加密後的通訊基碼解密得到原來的通訊基碼，再利用此通訊基碼將訊息密文解密即可得到訊息原文。
- 通訊基碼的大小通常遠比訊息原文小的多，所以運用公開金鑰密碼系統對通訊基碼加解密並不需花費太長的時間，而通訊基碼本身是一種秘鑰，所以運用秘密金鑰密碼系統對訊息本身加解密，在速度上可以大幅提昇。

數位信封


產生對稱金鑰
101001100
公鑰加密
101001111
(被加密對稱金鑰)
數位信封
傳送者
文件
101001100
加密
密文
密文
+
101001111
傳送給接收者
接收者
101001111
私鑰解密
101001100
數位信封
密文
101001100
解密
文件

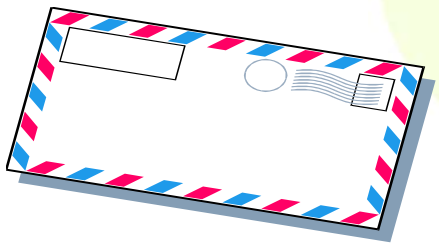
數位信封+數位簽章

- 在實務上，數位簽章的功能與數位信封的功能往往是搭配使用的
- 發文者可以先用自己的私鑰對訊息簽署數位簽章，然後利用收文者的公鑰將信息連同數位簽章一起加密成密文再傳送給收文者
- 收文者收到密文後，先用自己的私鑰將密文解密得到訊息原文及其數位簽章，然後再利用發文者的公鑰來檢驗數位簽章
- 如此可兼顧訊息的機密性、真確性、身分認證與不可否認的效果。



解決網路通訊資安問題的對策

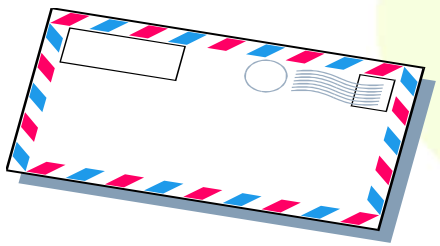
- 公開金鑰密碼技術（Public-Key Cryptography）
  - 數位簽章：防止篡改、冒名傳送、否認傳送
  - 數位信封：防止竊聽



Man-in-the-middle攻擊

解決網路通訊資安問題的對策

- 公開金鑰密碼技術（Public-Key Cryptography）
  - 數位簽章：防止竄改、冒名傳送、否認傳送
  - 數位信封：防止竊聽



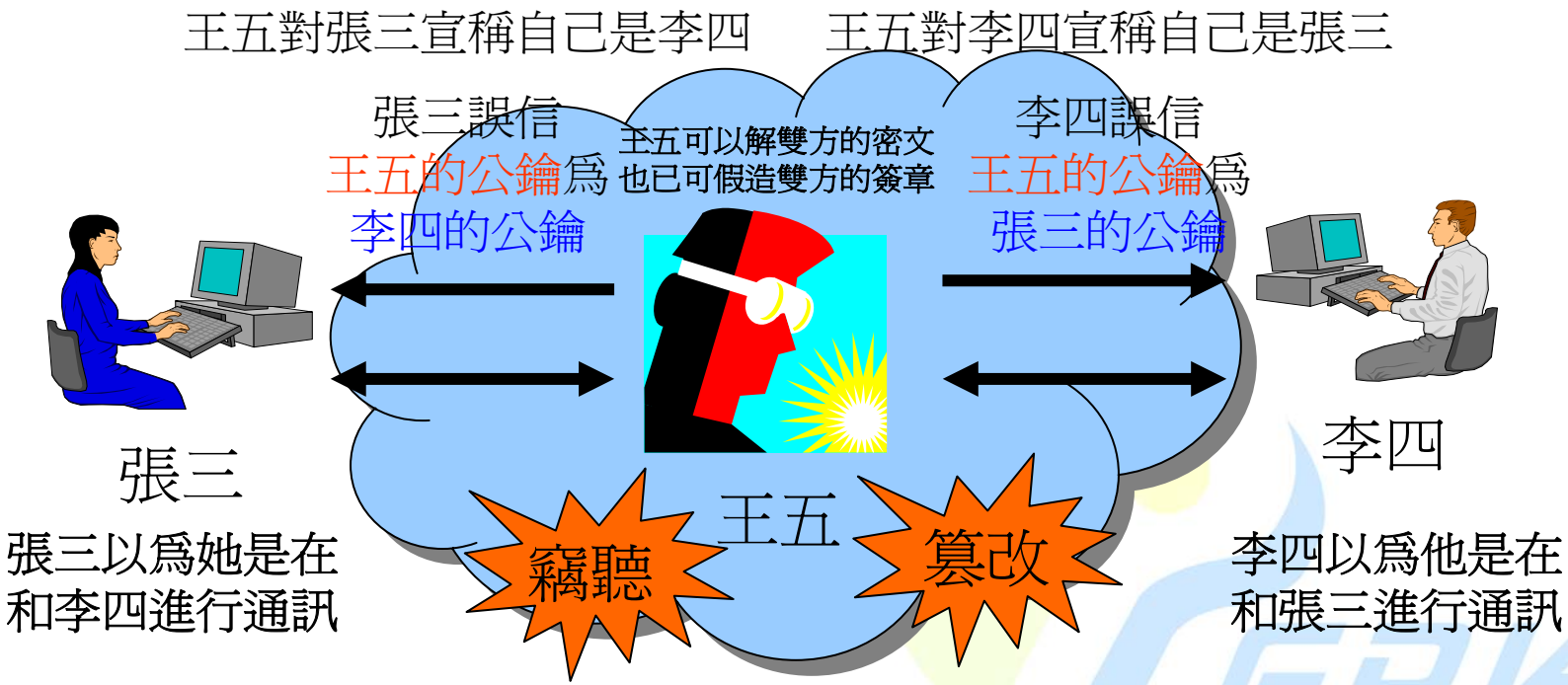
- 但問題真的有這麼簡單嗎？

On the internet, nobody knows you're a dog.

"On the Internet, nobody knows you're a dog."

By Peter Steiner on page 61 of July 5, 1993 issue of the New Yorker

Man-in-the-middle (MITM) Attack



- 公開金鑰密碼系統的運作是建立在「通訊雙方能夠正確地取得對方公鑰」的前提下，否則即有可能使訊息洩漏或收到偽造的訊息而不自知。

Man-in-the-middle (MITM) Attack

- From Wikipedia, the free encyclopedia.
  - In cryptography, a man in the middle attack (MITM) is an attack in which an attacker is able to read, and modify at will, messages between two parties without either party knowing that the link between them has been compromised. The attacker must be able to observe and intercept messages going between the two victims.
  - The possibility of a "man in the middle" attack remains a serious security problem for public-key based cryptosystems. A widely used mechanism for defeating such attacks is the use of digitally signed keys.
  - Such signed keys (eg, signed by a certificate authority) are one of the primary mechanisms used for secure world wide web traffic (eg, HTTPS SSL or Transport Layer Security protocols). However, lack of care in endorsing the match between identity information and public keys by certificate authorities is a problem for these systems.
  - While this example focuses on the MITM attack in a cryptographic context, MITM should be seen as a general problem resulting from any use of intermediate parties acting as a proxy for the clients on either side.

公開金鑰基礎建設
(Public-Key Infrastructure，PKI)

Certification Authority (CA)

- 公開金鑰密碼系統的運作是建立在「通訊雙方能夠正確地取得對方公鑰」的前提下，否則即有可能使訊息洩漏或收到偽造的訊息而不自知。
- 為了要確定「通訊雙方能夠正確地取得對方公鑰」，所以必須由通訊雙方都信任的可信賴第三者（Trusted Third-Party，TTP）經一定的程序，驗證個體之身分與金鑰對後簽發憑據，證明該個體確實擁有與其所宣稱的公鑰相對應之私鑰的憑據。
- 此種憑據稱為公鑰憑證（Public-Key Certificate），簡稱憑證，而簽發憑證的可信賴第三者稱為憑證機構（ Certification Authority，CA），或稱憑證管理中心。
- 因為CA是通訊雙方共同信賴的機構，所以雙方可在檢驗憑證的合法性後，經由proof-of-posession（PoP）的方法確認對方確實持有與其憑證相對應之私密金鑰（Private-Key Certificate），進而確認對方的身分。

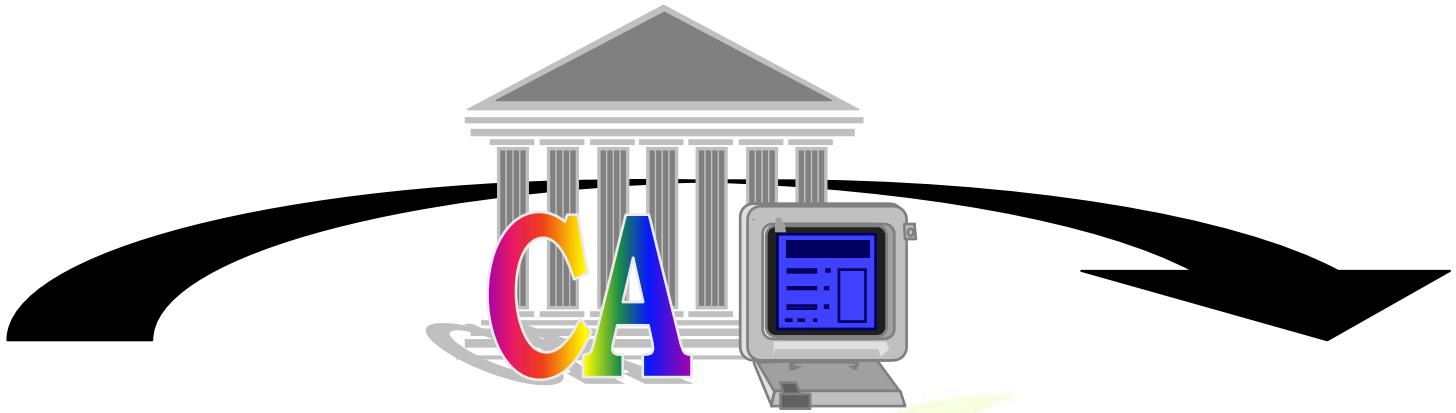
傳統的印鑑證明申請程序

1.刻印章

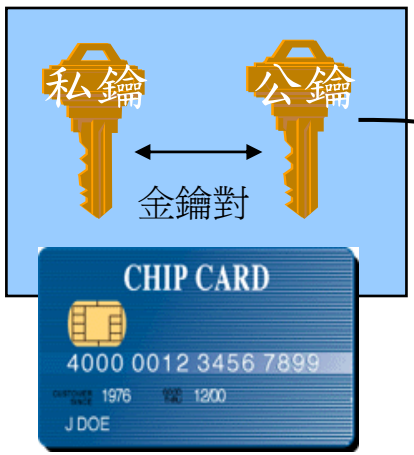
2.向戶政事務所
申請印鑑證明書

3.印鑑證明在手

公鑰憑證申請程序



1.產製金鑰對



2.向CA申請公鑰憑證

公鑰

100101010100010101
010010100101010100
010101010010100101
010100010101010010
101000101010110100
01010101

