科技部資訊安全技術研發專案計畫

『系統測試計畫書』

System Testing Plan

可信賴的 App 安全應用框架-App 應用服務可移轉性驗證

Trusted App Framework-Transferability Verification on App

MOST 103-2221-E-011 -090 -MY2

研究團隊

主持人：吳宗成教授(臺灣科技大學資管系)

共同主持人：羅乃維教授(臺灣科技大學資管系)

查士朝副教授(臺灣科技大學資管系)

葉國暉副教授(東華大學資管系)

蔡國裕助理教授(華夏科技大學資管系)

專任研究助理：藍袖瑛(臺灣科技大學資管系)

Department of Information Management,

National Taiwan University of Science and Technology, Taiwan

2016/6/14

目錄

版次變更記錄 3

1. 緒論 (Introduction) 4

1.1 測試範圍 (Scope of Testing) 5

1.2 接受準則 (Acceptance Criteria) 5

2. 測試環境 (Testing Environment) 8

2.1 硬體規格 (Hardware Specification) 8

2.2 軟體規格 (Software Specification) 9

2.3 測試資料來源 (Test Data Sources) 10

3. 測試時程、程序與責任 (Testing Schedule, Procedure, Responsibility) 12

3.1 測試時程 (Testing Schedule) 12

3.2 測試程序 (Testing Procedure) 12

3.2.1 整合測試(Integration Testing) 12

3.2.2 系統壓力測試(System Stress Testing) 17

3.3 人員職責分配 (Personnel Responsibility Assignment) 28

3.3.1 接受測試 (Acceptance Testing) 29

4. 測試案例 (Test Cases) 31

4.1 整合測試案例 (Integration Testing Cases) 31

4.1.1 IT1 測試案例 31

4.1.2 IT2 測試案例 32

4.1.3 IT3 測試案例 33

4.1.4 IT4 測試案例 34

4.1.5 IT5 測試案例 35

4.1.6 IT6 測試案例 36

4.1.7 IT7 測試案例 37

4.1.8 IT8 測試案例 38

4.1.9 IT9 測試案例 39

4.1.10 IT10 測試案例 40

4.2 接受測試案例 (Acceptance Testing Cases) 41

4.2.1 AT1 測試案例 41
4.2.3 AT2 測試案例 42
4.2.4 AT3 測試案例 43
5. 測試結果與分析 (Test Results and Analysis) 44
5.1 整合測試案例 (Integration Testing Cases) 44
5.2 接受測試案例 (Acceptance Testing Cases) 44
Appendix A： 追朔表 Traceability 45
A.1. 子系統 vs. 測試案例 (Subsystems vs. Test Cases) 45
A.2. 需求 vs. 測試案例 (Requirements vs. Test Cases) 46
Appendix B： 參考資料 (References) 47

版次變更記錄

版次	變更項目	變更日期
1.0	第一版	2016.06.14

1. 緒論 (Introduction)

可信賴的 App 安全應用框架(Trust App Framework)平台涉及 App 開發者、App 市集站(App Market)經營者、App 使用者，及相關需建置之安全伺服器，以支援不同的資訊安全應用與服務。本可信賴的 App 安全應用平台所需建置之安全伺服器包含 App 安全強化伺服器以防制 App 非授權複製與執行；App 移轉伺服器為提供 App 使用權移轉與驗證;App 檢測伺服器為提供 App 隱私風險檢測、App 完整性驗證、App 資料流限制驗證等功能。

本計畫第二年的進度是在可信賴的 App 安全應用框架平台上，呈現以下功能：(1) 整合移轉次數限制、時間區間移轉限制、硬體設備移轉限制之授權移轉模組，發展具多重限制條件之一對多 App 可移轉證明機制。(2) 建立潛隱通道機制，將 App 來源相關訊息隱藏於該通道中，以利追蹤非授權使用或篡改之 App。(3) 根據 App 資料敏感性分析與隱私風險評估之結果，建置自動化機敏資訊遮罩機制，以降低使用者隱私洩漏的可能性。(4) 利用反組譯技術，萃取 App 權限與 Metadata 資訊，快速產生特徵值，用以檢測 App 程式是否重新包裝。(5) 發展 App 隱私權政策與確認機制，使得 App 開發者能將隱私權政策嵌入在所開發之 App 中；App 使用者可運用 App 執行時產生之資料流分析結果，確認 App 隱私權政策與 App 行為是否一致。

本計畫已完成智慧型手機之安全且可信賴的 App 應用框架與相關驗證機制，並建置系統以驗證該系統之實用可行性，包含 App 應用服務可移轉性驗證模組、App 非授權複製模組、App 隱私風險檢測模組、App 重新包裝威脅與權限洩漏風險之防護模組、App 進階資料流限制性驗證模組。本整合型計畫中各子計畫的研究議題都是行動商務中相當重要的資訊安全議題，在實務上亦具有相當高的商業價值。

依據本應用框架的設計平台下列各項子系統將進行各子系統測試以及模組整合測試和接受測試(Acceptance Testing)：

(1) App 應用服務可移轉性驗證模組

(2) App 抗複製偵測模組

(3) App 隱私風險檢測模組

(4) App 源碼完整性驗證模組

(5) App 資料流限制性驗證模組

1.1 測試範圍 (Scope of Testing)

本可信賴的 App 安全應用框架的測試計畫，確認在系統整合前，所有的子系統均可正確輸出並順利整合。因此，將著重於整合測試(Integration Test)與接受度測試(Acceptance Test)，進一步找出本系統中的缺失，測試各個子系統於整合時衝突的情況是否會發生。本計畫未使用相關開放原始碼，無與商業軟體比較。進行整合測試的相關計畫與內容，並希望透過此文件的規劃與實踐，達到順利測試並正確執行各子系統，以正確完成本模組之系統整合。

1.2 接受準則 (Acceptance Criteria)

本測試計畫需要滿足下列的測試接受準則：

(1) Wi-Fi 裝置之 Wi-Fi 通訊功能需正常執行。

(2) 多款 Wi-Fi 行動裝置能夠正常於本平台執行。

(3) App 應用服務可移轉性驗證模組能與授權身分驗證模組溝通。

(4) App 應用服務可移轉性驗證模組能與可移轉證明產生及驗證模組溝通。

(5) 行動裝置端整合正常運作 App 動態載入模組與 App 抗複製偵測模組。

(6) App 安全強化伺服器端需整合且正常運作 App 檔案上傳模組、App 來源鑑別模組、個人金鑰管理模組及開發者智財權追蹤模組。

(7) App 安全強化伺服器端能與行動裝置間資料交換功能需正常運作。

(8) App 檔案上傳模組需正常執行安裝檔與分離程式檔上傳功能。

(9) 個人金鑰管理模組須能生成與分配個人金鑰的功能。

(10) 個人金鑰管理模組須能對分離程式檔進行對稱式區塊加密的功能。

(11) App 動態載入模組能與 App 來源鑑別模組進行身分鑑別功能。

(12) App 抗複製偵測模組需正常產生追蹤記錄的功能。

(13) App 抗複製偵測模組須能與開發者智財權追蹤模組資料交換功能。

(14) 開發者智財權追蹤模組需正常執行動態洩密者追蹤的功能。

(15) 開發者智財權追蹤模組須能廢止洩密者權利的功能。

(16) 網路爬蟲程式將指定的 APK 檔案下載回本地端需正常執行。

(17) 客戶端裝置取得特定 APK 檔案之特徵值組的功能需正常執行。

(18) 客戶端裝置將驗證 APK 檔案與訊息順利傳送至 Server 端需正常執行。

(19) APK 驗證 Server 接收來自客戶端裝置之待驗證 APK 訊息需正常執行。

(20) 客戶端裝置將欲驗證之 APK 特徵指紋與 Metadata 順利傳送至 Server 端需正常執行。

(21) APK 驗證 Server 接收來自客戶端裝置之待驗證 APK 訊息需正常執行。

(22) APK 驗證 Server 對資料庫之查詢操作需正常進行。

(23) APK 驗證 Server 將即時比對後的資料回傳給客戶端裝置需正常執行。

(24) 客戶端裝置將收到的比對結果轉換為圖示顯示在畫面上需正常執行。

(25) 行動裝置端上傳檢測檔案至檢測伺服器功能需正常執行。

(26) App 資料流限制性驗證模組功能需正常執行。

(27) 行動應用程式資料流分析結果以圖形化介面呈現功能需正常執行。

(28) App 檢測伺服器端需整合 App 資料流限制性驗證模組，且確認功能間運作正常。

(29) 行動裝置端 App 上傳至檢測伺服器端 App 資料流分析模組間資料交換功能需正常運作。

(30) 隱私風險評估模組可以正常擷取裝置安裝之 App 的資料庫。

(31) 隱私風險評估模組可以正常傳送及接受自 App 檢測伺服器之資訊。

(32) 原始資料擷取模組可以正常擷取裝置上的文件內容

(33) 資料遮罩加密模組可以正確將資料遮罩並加密原始資料。

(34) 資料遮罩加密模組可以正確將加密後的資料儲存於 App 伺服器。

(35) 資料遮罩加密模組可以正確從 App 檢測伺服器獲得先前儲存的資料，並解密還原。

(36) App 檢測伺服器能根據收到的資訊分析並傳回結果。

(37) 使用者介面能根據分析結果呈現數據及報告。

(38) 使用者介面能提醒使用者隱私洩漏的情況

(39) 使用者介面能提醒使用者遇到詐騙的情況。

(40) 使用者介面能提醒使用者裝置存在機敏文件的情況。

(41) 各項參數及數據在資料流中能正確的傳遞。

(42) 系統模型可依實際環境變更進行調整。

(43) 測試程序需要依循本測試計畫所訂定的程序進行，且測試結果必須符合

預期測試結果方能接受。

(44) 本系統需要對所有列為必要(Critical、Important、Desirable)之需求進行完整測試。

(45) 測試程序需要依照本測試計畫所訂定的程序進行,所有測試結果需要能符合系統需求規格書預期測試結果方能接受。

(46) 以測試案例為單位，當測試未通過時，需要進行該單元的測試，其接受的準則與前一項之規定相同。

2. 測試環境 (Testing Environment)

