Chapter 3

使用者身分鑑別

3.1 前言

- 使用者身分鑑別主要是要識別某人是否為合法的系統使用者。
- 使用者身分鑑別分為兩部分：使用者身分（Identity）及鑑別（Authentication）。不但要能夠唯一識別使用者身分，而且必須要有方法來預防歹徒冒充別人的身分。

3.2 身分鑑別類型

- 身分鑑別類型，證件驗證（Something You Have）、生物特性驗證（Something You Are）及通行密碼驗證（Something You Known）。

3.2.1 證件驗證

1.條碼卡 (Bar Code)

有一維條碼及二維條碼兩種。

二維條碼可存資料量比一維條碼多出許多。

圖 3.1 (a) 一維條碼與 (b) 二維條碼範例

3.2.1 證件驗證

2. QR Code

優點：

(1) 快速方便，使用者可以輕鬆地使用他們的行動設備生成和掃描 QR Code 以驗證身分。

(2) QR Code 可以透過加密或數位簽章來確保其交易安全，使得這種方法比使用密碼或 PIN 碼更安全。

(3) QR Code 可以嵌入多種類型和大量的數據資料，可應用範圍較廣泛。

3.磁卡

以磁場信號來表示資訊內容。磁卡易受陽光曝曬或外力影響而變形及電磁場之影響，可靠性較低，但其優點為成本較低。

3.2.1 證件驗證

4. IC 卡

是由一個或數個積體電路所組成。

IC 卡與目前所使用磁卡的最大差異，除了記憶容量及安全性高之外，資料還可重寫於 IC 卡。

5.智慧卡

體積小方便攜帶外，少量的記憶體及運算處理的能力。

智慧卡與 IC 卡最大的差別是早期的 IC 卡僅僅是記憶體，只輔助記憶個人機密資訊，智慧卡則如同一部缺少螢幕及鍵盤的迷你型電腦。

3.2.2 生物特性驗證

- 人的生理結構有些具有唯一性，例如每個人的指紋 (Finger Print)、手形 (Hand Shape)、眼紋 (Retina Print) 及臉型等等均不相同。
- 行為差異性主要是人的一些不同行為習慣，例如每個人因為音頻、音律及音量等等不盡相同，因此有各自獨特的聲音 (Voice)。每個人寫字力道、字型及字體等等不盡相同，因此每個人都有獨特的筆跡 (Handwriting)。每個人敲打鍵盤的速度及力道皆有所差異，而且使用滑鼠之習慣亦不盡相同。

https://www.youtube.com/watch?v=VNcArJ5ems8

3.2.2 生物特性驗證

- 生物特徵來驗證使用者身分之設備，評估準則：
  1. **錯誤接受率 (False Accept Rate)**：不合法使用者卻被此設備驗證為是合法使用者之錯誤比率。
  2. **錯誤拒絕率 (False Reject Rate)**：真正合法使用者卻被驗證誤認為是不合法使用者之比率。
  3. **活體驗證功能 (Live & Die Verify)**：此設備是否有能力辨識活的生理結構。
  4. **驗證時間 (Verify Time)**：亦即從使用者登入 (Login) 到系統驗證出結果所需的時間。

3.2.3 通行密碼驗證

- 系統需維護一個儲存所有經授權使用者的 ID 及通行密碼檔案，當使用者登入時，系統便到此檔案核對使用者通行密碼，以確認使用者是否經過授權。

- 如果這個通行密碼檔案未經保護，則使用者通行密碼很容易會被取得及篡改。
  - ✓因此，對此通行密碼檔案 (Password File) 做適當的保護是必要的，常見的保護方法有加密 (如 Unix 系統) 及對此檔案做存取控制等。

3.2.3 通行密碼驗證

- 利用通行密碼來進行使用者驗證雖然簡單，但使用者通常為了方便記憶不會選擇太長的通行密碼，因此通行密碼很容易被猜中。
- 一般使用者身分鑑別系統會同時使用 IC 卡及通行密碼技術。

3.3 身分鑑別流程

- 使用者身分鑑別系統有兩個重要觀念：

1.識別身分 (Personal Identification)

系統必須能夠唯一識別每一位合法使用者。

2.鑑別身分 (Authentication)

系統必須不含糊地驗證使用者所宣稱之身分。

3.3 身分鑑別流程

圖 3.2　使用者身分鑑別簡圖

3.3 身分鑑別流程

- 使用者身分鑑別系統分為註冊、進入系統及識別身分等三階段。

1.註冊階段

先向系統管理員申請帳號，將使用者身分識別碼及通行密碼分配給使用者個人保管，系統並將這些資訊儲存在「主機系統」

2.進入系統階段

每位使用者進入系統時，必須出示身分，並經由「終端設備」輸入通行密碼。

3.識別身分階段

主機系統驗證使用者身分及通行密碼是否為合法使用者。

3.4 通行密碼的安全威脅

1.字典攻擊法

以字典中之單字來測試使用者之通行密碼，一般常見單字有兩萬個，測試一組單字僅需 1 毫秒，因此以字典攻擊法 20 秒內即可得知使用者的通行密碼（ 假設使用者之通行密碼為字典中之單字 ）。