3.CA公佈公鑰憑證

公鑰憑證

……

發證者名稱

有效期限

憑證主體名稱

憑證主體公鑰

……

CA簽章

ITU-T X.509公鑰憑證格式

1.基本欄位

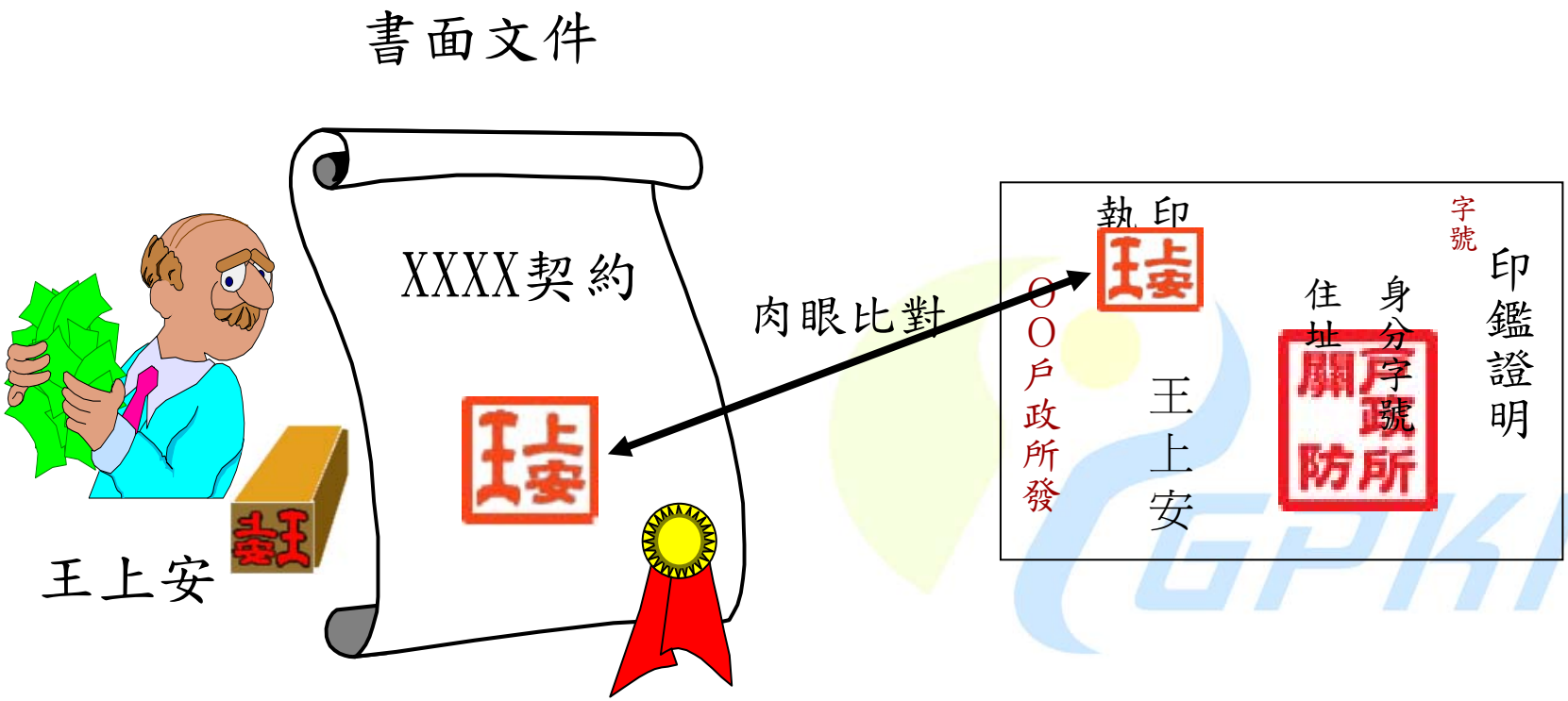
2.擴充欄位

憑證格式版本
憑證序號
簽章演算法識別碼
發證者名稱
憑證有效期限
憑證主體名稱
憑證主體公開金鑰
簽發者唯一識別碼
憑證主體唯一識別碼

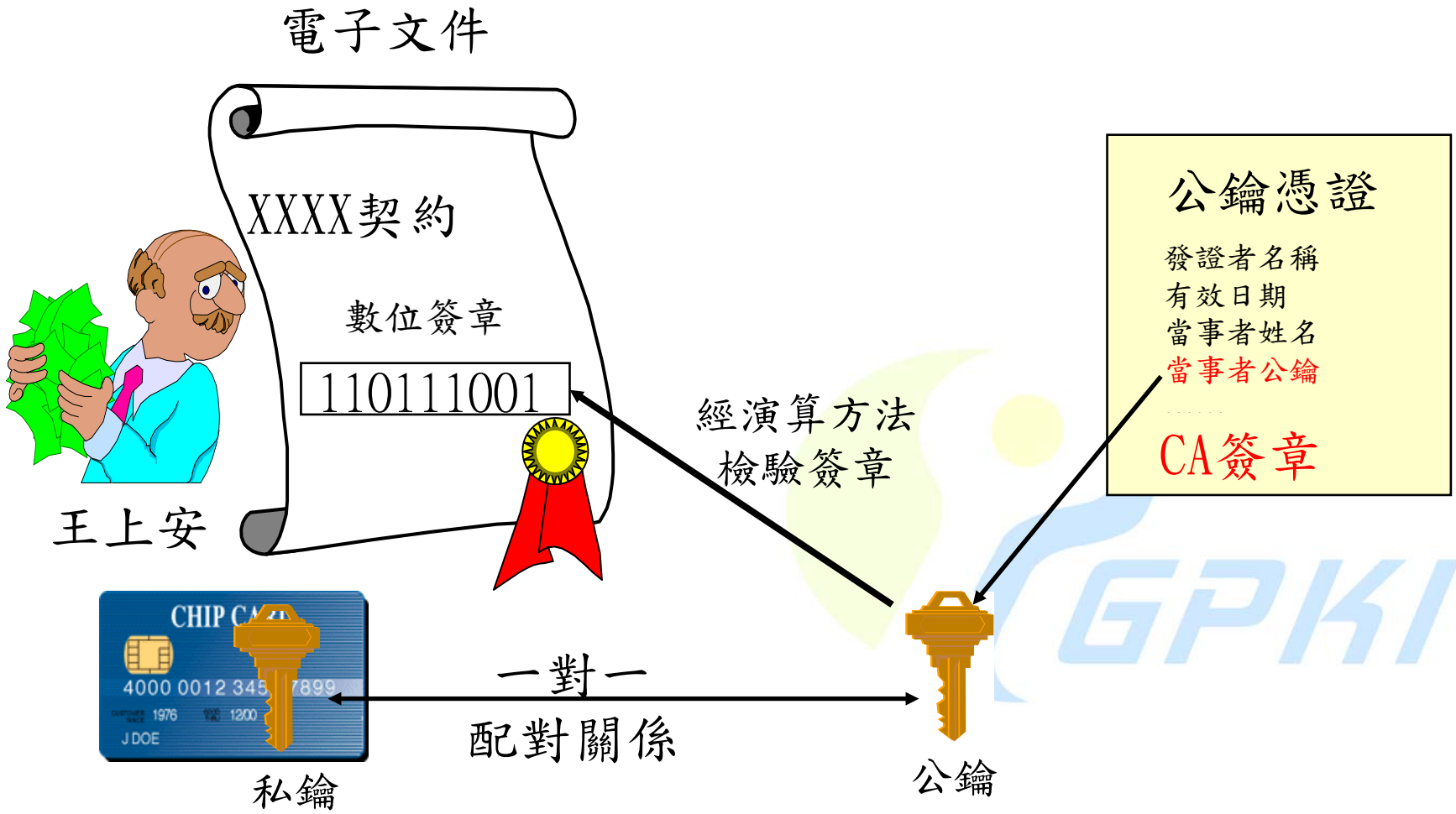
Issuer Key Identifier
Subject Key Identifier
Key Usage
Certificate Policies
Basic Constraints
CRL Distribution Points
Subject Directory Attribute
……