圖 1 為可信賴的 App 安全應用框架進行測試的環境描述：

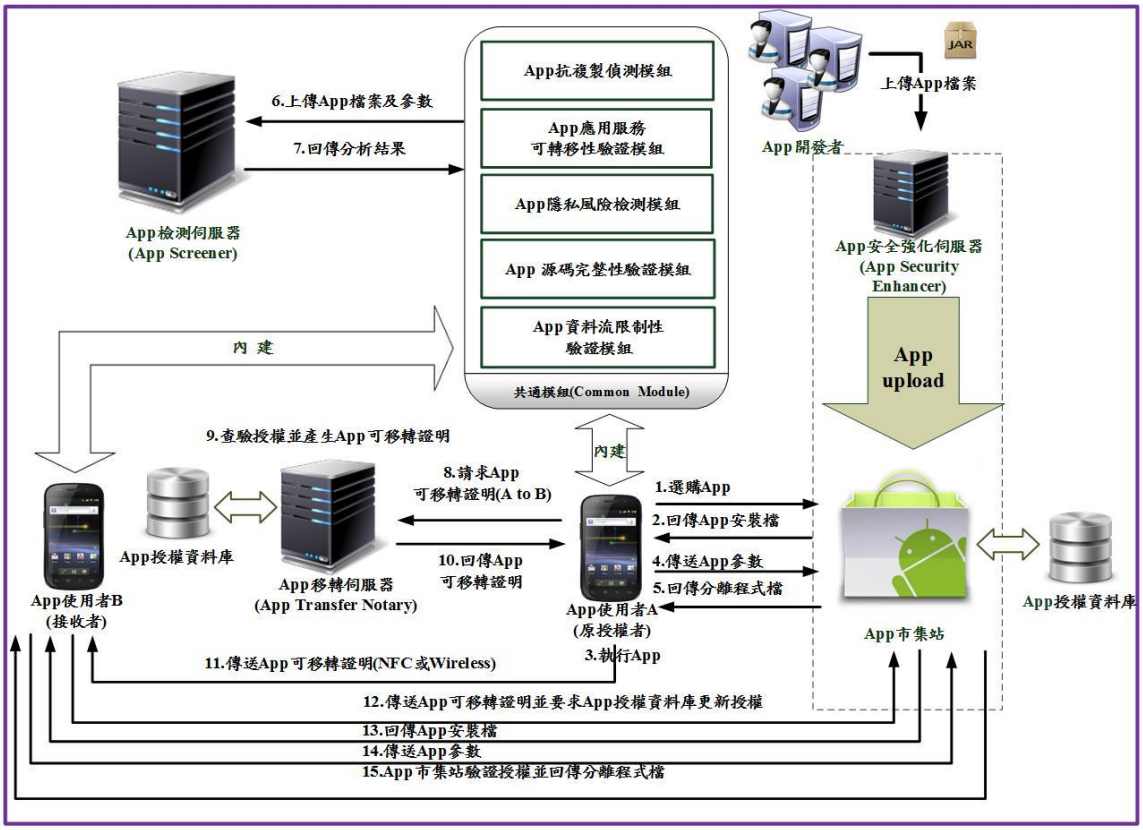


圖1. 可信賴的 App 安全應用框架測試環境架構圖

2.1 硬體規格 (Hardware Specification)

依據圖 1 測試環境架構圖內容，進行測試之硬體規格說明，如下列所示：

(1) App 使用者 A

- Tablet：HTC Nexus 9
- CPU：NVIDIA Tegra K1, 2.3GHz
- RAM：2GB
- ROM：32GB

(2) App 使用者 B

- Tablet：HTC Nexus 9
- CPU：NVIDIA Tegra K1, 2.3GHz
- RAM：2GB

- ROM：32GB

(3) App 移轉伺服器(App Transfer Notary)：含 App 應用服務可移轉性驗證模組

- CPU：Intel Core i7-3770 3.40 GHz
- RAM：4 GB
- 硬碟空間：500GB

(4) App 安全強化伺服器(App Security Enhancer)：含 App 抗複製偵測模組

- CPU：Intel Core i7-3770 3.40 GHz
- RAM：4 GB
- 硬碟空間：500GB

(5) App 檢測伺服器(App Screener)：含 App 隱私風險檢測模組、App 源碼完整性驗證模組、App 資料流限制性驗證模組

- CPU：Intel Core i7-3770 3.40 GHz
- RAM：4 GB
- 硬碟空間：500GB

2.2 軟體規格 (Software Specification)

依據圖 1 測試環境架構圖內容，測試環境之軟體規格說明，如下列所示：

(1) App 檢測伺服器(App Screener)：包含授權身分驗證模組、可移轉證明產生模組、可移轉證明驗證模組、行動 App 隱私風險分析與防護模組、進階資料流限制性進階驗證工具、App 特徵偵測應用程式

- 伺服器作業系統：Windows 7
- 網頁伺服器：PHP + Apache Tomcat 7.0+ Java-Spring+ Apache Server 2.4.9
- 應用程式開發語言：PHP 5.5+Java 1.7.0_55

(2) App 移轉伺服器

- 應用程式開發語言：Java
- 伺服器作業系統：Windows 7
- 網頁伺服器：Apache Tomcat 7.0+ Apache Server 2.4.9

(3) App 安全強化伺服器(App Security Enhancer)：含 App 抗複製偵測模組

- 應用程式開發語言：PHP5.5

- 伺服器作業系統：Windows 7
- 網頁伺服器：PHP + Apache Server 2.4.9

(4) 安裝於智慧型裝置之 App 應用服務可移轉性驗證模組

- 行動裝置作業系統：Android 4.3.1
- 應用程式開發語言：Java

(5) 安裝於智慧型裝置之 App 抗複製偵測模組

- 行動裝置作業系統：Android 4.3.1
- 應用程式開發語言：Java

(6) 安裝於智慧型裝置之 App 隱私風險檢測模組

- 手機作業系統：Android 4.3.1
- 應用程式開發語言：Java

(7) 安裝於智慧型裝置之 App 源碼完整性驗證模組

- 手機作業系統：Android 4.3.1
- 應用程式開發語言：Java

(8) 安裝於智慧型裝置之 App 資料流限制性驗證模組

- 手機作業系統：Android 4.3.1
- 應用程式開發語言：Java

2.3 測試資料來源 (Test Data Sources)

測試項目	數量	測試來源
Wi-Fi	一組	App 原持有者透過 Wi-Fi 傳送 App 可移轉證明
驗證授權次數限制	一組	App 原持有者請求 App 可移轉證明時，可信賴第三方伺服器檢核授權次數限制
驗證授權時間區間	一組	App 原持有者請求 App 可移轉證明時，可信賴第三方伺服器檢核授權時間區間
猜猜紅心 A(可下載之 App)	一支	產生之測試用 App，供本團隊驗證動態載入與動態洩密者追蹤機制。本 App 可公開驗證
電池計量(可下載之 App)	一支	產生之測試用 App，供本團隊驗證動態載入與動態洩密者追蹤機制。本 App 可公開驗證
記憶卡容量(可下載之 App)	一支	產生之測試用 App，供本團隊驗證動態載入與動態洩密者追蹤機制。本 App 可公開驗證
向左轉向右轉(可下載之 App)	一支	產生之測試用 App，供本團隊驗證動態載入與動態洩密者追蹤機制。本 App 可公開驗證
心電感應(可下載之 App)	一支	產生之測試用 App，供本團隊驗證動態載入

		與動態洩密者追蹤機制。本 App 可公開驗證
分離程式區段	5 個	猜猜紅心 A 、向左轉向右轉、心電感應、電池計量、記憶卡容量之 App 的分離程式區段
使用者購買記錄	10 筆	透過使用者購買記錄辨別哪幾位使用者為合法的購買者
分離程式區段	5 個	猜猜紅心 A 、向左轉向右轉、心電感應、電池計量、記憶卡容量之 App 的分離程式區段
追蹤記錄	10 筆	透過使用者購買追蹤辨別哪幾位使用者為可能的洩密者
廢止清單	2 個	經由動態洩密者追蹤機制，產生欲廢止洩密者權利的清單
使用者隱私項目	一組	符合相關格式之使用者個人資料
行動裝置即時通訊訊息	一組	使用者操作之通訊軟體的訊息紀錄
待驗證 APK 檔案(安全)	一組	1.從 Google Play 上取得之 APK 檔案 2.自行開發之 APK File
待驗證 APK 檔案(危險)	一組	1.第三方市集上取得被二次修改與簽章之 APK 檔案 2.由 OWASP 提供之惡意 APK File

3. 測試時程、程序與責任 (Testing Schedule, Procedure, Responsibility)

3.1 測試時程 (Testing Schedule)

(1) 時程

測試項目	時間
各子系統之內部元件整合測試 (Module Test)	2016/03/19~2016/05/31
TAF 系統整合測試 (Integration Test)	2016/06/01~2016/06/22
TAF 系統接受度測試 (Acceptance Test)	2016/06/23~2016/07/28

(2) 查核

測試項目	時間
各子系統之內部元件整合測試 (Module Test)	2016/06/01
TAF 系統整合測試 (Integration Test)	2016/06/23
TAF 系統接受度測試 (Acceptance Test)	2016/07/29

3.2 測試程序 (Testing Procedure)

各子系統的元件測試，由各子系統的開發負責人執行。待各子系統整合完成後，由全體人員協同執行整體計畫的整合測試。

3.2.1 整合測試(Integration Testing)

可信賴的 App 安全應用框架平台之使用案例(Use Case)，以測試 App 應用服務可轉移驗證模組、App 抗複製偵測模組、App 隱私風險檢測模組、App 源碼完整性驗證模組、App 資料流限制性驗證模組之整合為目的。

