科技部資訊安全實務研發計畫系統測試報告

System Integration and Testing Document

先進駕駛輔助系統之雲端輔助設計優化(II)
子計畫三：植基於免疫系統之安全的物聯網應用開發方法 - 以先進駕駛輔助系統為例(II)

MOST 106-2221-E-156-001

主持人蘇維宗
真理大學資訊工程學系(所)

Department of Computer Science and Information Engineering
Aletheia University, Taiwan

2018/05/07

目錄

圖目錄

表目錄

版本更變記錄

修改日期	版本	文件狀態	描述	負責人
2018/05/07	1.0	Release	測試報告完稿確認	蘇維宗
2018/04/17	0.2	Beta	加入效能測試結果	陳韋丞
2017/12/01	0.1	Draft	初稿	陳韋丞

1. 介紹

本子計畫 106 年度主要的研究議題是如何讓資源有限的物聯網裝置(即車機)能夠將複雜的運算，例如屬性加密技術(Attribute-Based Encryption)，卸載到鄰近可信賴的裝置進行運算。為了提高行動裝置運算卸載的安全性，我們提出了一個可以從聯網裝置過去的行為來評估其可信賴度來選擇運算卸載的目標。根據運算卸載標的歷史行為(Historical Behavior of Thing)選擇運算卸載目標的優點在於可以透過數據分析來量化信賴度以提高運算卸載的安全性。

本測試計畫主要是對圖 1 中的依據信賴值評估之安全卸載方法(2.0 Secure Offloading under TRUst Evaluation，簡稱 SOTRUE)軟體模組進行功能與效能測試。本子計畫所開發之運算卸載軟體原始碼(beebit-seco-api 專案)已發布在 GitHub 平台上提供下載測試與使用，專案網址為 https://github.com/ucanlab/beebit-seco-api。

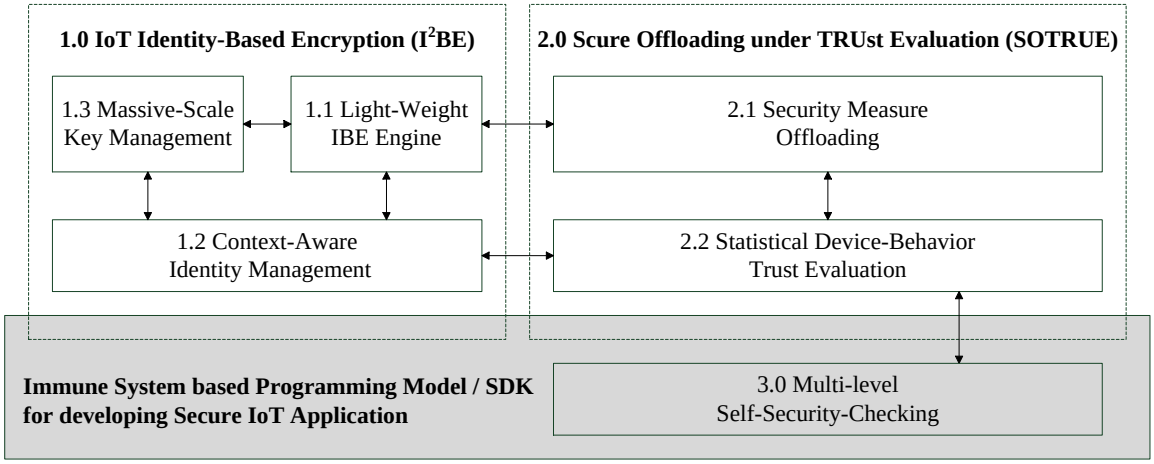

圖 1. 本計畫擬開發系統之架構圖(規劃三年完成)

1.1. 測試涵蓋範圍

本文件內容將依據系統需求規格書與系統設計文件，描述關於整合測試的相關計畫與內容。在確認本系統整合前，必須先確認所有的設計之子系統均能正確無誤的運作，因此著重於整合系統測試(Integration Test)及接受測試(Acceptance Test)，並透過此文件之描述與實踐，達到順利進行測試工作之目的。

1.2. 測試接受準則

本測試計劃需要滿足下列的測試接受準則：

- 本系統需要對所有列為必要(Critical、Important、Desirable)之需求作完整測試。
- 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預