簽章演算法識別碼
CA Signature

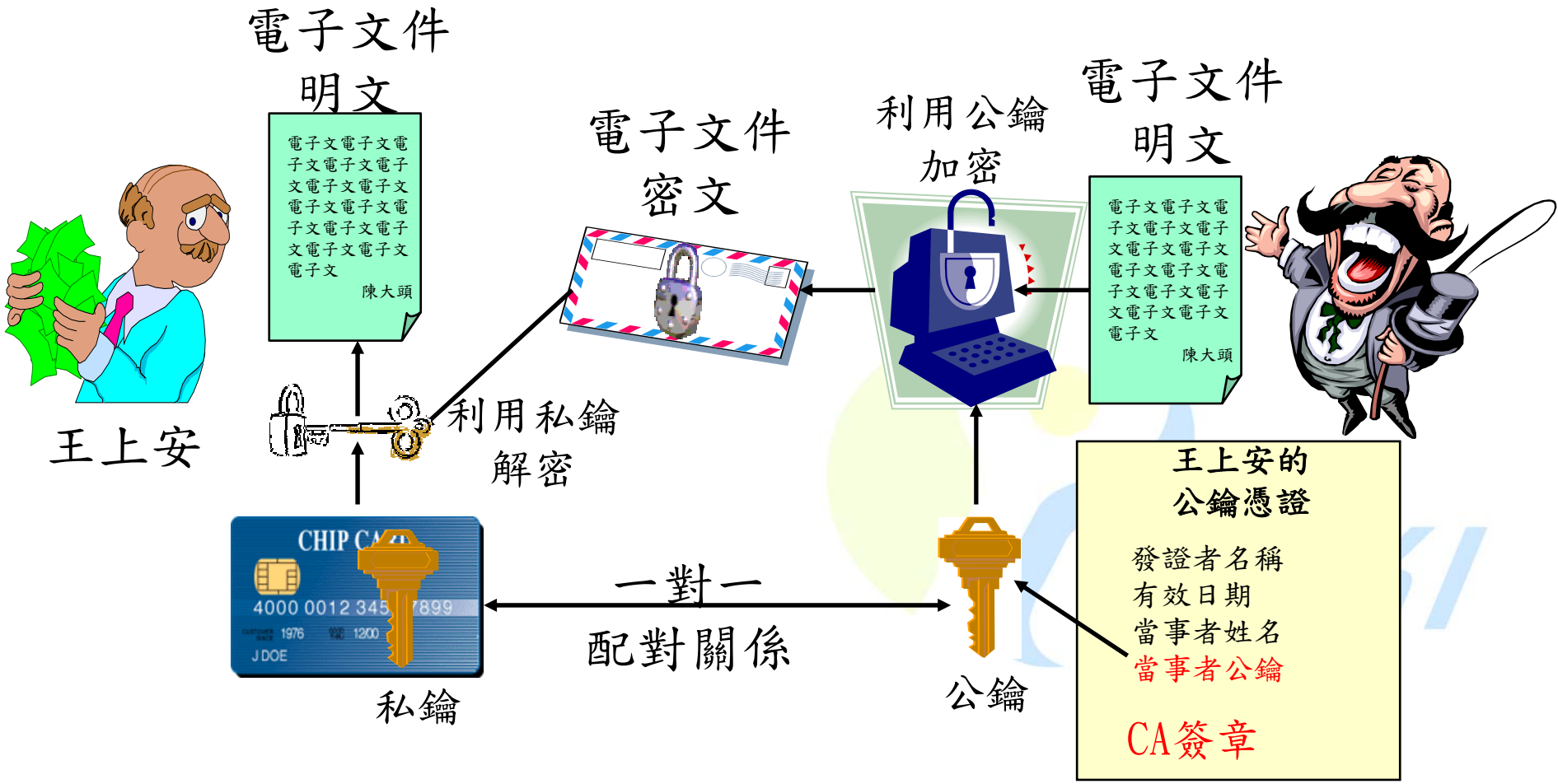
傳統印鑑之使用方式



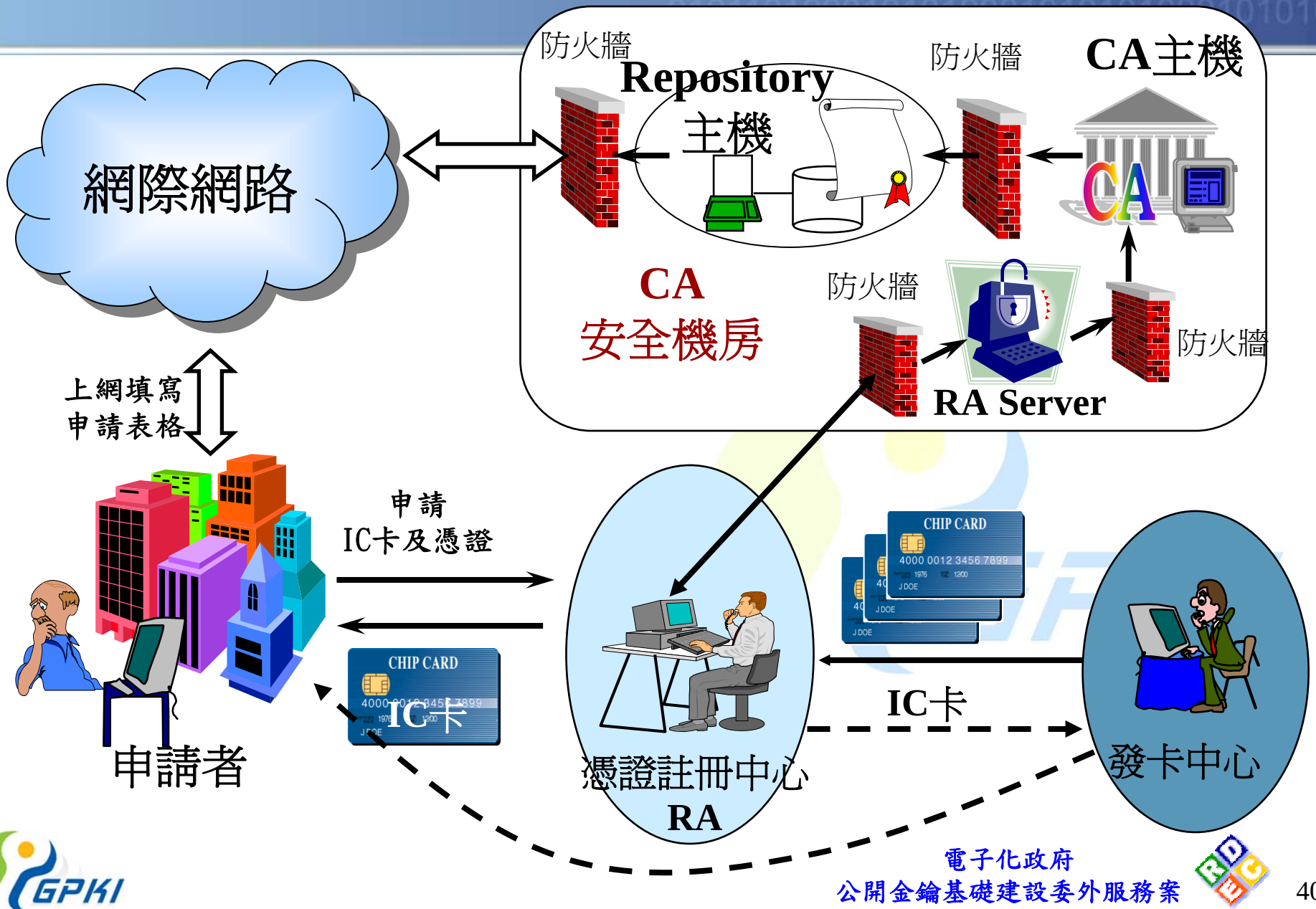
電子印鑑之使用方式


電子文件
XXXX契約
數位簽章
110111001
王上安
經演算方法
檢驗簽章
公鑰憑證
發證者名稱
有效日期
當事者姓名
當事者公鑰
CA簽章
CHIP CARD
4000 0012 345 7899
1976
1200
J DOE
一對一
配對關係
私鑰
公鑰

電子印鑑之額外功能—開啟數位信封

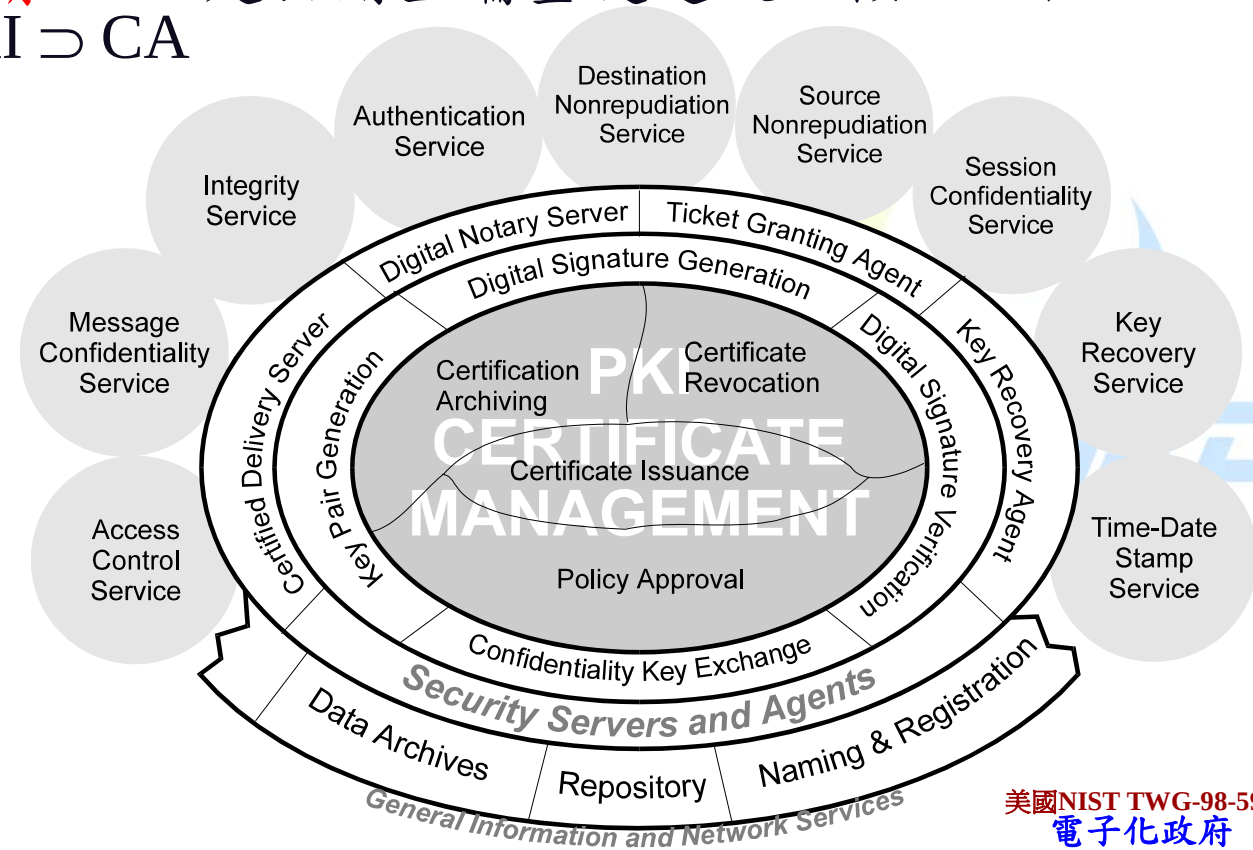


憑證管理系統架構


網際網路
防火牆
Repository
主機
防火牆
CA主機
CA
CA
安全機房
防火牆
防火牆
RA Server
上網填寫
申請表格
申請
IC卡及憑證
CHIP CARD
IC卡
IC卡
申請者
憑證註冊中心
RA
發卡中心

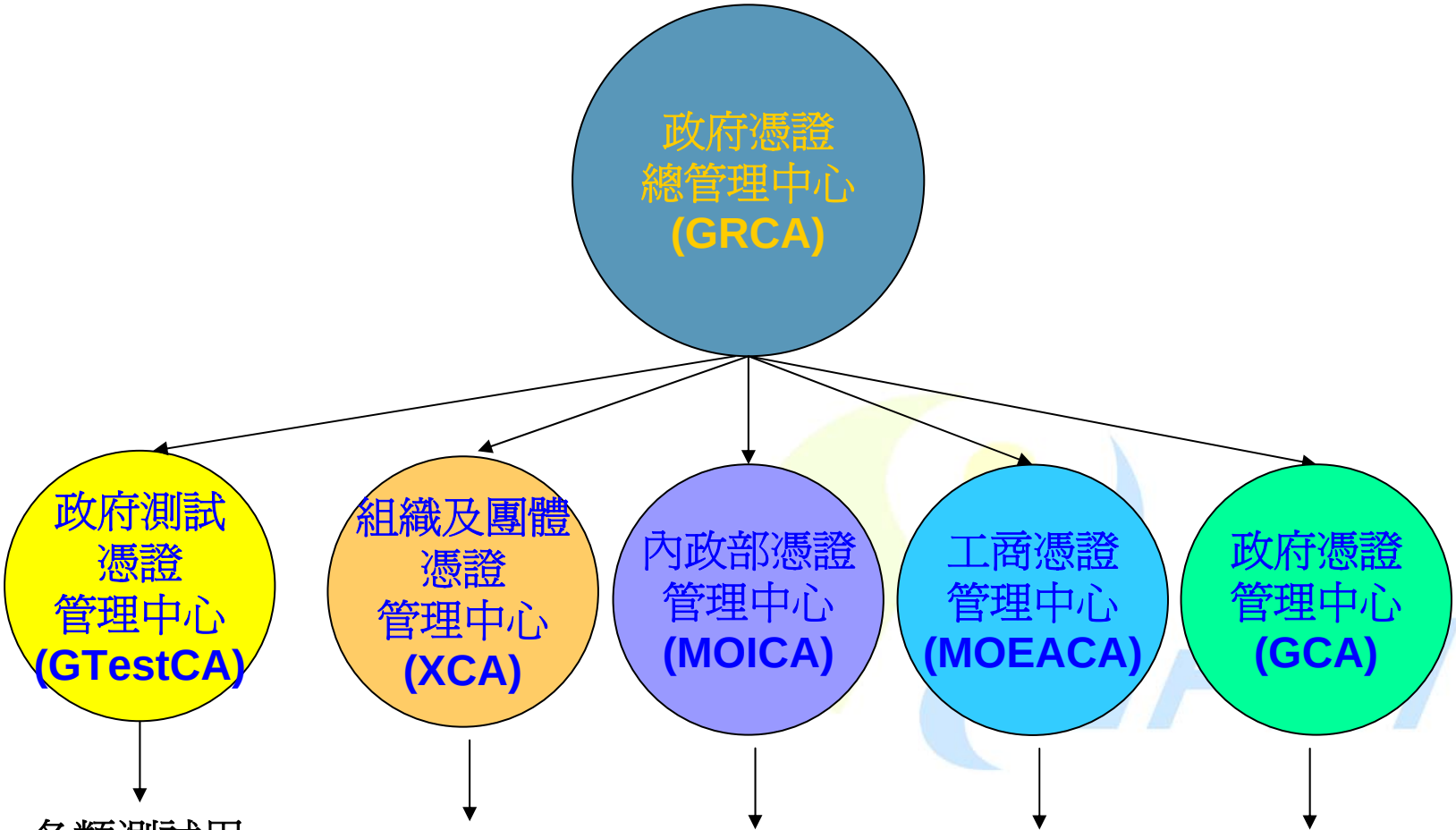
公開金鑰基礎建設(PKI)服務及功能

- What is PKI?
  - PKI是一種支持公開金鑰密碼系統正常運作的Infrastructure，所謂Infrastructure包含設備、設施、服務、人員、法律、政策、規範等
- 觀念澄清：CA是公開金鑰基礎建設之核心，但　　PKI≠CA，而是PKI⊃CA



我國PKI架構及應用

我國政府公開金鑰基礎建設 (Government PKI，GPKI)



