

Lecture 11

Data Integrity & Authentication

Jason Lin

Outline

- 資料的完整與認證
- 單向雜湊函數
- 介紹數位簽章並討論其特性
- 公開金鑰數位簽章
- 盲簽章
- 公開金鑰基礎建設
- 數位簽章的仲裁機制

資料的完整與認證

- **資料完整 (Data Integrity):** A variety of mechanisms used to assure the integrity of a data unit or stream of data units.
- **資料認證 (Data Authentication):** Ensure that data received is as sent by an authorized entity.
- **Data Integrity** and **Data Security** are related terms, each playing an important role in the successful achievement of the other.
- **Data Security** refers to the protection of data against unauthorized access or corruption and is necessary to ensure data integrity.
- Data can be compared to a hash value to determine its integrity

單向雜湊函數 (1/2)

- A hash function (雜湊函數) H accepts a variable-length block of data M as input and produces a *fixed-size* hash value

$$h = H(M)$$

- In general terms, the principal object of a hash function is **data integrity**.

A change to any bit or bits in M results, with high probability, in a change to the hash code.

單向雜湊函數 (2/2)

- A hash function H must have the following properties:
 - H can be applied to a block of data of any size.
 - H produces a fixed-length output.
 - $H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
 - For any given code h , it is computationally infeasible to find x such that $h(x) = h$.
 - For any given block x , it is computationally infeasible to find $y \neq x$ with $h(y) = h(x)$.
 - It is computationally infeasible to find any pair (x, y) such that $h(x) = h(y)$.

單向雜湊函數的應用

- It is used in a wide variety of security applications and Internet protocols such as:
 - **Digital Signatures** (We will discuss this application in the rest of the slides)
 - Message Authentication
 - Intrusion detection and Virus detection
 - Construction of pseudorandom number generator
 - ...

數位簽章的需求

- 資料完整性 (Integrity)
 - 不可偽造：能驗證訊息、日期與時間**完整性**
- 身份鑑別性 (Authentication)
 - 不可偽冒：能驗證簽章者**身分**
- 不可否認性 (Non-Repudiation)
 - 不可否認：能藉由第三方來**解決紛爭**



數位簽章法律與應用



全國法規資料庫
Laws & Regulations Database of The Republic of China

最新訊息 法規類別 法規檢索 司法判解 條約協定 兩岸協議 綜合查詢 跨機關檢索

現在位置：首頁 > 法規

法規	
名稱	電子簽章法 英
公布日期	民國 90 年 11 月 14 日

■ 法令規章

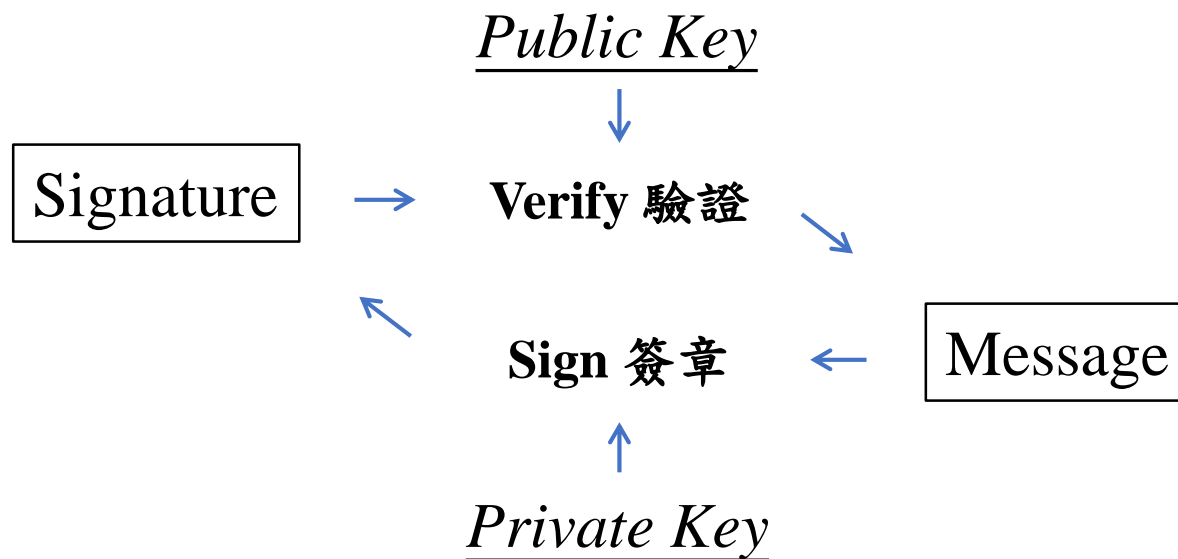
名稱：臺灣證券交易所股份有限公司證券商採用數位簽章注意要點 英
Taiwan Stock Exchange Corporation Directions for the Use of Digital Signatures by Securities Firms

公發布日：民國 91 年 10 月 24 日

修正日期：民國 103 年 12 月 19 日



以公開金鑰為基礎的數位簽章



公開金鑰數位簽章的方法

- RSA 數位簽章
- Rabin 數位簽章
- ElGamal 數位簽章
- Schnorr 數位簽章
- 數位簽章演算法

RSA 數位簽章 (1/2)

Sign

Plaintext:

$$M < n$$

Signature:

$$S = M^d \pmod{n}$$

Verify

Signature:

$$S$$

Verification:

$$M = S^e \pmod{n}$$

RSA 數位簽章 (2/2)

- 金鑰的產生
 - 準備兩個大質數 p 跟 q ，並計算 $n = p \times q$
 - 選擇一個正整數 e ，使得 $\gcd(\phi(n), e) = 1$ ，其中 $1 < e < \phi(n)$
 - 計算 $d \equiv e^{-1} \pmod{\phi(n)}$ ，也就是 $d \times e \equiv 1 \pmod{\phi(n)}$
- 如果簽章者 Alice 欲簽署訊息 M ($M < n$)，她便利用其私密金鑰 d ，對訊息 M 加以簽署而得到數位簽章 S
 - 簽署： $S = M^d \pmod{n}$
- 其餘使用者可以利用公開金鑰 $\{e, n\}$ 來驗證簽署的訊息 M'
 - 驗證： $M' = S^e \pmod{n}$
 - 若 $M' = M$ ，則驗證成功！