期測試結果方能接受。

- 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規定相同。

2. 測試環境

2.1. 操作環境

本文件主要測試 IoT Device (以 Raspberry Pi 3 模擬)將密文政策屬性加密(CP-ABE)卸載至 Trust and Power Device (TP Device)執行的功能正確性與效能。圖 2 為測試環境部署圖，Device 將資料以 CP-ABE 加密後透過 MQTT 通訊協定傳送給 IoT Device。IoT Device 因為運算能力不足將解密運算卸載到 TP Device 進行運算。為了確保資料的保密性，TP Device 只會降低存取政策內的屬性數量而無法得知明文的內容。接著，TP Device 將轉換存取政策的密文回傳給 IoT Device。最後，因為密文中存取政策的屬性數量降低，所以 IoT Device 可以更有效率地進行解密。

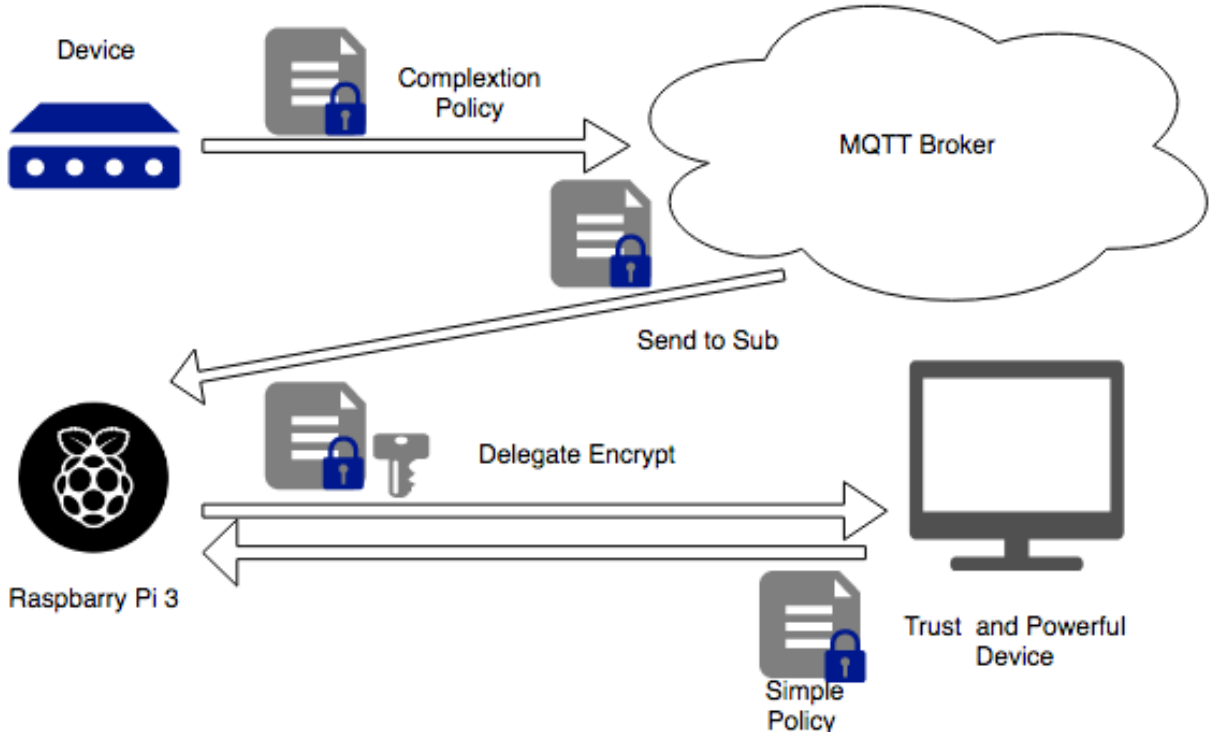

圖 2. 測試環境部署圖

2.2. 硬體規格

依據測試環境內容，關於測試環境所需的硬體規格說明，如表格 1 所示:

表格 1. 硬體規格

TP Device / PC	
CPU	Intel(R)Core(TM) i5-4200H CPU@ 2.80GHz 2.79GHz
RAM	8.00 GB
System type	64 bits
IoT Device / Raspberry Pi 3	
CPU	ARM Cortex-A7 (900 MHz)
RAM	1 GB
System type	64 bits

2.3. 軟體規格

依據測試環境內容，關於測試環境所需的軟體規格說明，如下所示:

- Linux (64 bits)與相關開發套件(如 gcc、make 等)
- 基本函式庫套件，包含 m4、flex、bison、libssl-dev、libgmp-dev(>4.0.0)與 glib(>2.0.0)
- 屬性加密函式庫套件(libpbc、libbswabe、與 cpabe)
- 本計畫開發之函式庫套件(libcpabe)