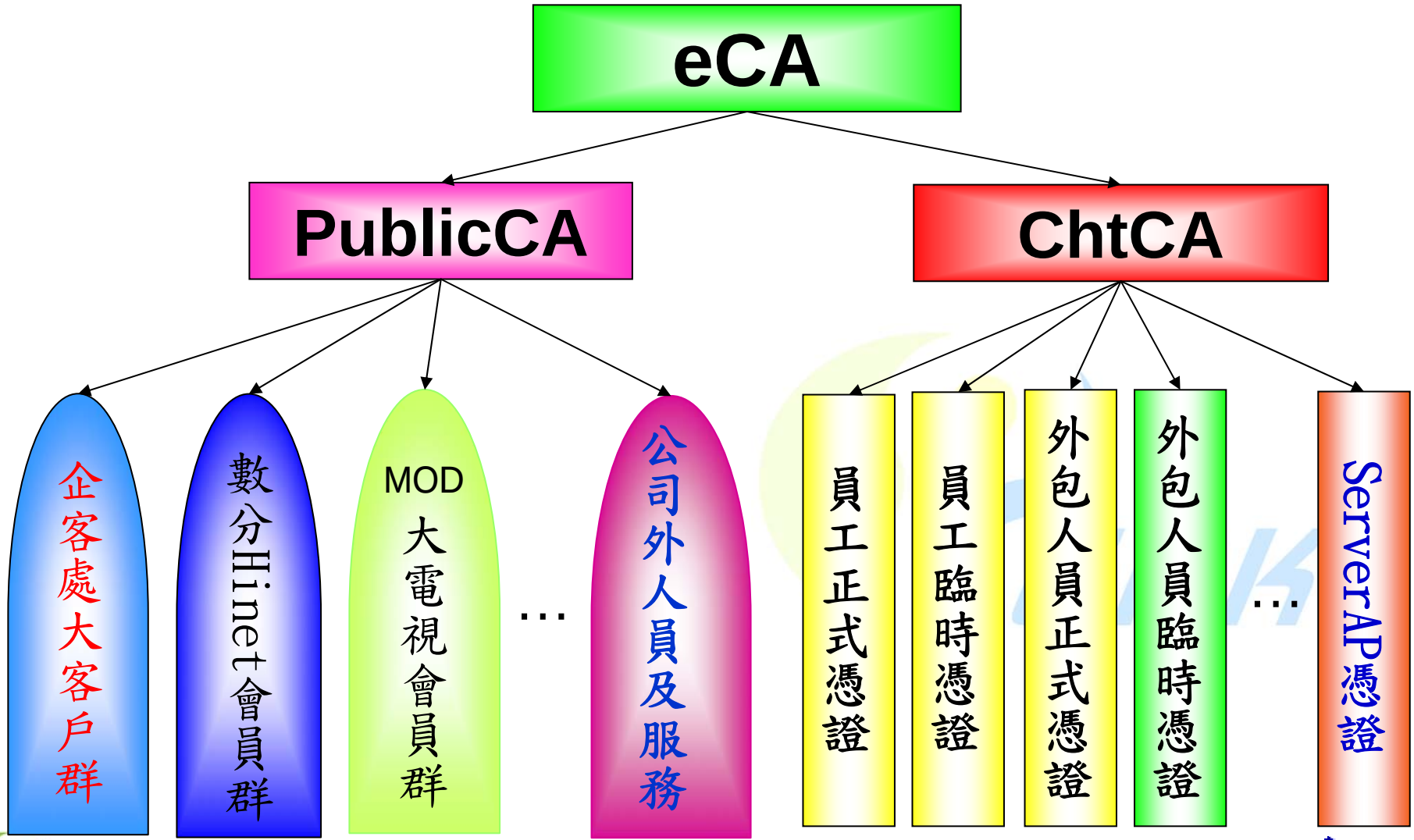
GPKI相關應用系統

主管機關	應用系統	自然人憑證	機關（單位）憑證	公司商號憑證	伺服器憑證	專屬憑證
司法院（資訊管理處）	法院囑託限制登記網路作業		V			
最高法院檢察署	全國檢察機關通訊監察子系統		V			
行政院（秘書處）	政府機關電子公文交換		V		V	
	立委質詢答復系統		V			
內政部	戶役政電子閘門		V		V	
	地政電子閘門	V	V		V	
	警政治安電子閘門資訊系統		V		V	
財政部	台北市國稅局網際網路資訊服務站	V			V	
	台北區支付處電子支付系統				V	V
	財稅中心五年財稅平台計畫（網路申報繳稅）：綜合所得稅結算申報繳稅	V			V	
	財稅中心五年財稅平台計畫（網路申報繳稅）：各類所得扣/免繳資料及股利憑單申報		V	V	V	
	財稅中心五年財稅平台計畫（網路申報繳稅）：營利事業所得稅結算申報、繳稅		V	V	V	
	財稅中心五年財稅平台計畫（網路申報繳稅）：營業稅申報、繳稅		V	V	V	
	財稅中心五年財稅平台計畫（網路申報繳稅）：營利事業所得稅暫繳申報、繳稅		V	V		
	關稅總局貨物通關電子閘門應用系統		V		V	

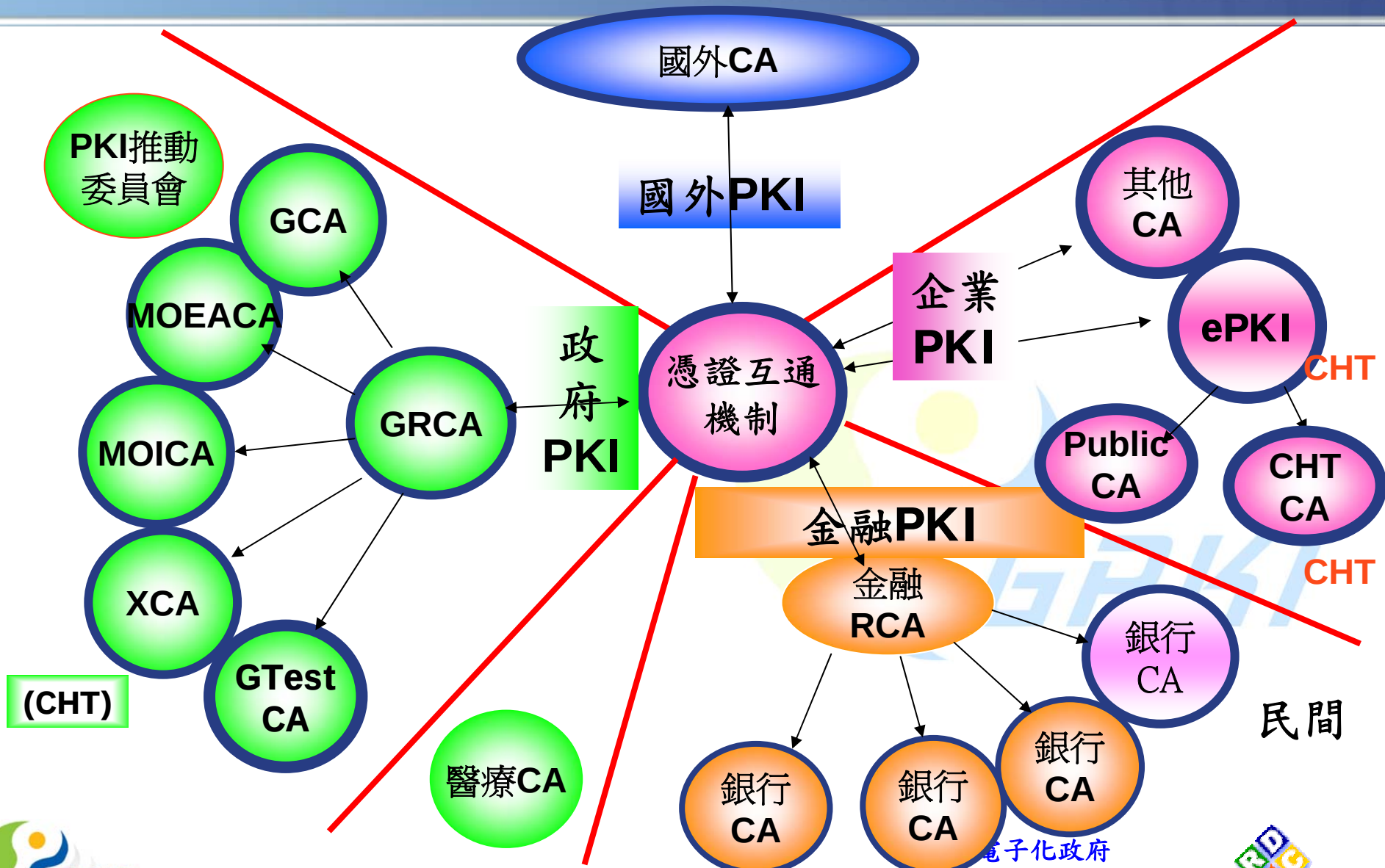
GPKI相關應用系統

	金融局文件簽章與數位信封上傳系統					V
	台灣省北區國稅局稅務ｅ網通線上申辦系統				V	
經濟部	工商電子閘門		V		V	
	標準檢驗局免驗作業線上申辦系統	V		V		
	水利署水權網路申辦系統	V				
交通部	電子公路監理	V			V	
	台中港務局資訊系統身分認證機制	V				
中央銀行	國庫局中央公債及國庫券連線投標作業系統				V	V
	金檢處金融監理資訊申報作業系統				V	V
	公開市場操作				V	V
輔導會台北榮民總醫院	電子病歷交換安全控管技術					
勞委會	勞保局農勞保網路申辦系統	V			V	V
	職訓局製造業外勞展延案網路申報系統					
工程會	廠商電子型錄、廠商電子詢報價及電子下訂系統			V	V	
	政府採購電子領標暨投標系統				V	V
台北市政府	消防局火災地點查詢系統		V			
台北縣政府	工務局建築執照管理系統	V		V		

中華電信ePKI 架構


eCA
PublicCA
ChtCA
企客處大客戶群
數分Hinet會員群
MOD
大電視會員群
…
公司外人員及服務
員工正式憑證
員工臨時憑證
外包人員正式憑證
外包人員臨時憑證
…
ServerAP憑證

我國PKI現況


國外CA
國外PKI
PKI推動委員會
GCA
MOEACA
GRCA
MOICA
XCA
GTest CA
(CHT)
政府PKI
憑證互通機制
企業PKI
其他CA
ePKI
CHT
Public CA
CHT CA
CHT
金融PKI
金融RCA
銀行CA
銀行CA
銀行CA
銀行CA
醫療CA
民間