- 模組一(App 應用服務可移轉性驗證模組)如下圖所示：

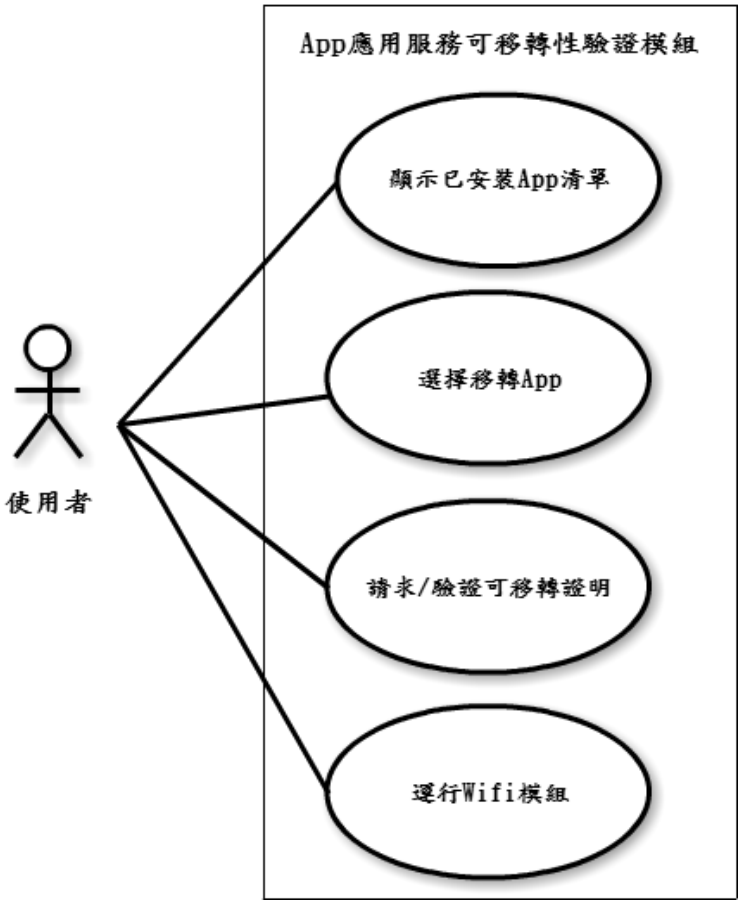


圖2.App 應用服務可移轉性驗證模組之使用案例

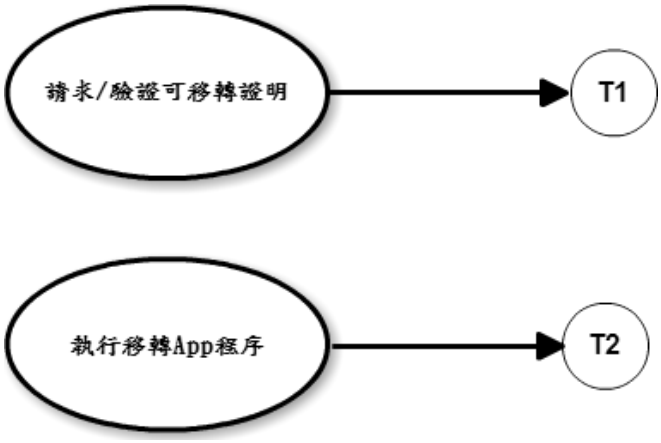


圖3.App 應用服務可移轉性驗證模組接受度測試

✓ 模組二(App 抗複製偵測模組)如下圖所示：

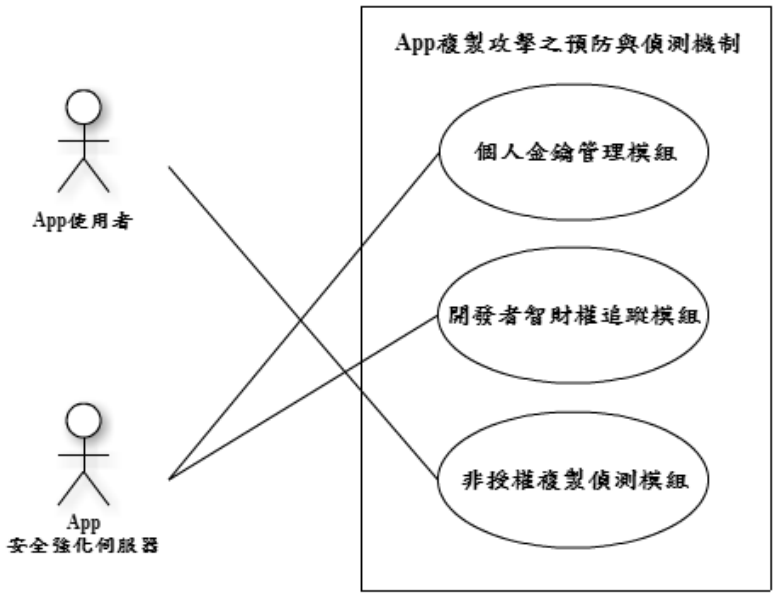


圖4. App 抗複製偵測模組之使用案例

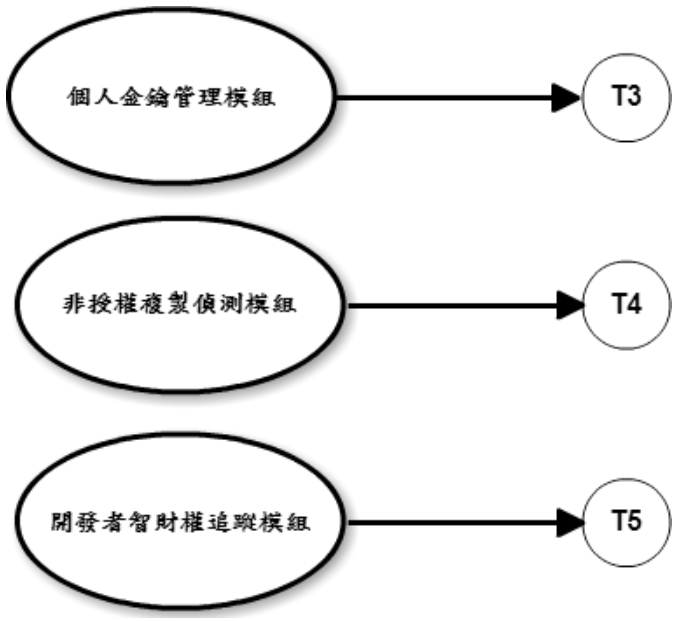


圖5. App 抗複製偵測模組接受度測試

✓ 模組三(App 隱私風險檢測模組)如下圖所示：

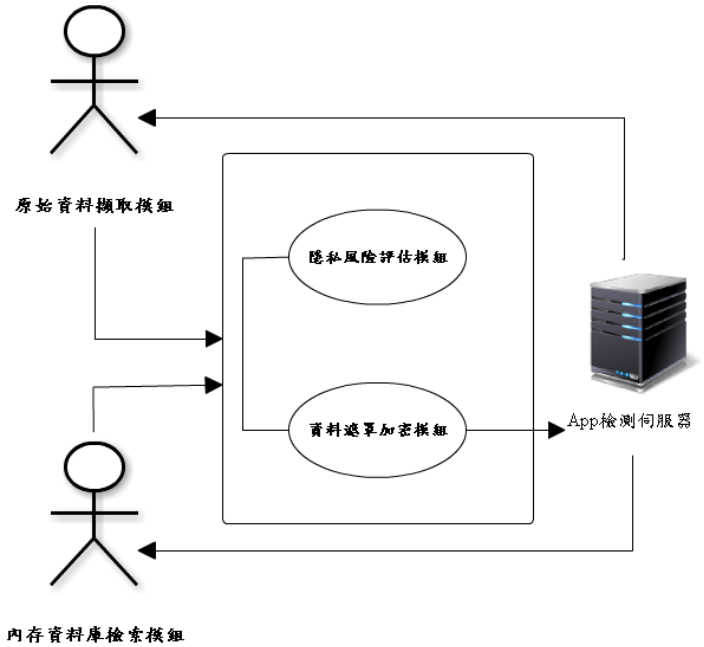


圖6. App 隱私風險檢測模組之使用案例

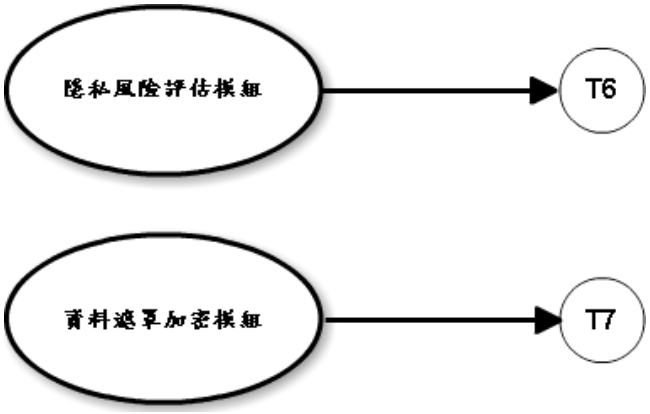


圖7. App 隱私風險檢測模組測試

✓ 模組四(App 源碼完整性驗證模組)如下圖所示：

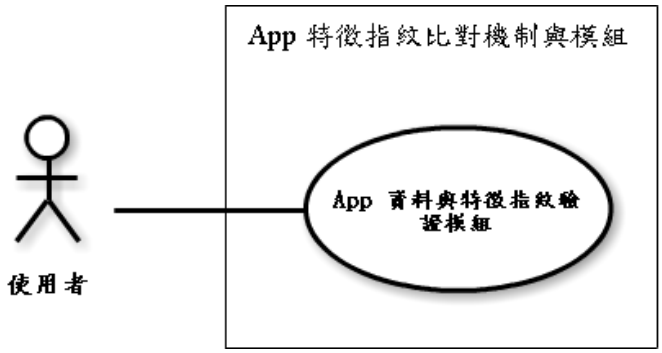


圖8. App 源碼完整性驗證模組之使用案例

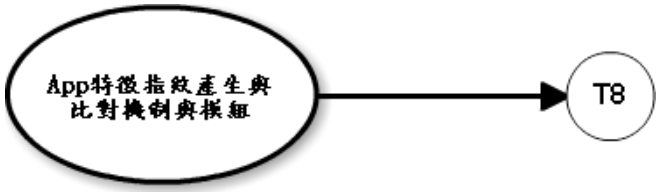


圖9. App 源碼完整性驗證模組之接受度測試

✓ 模組五(App 資料流限制性驗證模組)如下圖所示：

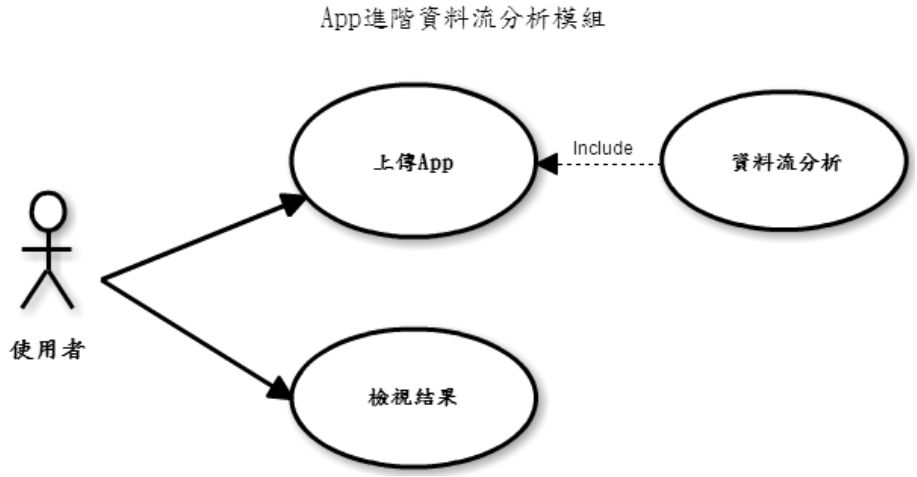


圖10. App 資料流限制性驗證模組之使用案例

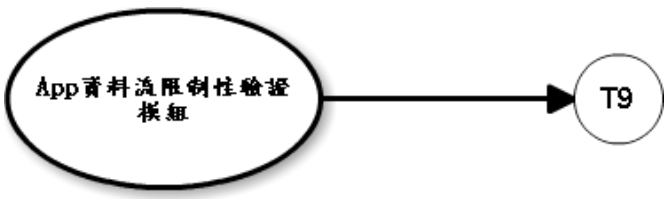


圖11. App 資料流限制性驗證模組之接受度測試

3.2.2 系統壓力測試(System Stress Testing)

本計畫使用 Apache JMeter 壓力測試工具，JMeter 是一套專門用來測試一個網際網路應用程式所能承受的壓力多寡的工具，透過此工具我們可以一個有效節省人力及時間成本的方式模擬多個使用者(Browsers)在同時連線、或是網站同時處理多個需求(Requests)時等情況，以進行網站的壓力或效能測試等工作。針對 Web Service 進行壓力測試，確保 App 伺服器服務之完整性與可用性，以提供便利和穩定的檢測服務。

■ **App Market 市集站**

本次壓力測試主要針對 App Market 所上傳 apk 檔的功能進行測試，將依據測試結果提出說明及建議，提供使用者在架設 App Market 網站服務的參考依據。

- 測試主機設備規格

  測試全程將於 100Mbps 區域網路環境下執行，並利用一台客戶端伺服器，對位於同一個區域網路內的測試伺服器進行App Market網站效能測試。以下，為本測試所使用的各測試設備規格：

✓ 伺服器硬體規格：
  - 作業系統：Windows 7 professional
  - 處理器：i7-4790 3.6GHz
  - 記憶體：8 GB
  - 硬碟：Seagate 500GB 讀取:115 MB/s 寫入:109 MB/s

✓ 客戶端硬體規格：
  - 作業系統：Windows 7 professional
  - 處理器：i7-4790 3.6GHz
  - 記憶體：8 GB
  - 硬碟：Seagate 500GB 讀取:115 MB/s 寫入:109 MB/s

- 效能測試流程

  模擬前端使用者透過同時對 App Market 伺服器送出多個網的請求，代表多個使用者對 App Market 送出上傳 apk 檔案的請求，並同時記錄每個千位元的平均處理量、每個千位元的平均傳輸量及紀錄的上傳成功的比率。首先，將先架設有 *n* 位使用者，各傳出 10 次請求開始，以遞迴地的方式向上累加遞增(*n*=10, 11, 12, …)，測試伺服器所能承受之最大請求數目。

- 壓力測試結果

  App Market 網站效能的測試之前，需訂定一個測試標準，並以此標準為基準，模擬伺服器在本計畫的測試伺服器環境下，測試多位使用

者同時上傳檔案大小約 6MB 的 App 時之效能，以及監控伺服器效能檢測出該模組所需要消耗的最大資源為何？測試結果如下表所示，在 100.0Mbps 的區域網路內，以網站上傳 App 的動作同時重複請求 10 次之結果，測試結果如表 1。

表1. App Market 伺服器壓力測試結果

使用者數目	重覆請求次數	總請求次數	平均值 (byte/sec)	處理量 (byte/sec)	錯誤率