2.4. 測試資料

測試資料為字串。字串的內容與大小另於測試案例描述。

3. 測試時程、程序、與責任歸屬

3.1. 測試時程

根據專案執行規劃書計畫書，測試時程 107 年 3 月起至 107 年 5 月止，詳細時程說明如下。

- 時程
  - 各子功能之內部元件整合測試(107/2/26 ~ 107/3/11)
  - 系統功能測試(107/3/12 ~ 107/4/8)
  - 系統效能測試(107/4/9 ~ 107/4/22)
- 查核點
  - 各子功能之內部元件整合測試確認(107/3/12)
  - 系統功能測試確認(107/4/8)
  - 系統效能測試確認(107/4/22)

3.2. 測試程序

3.2.1. 子計畫驗證

各子功能之內部元件測試，由各子系統開發負責人員完成，在此我們著重於所有子功能完成後之整合測試。

3.3. 測試人員與責任歸屬

表格 2. 測試人員角色與責任

Testing Cases	Testing Target	Personnel
FT01	IoT Device加解密	陳韋丞
FT02	TP Device加解密	陳韋丞
FT03	IoT Device卸載解密	陳韋丞
PT01	資料量對加解密效能的影響	魏祥宇
PT02	存取政策內屬性個數對加解密謝能的影響	魏祥宇
PT03	資料量對IoT Device本地解密與卸載解密的效能影響	魏祥宇

4. 測試案例

4.1. 整合測試案例

4.1.1. FT01 功能測試案例

目的: 驗證 TP Device 可以執行 CP-ABE 加解密功能且獲得正確結果。

表格 3. TP Device 可以執行 CP-ABE 功能測試案例說明

Identification	FT01
Name	驗證 TP Device 可以執行 CP-ABE 加解密功能
Assumption	函式庫 libcpabe-01.a 已整合到 PC 且功能正常
Pre-condition	已產生所有需要之金鑰
Target	呼叫 ct = enc(pt)函式能夠正確加密明文 pt 並產生密文 ct 呼叫 pt = dec(ct)函式能夠正確解密密文 ct 並產生明文 pt
Test Object	ct = enc(公開金鑰路徑, BASE64 編碼明文字串, 存取政策) pt = dec(公開金鑰路徑, 私密金鑰路徑, BASE64 編碼密文字串)
Severity	Critical
Test Source	TS 1. 對”Hello”進行 BASE64 編碼產生的 BASE64 字串(pt)
Instructions	Step 1. 執行 ct = enc(pt) Step 2. 執行 pt = dec(ct) Step 3. 對 pt 進行 BASE64 解碼 Step 4. 確認解碼後的字串是否為 Hello
Test Result	TR 1. 解碼後的字串為 Hello [SUCCESS]

4.1.2. FT02 功能測試案例

目的: 驗證 IoT Device 可以執行 CP-ABE 加解密功能且獲得正確結果。

表格 4. IoT Device 可以執行 CP-ABE 功能測試案例說明

Identification	FT02
Name	驗證 IoT Device 可以執行 CP-ABE 加解密功能
Assumption	函式庫 libcpabe-01.a 已整合到 Raspberry Pi 3 且功能正常
Pre-condition	已產生所有需要之金鑰
Target	呼叫 ct = enc(pt)函式能夠正確加密明文 pt 並產生密文 ct 呼叫 pt = dec(ct)函式能夠正確解密密文 ct 並產生明文 pt
Test Object	ct = enc(公開金鑰路徑, BASE64 編碼明文字串, 存取政策) pt = dec(公開金鑰路徑, 私密金鑰路徑, BASE64 編碼密文字串)
Severity	Critical
Test Source	TS 1. 對”Hello”進行 BASE64 編碼產生的 BASE64 字串(pt)
Instructions	Step 1. 執行 ct = enc(pt) Step 2. 執行 pt = dec(ct) Step 3. 對 pt 進行 BASE64 解碼 Step 4. 確認解碼後的字串是否為 Hello
Test Result	TR 1. 解碼後的字串為 Hello [SUCCESS]

4.1.3. FT03 功能測試案例

目的: 驗證 IoT Device 可以將 CP-ABE 加解密功能卸載到 TP Device 運算且獲得正確結果。

表格 5. IoT Device 卸載 CP-ABE 加解密功能

Identification	FT03
Name	IoT Device 可以卸載 CP-ABE 加解密功能
Assumption	函式庫 libcpabe-01.a 已整合到 PC 且功能正常 beebit-seco 網頁服務在 TP Device 執行且功能正常
Pre-condition	已產生所有需要之金鑰
Target	IoT Device 呼叫 TP Device 上的/cpabe/enc/data 服務進行加密卸載 IoT Device 呼叫 TP Device 上的/cpabe/dec/data 服務進行解密卸載
Test Object	/cpabe/enc/data /cpabe/dec/data
Severity	Critical
Test Source	TS 1. 對"Hello"進行 BASE64 編碼產生的 BASE64 字串(pt)
Instructions	Step 1. IoT Device 呼叫 TP Device 上的/cpabe/enc/data 並以 pt 為參數，TP Device 加密後回傳密文 ct Step 2. IoT Device 呼叫 TP Device 上的/cpabe/dec/data 並以 ct 為參數，TP Device 解密後回傳明文 pt Step 3. 對 pt 進行 BASE64 解碼 Step 4. 確認解碼後的字串是否為 Hello
Test Result	TR 1. 解碼後的字串為 Hello [SUCCESS]