2.猜測攻擊法

以使用者個人相關之資料（ 如生日 ）來猜測使用者之通行密碼，使用者所選擇與個人資料相關之通行密碼，除了生日外，常見的尚有親朋好友姓名、身分證號碼、居住地、電話號碼、紀念日、喜好之事物等等。

3.4 通行密碼的安全威脅

3.窮舉攻擊法或暴力攻擊 (Brute-Force Attack)

將所有可能之通行密碼一一測試，因此若使用者所選之通行密碼過短，很快就會被測出。

4.重送攻擊法

把攔截到的資訊重新輸入到主機系統，還是會通過驗證。

5.行騙法

偽造軟體來行騙 (Spoofing) 使用者，以獲取機密資訊。

3.5 通行密碼管理

- 一般產生通行密碼有以下兩種方式：
  - ✓使用者自選之通行密碼 (User-Generated Password)
  - ✓電腦隨機產生之通行密碼 (Computer-Generated Password)。

3.5 通行密碼管理

- 為了保護通行密碼不被輕易推導出，使用者必須遵循下列的原則來選擇及保管：
  1. 選擇一個不易猜中且長度合理的通行密碼，通常至少要有 8 個字母，合理長度為 8 至 12 個字母。
  2. 避免使用字典中的單字當作通行密碼，並最好在通行密碼中摻雜一些數字及分辨大小寫。
  3. 勿將通行密碼寫在筆記本及其他任何地方。
  4. 避免選擇單字、個人相關特性資料、以及鍵盤排列。
  5. 避免多台主機系統共用相同通行密碼。

3.5 通行密碼管理

- 使用通行密碼時需注意下列事項：
  1.切勿將通行密碼交給他人使用。
  2.每次登入時，先檢查系統相關訊息。
  3.嚴禁與其他使用者共用同一通行密碼。
  4.當你離開或暫時離開終端機時，一定要登出系統。

3.5 通行密碼管理

- 通行密碼有其生命週期 (Life Time)。為了安全起見，使用者應定期更改其通行密碼：

  1.時常改變你的通行密碼，每兩個月至三個月改變一次。

  2.如有任何的理由懷疑通行密碼已被他人知道，應該立刻進行更改。

  3.因臨時性任務申請之通行密碼，一旦任務結束後，使用者帳號應予以刪除。

  4.離職員工之帳號應予以刪除或停止使用。

3.6 各種通行密碼技術

- 設計通行密碼系統之一般需求：
  1.系統要訂定錯誤次數（ 如三次 ）。
  2.若同時有兩個裝置以相同使用者身分登入時，必須予以禁止或警告。
  3.系統要有強迫使用者定期更改通行密碼的功能，並檢示通行密碼是否合理。
  4.使用者輸入完所有資料（ 使用者識別名稱及通行密碼 ）後，才開始驗證其身分。
  5.需顯示上次登入之日期時間。
  6.所有登入之動作均需寫入紀錄 (Log) ，以做為日後稽核用。