10	10	100	1371	258	0.00%
11	10	110	1308	294	0.00%
12	10	120	1399	300	0.00%
13	10	130	1607	288	0.00%
14	10	140	1191	366	12.14%

註解如下：

- 每一個請求為約 6MB 的 APK 檔案
- 平均值：每個請求所需傳輸的平均封包大小，單位為千位元/秒
- 處理量:根據回應時間,計算出的每個請求所需處理的平均資料量大小,單位為千位元/秒

■ App 移轉驗證伺服器

透過壓力測試軟體 Jmeter，同時送出多個 HTTP POST，代表多個使用者對 App 移轉驗證伺服器送出的查詢要求，同時記錄起始時間、App 移轉驗證伺服器處理完所有請求所需時間、HTTP Request 總數以及 App 移轉驗證伺服器正確回應使用者的百分比。由於本計畫的實體設備有限，故採用在一台電腦上以程式模擬多個使用者同時對 App 移轉驗證伺服器最多送出共 1000 筆移轉證明的請求，所有請求都對應到真實存在於 DB 中的資料，以求每個請求都可以確實的讓程式對後端資料庫進行讀取並進行鑑別的動作(即回應成功或失敗)，鑑別成功後再進行移轉的動作。

- 測試主機設備規格

測試工作全程將於 100Mbps 的區域網路環境下執行，並利用一台客戶端機器，對位於同一個區域網路內的測試伺服器進行 App 移轉驗證伺服器效能測試。以下，為本測試所使用的各測試機器規格：

✓ 伺服器硬體規格：

- 作業系統：Windows 7 Professional
- 處理器：Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
- 記憶體：8 GB
- 硬碟：Seagate 500GB 讀取:115 MB/s 寫入:109 MB/s

✓ 客戶端硬體規格：

- 作業系統：Windows 7 Professional
- 處理器：Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
- 記憶體：8 GB
- 硬碟：Seagate 500GB 讀取: 115 MB/s 寫入:109 MB/s

- 效能測試流程

透過壓力測試軟體 JMeter，同時送出多個 HTTP POST 得請求，代表多個使用者對 App 移轉驗證伺服器送出的查詢要求，同時記錄起始時間、App 移轉驗證伺服器處理完所有請求所需時間、HTTP Request 總數以及 App 移轉驗證伺服器正確回應使用者的百分比。由於本計畫的實體設備有限，首先我們會先架設有 n 位使用者，各傳出 100 次請求開始，以遞迴地的方式向上累加遞增(n=500, 600, 700, …)，來測試伺服器所能承受之最大請求數目，以求每個請求都可以確實的讓程式對後端資料庫進行讀取並進行鑑別的動作(即回應成功或失敗)，鑑別成功後再進行移轉的動作，因此我們針對此兩種模組進行測試。

- 壓力測試結果

在開始進 App 移轉驗證伺服器網站效能的測試之前，我們需要訂定一個測試標準，並以此標準為基準，模擬伺服器在本計畫的測試伺服器環境下，測試多位使用者同時利用鑑別與轉移模組之效能，以及監控機器效能檢測出該模組所需要消耗的最大資源為何。

測試結果如表 2 所示，在 100.0Mbps 的區域網路內，對網站發送鑑別與轉移的動作同時重複請求 100 次之結果。

表2. App 移轉驗證伺服器壓力測試結果

使用者數目	重覆請求次數	總請求次數	平均值 (byte/sec)	處理量 (byte/sec)	錯誤率
500	100	50000	132.5	35.5	0.0%
600	100	60000	156.5	37.54	0.0%
700	100	70000	238	37	0.0%
800	100	80000	314.1	37.69	1.07%

註解如下：

- 每一個請求平均約為 50KB 的傳輸量。
- 平均值：每個請求所需傳輸的平均封包大小，單位為位元組/秒
- 處理量:根據回應時間，計算出的每個請求所需處理的平均資料量大小，單位為位元組/秒

由上表之測試結果得知，在測試伺服器環境下的 App 移轉驗證伺服器，當約 800 位使用者同時上傳 100 次 App 的情況下，網站的平均處理量會開始大量的遽增，且根據下圖之系統資源監視器顯示，我們可以看到各個硬體使用量已經出現了資源滿載的情況。因此，可推估在此測試條件下，若想讓 App 移轉驗證伺服器有正常且不出錯的效能表現，其建議的使用者負荷量約為 800 人左右，若須提升 App 移轉驗證伺服器的效能，將可參考下面的建議，依自身的需求調整網站伺服器的環境。

