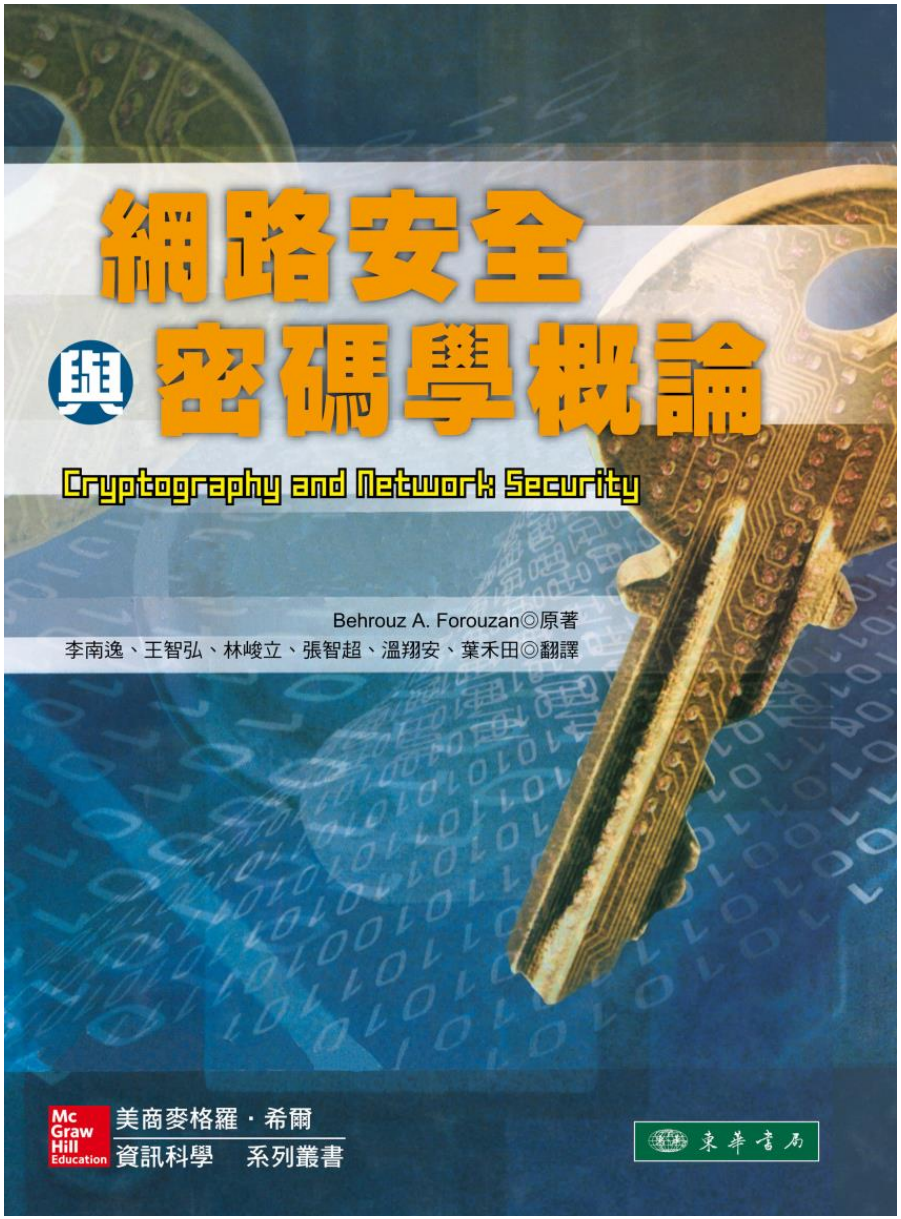


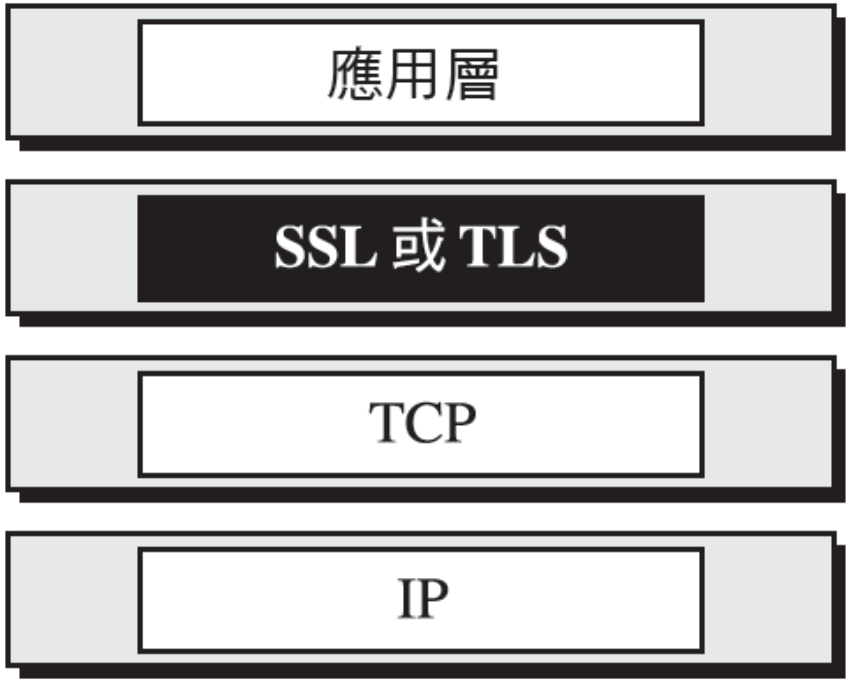
Chapter 17

傳輸層安全：SSL 與 TLS

學習目標

- 討論在網際網路模型中傳輸層的安全服務需求。
- 討論 SSL 的一般結構。
- 討論 TLS 的一般結構。
- 比較與對照 SSL 和 TLS。

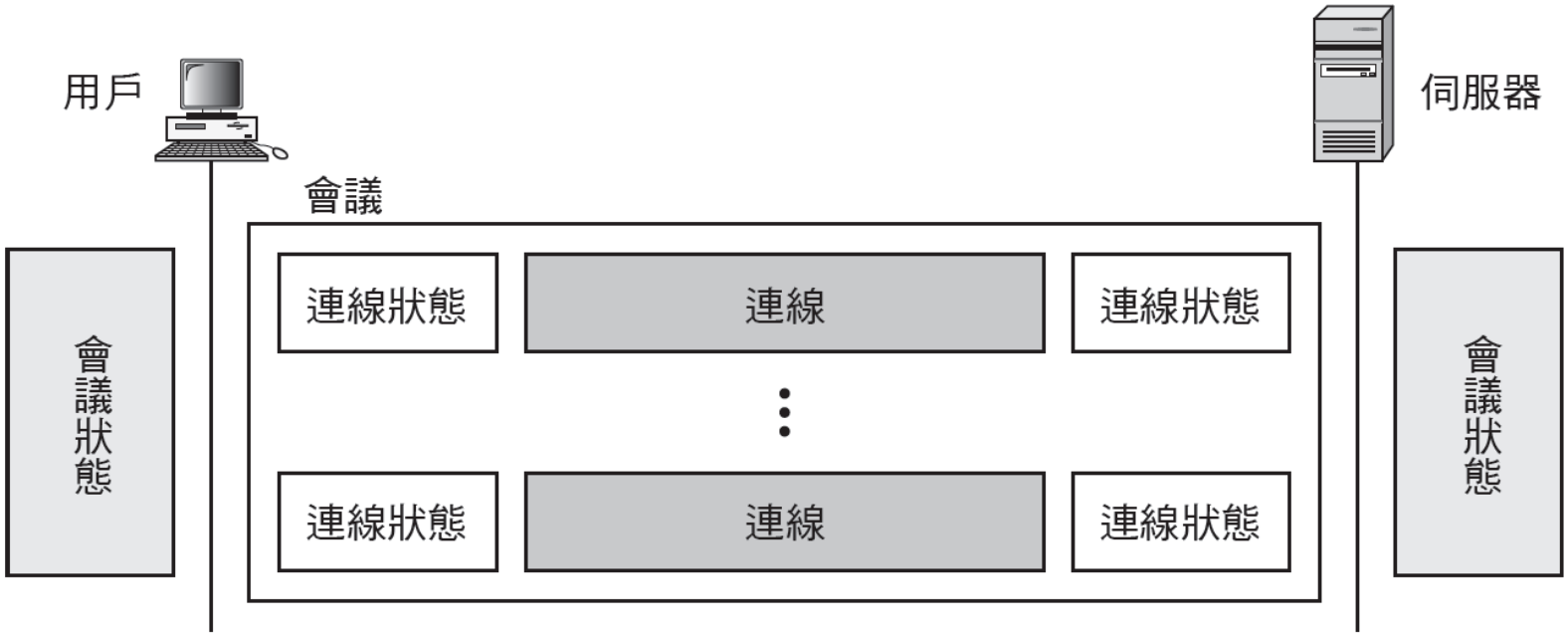
圖 17.1　SSL 和 TLS 在網際網路模型中的位置



17.1 SSL結構

- SSL 是設計對來自於應用層產生的資料提供安全和壓縮的服務。
- 本節討論主題