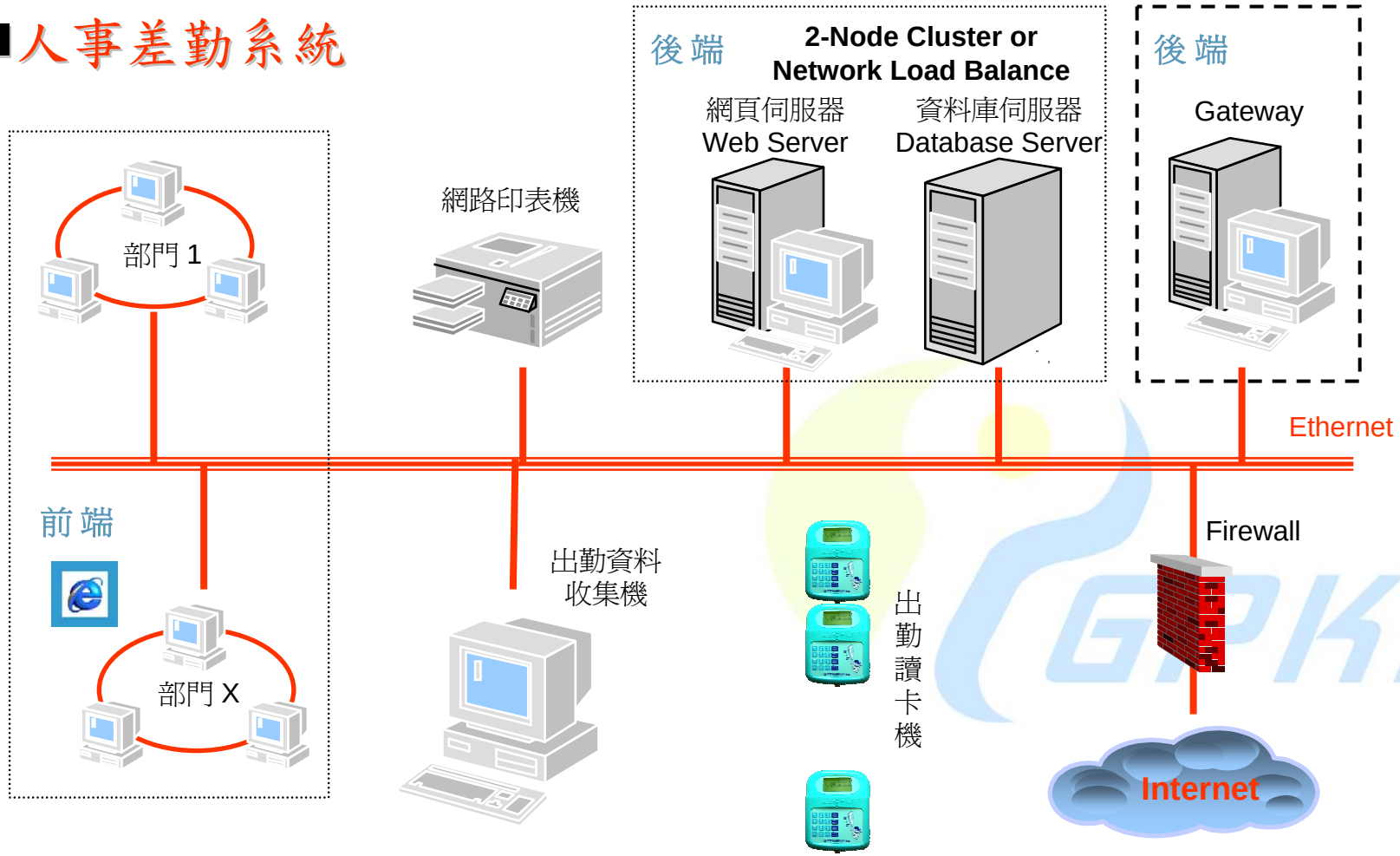
中華電信企業PKI之應用

差勤系統結合電子簽章

系統架構

差勤系統系統架構

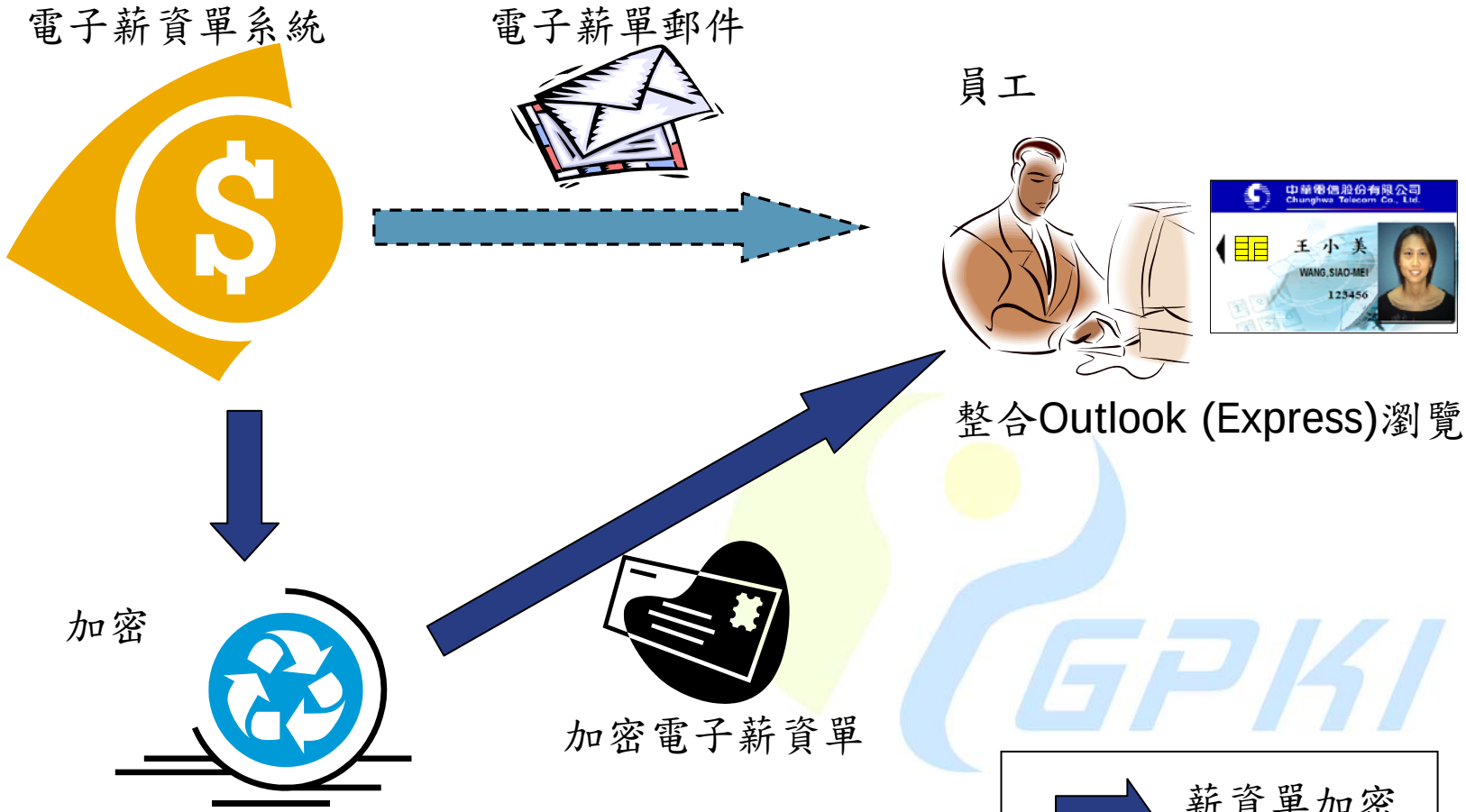
■人事差勤系統



榮譽

- 」中華電信所研發之「人事差勤管理系統 」榮獲2003年經濟部國際貿易局「第十一屆台灣精品獎」

電子薪資單加解密機制



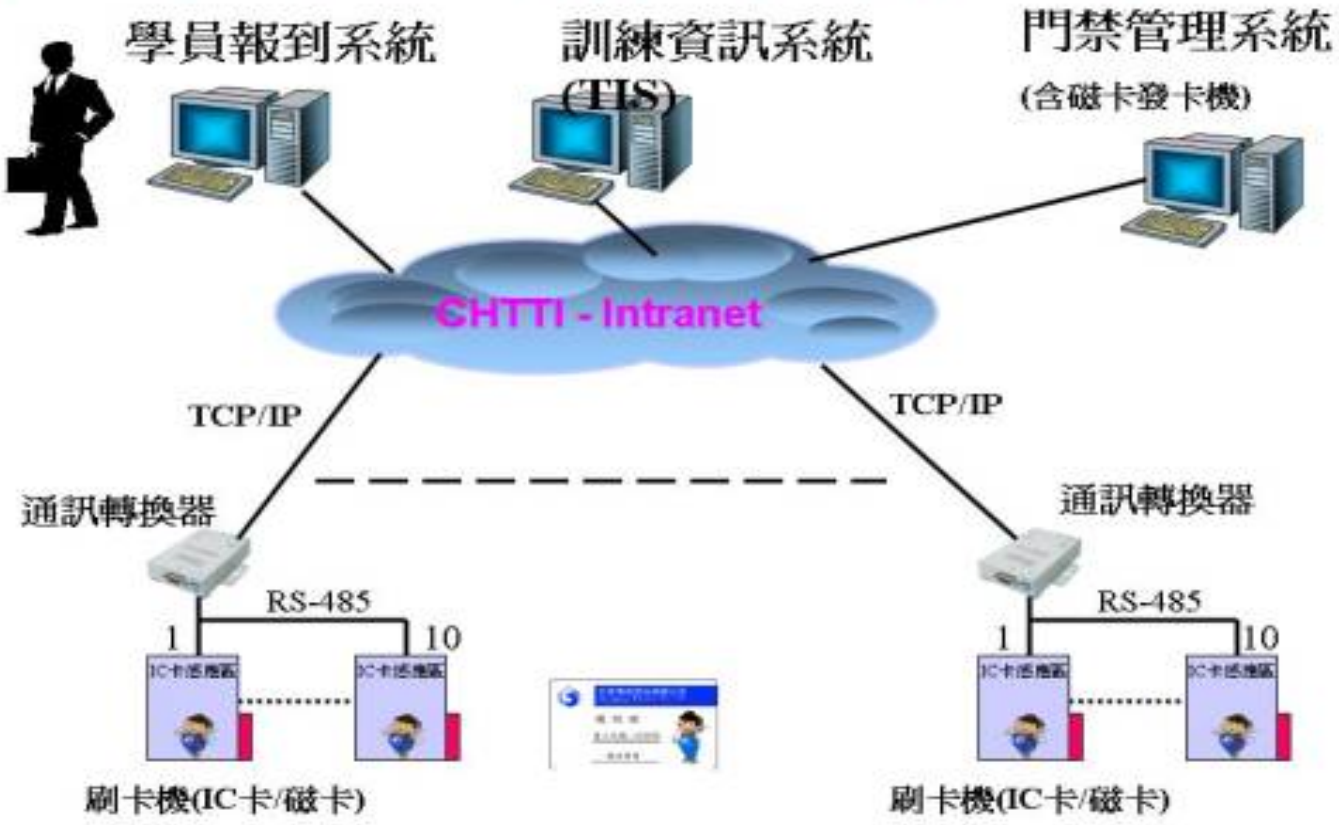
背景：

民營化後薪資獎金屬個人保密資訊

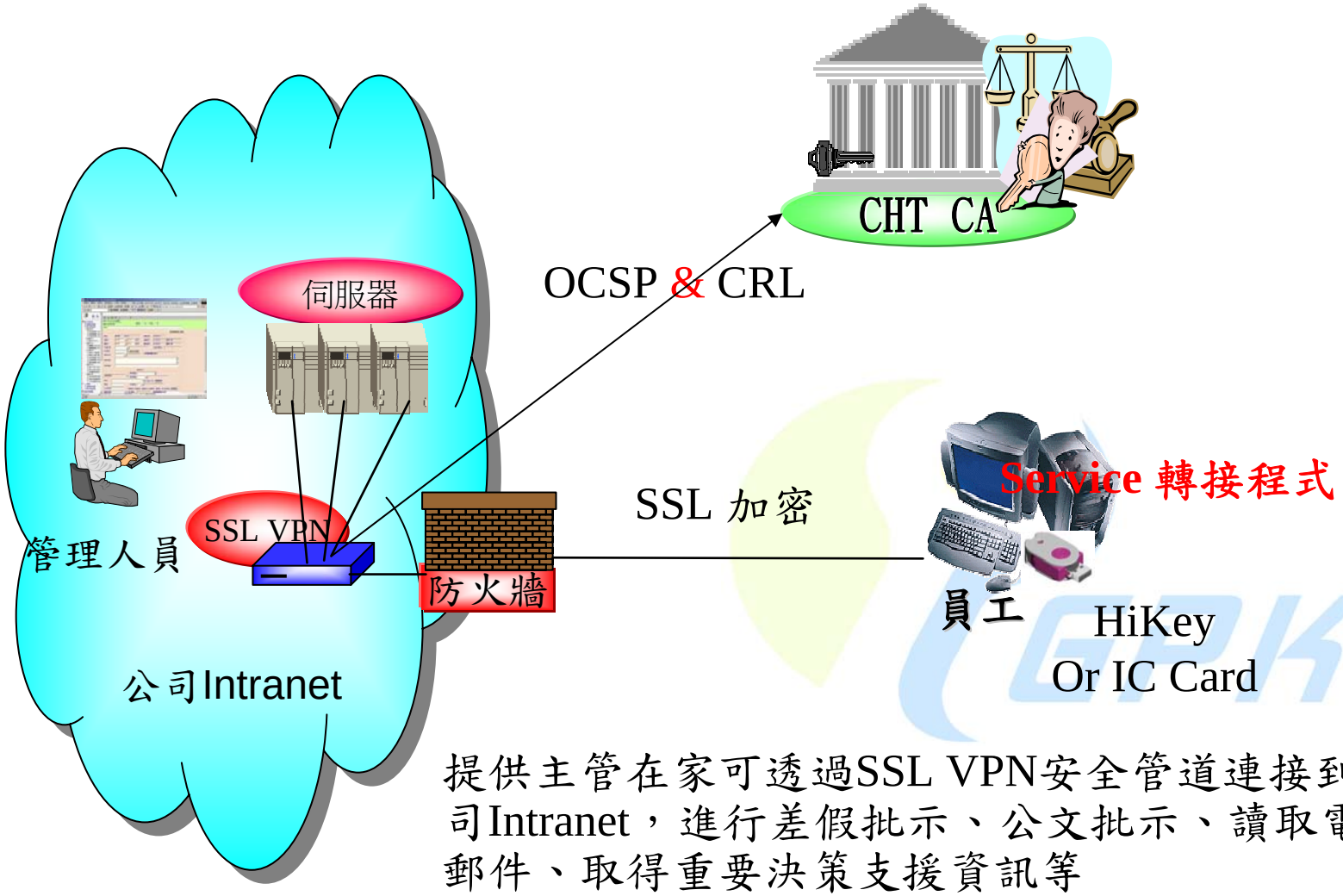
員工識別證已全區換發完成具PKI功能之晶片卡

電信訓練所學員刷卡報到及宿舍管理自動化服務系統

電信訓練所學員報到系統與門禁系統之整合架構圖

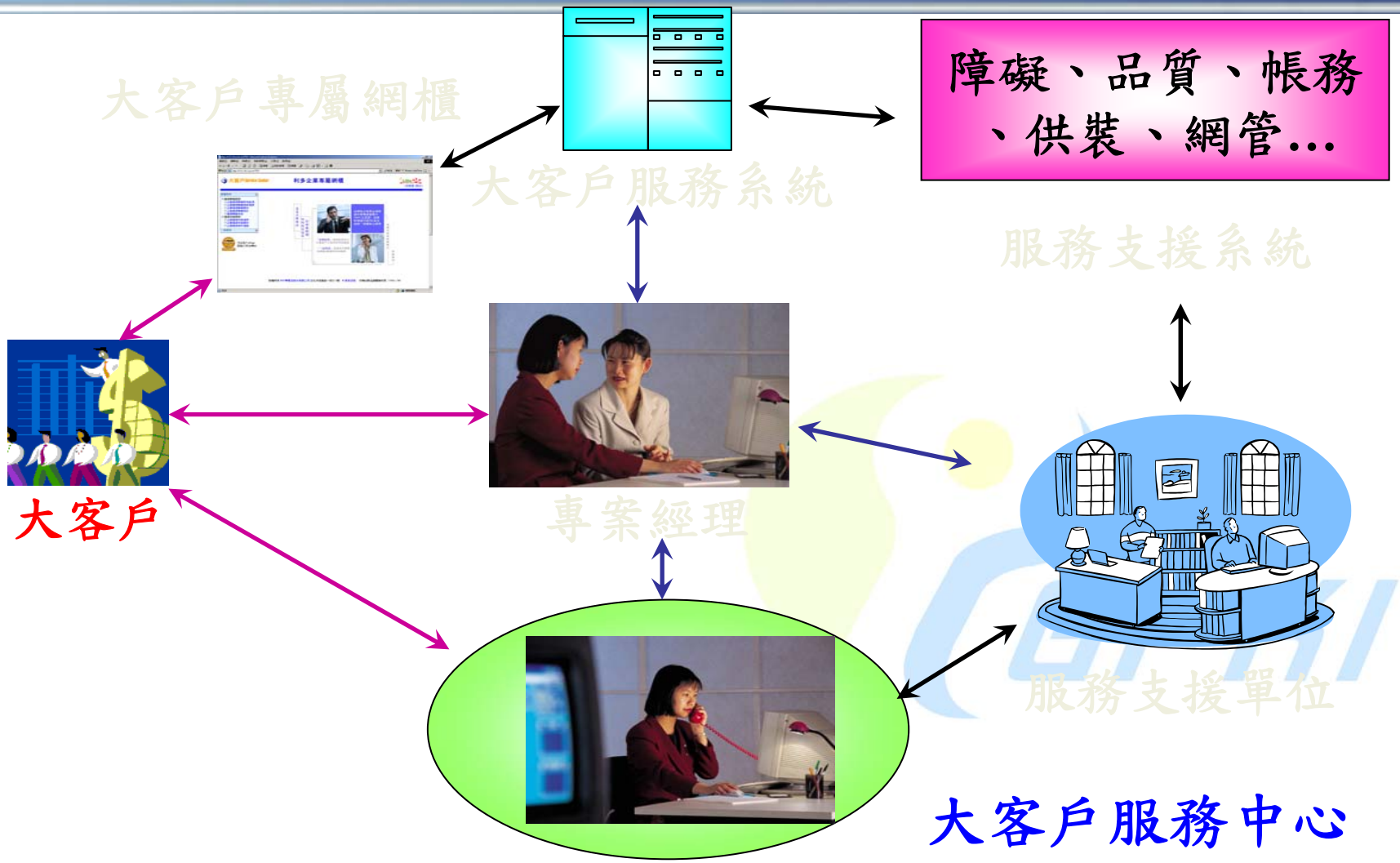


中華電信企業SSL VPN



提供主管在家可透過SSL VPN安全管道連接到公司Intranet，進行差假批示、公文批示、讀取電子郵件、取得重要決策支援資訊等

中華電信大客戶服務中心



中華電信大客戶服務系統的資訊安全機制

- 為了提供大企業客戶一個安全可靠的網路使用環境，大客戶服務系統特別導入一套由電信研究所自行研發之智慧卡（IC卡）認證及個人化安控機制
- 配合SSL、公鑰憑證及數位簽章等加解密與認證的技術
- 使得企業客戶租用電路的申告、供裝、查修、告警資料或大批受理及跨業務整批供裝等機密性資料，均可免除被竊取、被竄改、被冒名或否認傳送等各類的網路資訊安全事件發生。

中華電信大客戶服務系統的資訊安全機制

- 為了提供大企業客戶一個安全可靠的網路使用環境，大客戶服務系統特別導入一套由電信研究所自行研發之智慧卡（IC卡）認證及個人化安控機制
- 配合SSL、公鑰憑證及數位簽章等加解密與認證的技術
- 使得企業客戶租用電路的申告、供裝、查修、告警資料或大批受理及跨業務整批供裝等機密性資料，均可免除被竊取、被竄改、被冒名或否認傳送等各類的網路資訊安全事件發生。

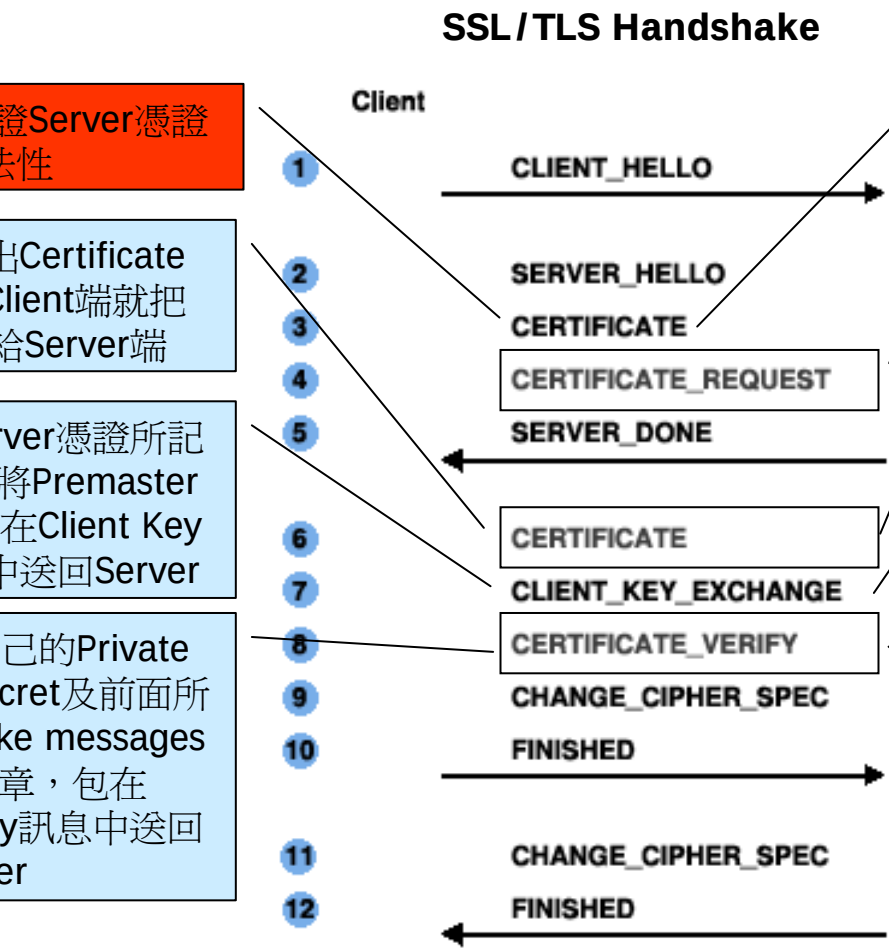
中華電信大客戶智慧卡認證應用

對抗Man-in-the-middle攻擊的對策

對抗MITM Attack的對策

- 通信雙方交換由可信賴之CA所簽發的公鑰憑證