I. 提高處理器的時脈。
II. 增加網路寬頻。

III. 增加伺服器實體記憶體。
IV. 更換高讀寫速度的硬碟裝置。

本份測試報告僅以範例的方式對 App 移轉驗證伺服器進行效能測試，再透過給予建議的方式，協助使用者在架設相關網站服務之時，可參考我們所提供的相關數據以自行評估所需的伺服器環境。

■ **App 檢測伺服器**

App 檢測伺服器包含 App 隱私風險檢測模組、App 源碼完整性驗證模組、App 資料流限制性驗證模組，同樣以 JMeter 壓力程式測試網際網路應用程式所能承受的壓力，確保 App 伺服器服務之完整性與可用性。

- 測試主機設備規格(App 隱私風險檢測模組)

✓ 伺服器規格：

- 作業系統：Ubuntu 16.04 LTS 64bit
- 處理器：Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
- 記憶體：8 GB

✓ 客戶端規格：

- 作業系統：Windows 10 64Bit
- 處理器：Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz
- 記憶體：8 GB
- 測試軟體：JMeter

- 效能測試流程(App 隱私風險檢測模組)

以 JMeter 程式同時送出多個 HTTP POST Request 模擬多個使用者將特定的資料儲存至伺服器中，並同時記錄使用者數目(Threads)、每個使用者重覆請求次數(Loop Count)、總請求次數(Samples)、平均值、處理量。

- 壓力測試結果(App 隱私風險檢測模組)

從 200 位使用者，各傳出 100 次請求開始，來測試伺服器所能承受的多個使用者同時發出的請求數目，總請求次數達到 60000 開始出現錯誤，表示已到達壓力點，測試結果如表 3。

表3. App Market 伺服器壓力測試結果

使用者數目	**重覆請求次數**	**總請求次數**	**平均值 (byte/sec)**	**處理量 (byte/sec)**	**錯誤率**
200	100	20000	0.496	352.5	0.0%
300	100	50000	0.742	364.7	0.0%
400	100	40000	0.996	358.7	0.0%
500	100	50000	1.265	360.4	0.0%
600	100	60000	1.438	378	3.76%

- 每一個 POST Request 為約 50Byte 的 HTML 表單格式。
- 平均值：回應每個 Request 需耗費時間的平均值，單位為千位元/秒
- 處理量：每秒鐘能回應的 Request 數目，單位為千位元/秒

在測試時，發送請求的客戶端與測試目標伺服器的系統資源使用情況，發現測試客戶端並無出現滿載現象。唯伺服器端的 CPU 使用率與記憶體有大幅提升之情況，錯誤率出現當下的記憶體已經到達完全滿載，使用率到達 98%，因回應過慢導致錯誤率出現。因此，若往後有更大量的服務需求，增加伺服器之記憶體為必要條件。

- 測試主機設備規格(App 源碼完整性驗證模組)

  測試工作全程將於 100Mbps 的區域網路環境下執行，並利用一台客戶端機器，對位於同一個區域網路內的測試伺服器進行效能測試。以下為此測試所使用的測試機器規格

✓ 伺服器規格：

- 作業系統：Windows 7 professional
- 處理器：Intel (R) Core (TM) i7 - 4790 @ 3.60 GHz
- 記憶體：8 GB
- 硬碟：SSD 128 GB、HDD 1024 GB

✓ 客戶端規格：

- 作業系統：Windows 7 professional
- 處理器：Intel (R) Core (TM) i5 - 650 @ 3.20 GHz
- 記憶體：10 GB
- 硬碟：HDD 500 GB

- 效能測試流程(App 源碼完整性驗證模組)

  我們以 JMeter 這套工具進行測試工作程式，同時送出多個 HTTP POST Request 模擬多個使用者將特定的 APK 相關資源特徵指紋送至 APVS，並同時記錄使用者數目(Threads)、每個使用者重覆請求次數(Loop Count)、總請求次數、平均值、處理量、總回應時間。本測試以兩個案例進行壓力測試，案例一假設每個使用者發出 1 次請求；案例二則為假設每個使用者重覆請求 100 次，我們將進行此兩個案例來測試 APVS 所能承受之最大請求數目。

- 壓力測試結果(App 源碼完整性驗證模組)

  測試的相關設定如下所示：

  - 用來進行壓力測試之 APK 檔案大小為 46.5 MB
  - 每一個請求所傳送至伺服器的資料集大小為 1.24 KB
  - 每個請求間的延遲時間設為 5 秒
  - 平均值：回應每個請求需耗費時間的平均值，單位為位元組/秒
  - 處理量：每秒鐘能回應的請求數目，單位為位元組/秒

案例一：將使用者重覆請求次數固定為 1，遞增使用者數目，以測試伺服器所能承受的多個使用者同時發出的請求數目。測試結果如表 4，由表中可得壓力點為總請求次數為 10000，因當總請求次數達到 10500 時，發現有錯誤率產生，部份請求無法正常被回應。

表4. App 源碼完整性驗證伺服器壓力測試結果(案例一)

使用者數目	重覆請求次數	總請求次數	平均值 (byte/sec)	處理量 (byte/sec)	錯誤率 (%)
100	1	100	0.061	18.85	0
500	1	500	1.02	72.87	0
1000	1	1000	4.54	72.27	0
5000	1	5000	32.20	70.12	0
7000	1	7000	45.15	69.09	0
10000	1	10000	63.00	69.16	0
10500	1	10500	64.36	70.35	2.62

案例二：將使用者重覆請求次數固定為 100，以測試伺服器所能承受的多個使用者同時發出的請求數目。測試結果如表 5，由表中可得當壓力點為總請求次數等於 6500，因當總請求次數達到 7000 時，發現有錯誤率產生，部份請求無法正常被回應。

表5. App 源碼完整性驗證伺服器壓力測試結果(案例二)

使用者數目	重覆請求次數	總請求次數	平均值 (byte/sec)	處理量 (byte/sec)	錯誤率 (%)
100	100	10000	1.29	74.05	0
500	100	50000	6.74	73.11	0
1000	100	100000	13.97	70.73	0
2000	100	200000	26.41	68.94	0
3000	100	300000	42.55	69.81	0
4000	100	400000	56.12	70.62	0
6000	100	600000	86.23	69.03	0
6500	100	650000	93.11	69.25	0
7000	100	700000	99.63	69.33	0.135

由上面之測試結果得知，在測試環境下的 App 源碼完整性驗證伺服器，當出現錯誤率時,由系統工作管理員可以觀察到CPU與記憶體資源耗用上升現象，因此，可推估在此測試條件下，若想增加 App 源碼完整性驗證伺服器所能承受之最大請求數目，建議可增加 CPU 與記憶體資源或是網路頻寬資源。

- 測試主機設備規格(App 資料流限制性驗證模組)

測試工作全程將於 100Mbps 的區域網路環境下執行，並利用一台客戶端機器，對位於同一個區域網路內的測試伺服器進行 App 檢測伺服器效能測試。以下，為本計畫測試所使用的各測試機器規格

✓ 伺服器規格：

- 作業系統：Windows 7 professional
- 處理器：Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
- 記憶體：8 GB
- 硬碟：Seagate 500GB 讀取:3886.44 MB/s 寫入:3809.31 MB/s

✓ 客戶端規格：

- 作業系統：Windows 7 professional
- 處理器：Intel (R) Core (TM) i5 - 650 @ 3.20 GHz
- 記憶體：8 GB
- 硬碟：Seagate 500GB 讀取:3886.44 MB/s 寫入:3809.31 MB/s

- 效能測試流程(App 資料流限制性驗證模組)

模擬前端使用者透過同時對 App 檢測伺服器送出多個網頁請求，這代表多個使用者對 App 檢測伺服器送出上傳 apk 檔案的請求，並同時記錄每個千位元的平均處理量、每個千位元的平均傳輸量及紀錄的上傳成功的比率。首先我們會先架設有 *n* 位使用者，各傳出 10 次請求開始，以遞迴地的方式向上累加遞增(n=10, 20, 30, …)，來測試伺服器所能承受之最大請求數目。

- 壓力測試結果(App 資料流限制性驗證模組)

進行 App 檢測伺服器效能測試前，需要訂定一個測試標準，並以此標準為基準，模擬 App 檢測伺服器在本計畫的測試伺服器環境下，多位使用者同時上傳檔案大小約 2MB 的 App 時之效能，以及監控機器效能檢測出該模組所需要消耗的最大資源為何？

測試結果如下表所示，在 100.0Mbps 的區域網路內，以上傳 App 的動作同時重複請求 10 次之結果。

表6. App 資料流限制性驗證模組壓力測試結果

使用者數目	重覆請求次數	總請求次數	平均值 (byte/sec)	處理量 (byte/sec)	錯誤率
10	10	100	255	24.6	0.0%
20	10	200	346	43.8	0.0%
30	10	300	508	45.0	0.0%
40	10	400	590	44.4	0.0%
50	10	500	937	44.6	0.0%

由上表之測試結果得知，在測試伺服器環境下的 App 檢測伺服器，當約 30 位使用者同時上傳 10 次 App 的情況下，伺服器的平均處理量會開始大量的遽增，且根據系統資源監視器顯示，我們可以看到硬碟已經出現了資源滿載的情況。因此，可推估在此測試條件下，若想讓 App 檢測伺服器有正常且不出錯的效能表現，其建議的使用者負荷量約為 30 人左右，若須提升 App 檢測伺服器(子五部分)的效能，將可參考下面的建議，依自身的需求調整伺服器的環境。

V. 增加伺服器實體記憶體
VI. 更換高讀寫速度的硬碟裝置

本份測試報告僅以範例的方式對 App 檢測伺服器進行效能測試，再透過給予建議的方式，協助使用者在架設相關服務時，可參考我們所提供的相關數據以自行評估所需的伺服器環境。

3.3 人員職責分配 (Personnel Responsibility Assignment)