3.6.1 直接儲存法

表 3.1　直接儲存通行密碼表

使用者識別名稱 (ID_i)	通行密碼 (PW_i)
黃品潔	is231765
林逸喬	Insecde
⋮	⋮
邱瓊儀	Unno4321

3.6.1 直接儲存法

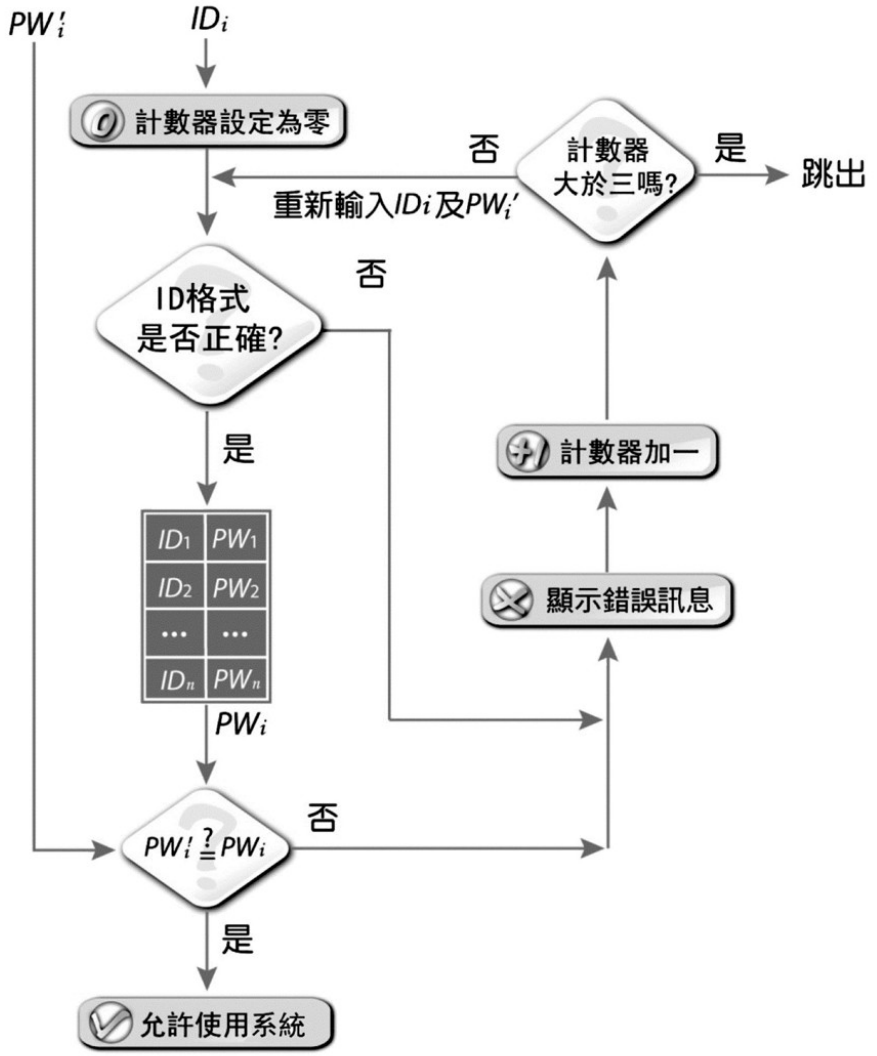

圖 3.3　以直接儲存通行密碼法之使用者身分鑑別流程圖

3.6.1 直接儲存法

- **步驟一**：計數器先設定為零。
- **步驟二**：判斷使用者輸入之識別名稱格式是否正確。
- **步驟三**：系統從通行密碼表找出識別名稱之相對通行密碼 PW_i。
- **步驟四**：將此 PW_i 與使用者輸入之通行密碼 PW_i' 做比對。
- **步驟五**：在步驟二及步驟四中，若判定為不合法使用者，將計數器累增一次。
- **步驟六**：判斷計數器是否已超過三次。

3.6.2 單向函數法

表 3.2　單向函數通行密碼表

使用者識別名稱 (ID_i)	單向函數通行密碼 $Y_i = F(PW_i)$
黃品潔	F (is231765)
林逸喬	F (Insecde)
⋮	⋮
邱瓊儀	F (unno4321)