  - 注意：對方送來某甲的公鑰憑証，並不代表他就是某甲，一定要搭配PoP驗證(即Authentication)及憑證驗證
- 經由proof-of-possession（PoP）的方法去Authenticate對方的身分(即確認對方確實持有與其憑證相對應之私密金鑰)
  - PoP的方法：
    - 簽章/驗簽：送一段訊息請對方使用其私密金鑰加密簽章，然後用其公開金鑰來驗證其簽章是否正確
    - 加密/解密：使用對方的公開金鑰加密一段訊息，再請對方使用其私密金鑰將該訊息解密
  - 注意：確認對方確實持有與憑證對應的私密金鑰，並不代表他就是憑證中所記載的個體，因為憑證可能是假造的
- 必須驗證對方憑證的合法性(Validity)
  - 注意：如果憑證驗證的過程不夠嚴謹，則protocol設計得再好也沒有用。大部分的protocol對於如何驗證憑證都少有著墨，因為protocol的設計者都假設implementation的憑證驗證過程是正確的。

對抗MITM Attack的對策
-- 以SSL/TLS Handshake為例



Client端必須驗證Server憑證的合法性

如果Server送出Certificate Request ，則Client端就把Client憑證送給Server端

Client端使用Server憑證所記載的Public Key將Premaster Secret加密，包在Client Key Exchange訊息中送回Server

Client端使用自己的Private Key對master secret及前面所傳送之handshake messages的Hash值簽章，包在Certificate Verify訊息中送回Server

Server端把Server憑證送給Client端

如果Server也想認證Client端的身分，則送出Certificate Request要求Client送回Client憑證

Server端必須驗證Client憑證的合法性

Server端如果能夠使用其Private Key解開Client Key Exchange訊息中的Premaster Secret密文，表示Server確實是Server憑證中所記載的主機

Server端將使用Client憑證中所記載的Public Key來檢驗Certificate Verify的簽章，如果簽章檢驗通過，則表示Client端確實是憑證所記載之用戶