表7. 人員職責分配表

Testing Activities	Personnel
Acceptance Testing (AT1)	藍袖瑛 陳心廉
Acceptance Testing (AT2)	藍袖瑛 吳書樂
Acceptance Testing (AT3)	藍袖瑛 陳昶廷 吳書樂
Integration Testing (IT1)	吳淩
Integration Testing (IT2)	蕭詔安 李濬志
Integration Testing (IT3)	謝欣余
Integration Testing (IT4)	駱建安
Integration Testing (IT5)	吳文傑 賴勁瑋
Integration Testing (IT6)	林子鈞
Integration Testing (IT7)	林子鈞 薛宇凡 邱維揚
Integration Testing (IT8)	莊祐軒 唐偲瑋 許勝翔 劉佳琳 Alexander Yohan
Integration Testing (IT9)	陳俊甫 劉子慶 蔡宗穎 鐘士昌

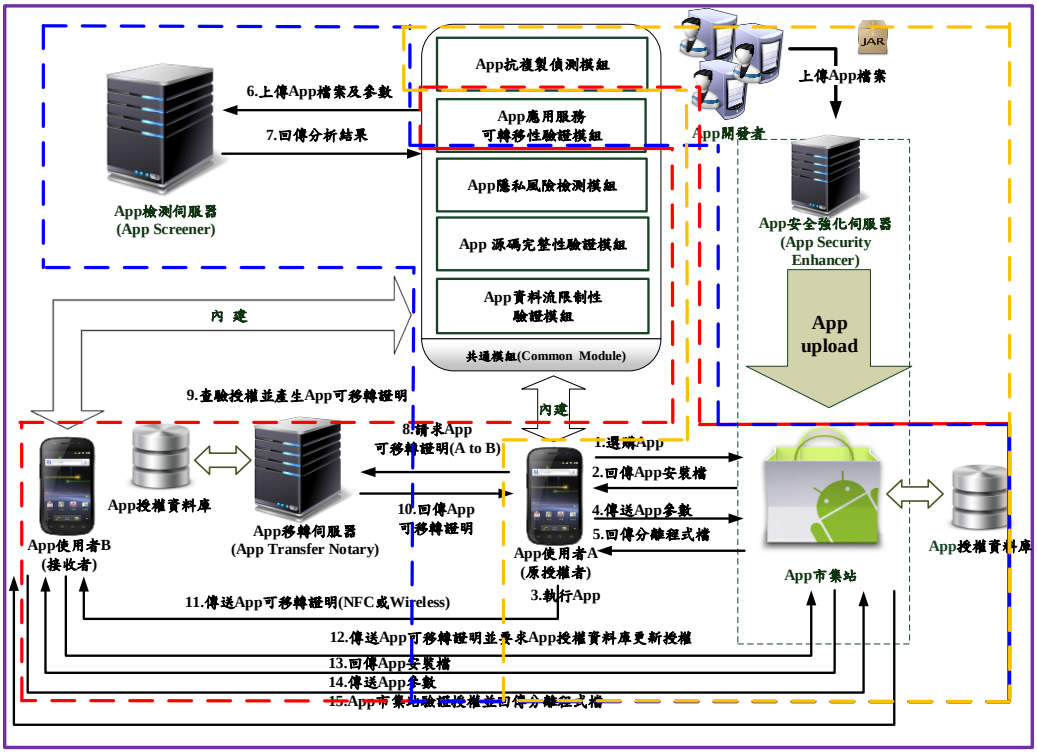
3.3.1 接受測試 (Acceptance Testing)

本計畫將進行使用者相關資訊進行系統測試，系統測試流程如下：

建立 App 市集站：選擇可信賴的 App 安全應用框架系統：App 使用者 A 持有行動裝置向 App 市集站提出下載 App 安裝檔的請求，以及驗證授權，App 複製攻擊之預防與偵測模組:透過 App 安全強化伺服器，市集站提供 App 使用者 A 回傳分離程式檔。

App 安全檢測：App 使用者 A 透過行動裝置內 App 應用服務可轉移性驗證模組、App 複製攻擊之預防與偵測模組、行動 App 隱私風險分析與防護模組、App 特徵產生與比對機制與模組、App 進階資料流分析模組，上傳 App 檔案至 App 檢測伺服器，並回傳分析結果。

App 移轉及驗證：App 使用者 A 向可信賴第三方請求 App 移轉證明，並向授權資料庫查詢授權並產生 App 可移轉證明，可信賴第三方將回傳 App 可移轉證明至 App 使用者 A。將以 NFC 或 Wireless 方式傳送 App 可移轉證明給 App 使用者 B，並向 App 市集站傳送 App 可移轉證明並更新授權，回傳 App 安裝檔至 App 使用者 B 後傳送 App 參數至 App 市集站，App 市集站將回傳下載 App 安裝檔及回傳分離程式檔 App 使用者 B。



Trusted App Framework

圖12. 可信賴的 App 安全應用框架系統流程

再者，我們進行系統接受測試，其次序為：AT1→AT2→AT3，AT1 驗證各子系統 App 應用服務可移轉性驗證及複製攻擊之預防與偵測可正確輸出並順利整合運作，AT2 分析各系統的隱私保護與進階資料流限制性驗證明確列出相關風險與分析，AT3 各子系統需正確執行並符合系統需求規格書[1]之預期測試結果。

TAF：可信賴的 App 安全應用框架

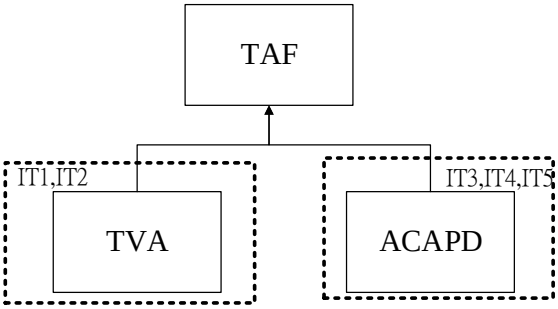
TVA：App 應用服務可移轉性驗證

ACAPD：App 複製攻擊之預防與偵測

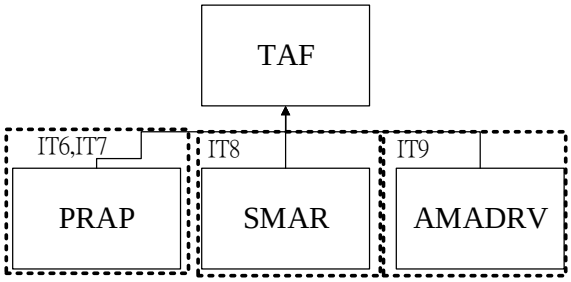
PRAP：行動 App 之隱私風險分析與防護

DETREP：行動應用程式重新包裝安全威脅與權限洩漏風險之防護

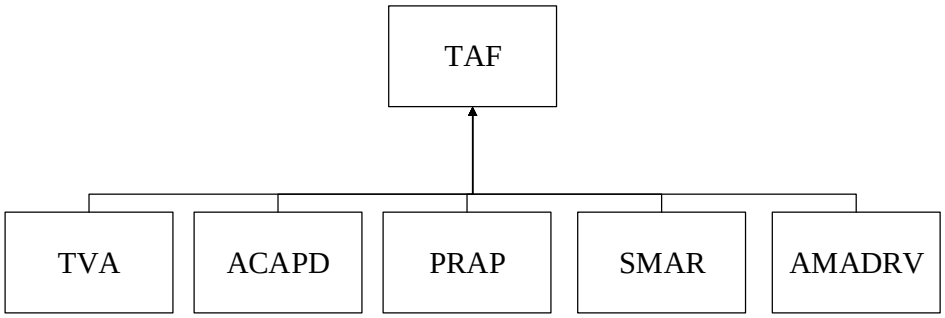
AMADRV：進階手機應用程式資料流限制驗證系統



(a)接受度測試 AT1



(b)接受度測試 AT2



(c)接受度測試 AT3

圖13. 可信賴的 App 安全應用框架系統整合與可接受度測試架構圖

4. 測試案例 (Test Cases)

4.1 整合測試案例 (Integration Testing Cases)

4.1.1 IT1 測試案例

目的：

(1) 驗證使用案例 1。

(2) 驗證使用者透過 Wi-Fi 移轉 App 可移轉證明。

(3) App 應用服務可移轉性驗證模組(TVA)能與可信賴第三方伺服器(Transfer Notary)連結。

表8. IT1 測試案例

Identification	IT1	
Name	TVA 整合至 Transfer Notary	
Tested target	App 應用服務可移轉性驗證模組	
Reference	TVA-EIR-002、TVA-EIR-003、TVA-EIR-004、TVA-IIR-004、TVA-IIR-005	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 開啟App應用服務可移轉性驗證模組	
	2. 使用者於選單選擇欲轉移之App並按下確定鍵	
		3. 系統顯示登入訊息
	4. 使用者輸入帳號、密碼及接受者帳號並按下確定鍵	
		5. 系統鑑別身分並回應登入成功與否
		6. 上傳 App
		7. 產生可移轉證明
		8. 傳送可移轉證明
	9. 進入初始化準備階段	
	10. 使用者使用(Wi-Fi)傳送到接收者的行動裝	

	置並配對裝置	
	11. 顯示傳輸完成訊息	
Expected result	當使用者成功登入，且與接收者的行動裝置配對裝置並傳送可移轉證明後，即顯示傳輸完成。	
Cleanup	無	