  - 服務
  - 金鑰交換演算法
  - 加密／解密演算法
  - 雜湊演算法
  - 加密套件
  - 壓縮演算法
  - 密碼參數的產生
  - 會議與連線

圖 17.11 一個會議和多個連線



17.2 四個協定

- 我們已經討論過 SSL 的想法，而沒有說明SSL如何完成它的工作。如圖17.12 所示，SSL在兩層中定義四個協定。
- 本節討論主題
  - 握手協定
  - 密文變更協定
  - 警示協定
  - 記錄協定

圖 17.12 四個 SSL 協定

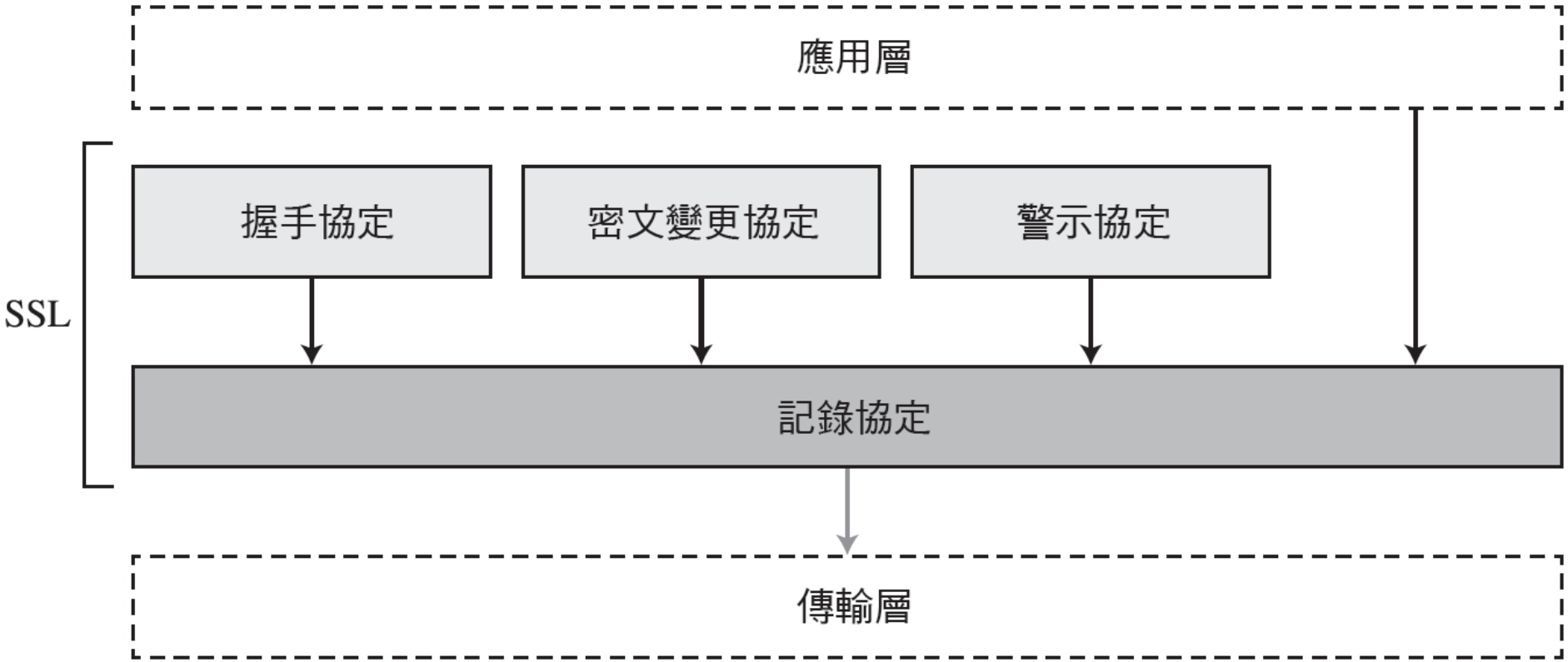
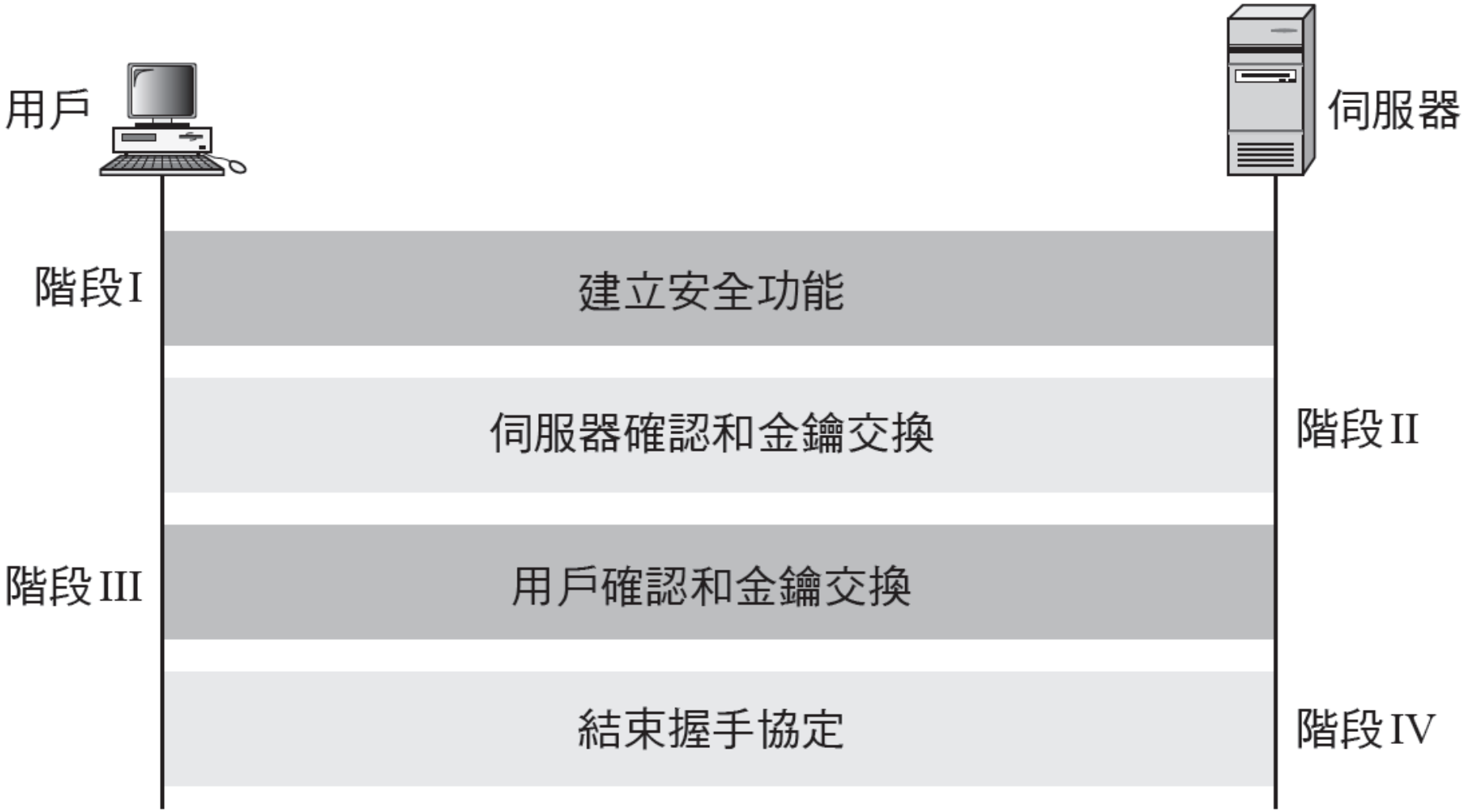