RSA Signature Example

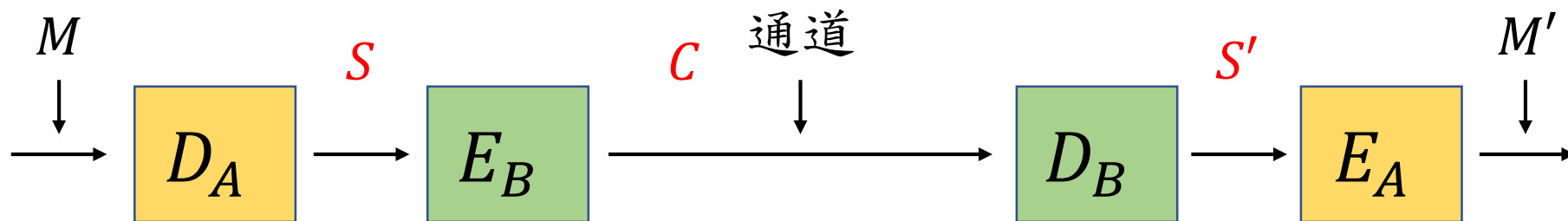
- 任意選擇質數： $p = 17$, $q = 11$
 - 計算 $n = p \times q = 17 \times 11$
 - 計算 $\phi(n) = \phi(p)\phi(q) = (p - 1) \times (q - 1) = 16 \times 10 = 160$
- 任意選擇正整數 $e = 7$
 - $\gcd(e, 160) = 1$
- 計算 $d = 23$
 - $de \equiv 1 \pmod{160}$ 且 $d < 160$
- 保留私密金鑰 $d = 23$, 並公布公開金鑰 $\{e, n\} = \{7, 187\}$

RSA Signature Example (續)

- 若要簽署訊息，例如： $M = 88 = (01011000)_2$
 - 計算 $S = 88^{23} \bmod 187 = 11$ 後，送出 $11 = (00001011)_2$
- 若要驗證數位簽章 $S = 11$
 - 計算 $M' = 11^7 \bmod 187$ ，可得 $M' = 88 = M$

同時達到秘密通訊與數位簽章

- 兩對 RSA 的公開與私密金鑰對： $\{(e_A, n_A), d_A\}$ 、 $\{(e_B, n_B), d_B\}$
- 簽章： $S = D_A(M) = M^{d_A} \pmod{n_A}$
- 加密： $C = E_B(S) = S^{e_B} \pmod{n_B}$
- 解密： $D_B(C) = C^{d_B} = S' \pmod{n_B}$
- 驗證： $E_A(S') = (S')^{e_A} \pmod{n_A} = M' \pmod{n_A}$



減少數位簽章系統之時間與空間複雜度

- 利用單向雜湊函數 H ，將任意區段的明文 M_1, M_2, \dots, M_t 壓縮成一區段的明文 m ，即 $H(M_1, M_2, \dots, M_t) = m$ 。然後簽章者只需對 m 簽章即可，如此即可大量減少簽屬文之儲存空間即計算時間
- 為了避別人偽造明文，單向雜湊函數必須滿足在已知 H 的情況下，很難找到兩個許多區段的明文 M_1, M_2, \dots, M_t 及 M'_1, M'_2, \dots, M'_t ，使得其壓縮值相同，即 $H(M_1, M_2, \dots, M_t) = H(M'_1, M'_2, \dots, M'_t)$

Rabin 數位簽章 (1/3)

- 系統參數：
 - 與 RSA 數位簽章相同，在系統中存在兩個大質數 p 與 q
- 私密金鑰：
 - 簽章者 Alice 以 p 與 q 為其私鑰
- 公開金鑰：
 - 簽章者 Alice 計算出 $n = p \times q$ ，並發布其公鑰為 n

Rabin 數位簽章 (2/3)

- 簽署：

- 對於明文 m ， $0 < m < n$ 。若 $m \in QR_p \cap QR_q$ ，則設定整數 $m' = m$ ；但若 $m \notin QR_p \cap QR_q$ ，則可利用許多方法將 m 映射至 $f(m) = m'$ ，使其值滿足 $m' \in QR_p \cap QR_q$
- Alice 求出對 m 之簽章文 S 為 m' 之平方根

Rabin 數位簽章 (3/3)

- 驗證：
 - 任何人可驗證 $S^2 \bmod n$ 是否等於 m' ；若是，則 S 為 m 的合法數位簽章
- 安全性分析：
 - 本系統之安全性可證明等同於質因數分解，但如同 Rabin 加密系統一樣，存在一種選擇密文攻擊法

ElGamal 數位簽章 (1/3)

- **系統參數：**

- 在系統中存在一大質數 p 以及模 p 之原根 e_1 ，使得解離散對數成為相當困難的問題

- **私密金鑰：**

- 簽章者 Alice 任意選擇一個整數 d ($1 < d < (p - 1)$) 為其私鑰

- **公開金鑰：**

- 簽章者 Alice 計算出 $e_2 = e_1^d \bmod p$ ，並發布其公鑰為 (e_1, e_2, p)

ElGamal 數位簽章 (2/3)

- 簽署：

- 假設簽章者 Alice 欲簽署一個訊息 m
- 步驟一：Alice 先任選一整數 k ，滿足 $\gcd(k, (p - 1)) = 1$
- 步驟二：Alice 計算 $r = e_1^k \bmod p$
- 步驟三：Alice 求出 $s = k^{-1}(m - dr) \bmod (p - 1)$
- 最後，Alice 算出 m 相對應的數位簽章 (r, s)

ElGamal 數位簽章 (3/3)

- **驗證：**

- 假設 Bob 欲驗證 Alice 的簽章是否有效
- Bob 透過明文 m 與數位簽章 (r, s) 檢查 $e_1^m ? \equiv e_2^r r^s \pmod{p}$

Schnorr 數位簽章 (1/3)

- 系統參數：

- 兩個大質數 p 與 q 滿足 $q|(p-1)$ 、 $q \geq 2^{160}$ 及 $p \geq 2^{512}$ ， $g \in Z_p$ 且滿足 $g^q \bmod p = 1$ ，注意 $g \neq 1$
- $H(\cdot)$ 為單向雜湊函數