框在 [ ] 中的步驟，表示此步驟爲**optional**

5. 憑證驗證代理機制簡介

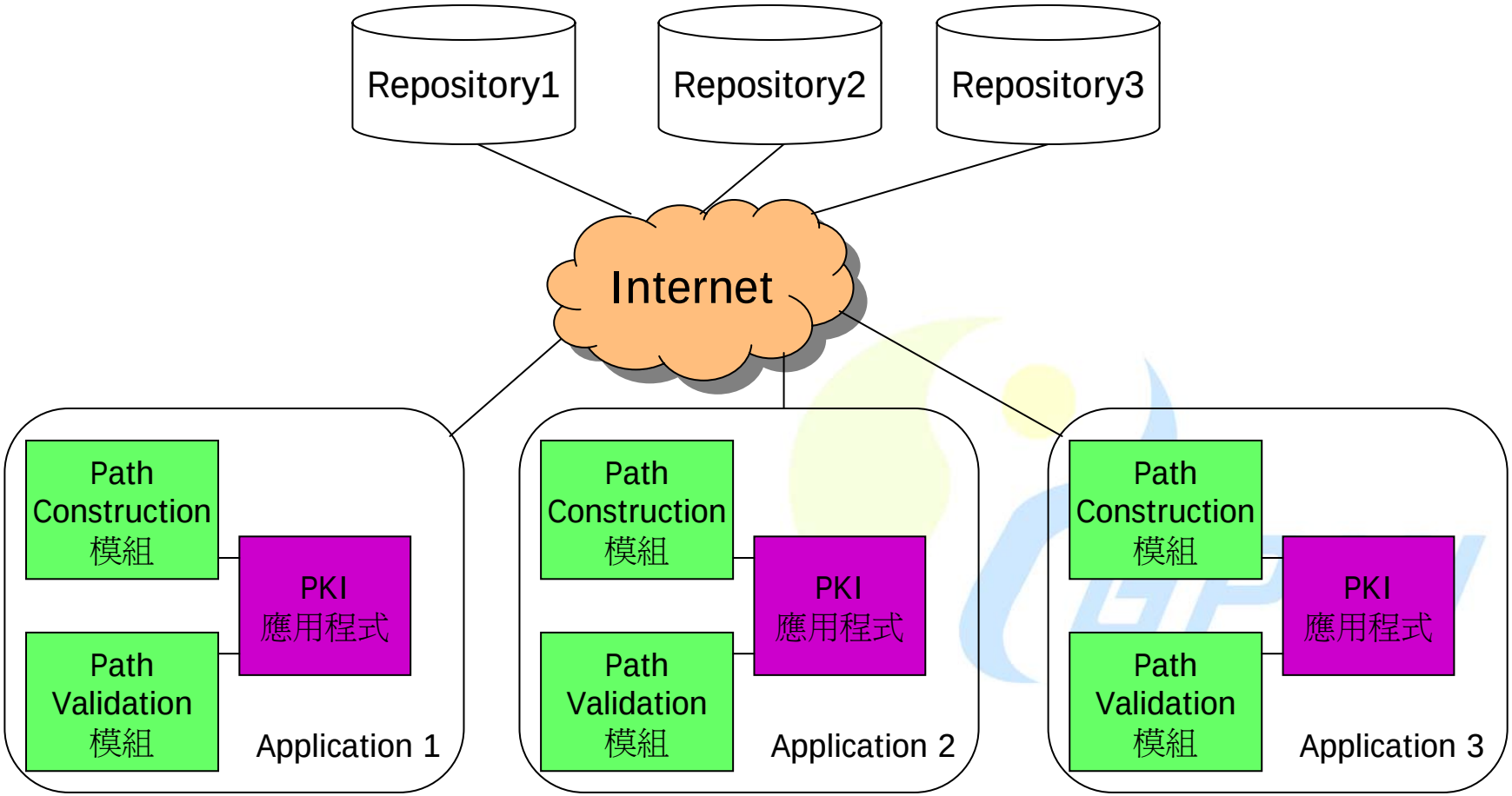
應用系統在驗證憑證時的問題

- 建構憑證路徑(Certification Path Discovery)複雜性高。
- 憑證路徑驗證(Certification Path Validation)複雜性高。
- PKI建構者與應用系統供應商間的整合應用之隔閡(Gap)。
- 應用系統的憑證路徑驗證方法可能不符合PKI的規範。
- 各應用系統的憑證驗證政策(Validation Policy，VP)可能不一致。(或是根本沒有憑證驗證政策)
- 應用系統無法有彈性地驗證所有類別的憑證，尤其是新種類的憑證或跨領域的憑證。

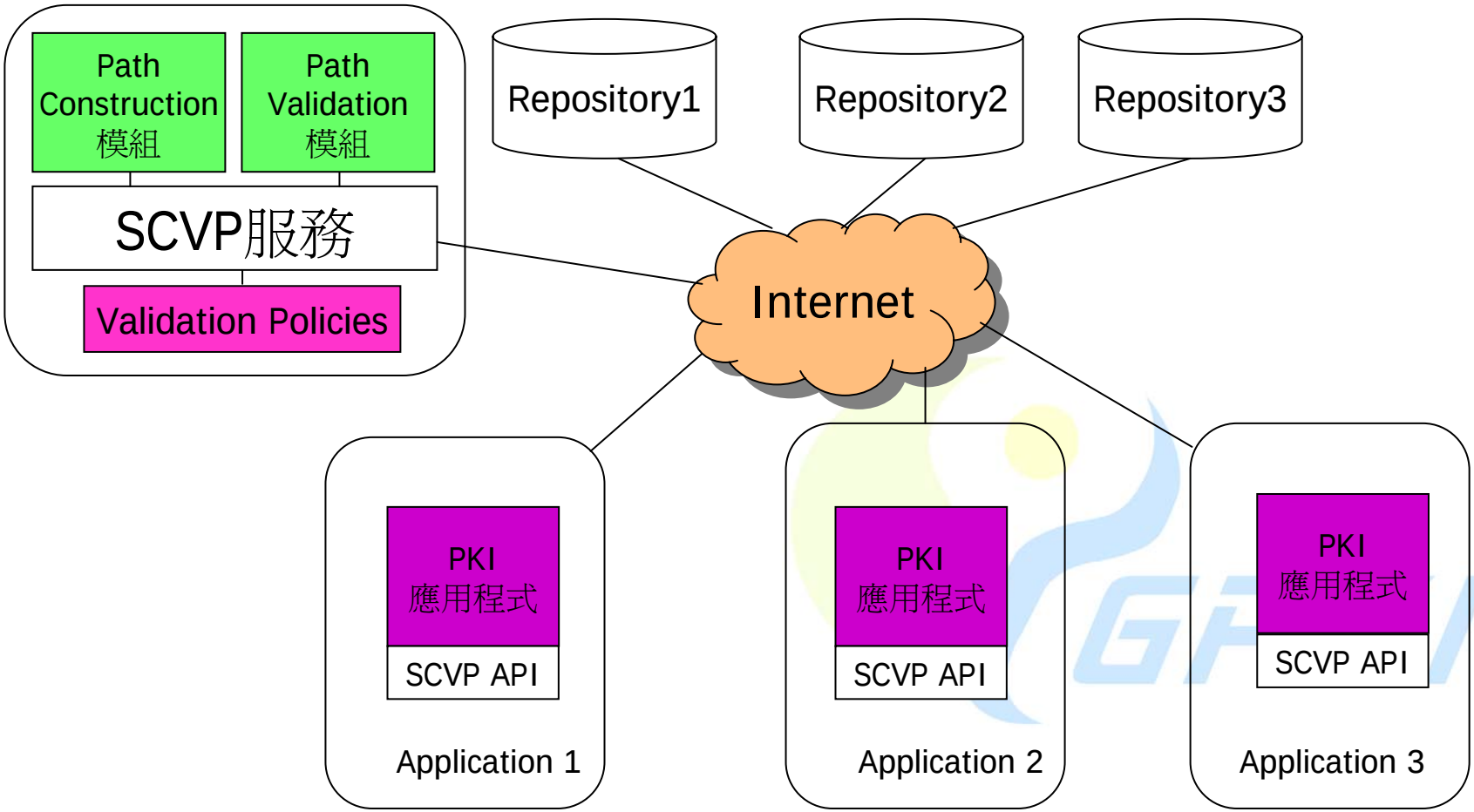
憑證驗證服務(SCVP)的目標

- Server-Based Certificate Validation Protocol (SCVP)的目標：
  - 提供代理憑證路徑建構（Delegated Path Discovery，DPD）服務。
  - 提供代理憑證路徑驗證（Delegated Path Validation，DPV）服務。
  - 採用符合國際規範的憑證路徑建構及驗證演算法。
  - 提供憑證驗證服務Client端程式介面（API）。
  - 減低Application開發廠商的負擔。
  - 確保Application依照統一的政策(VP)進行憑證路徑處理。
  - 可驗證GPKI所有類別的憑證，並可驗證新類別的憑證，未來並可驗證跨領域的憑證。

Without SCVP Service



With SCVP Service



憑證驗證相關國際規範

- ITU-T Recommendation X.509
  - Information Technology - Open Systems Interconnection – The Directory: Authentication Framework
- RFC 3280
  - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
- RFC 3379
  - Delegated Path Validation and Delegated Path Discovery Protocol Requirements
- RFC 5055
  - Server-Based Certificate Validation Protocol (SCVP)
- Appendix 3 & 4 of JKST-IWG 2003 Final Report
  - Certificate Path Processing Implementation Guideline
  - JKST-IWG Certificate Path Processing Testing Guideline

Useful Resources

- GRCA網站：http://grca.nat.gov.tw
- GCA網站：http://gca.nat.gov.tw
- MOICA網站：http://moica.nat.gov.tw
- MOEACA網站：http://moeaca.nat.gov.tw
- XCA網站：http://xca.nat.gov.tw
- GTestCA網站：http://gtestca.nat.gov.tw
- 政府機關公開金鑰基礎建設憑證政策（http://grca.nat.gov.tw）
- GPKI憑證及憑證廢止清冊格式剖繪(http://grca.nat.gov.tw)
- 公鑰憑證處理安全檢查表（http://gca.nat.gov.tw）

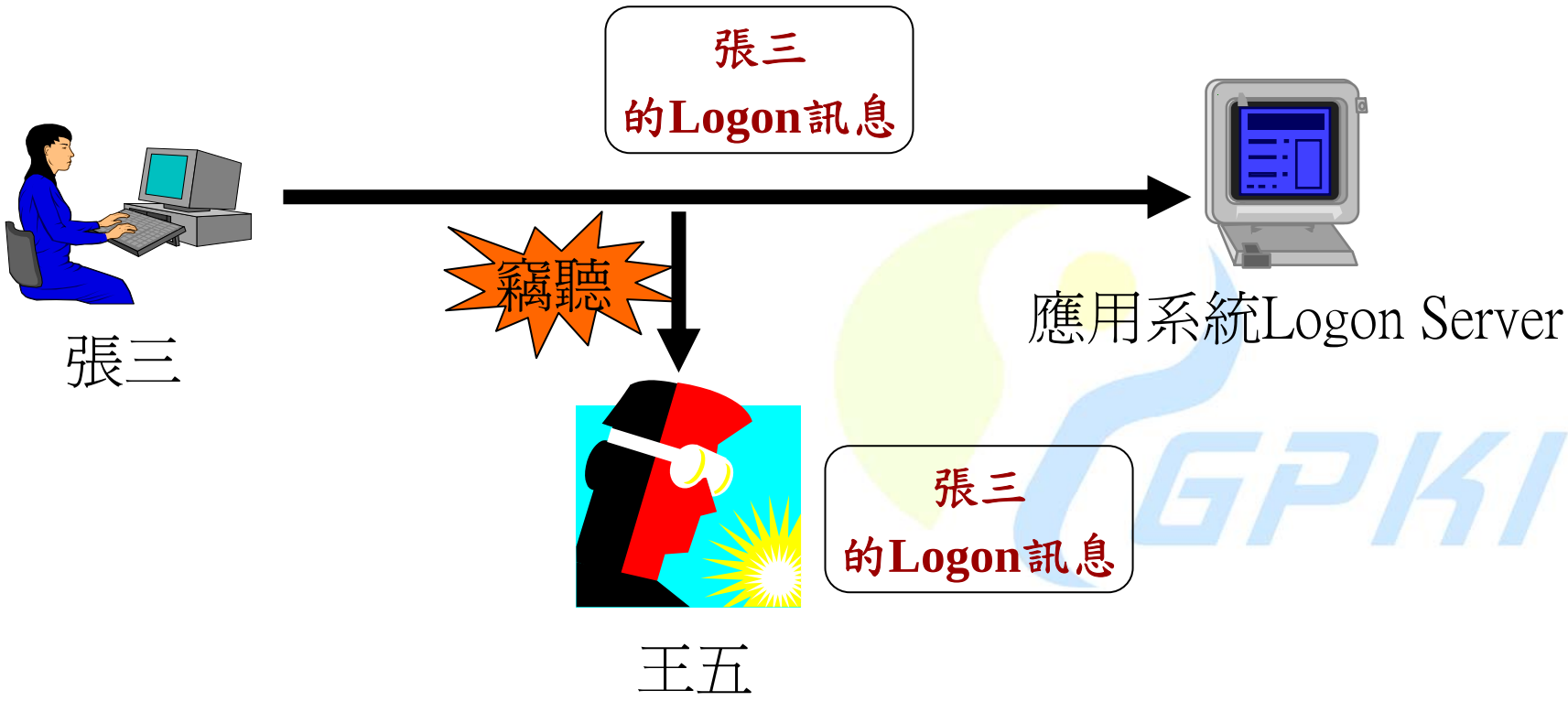
Replay攻擊及其對策

數位簽章可以防止冒名傳送？

- 理論上：
  - 無法由Public Key推算得到其相對應的Private Key
    - 理論上可以推算，但必需花很高的成本及很長的時間
  - 唯有持有相對應的Private Key的個體能夠產生能通過相對應之Public Key檢驗之數位簽章
  - 簽章演算法沒有瑕疵，所以不可能靠演算法的瑕疵來假造數位簽章
  - 數位簽章可以用來鑑定資料來源（authenticate the origin of data），因此可以防止冒名傳送
- 實務上：
  - What if 所傳送的數位簽章資料被意圖不法者側錄下來，然後重送（Replay）呢？
  - Replay Attack所需要的成本比去破解Private Kay或簽章演算法小太多了！