圖 17.13 握手協定



Handshake Protocol Action

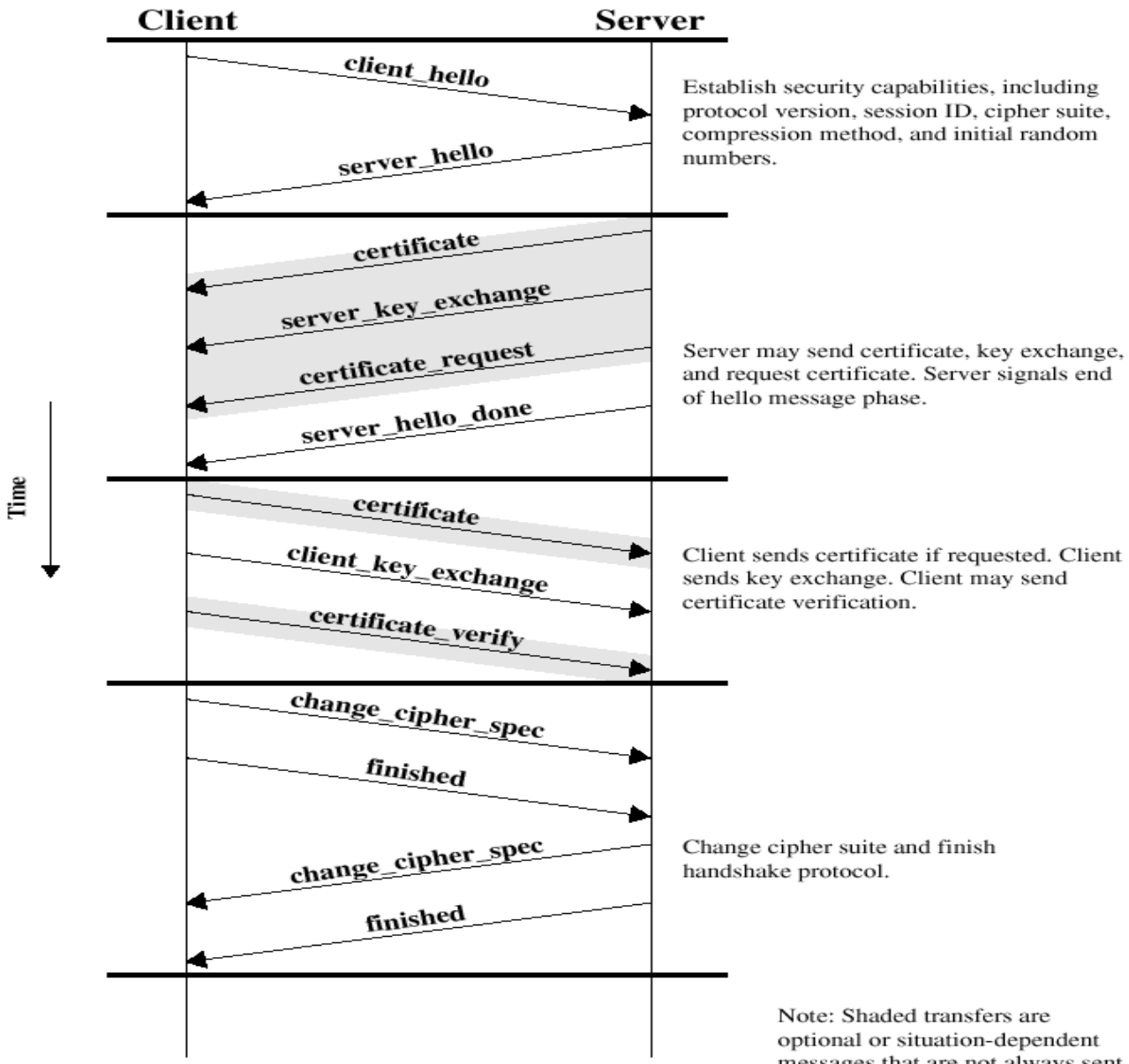
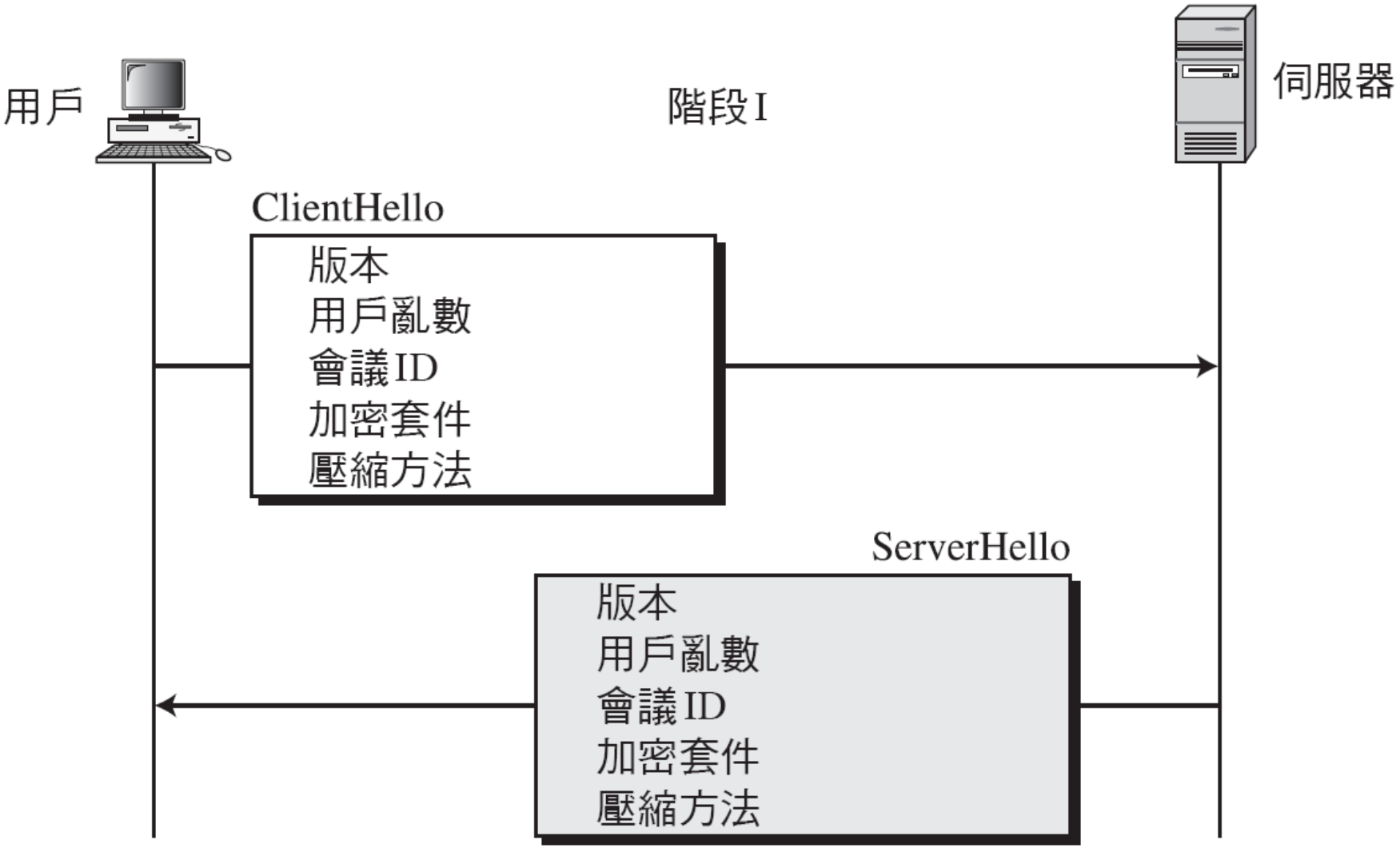