- 私密金鑰：

- 簽章者 Alice 任意選擇一整數 x 為其私密金鑰

- 公開金鑰：

- 簽章者 Alice 計算其公開金鑰 $y = g^x \bmod p$

Schnorr 數位簽章 (2/3)

- 簽署：

- 假設簽章者 Alice 欲簽署訊息 m ，她會執行下列的步驟
- 步驟一：Alice 任選一整數 k 滿足 $1 < k < q$ ，並算出 $t = g^k \bmod p$
- 步驟二：Alice 計算 $r = H(t, m)$
- 步驟三：Alice 求出 $s = k - xr \bmod q$
- 最後，Alice 得到 m 的數位簽章 (r, s)

Schnorr 數位簽章 (3/3)

- 驗證：

- 假設 Bob 欲驗證 Alice 的簽章是否有效
- 步驟一：Bob 求出 $t' = g^s y^r \bmod p$
- 步驟二：Bob 檢查 $H(t', m) \stackrel{?}{=} r$
- 如果該式滿足，則 (r, s) 為 m 的合法數位簽章

數位簽章演算法

- 美國國家標準技術局（NIST）於 1991 年提出將數位簽章演算法 DSA（Digital Signature Algorithm）作為其數位簽章標準 DSS（Digital Signature Standard）
- DSA 於 1994 年為美國聯邦資訊處理標準 FIPS 186 所採用
- DSA 是 Schnorr 與 ElGamal 數位簽章的變體
- DSA 的橢圓曲線密碼學版本是 ECDSA（Elliptic Curve Digital Signature Algorithm）
- DSA 包含了四個部分：金鑰生成、金鑰分發、簽章、驗證

DSA (1/3)

- 系統參數：
 - p ：512 位元的質數
 - q ：160 位元的質數且滿足 $q|(p-1)$
 - g ：滿足 $g = h^{(p-1)/q} \bmod p$ ，其中 $h \in [2, (p-2)]$ 之任意整數
 - $H(\cdot)$ ：單向雜湊函數
- 私密金鑰：
 - 簽章者 Alice 任意選擇一整數 x 為其私密金鑰
- 公開金鑰：
 - 簽章者 Alice 計算其公開金鑰 $y = g^x \bmod p$

DSA (2/3)

- 簽署：

- 假設欲簽署的明文為 m
- 步驟一：簽章者 Alice 任選一整數 k 滿足 $0 < k < q$
- 步驟二：Alice 計算 $r = g^k \bmod p$
- 步驟三：Alice 求出 $s = k^{-1}(H(m) + xr) \bmod q$ 。注意： $k^{-1}k \bmod q = 1$
- 最後，Alice 得到 m 的數位簽章 (r, s)

DSA (3/3)

- **驗證：**

- 假設 Bob 欲驗證 Alice 的簽章是否有效
- 步驟一：Bob 檢查 r 和 s 是否均屬於 $[0, (q - 1)]$ ，如果不是，則表示 (r, s) 非簽章
- 步驟二：Bob 計算 $t = s^{-1} \bmod q$
- 步驟三：Bob 計算 $r' = [(g^{H(m)t})y^{rt} \bmod p] \bmod q$
- 如果 $r' = r$ ，則 (r, s) 為 m 的合法數位簽章

ECDSA (1/4)

- **系統參數：**
 - CURVE：所有參與者同意所使用的橢圓曲線體（Field）與方程式
 - G ：橢圓曲線的基點（Base Point），用以生成大質數 n 階的子群
 - n ：基點 G 的整數乘冪，使得 $nG = O$ ，其中 O 為單位元素（無窮遠點）
 - $H(\cdot)$ 為單向雜湊函數
- **私密金鑰：**
 - 使用者 Alice 任意選擇一整數 d_A 為其私密金鑰
- **公開金鑰：**
 - 使用者 Alice 透過橢圓曲線的純量乘法計算出公開金鑰 $Q_A = d_A G$

ECDSA (2/4)

- 簽署：

- 假設欲簽署的明文為 m
- 步驟一：簽章者 Alice 計算 $e = H(m)$
- 步驟二：令 z 為 e 的 L_n 個最左邊的位元，其中 L_n 為 n 的位元長度
- 步驟三：Alice 自區間 $[1, n - 1]$ 選擇一個隨機的整數 k
- 步驟四：Alice 計算曲線上的點 $(x_1, y_1) = kG$
- 步驟五：Alice 計算 $r = x_1 \bmod n$ 。若 $r = 0$ ，則返回步驟三執行
- 步驟六：Alice 計算 $s = k^{-1}(z + rd_A) \bmod n$ 。若 $s = 0$ ，則返回步驟三執行
- 最後，Alice 得到 m 的數位簽章 (r, s)

ECDSA (3/4)

- 驗證的前置作業：

- 在驗證簽章以前，可以先檢查簽章者 Alice 公鑰 Q_A 的合法性
- 步驟一：確認 Q_A 是否不等於 O
- 步驟二：確認 Q_A 是否有在曲線上
- 步驟三：確認 $nQ_A = O$

ECDSA (4/4)

- 驗證：

- 假設 Bob 欲驗證 Alice 的簽章是否有效
- 步驟一：Bob 驗證 r 和 s 是否為區間 $[1, n - 1]$ 的整數，若不是，則簽章無效
- 步驟二：Bob 計算 $u_1 = zs^{-1} \bmod n$ 和 $u_2 = rs^{-1} \bmod n$ ，其中 z 的計算與 Alice 相同
- 步驟三：Bob 計算曲線上的點 $(x_1, y_1) = u_1G + u_2Q_A$
- 若 $r \equiv x_1 \pmod{n}$ ，則簽章有效

盲簽章 (1/3)

- 系統參數：

- 簽章者 Alice 任意選擇兩個大質數 p 及 q ，並求出其乘積 $n = p \times q$
- Alice 任意選擇一整數 e 為公開金鑰，使得 e 與 $\phi(n)$ 互質，也就是 $\gcd(e, \phi(n)) = 1$
- Alice 求出其私密金鑰 d 使得 $d \times e \equiv 1 \pmod{\phi(n)}$
- Alice 公佈公開金鑰 (e, n) ，而保留私密金鑰 d

盲簽章 (2/3)