3.6.2 單向函數法

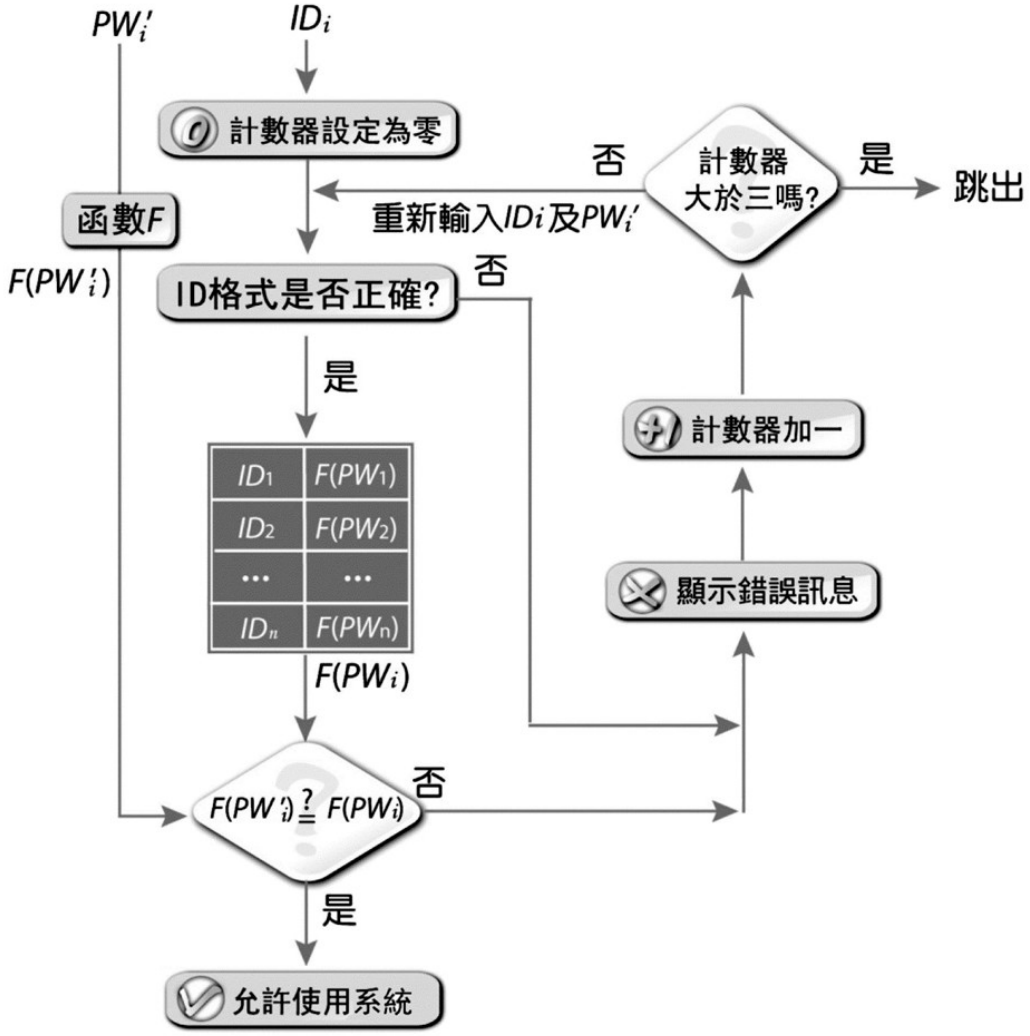

圖 3.4　以單向函數法之使用者身分鑑別流程圖

3.6.3 通行密碼加密法

表 3.3　通行密碼加密法對照表

使用者識別名稱 (ID_i)	單向函數通行密碼 $C_i(=E_k(PW_i))$
黃品潔	E_k (is231765)
林逸喬	E_k (Insecde)
⋮	⋮
邱瓊儀	E_k (unno4321)

3.6.3 通行密碼加密法

- 現行 UNIX 系統即是採用通行密碼加密法，其中密碼系統採用 DES 加密法。
- 首先遠端登入該主機 (telnet mis.nchu.edu.tw) ，再輸入使用者名稱 (設 s9414613) 與密碼，成功登入後，再鍵入

  cat /etc/passwd} [Enter]

  s9414613:x:2374:503::/home/s9414613:/bin/bash
- 其中， x 代表使用者 s9414613 之通行密碼，此一通行密碼已經過加密處理。

3.6.3 通行密碼加密法

- 使用者通行密碼應定期或不定期更新。在 UNIX 系統更改通行密碼。

  > passwd

  Enter login password: sd123456 [Enter]

  New password: h1w3ankg [Enter]

  Re-enter new password: h1w3ankg [Enter]

3.7 多因子身分鑑別機制

- 先前所提到的身分鑑別機制分別基於證件驗證（Something You Have)、生物特性驗證（Something You Are）及通行密碼驗證（Something You Known）等單一因子來鑑別使用者的身分。
- 對於單因子身分鑑別機制的安全性開始有所疑慮。有鑒於此，許多系統開始導入雙因子身分鑑別（Two-factor Authentication ，縮寫即 2FA）或多因子身分鑑別（Multi-factor Authentication ，縮寫即 MFA)。
- 雙因子或多因子身分鑑別是使用者登入時除了以帳號和密碼作為身分鑑別的方式(Something You Know）外，還會搭配另外一種身分鑑別的方式來進行登入。