圖 17.14 握手協定的階段 I



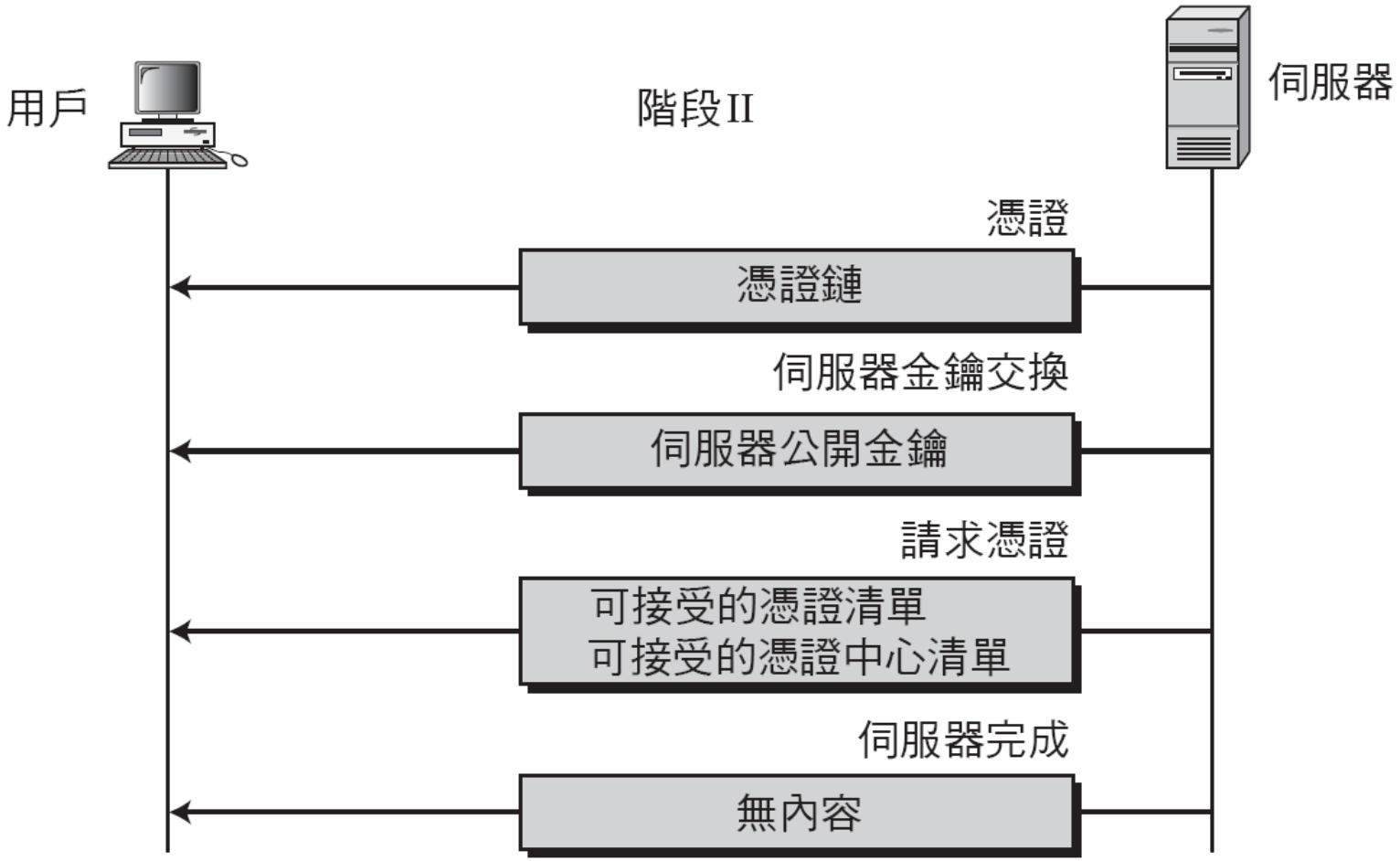
17.2.1 握手協定

注意

在階段I之後，用戶和伺服器知道如下內容：

- SSL的版本
- 金鑰交換、訊息確認和加密的演算法
- 壓縮方法
- 產生金鑰的兩個亂數

圖 17.15 握手協定的階段 II



17.2.1 握手協定 (續)

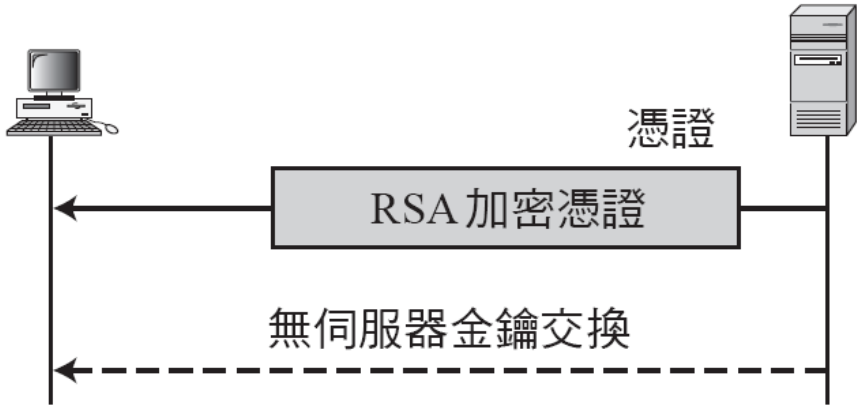
注意

在階段 II之後，

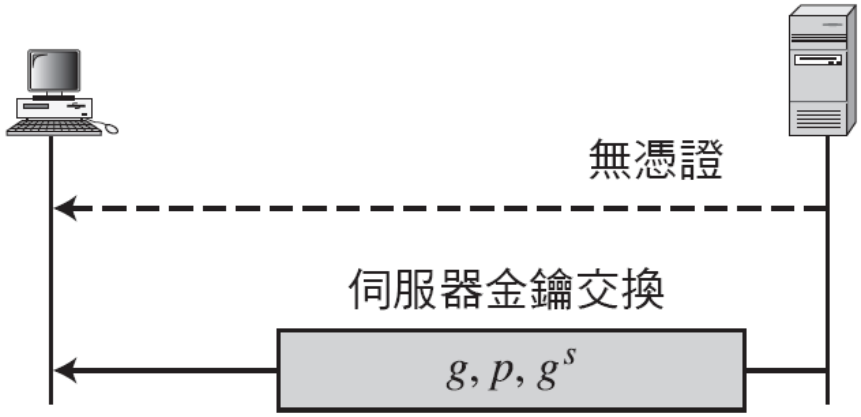
- 用戶已確認伺服器的身份
- 如需要，用戶已知伺服器的公鑰

圖 17.16 階段 II 的四種實例 (參考前面介紹)