4.1.2 IT2 測試案例

目的：

(1) 驗證使用案例 2。

(2) 驗證授權次數限制、驗證授權時間區間。

(3) App 應用服務可移轉性驗證模組(TVA)能與可信賴第三方伺服器(Transfer Notary)連結。

表9. IT2 測試案例

Identification	IT2	
Name	TVA 整合至 Transfer Notary	
Tested target	App 應用服務可移轉性驗證模組	
Reference	TVA-EIR-002、TVA-EIR-003、TVA-EIR-004、TVA-IIR-004、TVA-IIR-005	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 開啟 App 應用服務可移轉性驗證模組	
	2. 使用者於選單選擇欲轉移之 App 並按下確定鍵	
		3. 系統顯示登入訊息
	4. 使用者選擇帳號、密碼及接受者帳號並按下確定鍵	
		5. 系統鑑別身分與授權次數限制，並回應登入成功與否
		6. 上傳 App
		7. 產生可移轉證明
		8. 傳送可移轉證明
	9. 進入初始化準備階段	
	10. 使用者使用 (Wi-Fi)	

	傳送到接收者的行動裝置並配對裝置	
	11. 顯示傳輸完成訊息	
Expected result	當使用者選擇身分鑑別資料後，可信賴第三方伺服器驗證授權次數限制成功，即顯示使用者登入完成	
Cleanup	無	

4.1.3 IT3 測試案例

目的：

(1) 驗證使用案例 3。

(2) App 安全強化伺服器可以成功產生並傳遞一組使用者個人金鑰給 App 授權使用者。

(3) App 安全強化伺服器成功加密分離程式檔並傳遞給 App 授權使用者。

表10. IT3 測試案例

Identification	IT3	
Name	整合執行至 ACAPD 模組	
Tested target	個人金鑰管理模組	
Reference	ACAPD-EIR-001、ACAPD-EIR-002	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者購買 App	
		2. 個人金鑰管理模組將該使用者加入授權子集，產生一組使用者個人金鑰，將成功訊息與使用者個人金鑰傳送給 App 使用者
	3. 使用者執行具動態載入模組之 App	
		4. 系統顯示使用者身分鑑別成功與否
		5. 系統顯示 App 動態載入成功
		6. 個人金鑰模組將分離程式檔進行廣播加密，傳遞給 App 使用者
Expected result	1. 行動裝置存有個人金鑰 2. 使用者啟用具動態載入模組之 App 3. 完成 App 來源鑑別後，需提供鑑別結果訊息，以	

	供使用者確認與執行 4. App 市集站必須可以透過安全通道將加密過的分離程式檔傳送給 App 使用者 5. App 使用者必須可以使用個人金鑰解密分離程式檔
Cleanup	無

4.1.4 IT4 測試案例

目的：

(1) 驗證使用案例 4。

(2) 使用者進行 App 來源鑑別。

(3) 非授權複製偵測模組將紀錄 App 使用者解密分離程式檔的過程，並產生追蹤紀錄回傳至開發者智財權追蹤模組。

表11. IT4 測試案例

Identification	IT4	
Name	整合追蹤與偵測至 ACAPD 模組	
Tested target	非授權複製偵測模組	
Reference	ACAP-FNR-001、ACAP-FNR-002	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者執行非授權複製偵測模組解密分離程式檔	
		2. 系統提交該 App 使用者追蹤紀錄至 App 市集站
		3. App 市集站回傳追蹤紀錄至 App 強化伺服器
Expected result	1. App 使用者執行 IPRDSM 模組時，將提交 App 使用者追蹤紀錄至 App 市集站 2. App 市集站回傳追蹤紀錄至 App 安全強化伺服器 3. App 安全強化伺服器回傳分析結果至 App 市集站	
Cleanup	清除解密過的分離程式檔	

4.1.5 IT5 測試案例

目的：

(1) 驗證使用案例 5。

(2) 開發者智財權追蹤模組執行動態洩密者追蹤，找出有洩密行為的 App 使用者並廢止其授權狀態。

表12. IT5 測試案例

Identification	IT5	
Name	整合廢止至 ACAPD 模組	
Tested target	開發者智財權追蹤模組	
Reference	ACAP-FNR-001、ACAP-FNR-002	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
		1. 開發者智財權追蹤模組執行動態洩密者追蹤
		2. 若偵測到 App 使用者非授權的行為，將停止 App 執行，並將該使用者從使用者子集中刪除
Expected result	當 IPRDSM 模組偵測到使用非授權複製之 App，將停止 App 執行，並廢止其授權狀態。	
Cleanup	無	

4.1.6 IT6 測試案例

目的：

(1) 驗證使用案例 6。

(2) 驗證模組可擷取欲檢驗隱私風險之行動裝置應用程式的內存資料庫資訊，並根據使用者設定之隱私篩選分級，傳送至隱私風險評估程序，根據分析結果警示使用者。

(3) PRAP 可與 App Screener 正確的互相收發資訊。

表13. IT6 測試案例

Identification	IT6	
Name	PRAP 和 App Screener 整合	
Tested target	PRAP [1.0.0]	
Reference	PRAP-UIR-001、PRAP -UIR-002、PRAP-EIR-001、PRAP-EIR-002、PRAP-IIR-001、PRAP-IIR-002、PRAP-IIR-003	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者執行客戶端程式	
	2. 使用者鍵入相關個人隱私資訊，進行資料篩選分級設定	
	3. 使用者選擇欲檢驗隱私風險之行動裝置應用程式	
		4. 內存資料庫檢索模組進行指定之應用程式內存資料庫資訊擷取
		5. 資訊送至隱私風險評估模組
		6. 將隱私風險評估分析結果，回傳並警示使用者其隱私風險程度
Expected result	當使用者送出含有個人隱私資訊的篩選分級資料，內存資料庫檢索模組擷取指定之應用程式內存資料庫資訊後送至 App Screener，App Screener 收到資訊後，進行隱私風險評估分析計算其風險值，將分析結果回傳至終端載具，警示使用者其隱私風險程度	

Cleanup	無

4.1.7 IT7 測試案例

目的：

(1) 驗證使用案例 7。

(2) 驗證模組可即時取得行動裝置通訊訊息並送至詐騙比對分析模組，根據分析結果判定是否警示使用者該通訊內容為詐騙。

(3) PRAP 可與 App Screener 正確的互相收發資訊。

表14. IT7 測試案例

Identification	IT7	
Name	PRAP 詐騙分析比對模組和 App Screener 整合	
Tested target	PRAP [1.0.0]	
Reference	PRAP-UIR-001、PRAP -UIR-002、PRAP-EIR-001、PRAP-EIR-002、PRAP-IIR-001、PRAP-IIR-002、PRAP-IIR-003	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者執行客戶端程式	
	2. 執行原始資料擷取模組，進行通訊訊息監測	
	3. 使用者執行通訊應用程式，進行聊天對話	
		4. 原始資料擷取模組進行通訊訊息擷取
		5. 訊息送至詐騙比對分析模組
		6. 進行詐騙比對分析
		7. 根據分析結果，判定是否為詐騙訊息與是否警示使用者
Expected result	當使用者收到或送出通訊訊息時，原始資料擷取模組擷取通訊訊息後，傳送至 App Screener，App Screener 收到通訊訊息後，進行詐騙比對分析，根據分析結果判定是否為詐騙，若確實有詐騙情形則回傳結果至終端載具並警示使用者	
Cleanup	無	

4.1.8 IT8 測試案例

目的：

(1) 驗證使用案例 8。

(2) 驗證 PRAP 詐騙分析比對模組之使用者案例各項流程是否被使用者所接受。

表15. IT8 測試案例

Identification	IT8	
Name	PRAP 個資分析比對模組和 App Screener 整合	
Tested target	PRAP [1.0.0]	
Reference	PRAP-UIR-001、PRAP -UIR-002、PRAP-EIR-001、PRAP-EIR-002、PRAP-IIR-001、PRAP-IIR-002、PRAP-IIR-003	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者執行客戶端程式	
	2. 使用者輸入個資與機敏資訊的關鍵字	
	3. 執行原始資料擷取模組	
		4. 原始資料擷取模組進行檔案文件資訊擷取
		5. 資訊送至個資比對分析模組
		6. 進行個資與機敏資訊評估
		7. 判定是否包含個資或機敏資訊
		8. 判定是否警示使用者
Expected result	當個資分析比對模組偵測到符合關鍵字的文件時，會將檔案內容後送至 App Screener，App Screener 收到資訊後判定是否為個資或機敏資訊，若確實包含個資或機敏資訊情形則回傳結果至終端載具並警示使用者	
Cleanup	無	

4.1.9 IT9 測試案例

目的：

(1) 驗證使用案例 9。

(2) 驗證使用者可驗證指定的 APK File 的完整性。

(3) AFDA 可與 APVS 正確的互相收發資訊。

表16. IT9 測試案例

Identification	IT9	
Name	AFDA 和 APVS 整合	
Tested target	DETREP [1.0.0]	
Reference	DETREP-FNR-001、DETREP-FNR-002	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者執行客戶端程式	
	2. 選擇欲驗證的 App，執行驗證動作	
		3. 客戶端程式針對被選取的 App 進行資料萃取，並且將資料和 APK 檔案傳至 App 檢測伺服器進行檢測
		4. App 檢測伺服器會先將 APK 檔案進行特徵指紋 (Feature Fingerprint)的產生，再根據資料庫中已經準備好的 App 相關資料與特徵指紋作為基準，與客戶端傳送過來的資料進行比對，並且回傳驗證結果
	5. 使用者可直接透過客戶端程式的介面得知 App 檢測伺服器回傳的結果來確認欲驗證之 App 是否經過竄改	
Expected result	當使用者送出正常、未經竄改的 APK 檔案，回傳結	

	果為綠燈；若該 App 於 App 檢測伺服器的資料庫中無相對應之比對基礎，故無法確認其完整性，結果為黃燈；若該 App 和資料庫中的樣本有異，表示已被竄改過，結果為紅燈；若該 App 和資料庫中的版號不同，表示 App 不為最新版，結果為藍燈
Cleanup	無