- 簽署：

- 當使用者 Bob 希望簽章者 Alice 幫他簽署文件 m 的有效簽章，又不希望 Alice 知道文件 m 的內容
- 步驟一：使用者 Bob 選擇一個隨機亂數 r ，並計算 $C = r^e m \bmod n$
- 步驟二：使用者 Bob 將 C 送給 Alice
- 步驟三：當 Alice 接收到 C 後，便求出 m 的盲簽章 $s' = C^d \bmod n$
 $= (r^e m)^d \bmod n = r m^d \bmod n$
- 步驟四：接下來，Alice 將 s' 傳送予使用者 Bob
- 步驟五：當 Bob 得到 s' ，便求出 m 的有效簽章 $s = s' r^{-1} \bmod n$
 $= m^d \bmod n$

盲簽章 (3/3)

- 驗證：

- 步驟一：先計算 $m' = s^e \bmod n$
- 步驟二：如果 $m' = m$ ，則 s 為 m 的合法數位簽章


公開金鑰基礎設施

- 公開金鑰基礎設施（Public Key Infrastructure，PKI）用於創建、管理、分發、使用、儲存和撤銷數位憑證（Digital Certificate）
- 數位憑證又稱為公開金鑰認證（Public Key Certificate），是用來證明公開金鑰擁有者的身分
 - 透過第三方可信任的數位憑證認證機構（Certificate Authority，CA）來簽發憑證
 - 數位憑證的內容主要包含了公開金鑰資訊、擁有者的身分資訊、以及憑證發行者 CA 對這份檔案的數位簽章，以確保檔案的內容正確無誤

數位憑證 (1/2)

Document containing the
public key and identity for
Jason Lin

Name: Jason
Surname: Lin
Address: --- St.
.....



Jason Lin's
public key




Certificate Authority's
private key

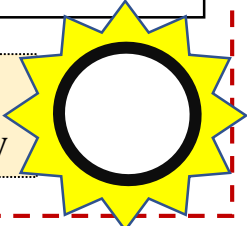


Jason Lin's Certificate

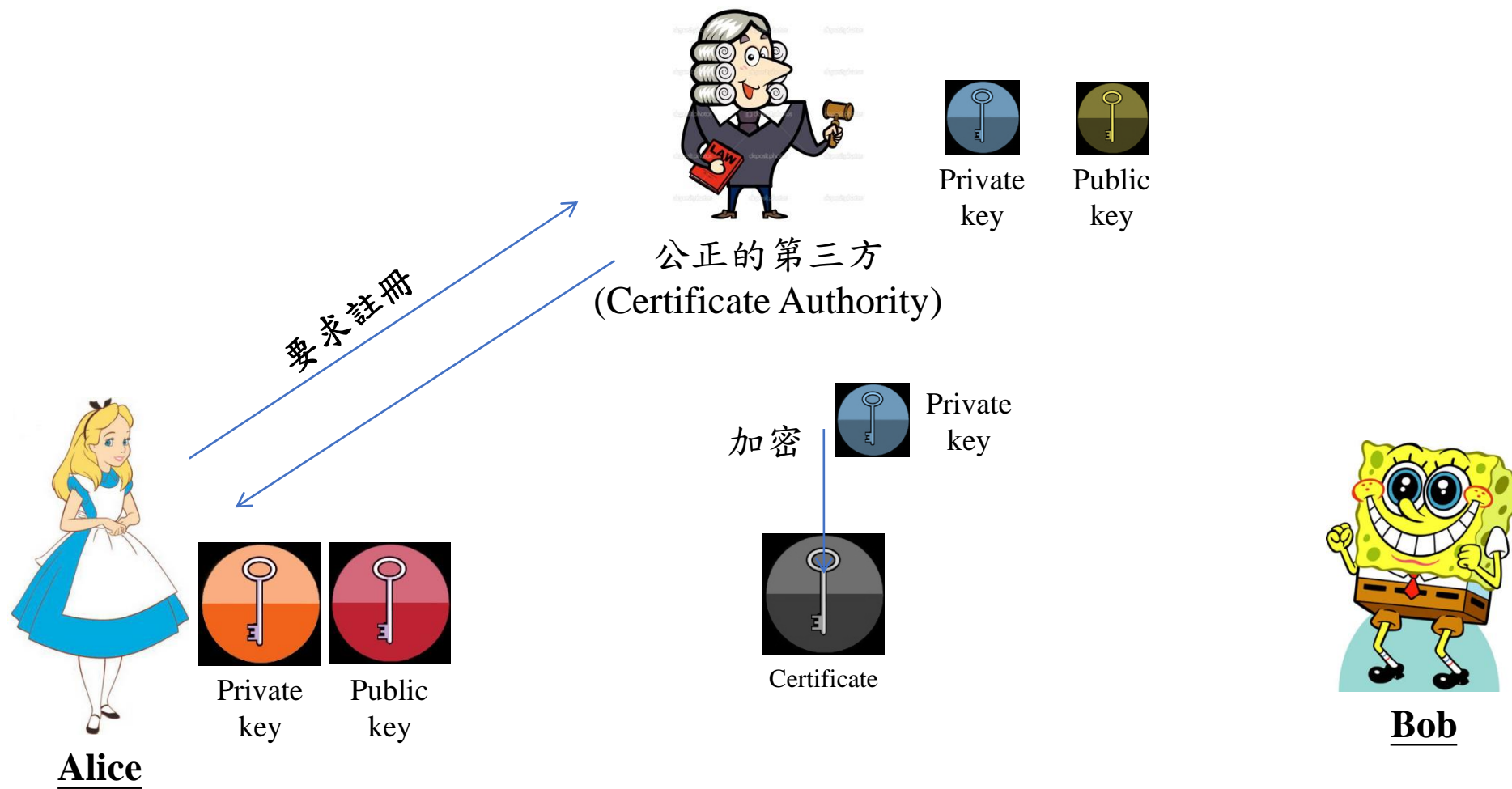
Name: Jason
Surname: Lin
Address: --- St.
.....