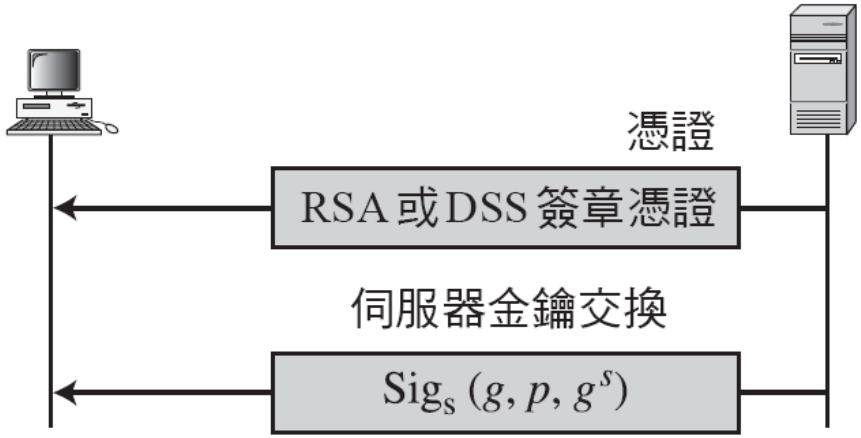
進階--後面解釋



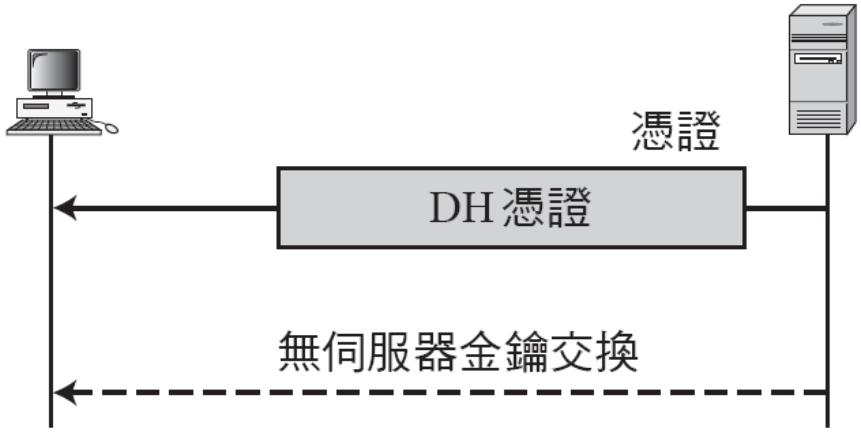
a. RSA



b. 匿名式 DH

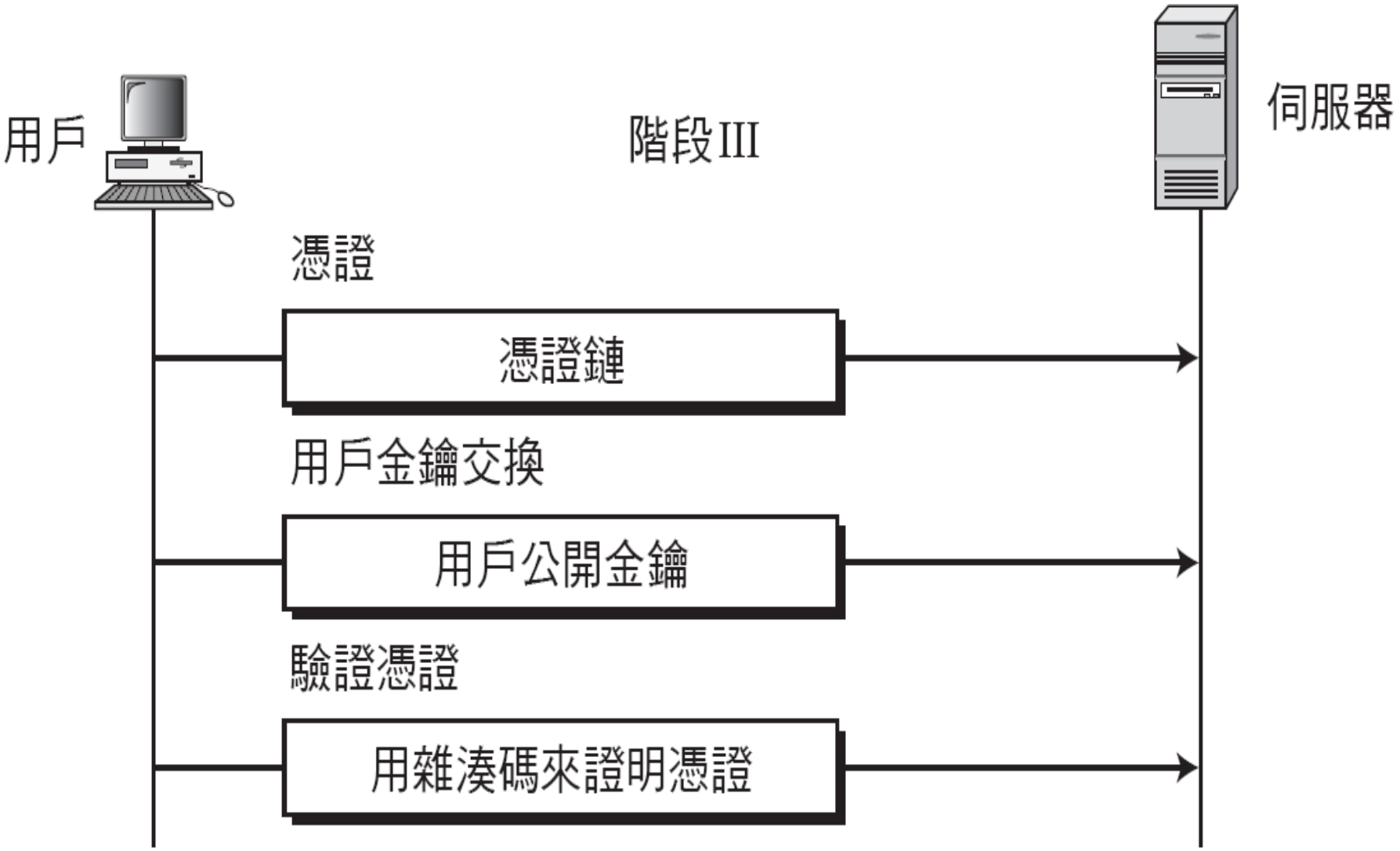


c. 暫時式 DH



d. 固定式 DH

圖 17.17 握手協定的階段 III



17.2.1 握手協定 (續)