4.1.10 IT10 測試案例

目的：

(1) 驗證使用案例 10。

(2) 使用者透過進階資料流分析工具(AMADRV)將檢測檔案上傳至 App 檢測伺服器(ASS)，檢測伺服器需正確收到檔案。

(3) 使用者透過進階資料流分析工具(AMADRV)檢視分析結果，App 檢測伺服器(ASS)能正確地提供分析結果。

表17. IT10 測試案例

Identification	IT10	
Name	進階資料流分析工具(AMADRV)與 App 檢測伺服器(ASS)間之整合測試	
Tested target	[AMADRV 1.5.0]	
Reference	AMADRV-FNR-001、AMADRV-FNR-002	
Severity	Level 1 (Critical)	
Instructions	*Actor actions*	*System responses*
	1. 使用者透過 AMADRV 選擇欲檢測之檔案，並上傳。	
		2. ASS 收到檔案後，回傳上傳成功訊息，並開始進行資料流分析。
	3. 使用者欲檢視分析結果，選擇檢視分析紀錄。	
		4. ASS 依據使用者選擇的紀錄提供該分析結果。
	5. 結束，關閉 AMADRV 程式。	
Expected result	1. 使用者上傳檔案後，ASS 會正確收到該檔案。	

	2. 使用者可以正確檢視分析結果。
Cleanup	無

4.2 接受測試案例 (Acceptance Testing Cases)

4.2.1 AT1 測試案例

目的：

(1) 驗證 TVA、ACAPD 能依流程正確整合運作。

(2) App 安全強化伺服器成功加密分離程式檔並傳遞給 App 授權使用者。

(3) 確認 App 應用服務可移轉性驗證模組(TVA)能與可信賴第三方伺服器(Transfer Notary)連結。

表18. AT1 測試案例

Identification	AT1
Name	TVA、ACAPD 整合至 TAF
Tested target	TVA、ACAP
Reference	TVA-EIR-002 、 TVA-EIR-003 、 TVA-EIR-004 、 TVA-IIR-005 、ACAPD-EIR-001、ACAPD-EIR-002 、 ACAPD-FNR-001、ACAPD-FNR-002
Severity	Level 1 (Critical)
Instructions & Expected result	1. 使用者購買 App 2. 個人金鑰管理模組將該使用者加入授權子集，產生一組使用者個人金鑰，將成功訊息與使用者個人金鑰傳送給 App 使用者 3. 使用者執行具動態載入模組之 App 4. 系統顯示使用者身分鑑別成功與否 5. 系統顯示 App 動態載入成功 6. 個人金鑰模組將分離程式檔進行廣播加密，傳遞給 App 使用者 7. 開啟 App 應用服務可移轉性驗證模組 8. 使用者於選單選擇欲轉移之 App 並按下確定鍵 9. 進入初始化準備階段 10. 使用者使用(Wi-Fi)傳送到接收者的行動裝置並配對裝置 11. 顯示傳輸完成訊息
Cleanup	無

4.2.3 AT2 測試案例

目的：

(1) 驗證 PRAP、DETREP、AMADRV 能正確執行風險評估功能。

(2) 驗證模組可即時取得行動裝置通訊訊息並送至詐騙比對分析模組，根據分析結果判定是否警示使用者該通訊內容為詐騙。

(3) 驗證 PVDM 使用者案例各項流程是否被使用者所接受。

(4) 驗證使用者可驗證指定的 APK File 的完整性。

(5) 使用者透過進階資料流分析工具(AMADRV)將檢測檔案上傳至 App 檢測伺服器(ASS)，檢測伺服器需正確收到檔案。

(6) 使用者透過進階資料流分析工具(AMADRV)檢視分析結果，App 檢測伺服器(ASS)能正確地提供分析結果。

表19. AT2 測試案例

Identification	AT2
Name	PRAP、DETREP、AMADRV 整合至 TAF
Tested target	PRAP、DETREP、AMADRV
Reference	PRAP-UIR-001、PRAP -UIR-002、PRAP-EIR-001、PRAP-EIR-002、PRAP-IIR-001、PRAP-IIR-002、PRAP-IIR-003、PRAP-UIR-001、PRAP -UIR-002、PRAP-EIR-001、PRAP-EIR-002、PRAP-IIR-001、PRAP-IIR-002、PRAP-IIR-003、DETREP-FNR-001、DETREP-FNR-002、AMADRV-FNR-001、AMADRV-FNR-002
Severity	Level 2 (Important)
Instructions & Expected result	1. 使用者執行客戶端程式，並執行原始資料擷取模組，進行通訊訊息監測 2. 使用者執行通訊應用程式，進行聊天對話 3. 原始資料擷取模組進行通訊訊息擷取，訊息送至詐騙比對分析模組，根據分析結果，判定是否為詐騙訊息與是否警示使用者 4. 使用者執行客戶端程式，並使用者執行客戶端程式，選擇欲驗證的App，執行驗證動作 5. 使用者可直接透過客戶端程式的介面得知App檢測伺服器回傳的結果來確認欲驗證之App是否經過竄改 6. 使用者透過AMADRV選擇欲檢測之檔案，並上傳 7. 使用者欲檢視分析結果，選擇檢視分析紀錄
Cleanup	無

4.2.4 AT3 測試案例

目的：

(1) 在 AT1 與 AT2 測試成功的基礎上，再次重覆驗證 TVA、ACAPD、PRAP、DETREP 與 AMADRV 能正確地整合於 TAF 架構。

(2) 驗證 TAF 所有功能與流程能正確整合運作。

表20. AT3 測試案例

Identification	AT3
Name	TVA、ACAPD、PVDM、IAPSDE、MADRV 整合至 TAF
Tested target	TVA、ACAPD、PVDM、IAPSDE、MADRV
Reference	TVA-EIR-002、TVA-EIR-003、TVA-EIR-004、TVA-IIR-005 、ACAPD-EIR-001、ACAPD-EIR-002 、ACAPD-FNR-001、ACAPD-FNR-002、PRAP-UIR-001、PRAP -UIR-002、PRAP-IIR-001、PRAP-IIR-002、PRAP-IIR-003、PRAP-EIR-001、PRAP-EIR-002、DETREP-FNR-001、DETREP-FNR-002、AMADRV-FNR-001、AMADRV-FNR-002
Severity	Level 2 (Important)
Instructions & Expected result	1. 使用者購買 App 2. 個人金鑰管理模組將該使用者加入授權子集，產生一組使用者個人金鑰，將成功訊息與使用者個人金鑰傳送給 App 使用者 3. 使用者執行具動態載入模組之 App 4. 系統顯示使用者身分鑑別成功與否 5. 系統顯示 App 動態載入成功 6. 個人金鑰模組將分離程式檔進行廣播加密，傳遞給 App 使用者 7. 開啟 App 應用服務可移轉性驗證模組 8. 使用者於選單選擇欲轉移之 App 並按下確定鍵 9. 進入初始化準備階段 10. 使用者使用(Wi-Fi)傳送到接收者的行動裝置並配對裝置 11. 顯示傳輸完成訊息 12. 使用者執行客戶端程式，並執行原始資料擷取模組，進行通訊訊息監測 13. 使用者執行通訊應用程式，進行聊天對話 14. 原始資料擷取模組進行通訊訊息擷取，訊息送至詐騙比對分析模組，根據分析結果，判定是否為詐騙訊息與是否警示使用者 15. 使用者執行客戶端程式，並使用者執行客戶端程式，選擇欲驗證的App，執行驗證動作

	16. 使用者可直接透過客戶端程式的介面得知App檢測伺服器回傳的結果來確認欲驗證之App是否經過竄改 17. 使用者透過AMADRV選擇欲檢測之檔案，並上傳 18. 使用者欲檢視分析結果，選擇檢視分析紀錄
Cleanup	無

5. 測試結果與分析 (Test Results and Analysis)

5.1 整合測試案例 (Integration Testing Cases)

表21. 整合測試案例結果

Test Case #	Results (Pass/Fail)	Comment
IT1	Pass	測試通過
IT2	Pass	測試通過
IT3	Pass	測試通過
IT4	Pass	測試通過
IT5	Pass	測試通過
IT6	Pass	測試通過
IT7	Pass	測試通過
IT8	Pass	測試通過
IT9	Pass	測試通過
IT10	Pass	測試通過

5.2 接受測試案例 (Acceptance Testing Cases)

表22. 接受度測試案例結果

Test Case #	Results (Pass/Fail)	Comment
AT1	Pass	測試通過
AT2	Pass	測試通過
AT3	Pass	測試通過

Appendix A： 追朔表 Traceability

A.1. 子系統 vs. 測試案例 (Subsystems vs. Test Cases)

表23. Subsystems vs. Test Cases Traceability Table

Test Cases / Subsystems	IT1	IT2	IT3	IT4	IT5	IT6	IT7	IT8	IT9	IT10
TVA-EIR-002	✓	✓								
TVA-EIR-003	✓	✓								
TVA-EIR-004	✓	✓								
TVA-IIR-004	✓	✓								
TVA-IIR-005	✓	✓								
ACAPD-EIR-001			✓							
ACAPD-EIR-002			✓							
ACAPD-FNR-001				✓	✓					
ACAPD-FNR-002				✓	✓					
PRAP-UIR-001						✓	✓	✓		
PRAP-UIR-002						✓	✓	✓		
PRAP-IIR-001						✓	✓	✓		
PRAP-IIR-002						✓	✓	✓		
PRAP-IIR-003						✓	✓	✓		
PRAP-EIR-001						✓	✓	✓		
PRAP-EIR-002						✓	✓	✓		
DETREP-FNR-001									✓	
DETREP-FNR-002									✓	
AMADRV-FNR-001										✓
AMADRV-FNR-002										✓

A.2. 需求 vs. 測試案例 (Requirements vs. Test Cases)

表24. Requirements vs. Test Cases Traceability Table

Test Cases / Requirements	TVA	ACAPD	PRAP	DETREP	AMADRV
AT1	✓	✓			
AT2			✓	✓	✓
AT3	✓	✓	✓	✓	✓

Appendix B： 參考資料 (References)

[1] 國立台灣科技大學吳宗成教授研究團隊，可信賴的 App 安全應用框架需求規格書，民國一零四年十二月一日。