Jason Lin's
public key

Signature of the
Certificate Authority 

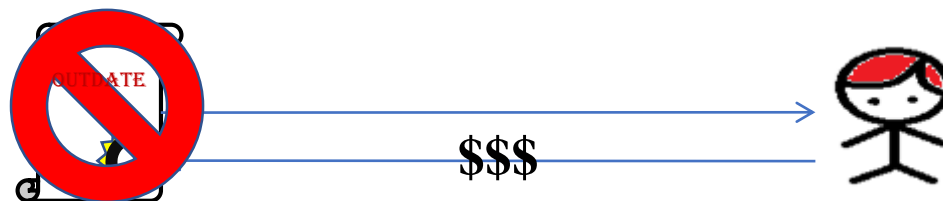
數位憑證 (2/2)



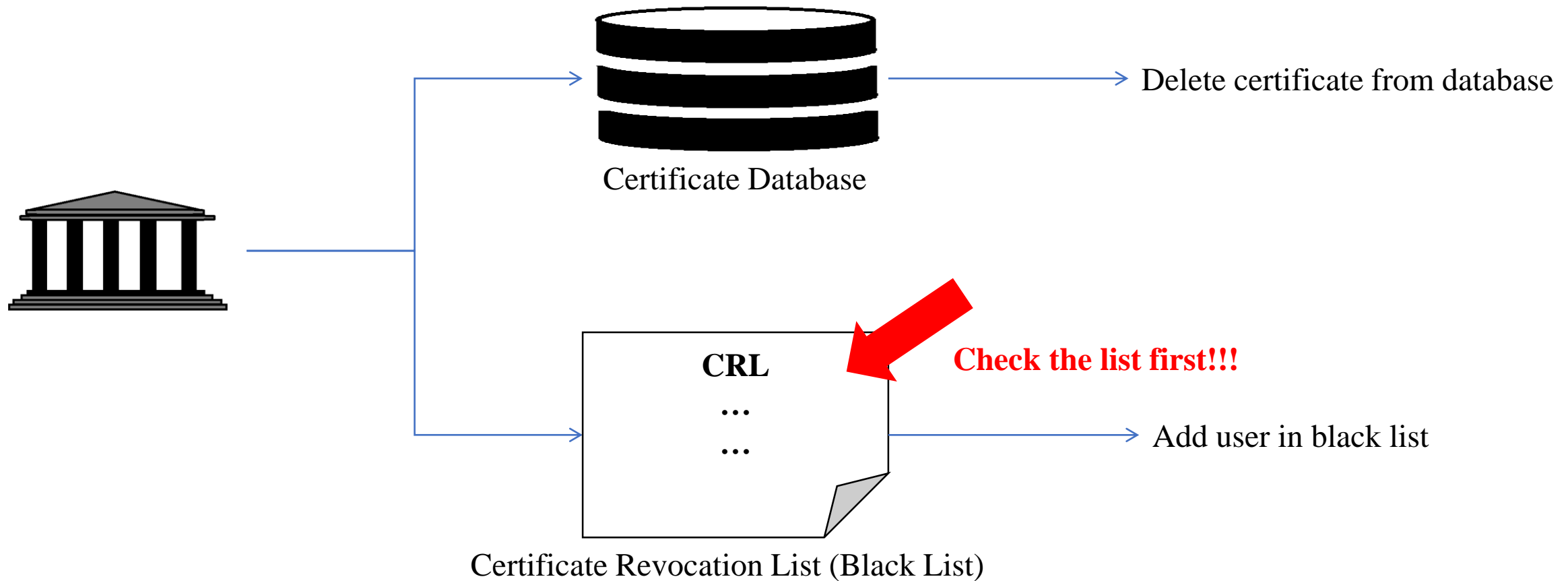
數位憑證的撤銷 (1/2)



1. Time up!!! ...more money
2. Lost private key
3. Blacklist



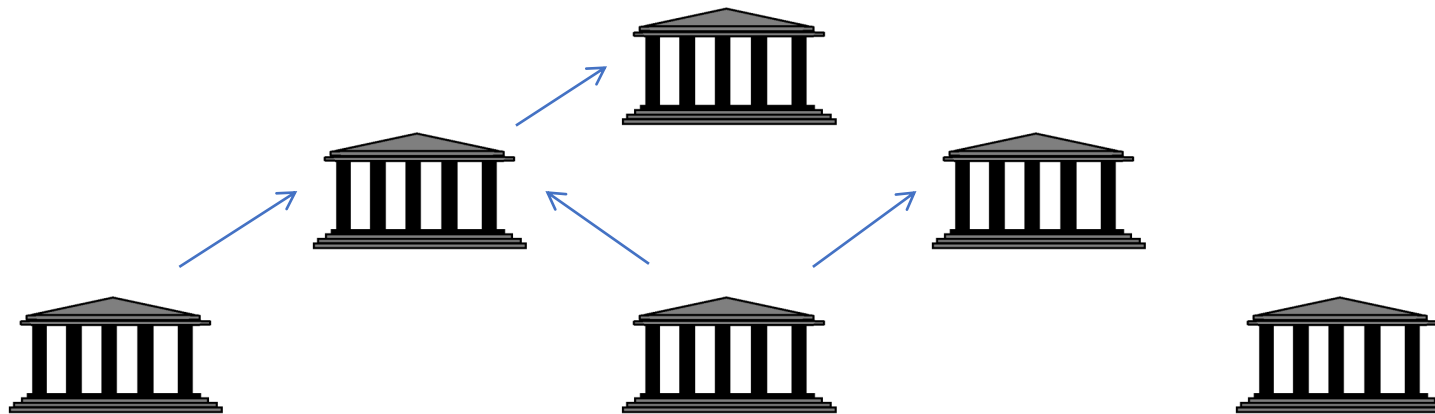
數位憑證的撤銷 (2/2)



PKI 的挑戰與問題 (1/3)