注意

在階段 III之後，

- 伺服器已經確認用戶的身份。
- 用戶與伺服器已經知道預先主金鑰。

圖 17.18 階段 III 的四個實例

S 使用伺服器的公開金鑰加密

Sig_c: 使用用戶的公開金鑰簽章

進階--後面解釋

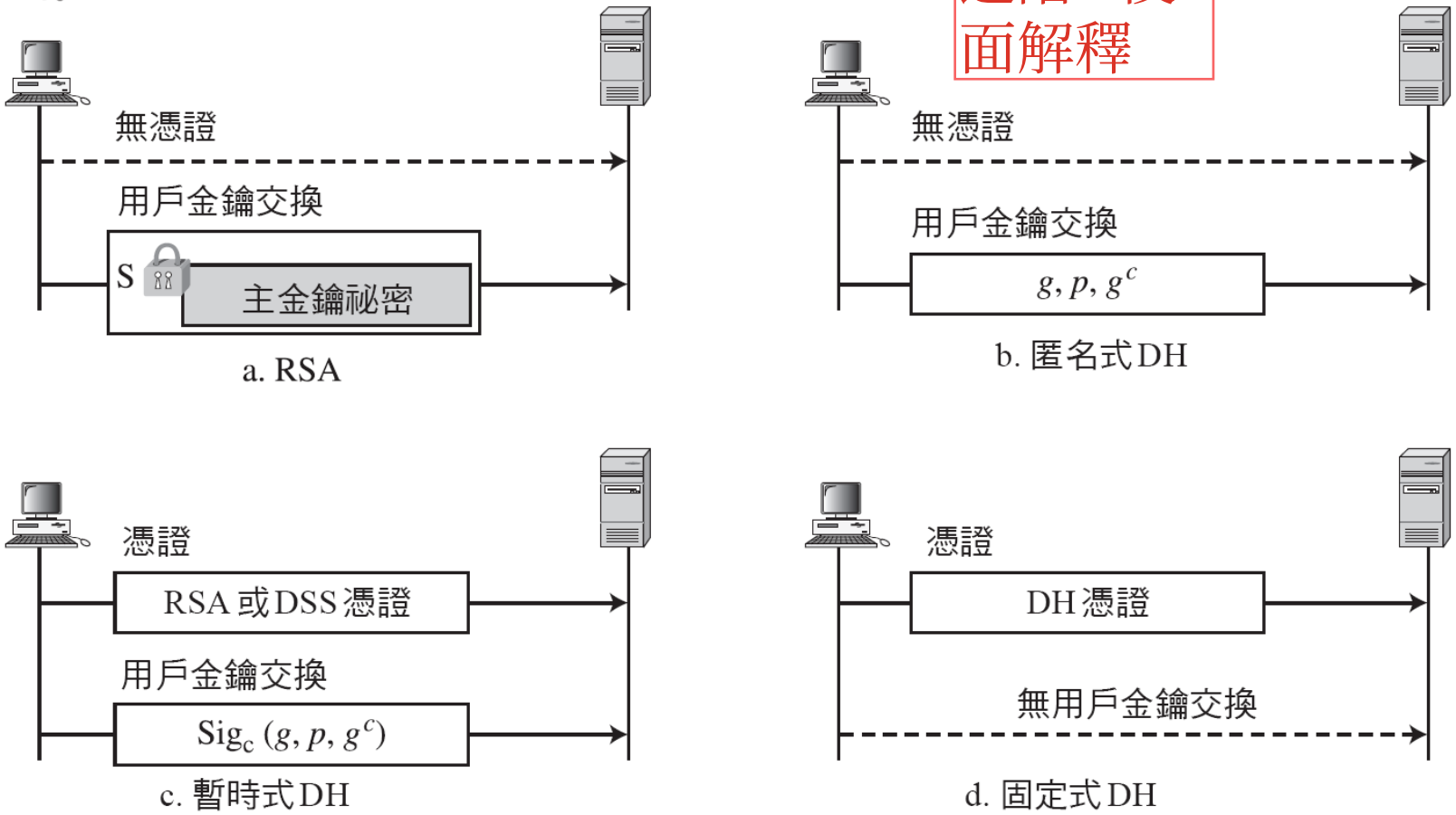


圖 17.19 握手協定的階段 IV

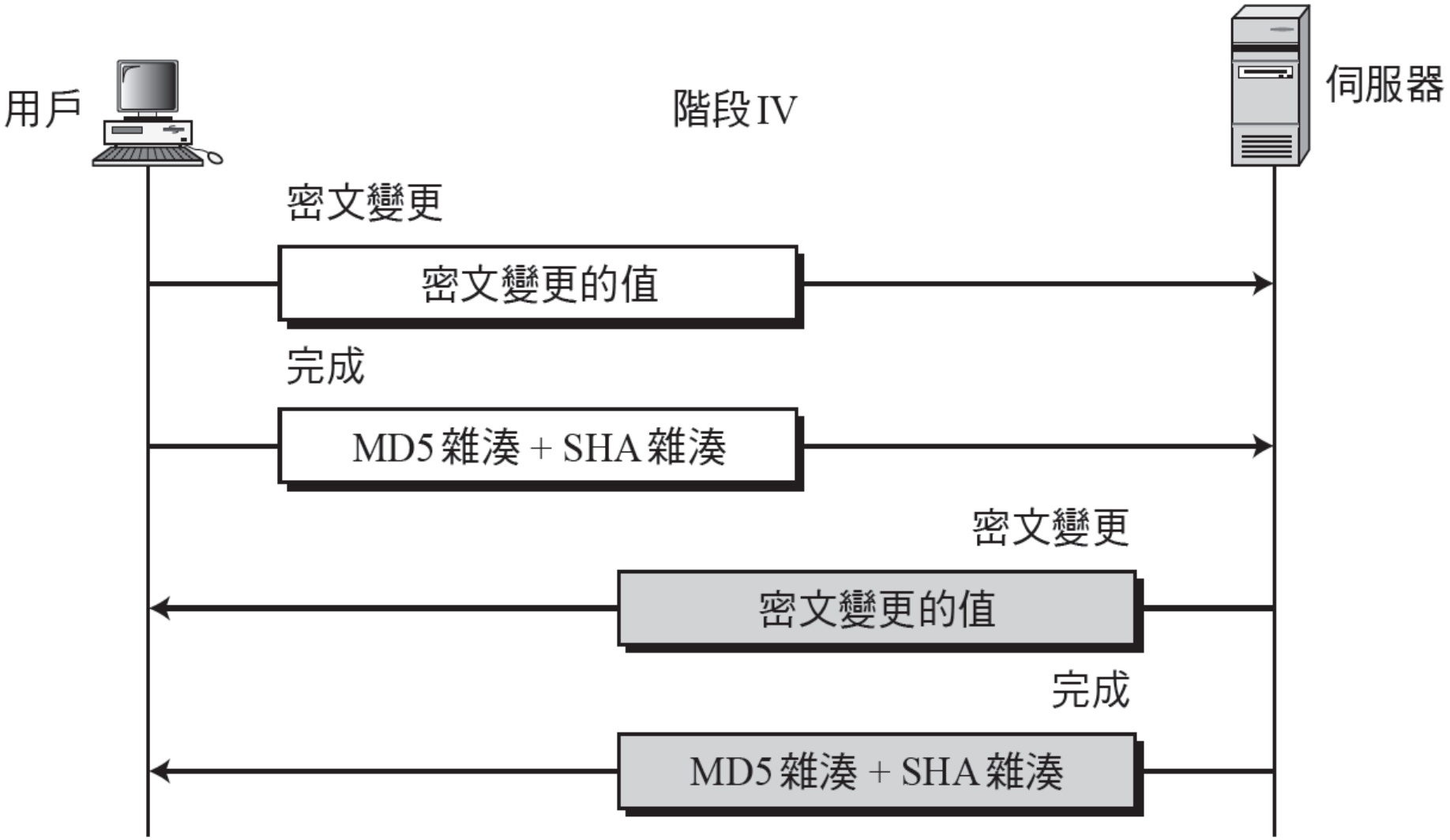
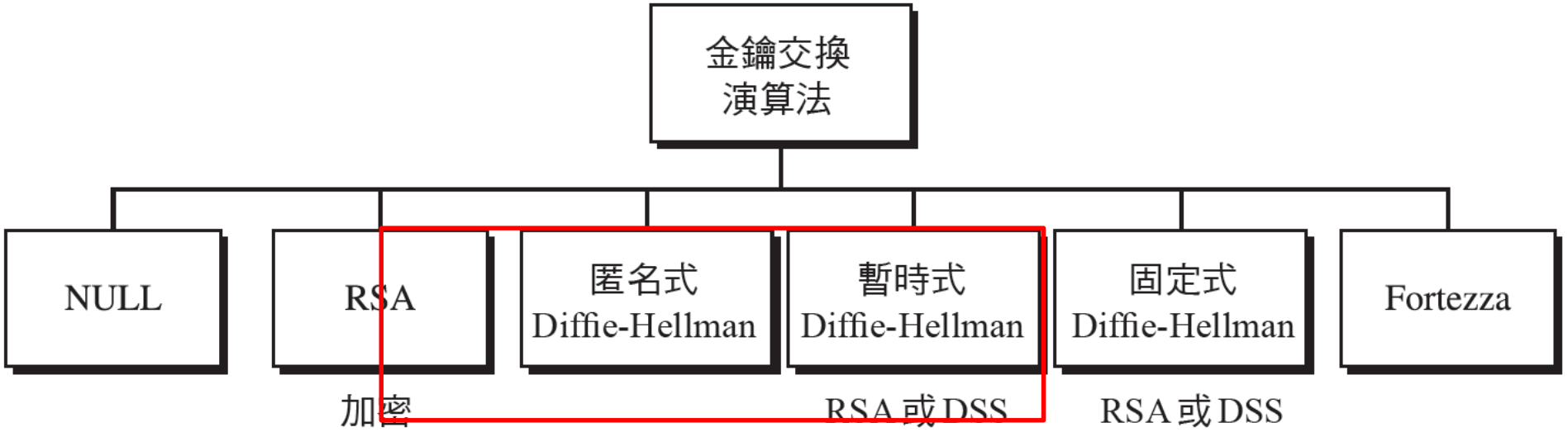