Replay Attack：以Logon為例(1/2)

Earlier Time：王五竊聽並將張三的Logon訊息存起來



Replay Attack：以Logon為例(2/2)

Later Time：王五將先前存起來之張三的Logon訊息重送給應用系統Logon Server，結果以張三之名成功登入系統，系統允許王五執行原本只有張三才能執行的作業項目

Replay Attack

- From Wikipedia, the free encyclopedia.
  - An attack in which a valid data transmission is maliciously or fraudulently repeated, either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack.
  - A way to avoid replay attacks is using session tokens.
  - Session tokens can be chosen at random or using any algorithm that prevents duplicates.
  - Timestamping is another way of preventing a replay attack.

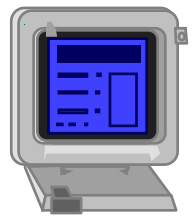
對抗Replay Attack的對策

- 建立通訊Session的觀念。
- 在通訊協定的設計上，採用Challenge-Response的機制，確定Request訊息與Response訊息是成對的關係。
  - 常見的作法是一方在Request訊息中加上nonce（通常是一段亂數）做為Challenge，並要求另一方送回來的Response訊息中也必需包含該nonce，並且簽章。
  - 必要時可以雙方互相Challenge，已達到雙向防止Replay Attack的目的。
- 在通訊訊息中加入time stamp，收方可以檢查訊息時間的合理性（a window of time，例如10分鐘內），有助於防止Replay Attack。
  - 但收送雙方的系統時間誤差不能太大。
- 在通訊訊息中加入sequence number，使每個訊息的sequence number逐次遞增或遞減，收方可以檢查訊息的sequence number是否正確， 有助於防止Replay Attack。
- 以上的機制可以合併使用。

對抗Replay Attack的對策
-- 以Logon為例(1/2)

1. 張三向應用系統發出logon請求

2. 系統產生一個隨機亂數R1，傳給張三當成其Login之Nonce



應用系統
Logon Server

3. 張三以下登入信息作數位簽章，傳送給系統：
【登入時間+隨機亂數R1+張三之識別名稱】+數位簽章

張三

竊聽

王五

王五所竊聽到之張三的登入訊息 +數位簽章

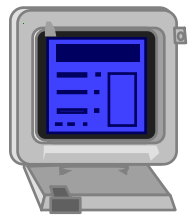
4. 系統檢查隨機亂數R1否符合，且簽章正確，憑證有效，所以登入成功

對抗Replay Attack的對策
-- 以Logon為例(2/2)

王五

1. 王五冒張三之名向應用系統發出logon請求

2. 系統產生一個隨機亂數R2，傳給張三(王五)當成其Login之Nonce



應用系統
Logon Server

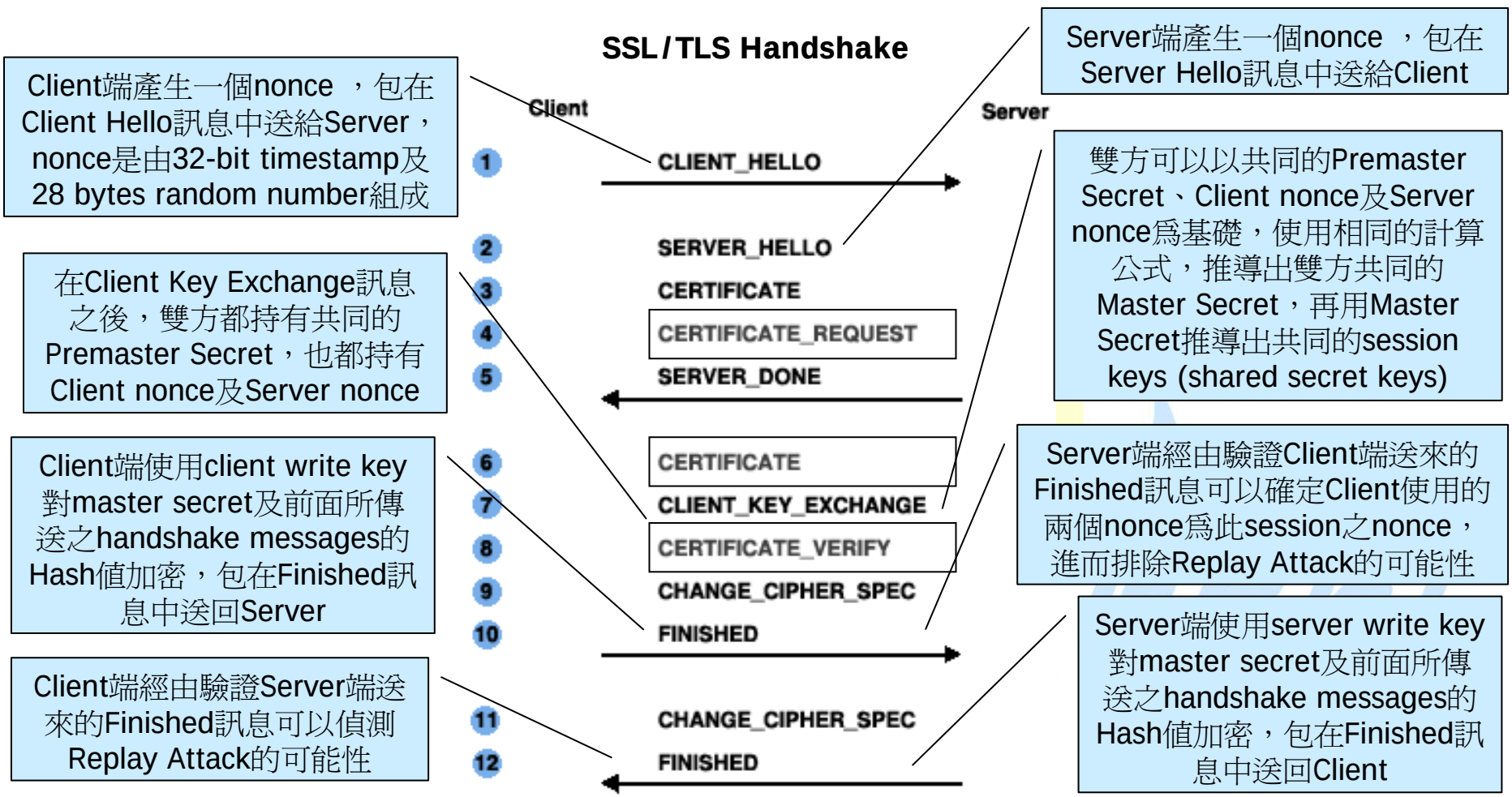
3. 王五重送上次所竊聽到之張三的登入訊息與數位簽章，傳送給系統：
【登入時間+隨機亂數R1+張三之識別名稱(DN)】+數位簽章

4. 雖然簽章正確，但系統檢查會發現以下的異常狀況：
   i. 隨機亂數R1與系統要求的R2不符合
   ii. 登入時間可能和現在系統時間差距過大

所以系統會偵測到Replay而拒絕登入

對抗Replay Attack的對策

-- 以SSL/TLS Handshake為例



框在 [ ] 中的步驟，表示此步驟爲**optional**

對抗Replay Attack的對策
-- 以IKE (ISAKMP/Oakley)為例(1/3)

- Example of Oakley Key Exchange:
  - **I → R:** CKY_I, 0, OK_KEYX, GRP, g^x, EHAO, NIDP, ID_I, ID_R, N_I, $S_{KI}[ID_I \| ID_R \| N_I \| GRP \| g^x \| EHAO]$
  - **R → I:** CKY_R, CKY_I, OK_KEYX, GRP, g^y, EHAS, NIDP, ID_R, ID_I, N_R, N_I, $S_{KR}[ID_R \| ID_I \| N_R \| N_I \| GRP \| g^y \| g^x \| EHAS]$
  - **I → R:** CKY_I, CKY_R, OK_KEYX, GRP, g^x, EHAS, NIDP, ID_I, ID_R, N_R, N_I, $S_{KR}[ID_I \| ID_R \| N_I \| N_R \| GRP \| g^x \| g^y \| EHAS]$

對抗Replay Attack的對策
-- 以IKE (ISAKMP/Oakley)為例(2/3)

- Notation:
  - I = Initiator
  - R = Responder
  - CKY_I, CKY_R = Initiator, Responder Cookies
  - OK_KEYX = Key exchange message type
  - GRP = Name of Diffie-Hellman group for this exchange
  - g^x, g^y = Public key of initiator, responder
  - EHAO, EHAS = Encryption, hash, authentication functions, offered and selected
  - NIDP = Indicates encryption is not used for remainder of this message
  - ID_I, ID_R = Identifier for initiator, responder
  - N_I, N_R = Random nonce supplied by initiator, responder for this exchange
  - $S_{KI}[X]$, $S_{KR}[X]$ = Indicates the signature over X using the private key (signing key) of initiator, responder

對抗Replay Attack的對策
-- 以IKE (ISAKMP/Oakley)為例(3/3)

- Oakley是一個以Diffie-Hellman演算法為基礎的Key Exchange Protocol，但是Oakley加上了以下的改良：
  - It employ a mechanism known as cookie to thwart clogging attack.
  - It enables the two party to negotiate a group; this in essence, specifies the global parameters of the Diffie-Hellman key exchange.
  - It uses nonces to ensure against replay attacks.
  - It enables the exchange of Diffie-Hellman public key value.
  - It authenticates the Diffie-Hellman exchange to thwart man-in-the-middle attacks.

總結

- 不要以為使用了簽章及加解密的技術，就一定能夠確保資料不被竊聽、篡改、或冒名傳送。
- 要對PKI-enabled System/Protocol進行攻擊並不一定要靠破解金鑰或簽章/加密演算法。
- Man-in-the-middle attack及Replay attack是便宜有效的攻擊方法，但PKI-enabled System/Protocol的Designer確很容易忽略這些攻擊。
- 可以靠Authentication及Challenge-Response機制來對抗Man-in-the-middle attack及Replay attack。
- 凡是依賴PKI技術的防禦機制都是建立在「通訊雙方能夠正確地取得對方公鑰」的前提下，所以一定要正確驗證憑證，也就是必需要正確進行Certification Path Processing。System/Protocol的Designer卻很容易忽略正確驗證憑證的重要性。
- Security是由各種不同的防護機制堆疊起來的，這些機制環環相扣，只要任一環節有漏洞，就會讓攻擊者有可乘之機。