3.8 Kerberos 身分鑑別系統

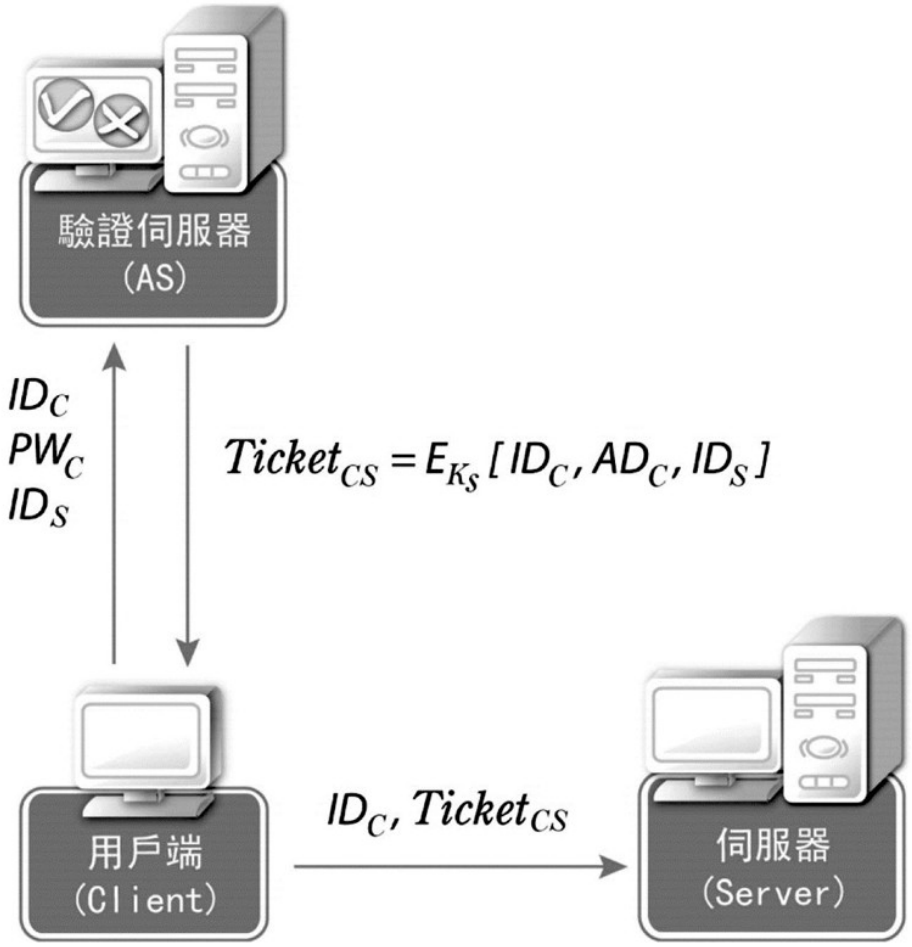

圖 3.5　Kerberos 使用者身分鑑別系統

取得票證授權票證

步驟 1：證明身份

用戶端將其 ID 和用密碼雜湊加密的時間戳發送給 KDC 的驗證伺服器 (AS)。密碼本身絕不傳送。

步驟 2：KDC 驗證

AS 檢查使用者身份，並創建一個秘密的「會話金鑰」和一張特殊票證。

步驟 3：接收 TGT

AS 回傳用 KDC 主密鑰加密的票證授權票證 (TGT)，以及用用戶端密碼雜湊加密的會話金鑰副本。

取得服務票證

步驟 1：請求存取

用戶端告知 KDC 的票證授權伺服器 (TGS) 它想存取的服務。

步驟 2：出示憑證

用戶端發送其 TGT（主識別證）和一個稱為「驗證器」的新加密時間戳。

步驟 3：TGS 驗證

TGS 驗證 TGT 和驗證器，確認請求合法且非重播攻擊。

步驟 4：接收服務票證

TGS 核發一張用*服務*的秘密金鑰加密的服務票證，以及供用戶端和伺服器使用的新會話金鑰。

3.8 Kerberos 身分鑑別系統

- 簡單的 Kerberos 使用者身分鑑別系統有兩個缺點：
  1. 用戶端傳送給 AS 之通行密碼是以明文（ 未經加密處理 ）方式傳送，若遭截取其安全堪慮。
  2. 通行票僅能使用在一伺服器，當要到不同的伺服器（ 如印表機伺服器、 Mail 伺服器 ）要求提供服務時，則需重新向 AS 申請新的通行票。