圖 17.2 金鑰交換方法

進階--後面解釋



17.1.2 金鑰交換演算法

- Null
  - 在此方法中並沒有金鑰交換，用戶和伺服器間沒有建立預先主金鑰。

用戶和伺服器兩者需先知道預先主金鑰的值。

圖 17.3　RSA 金鑰交換；伺服器公開金鑰

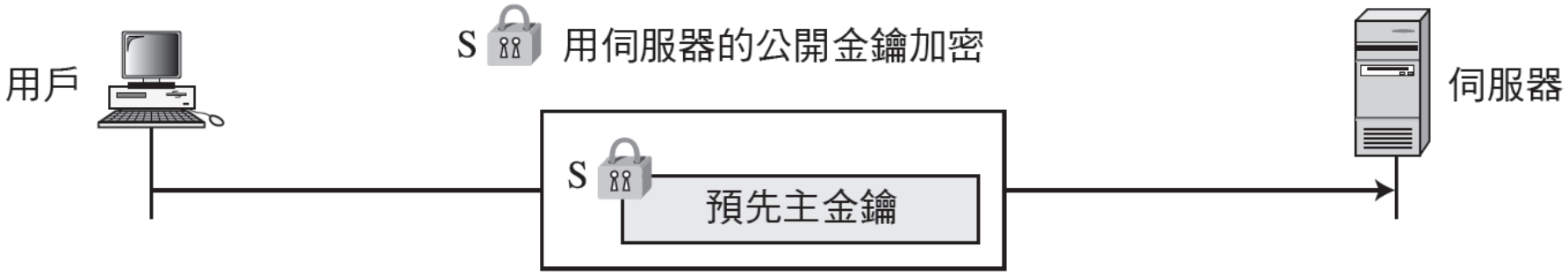
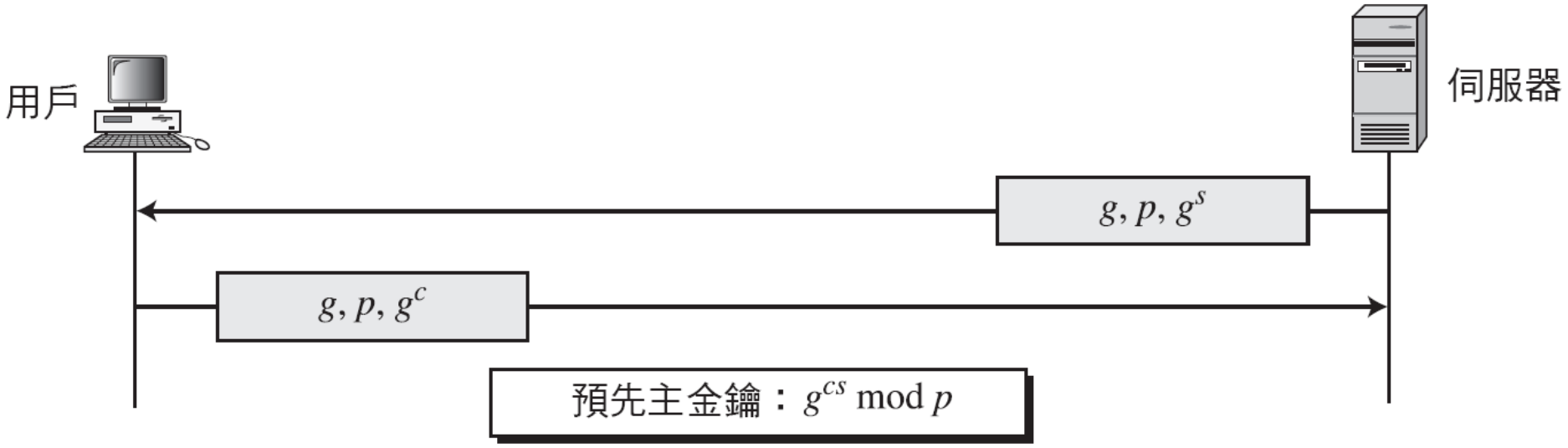
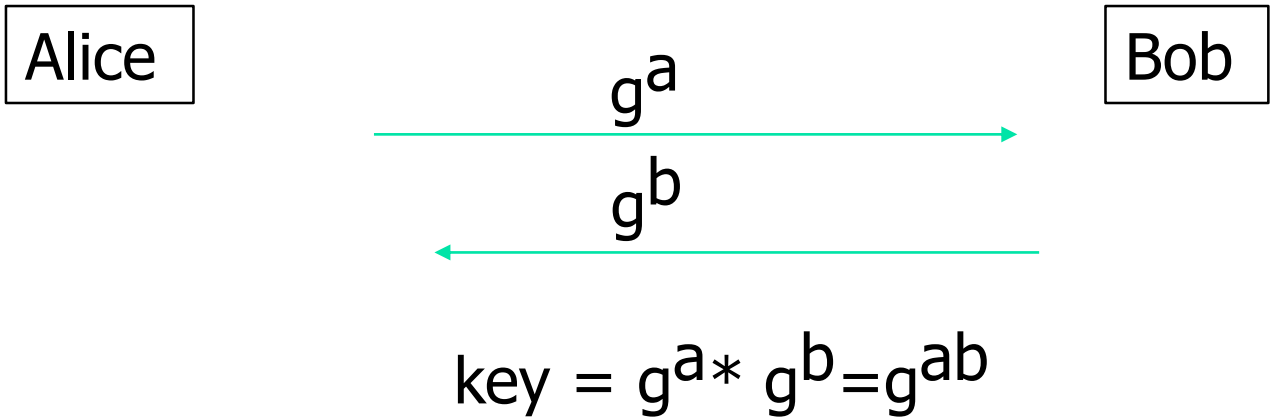


圖 17.4　匿名式 Diffie-Hellman 金鑰交換演算法



Anonymous Diffie-Hellman

- 開始之前， Alice和Bob都會創建自己的公鑰(g^a、g^b)、私鑰(a、b)
- Anonymous Diffie-Hellman
  - Alice和Bob都會得到一個未驗證身分Client的公鑰，不認識對方的時候進行嘗試通訊
  - 由於未驗證對方的身分，容易受到中間人攻擊 Alice→Mallory→Bob
    - Mallory是第三者，他會假裝是Alice和Bob



Fixed Diffie-Hellman

- 開始之前， Alice和Bob都會創建自己的公鑰(g^a、g^b)、私鑰(a、b)
- 在證書中提供他們的公鑰
- Alice和Bob都會收到被簽名的公鑰
- 由Server來認證的身分