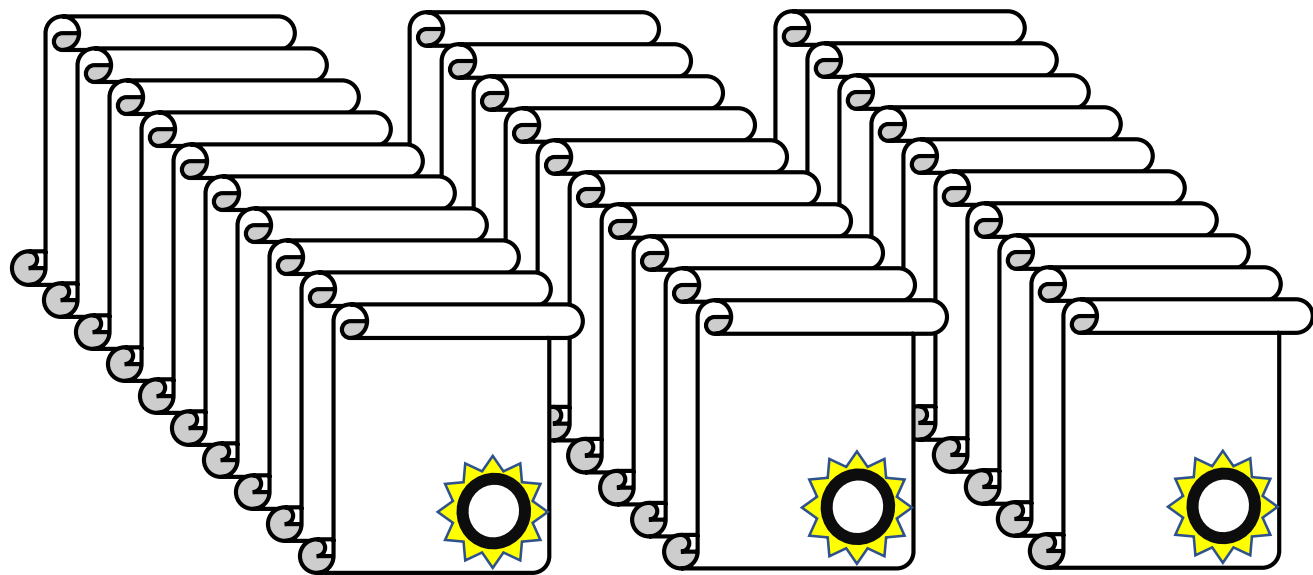
- 信任模型的複雜度

- 公開金鑰認證依賴於 CA 的信任，若 CA 遭到破壞或者發行了不可靠的憑證，整個信任鏈都會受到影響
- 樹狀的信任結構要求每一層級的 CA 都要完全可靠，但這在實際操作中難以完全保證



PKI 的挑戰與問題 (2/3)

- 憑證管理的複雜度
 - 憑證的發放、更新、撤銷以及管理需要大量的資源和時間
 - 使用者或企業需定期更新憑證，以避免因憑證過期而造成的安全隱患



PKI 的挑戰與問題 (3/3)

- 撤銷列表和在線證書狀態協定
 - 當憑證需要撤銷時，必須即時更新撤銷列表 CRL，但是這些列表可能會變得非常大而影響效能
 - 在線證書狀態協定（Online Certificate Status Protocol，OCSP）雖然提供了更即時的撤銷狀態查詢，但仍會引入額外的延遲和網路資源

公開金鑰系統存在的問題

- 計算速度慢
- 計算複雜度高
- 會被量子電腦破解

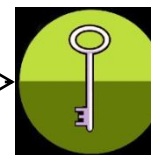
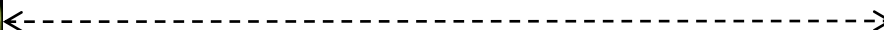
基於對稱式金鑰密碼系統的數位簽章 (1/3)