3.8 Kerberos 身分鑑別系統

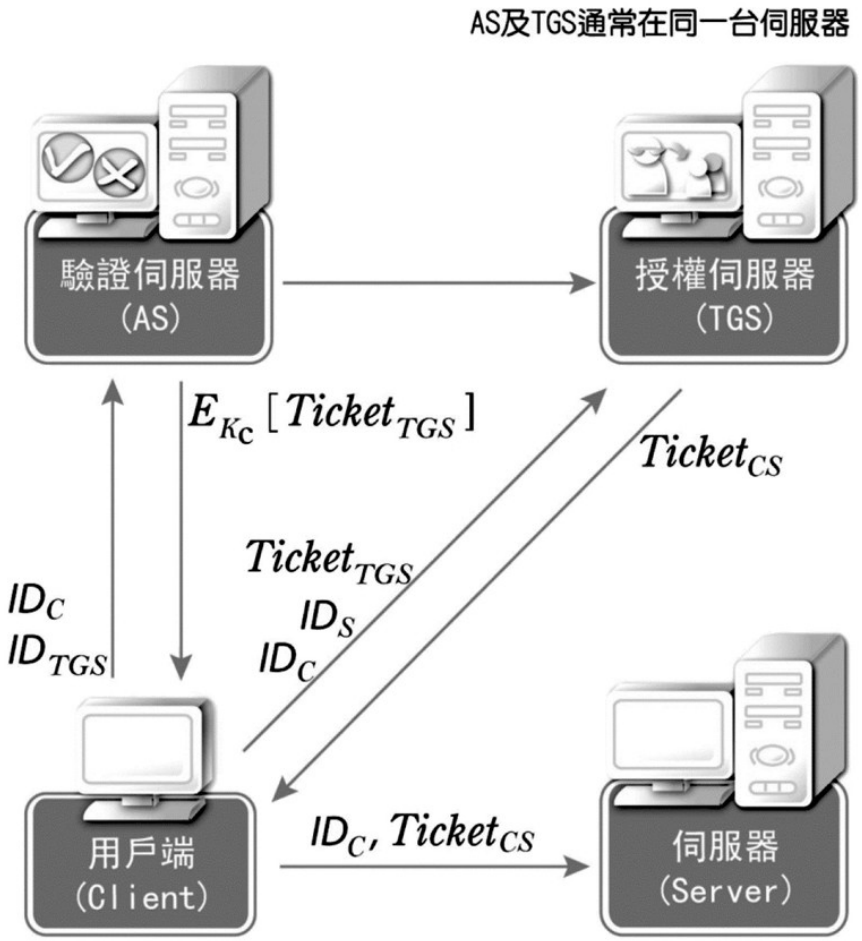

圖 3.6　更安全的 Kerberos 使用者身分鑑別系統

3.9 FIDO 身分鑑別機制

- FIDO 的作法是採用公鑰密碼學技術來取代傳統需要傳輸或儲存用戶的帳號密碼的作法，可有效解決密碼被盜、暴力破解、猜測攻擊或被用戶遺忘等問題。

3.9.1 FIDO 註冊流程

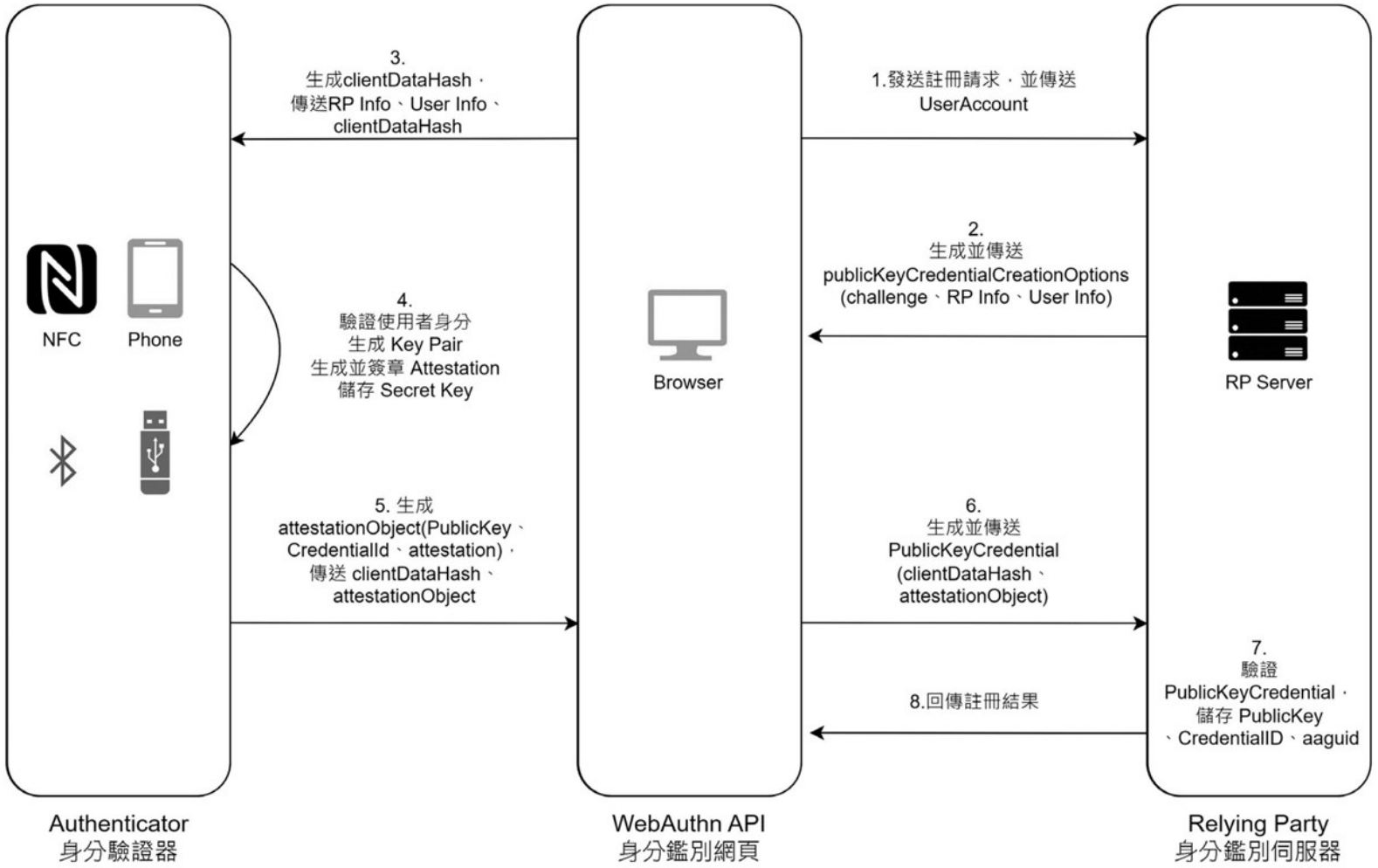

圖 3.7　FIDO 註冊流程

3.9.1 FIDO 註冊流程

- 註冊流程過程描述如下。

1.註冊請求

2.生成 PublicKeyCredentialCreationOptions

3.生成 clientData

4.驗證使用者身分

5.回傳 attestationObjcet 和 clientData 至 Browser

6.解析 PublicKeyCredential

7.驗證

3.9.1 FIDO 註冊流程

表 3.4　PublicKeyCredentialCreationOptions 資料內容

challenge	由 RP Server 產生的隨機亂數值。
RP Info	RP ID：是目前網站的 domain。 RP Name：RP 的名稱。
User Info	id：由 RP 生成的使用者識別碼。 name：使用者名稱。(例如：Mars@nchu.edu.tw) displayName：使用者的暱稱。(例如：Mars)
pubKeyCredParams	RP Server 可以支援的公開金鑰演算法。
timeout	等待 RP Server 回應的時間。
attestation	有三種選項：none、indirect 與 direct 來決定要跟使用者驗證器取得多少資訊。 none：不需要驗證驗證器是否為合法的，不回傳任何 attestation data。 Indirect：只拿匿名的 attestation data。 direct：回傳所有 attestation data。

3.9.1 FIDO 註冊流程

表 3.5　clientData 資料內容

challenge	RP Server 透過 Browser 傳遞過來的 challenge。