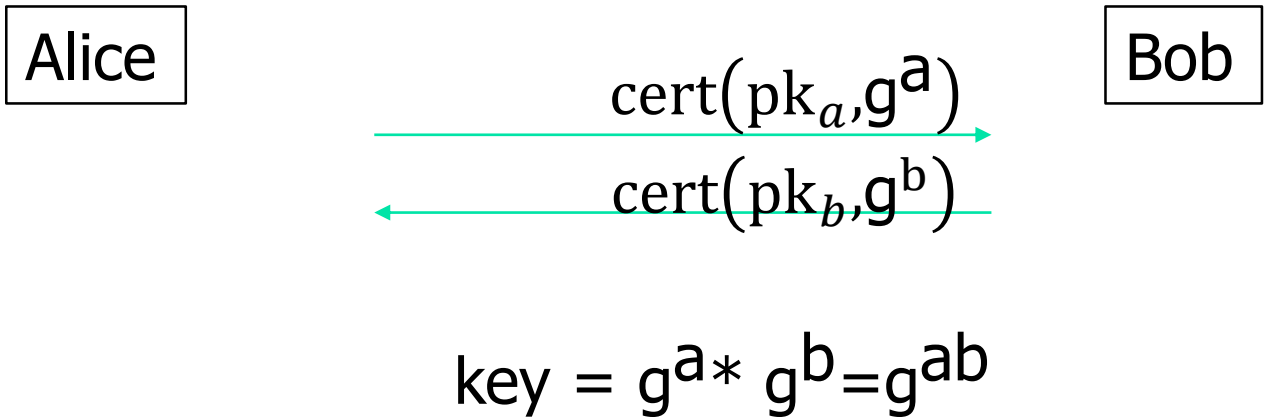
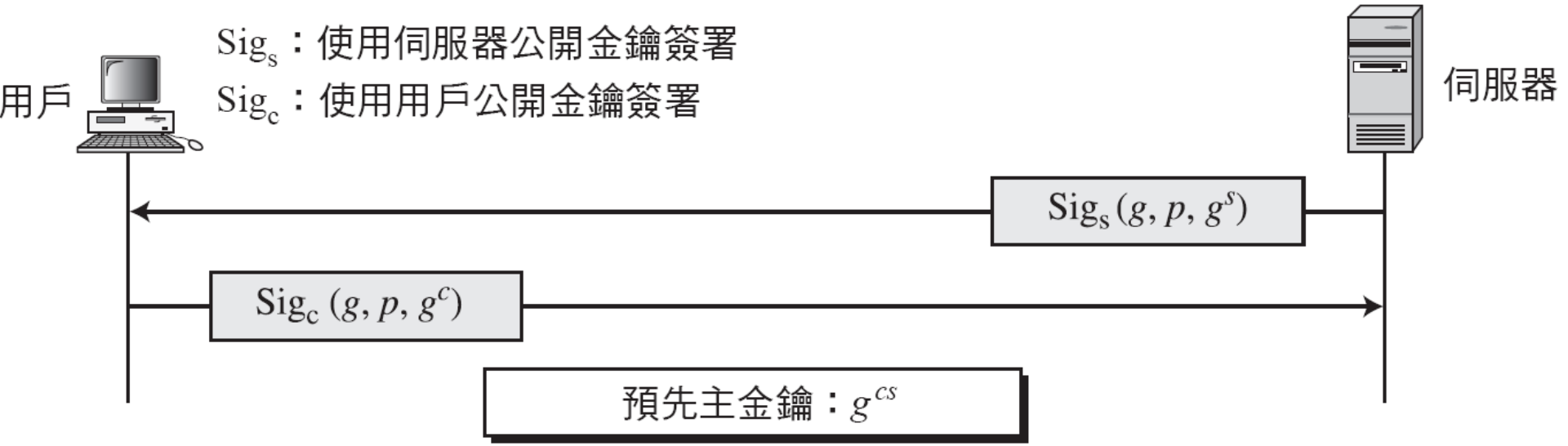
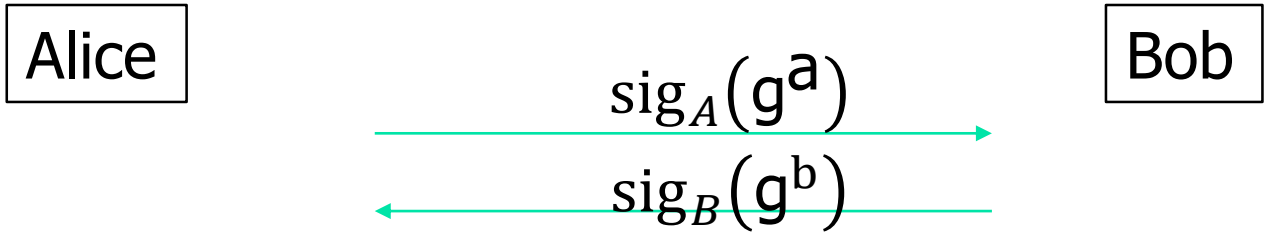


圖 17.5　暫時式 Diffie-Hellman 金鑰交換演算法



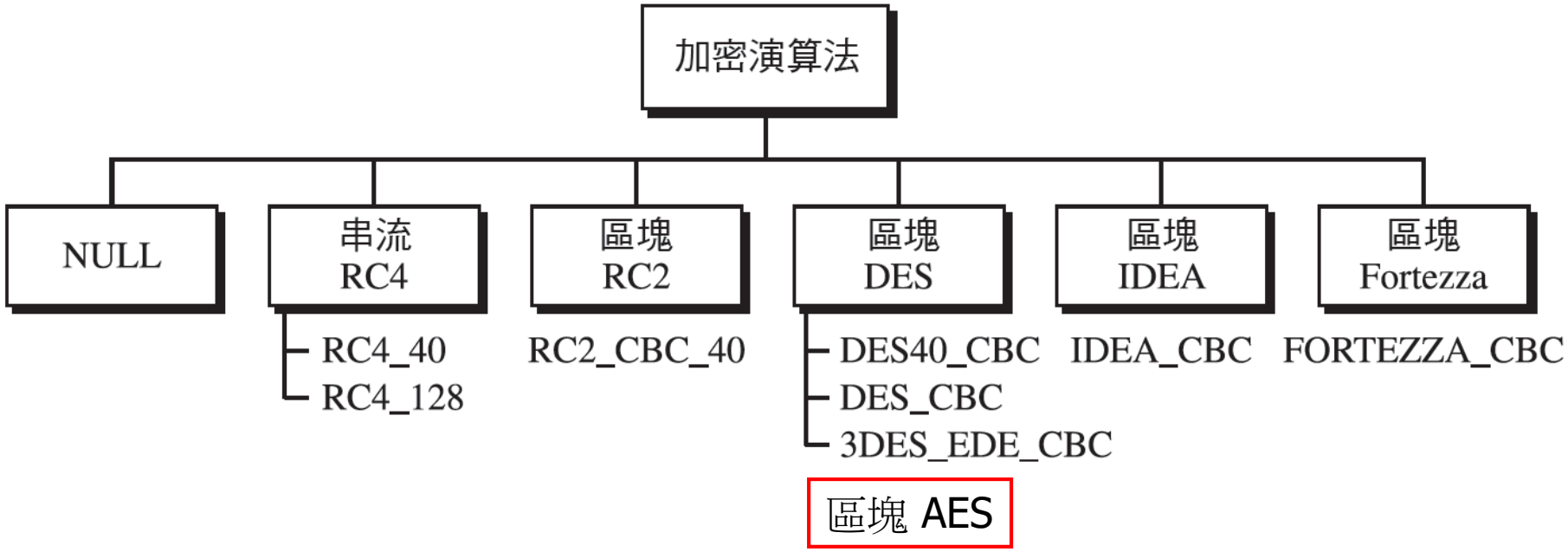
Ephemeral Diffie-Hellman

- 開始之前， Alice和Bob都會創建自己的公鑰(g^a、g^b)、私鑰(a、b)
- 與Fixed Diffie-Hellman不同之處
  - Fixed Diffie-Hellman密鑰交換始終使用相同的 Diffie-Hellman 金鑰
  - 每次連接時都會重新創建公鑰(g^a、g^b)、私鑰(a、b)
  - 因此永遠不會使用相同的密鑰兩次
  - 就算未來Server被破解，被破解的Server無法重新取得過去產生的公鑰



$$\text{key} = g^a * g^b = g^{ab}$$

圖 17.6 加密／解密演算法



17.1.5 加密套件

- 金鑰交換、雜湊函數、加密演算法的組合定義了一個為 SSL 會議的加密套件（cipher suite）。

SSL_DHE_RSA_WITH_DES_CBC_SHA

Videos

1. SSL Certificate Explained
   http://www.youtube.com/watch?v=SJJmoDZ3il8&feature=related
2. How SSL works tutorial - with HTTPS example
   http://www.youtube.com/watch?v=iQsKdtjwtYI&feature=related
3. What is HTTPS?
   http://www.youtube.com/watch?v=JCvPnwpWVUQ&feature=related
4. HTTPS and SSL tutorial

https://www.youtube.com/watch?v=_p-LNLv49Ug