Alice



Secret key

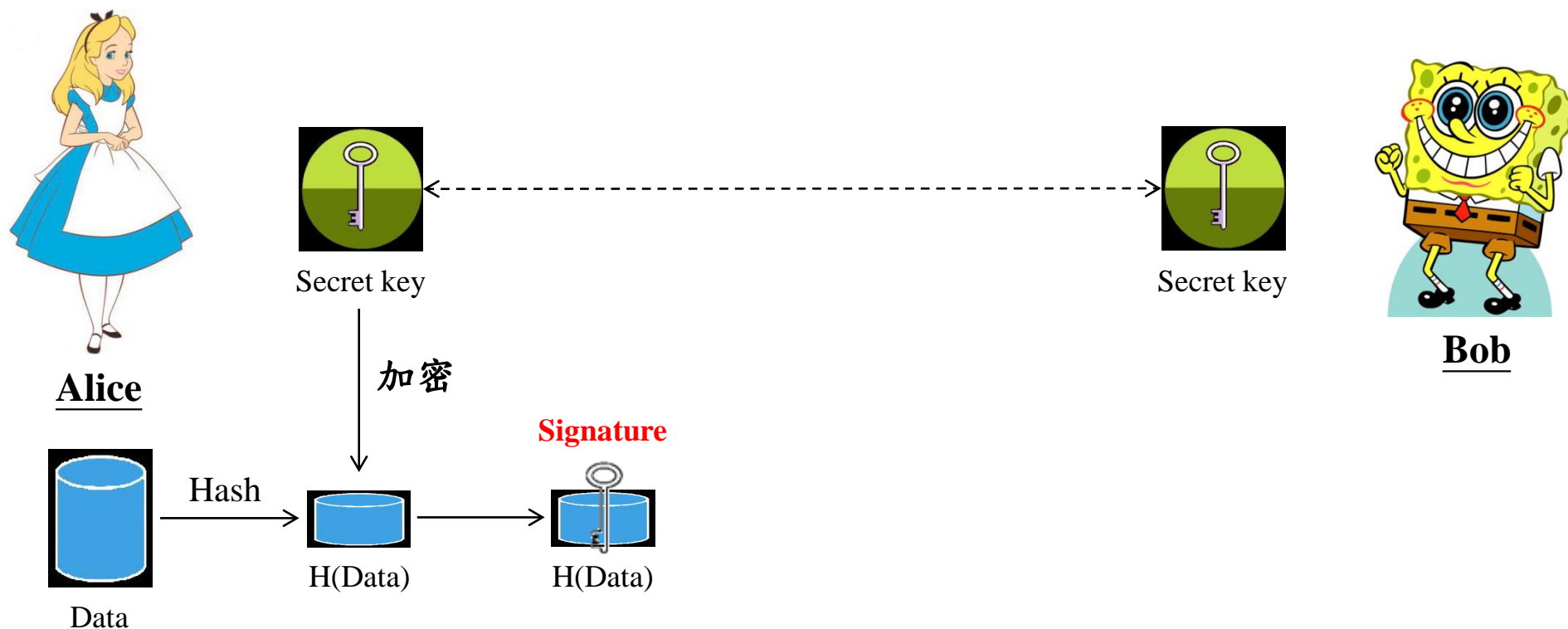


Secret key

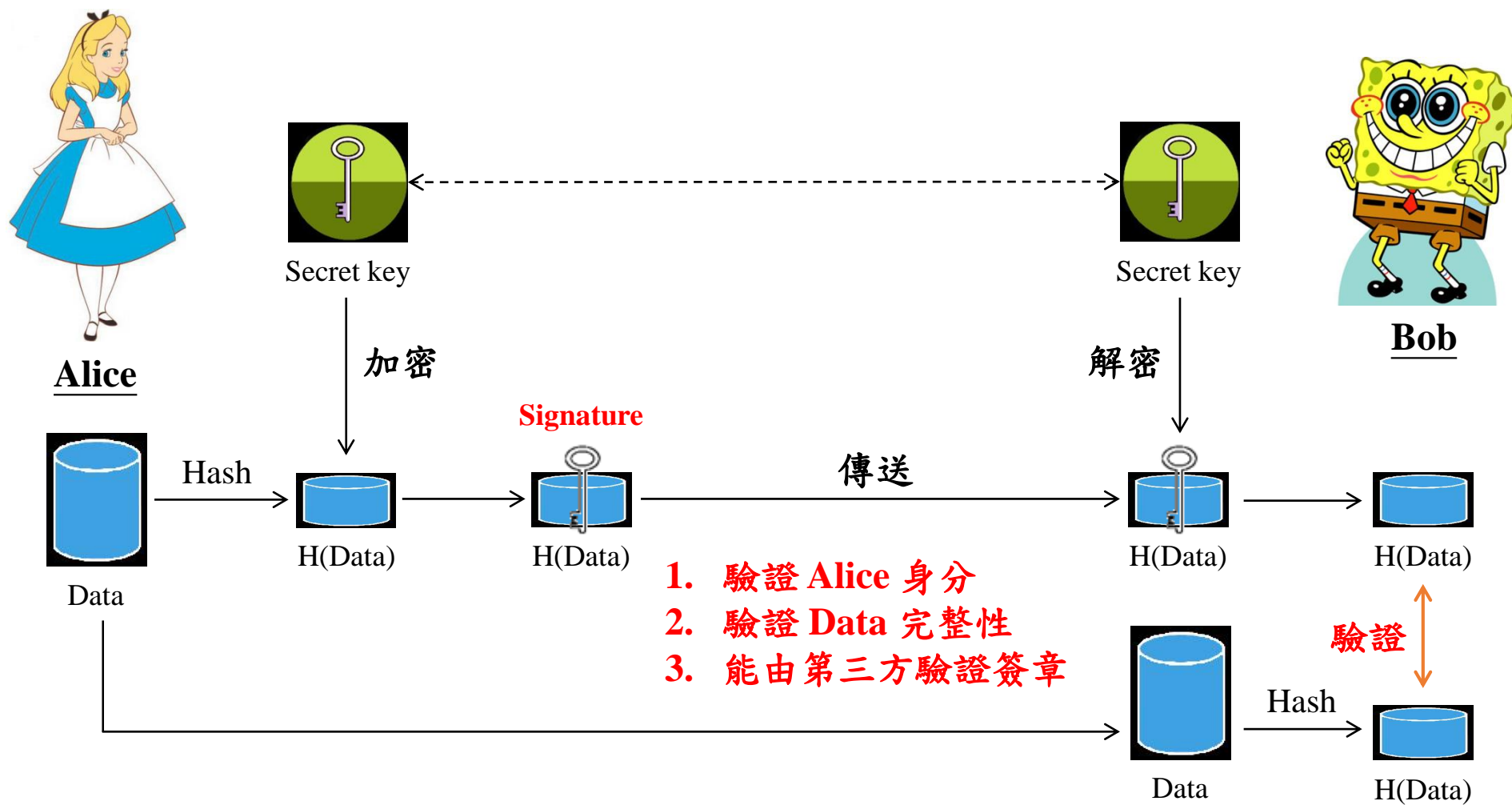


Bob

基於對稱式金鑰密碼系統的數位簽章 (2/3)



基於對稱式金鑰密碼系統的數位簽章 (3/3)



對稱式金鑰密碼系統的問題



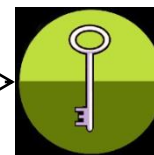
Alice



Secret key



公正的第三方



Secret key



Bob

加密

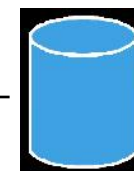
Signature



H(Data)



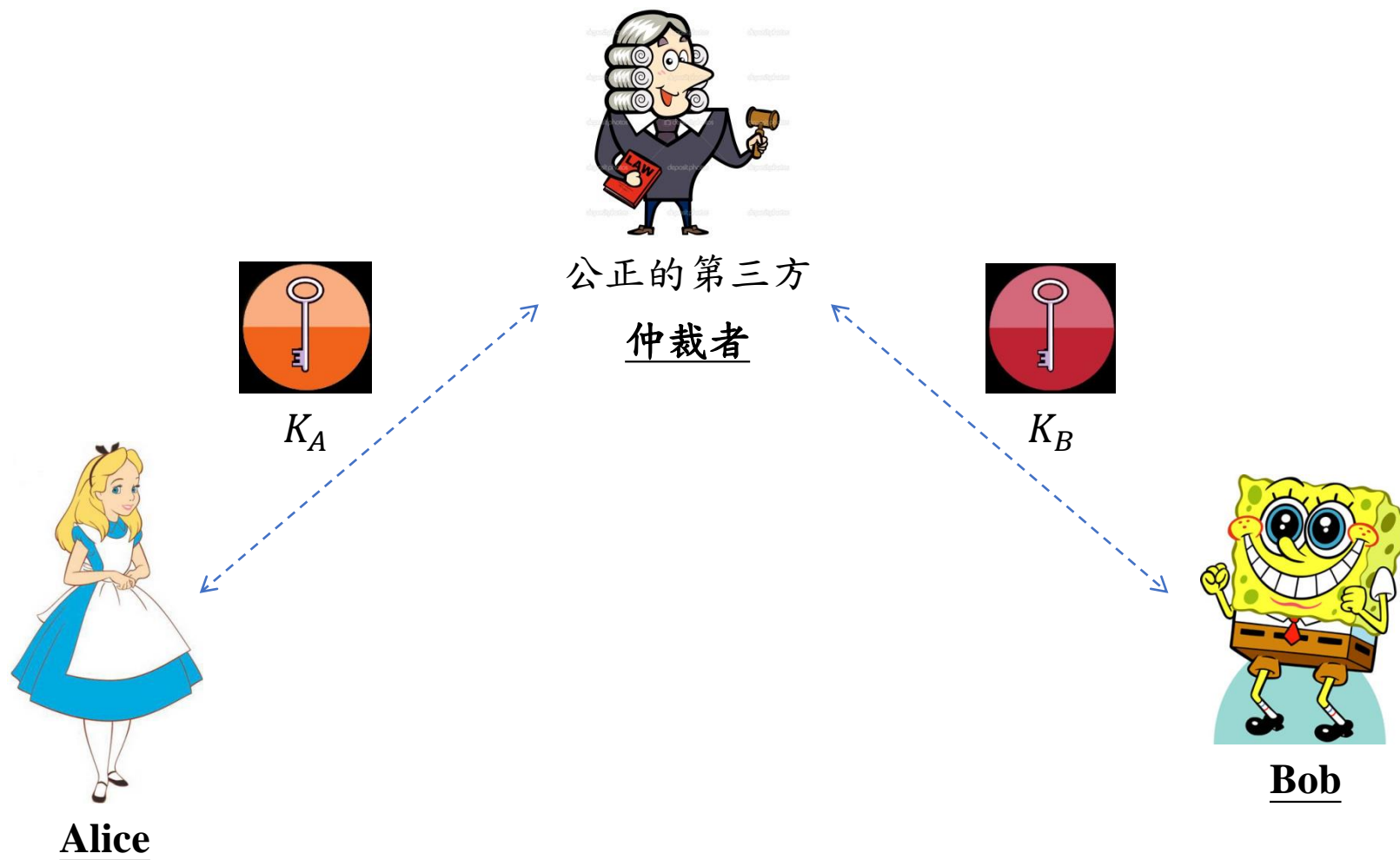
H(Data)