origin	RP Server 必須驗證此 origin 字串是否與應用程式的 origin 相符。
type	RP Server 會驗證此字串實際上是否符合「webauthn.create」。
tokenBinding	確保正在通訊的 RP Server 能夠識別它與的是正確的 Browser 進行通訊。

3.9.1 FIDO 註冊流程

表 3.6　attestationObject 資料內容

fmt	指示 RP Sever 應該如何去解析和驗證 attestation Data。
authData	包含有關註冊事件的 metaData，以及用於身分驗證的公鑰，資料內容如表 3.7。
attStmt	驗證公鑰是不是來自預期的驗證器。

3.9.1 FIDO 註冊流程

表 3.7　authData 資料內容

aaguid	Authenticator 的 GUID，可以用於驗證 Authenticator 的真實性。
rpIdHash	是 Credential 裡 RP ID 的 SHA-256 Hash。
flags	AT：驗證器是否已添加 attested credential data。 ED：驗證器資料是否具有擴展。 UP：User 是否存在。 UV：User 是否已驗證（使用 PIN 或生物識別等)。
credentialID	由認證器生成的，用於唯一標識特定的 Credential。
counter	偵測複製的驗證器。防止攻擊者使用複製的驗證器進行身分驗證。
credentialPublicKey	以 COSE_Key 格式編碼的憑證公鑰。 1：是密鑰類型。值為 2 是 EC2 類型。 3：是簽章算法。值為 –7 是 ES256 簽章算法。 –1：是曲線類型。值為 1 是 P-256 曲線。 –2：公鑰的 x 座標。 –3：公鑰的 y 座標。

3.9.1 FIDO 註冊流程

表 3.8　PublicKeyCredential 資料內容

id	Credential id。
response	attestationObject：包含 authenticator data 和 attestation statement。 clientData：客戶端傳給驗證器，用於生成 Credential 的資料。
type	類型為公鑰。

3.9.2 FIDO 登入驗證流程

1.登入請求

2.回傳 PublicKeyCredentialRequestOptions

3.驗證 origin

4.驗證使用者的身分

5.將簽章後的資訊回傳給 Browser

6.解析 PublicKeyCredential

7.解析和驗證

3.9.2 FIDO 登入驗證流程

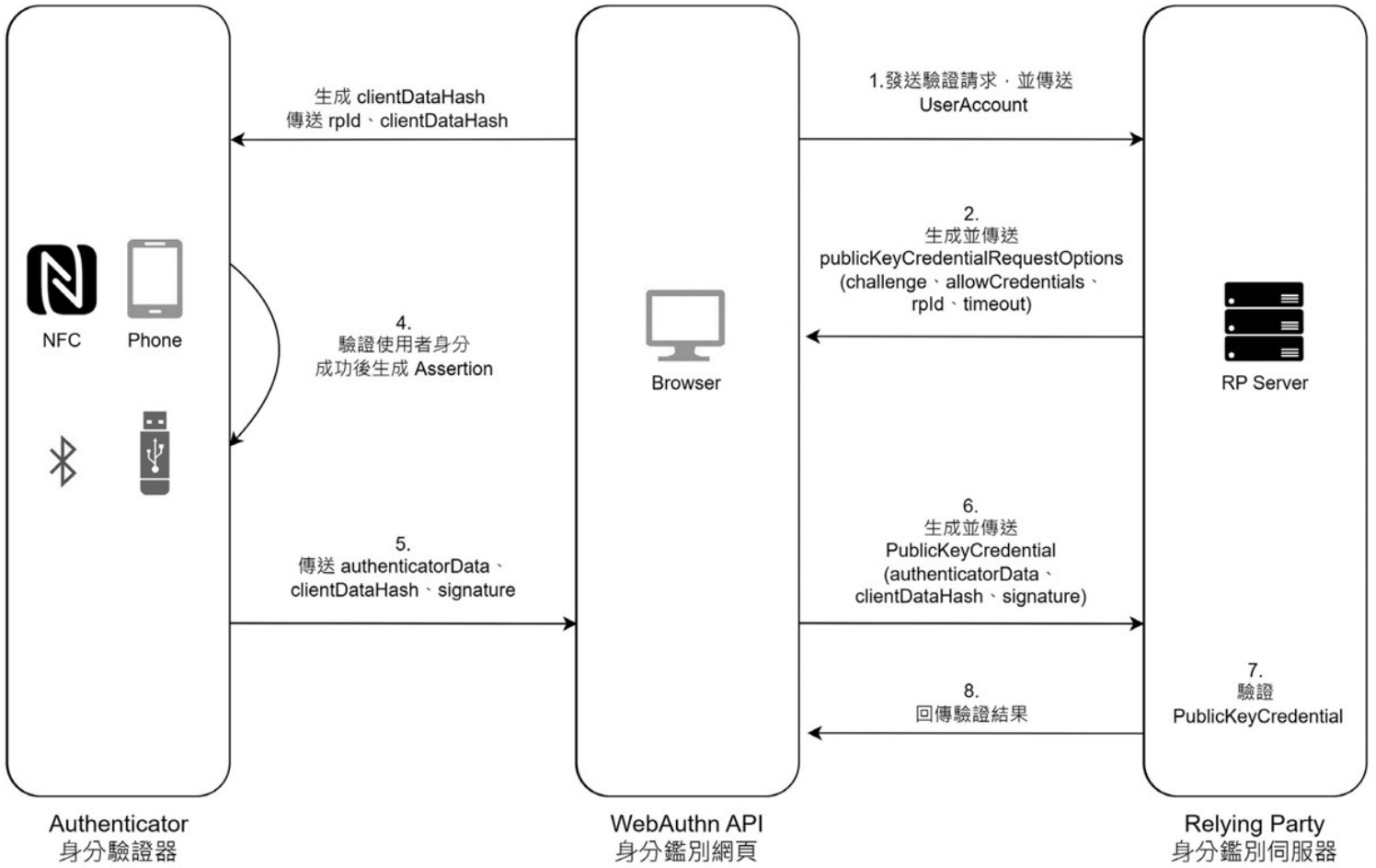

圖 3.8　FIDO 登入驗證流程

3.9.2 FIDO 登入驗證流程

表 3.9　PublicKeyCredentialRequestOptions 資料內容

challenge	RP Server 隨機產生的亂數值。
allowCredentials	RP Server 認可的公鑰憑證列表。
rpid	需要和註冊時的 rpid 一致，確保 Client 與正確的 RP Server 進行身分鑑別。
timeout	等待 RP Server 回應的時間。

3.9.2 FIDO 登入驗證流程

表 3.10　authenticatorData 資料內容

RpIdHash	是 Credential 裡 RP ID 的 SHA-256 Hash。
flags	AT：驗證器是否已添加 attested credential data。 ED：驗證器資料是否具有擴展。 UP：User 是否存在。 UV：User 是否已驗證（ 使用 PIN 或生物識別等 ）。
counter	偵測複製的驗證器。防止攻擊者使用複製的驗證器進行身分驗證。
extensionsData	返回零個或多個擴展的值。

3.9.3 FIDO 機制的安全評估

- FIDO 是採用公開金鑰架構的驗證模式，在 FIDO 認證伺服器端只保存相對應的公鑰，而私鑰則僅保存在使用者裝置端的驗證器中。
- FIDO 也支持多因子身分鑑別，提高身分鑑別的安全性，並可以在任何網站或應用程序上使用同一個身分鑑別方式。

3.9.3 FIDO 機制的安全評估

- 總結， FIDO 機制有以下幾項優點，也使得 FIDO 成為現今身分鑑別的主流技術之一。

1.完善的無密碼登入驗證標準。

2. Server 只保存公鑰 / 驗證，不會上傳使用者的個資。

3.採用公開金鑰加密的技術，可以防止密碼洩漏及釣魚攻擊。

4.可以在不同的設備上使用同一個身分鑑別的方法。

5.可支援 Multi-server 的架構，降低金鑰管理的負擔。

3.10 實務操作

- **練習一**：請透過下列網址進入該網頁，輸入欲測試的密碼組合，來檢視使用一部主機來破解該密碼需要的時間。

  請連結： https://www.security.org/how-secure-is-my-password/

3.10 實務操作

圖 3.9　使用者身分鑑別：練習一