Data

Alice's Signature?
Bob's Signature?

數位簽章的仲裁機制



仲裁式數位簽章 (1/5)



Alice



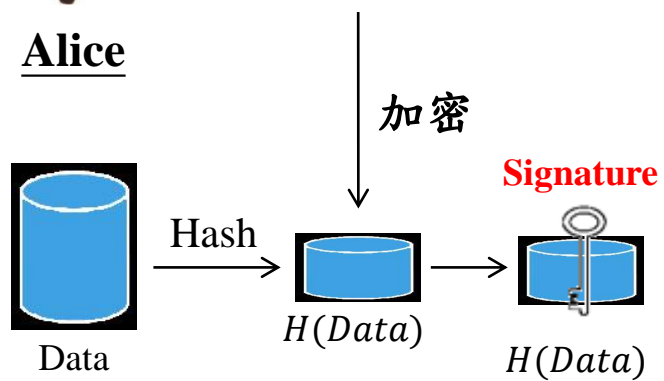
K_A



Bob



仲裁者



仲裁式數位簽章 (2/5)



Alice



K_A

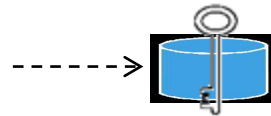


Bob



仲裁者

Signature



$H(Data)$



Data

仲裁式數位簽章 (3/5)



Bob



$H(\text{Data})$



Data



K_A

解密

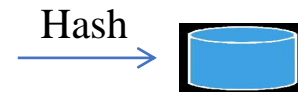


仲裁者



$H(\text{Data})$

驗證



$H(\text{Data})$

Hash

仲裁式數位簽章 (4/5)

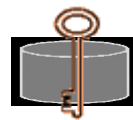


Bob



K_B

加密



$H(S, Data)$

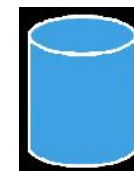


$H(S, Data)$

Hash



$H(Data)$

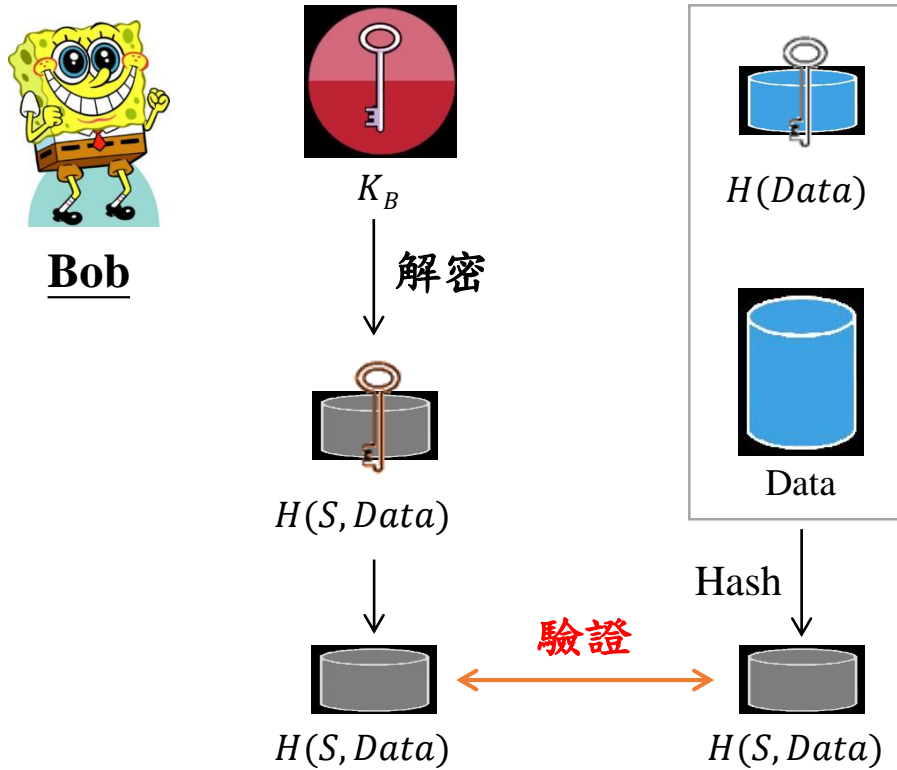


Data



仲裁者

仲裁式數位簽章 (5/5)



1. 驗證 Alice 身分
2. 驗證 Data 完整性
3. 能由第三方驗證簽章