4.1.4. PT01 效能測試案例

目的：測試資料量對 CP-ABE 加解密效能的影響。

表格 6. 資料量對加解密效能的影響

Identification	PT01
Name	測試資料量對 CP-ABE 加解密效能的影響
Assumption	函式庫 libcpabe-01.a 已整合到 PC 與 Raspberry Pi 3 且功能正常
Pre-condition	已產生所有需要之金鑰
Target	呼叫 ct = enc(pt)函式能夠正確加密明文 pt 並產生密文 ct 呼叫 pt = dec(ct)函式能夠正確解密密文 ct 並產生明文 pt
Test Object	ct = enc(公開金鑰路徑, BASE64 編碼明文字串, 存取政策) pt = dec(公開金鑰路徑, 私密金鑰路徑, BASE64 編碼密文字串)
Severity	Critical
Test Source	TS 1. 10 KB 的資料/1 個屬性的存取政策 TS 2. 50 KB 的資料/1 個屬性的存取政策 TS 3. 100 KB 的資料/1 個屬性的存取政策 TS 4. 1000 KB 的資料/1 個屬性的存取政策 TS 5. 5000 KB 的資料/1 個屬性的存取政策 TS 6. 10000 KB 的資料/1 個屬性的存取政策
Instructions	Step 1. 在 IoT Device 上分別從 TS1 至 TS6 執行 ct = enc(pt)並記錄時間 Step 2. 在 IoT Device 上分別從 TS1 至 TS6 執行 pt = dec(ct)並記錄時間 Step 3. 在 TP Device 上分別從 TS1 至 TS6 執行 ct = enc(pt)並記錄時間 Step 4. 在 TP Device 上分別從 TS1 至 TS6 執行 pt = dec(ct)並記錄時間
Test Result	根據實驗結果繪製資料量與加解密時間的關係圖

4.1.5. PT02 效能測試案例

目的：測試存取政策中屬性數量對 CP-ABE 加解密效能的影響。

表格 7. 存取政策中屬性數量對加解密效能的影響

Identification	PT02
Name	測試存取政策中屬性數量對 CP-ABE 加解密效能的影響
Assumption	函式庫 libcpabe-01.a 已整合到 PC 與 Raspberry Pi 3 且功能正常
Pre-condition	已產生所有需要之金鑰
Target	呼叫 ct = enc(pt)函式能夠正確加密明文 pt 並產生密文 ct 呼叫 pt = dec(ct)函式能夠正確解密密文 ct 並產生明文 pt
Test Object	ct = enc(公開金鑰路徑, BASE64 編碼明文字串, 存取政策) pt = dec(公開金鑰路徑, 私密金鑰路徑, BASE64 編碼密文字串)
Severity	Critical
Test Source	TS 1. 10 個屬性的存取政策/1000KB 的資料 TS 2. 20 個屬性的存取政策/1000KB 的資料 TS 3. 20 個屬性的存取政策/1000KB 的資料
Instructions	Step 1. 在 IoT Device 上分別從 TS1 至 TS3 執行 ct = enc(pt)並記錄 CPU 使用率 Step 2. 在 IoT Device 上分別從 TS1 至 TS3 執行 pt = dec(ct)並記錄 CPU 使用率 Step 3. 在 TP Device 上分別從 TS1 至 TS3 執行 ct = enc(pt)並記錄 CPU 使用率 Step 4. 在 TP Device 上分別從 TS1 至 TS3 執行 pt = dec(ct)並記錄 CPU 使用率
Test Result	根據實驗結果繪製存取政策中屬性數量與加解密時 CPU 使用率的關係圖

4.1.6. PT03 效能測試案例

目的：測試資料量對於 IoT Device 本地解密與卸載解密效能的影響。

表格 8.資料量對於 IoT Device 本地解密與卸載解密效能的影響

Identification	PT03
Name	資料量對於 IoT Device 本地解密與卸載解密效能的影響
Assumption	函式庫 libcpabe-01.a 已整合到 PC 與 Raspberry Pi 3 且功能正常
Pre-condition	已產生所有需要之金鑰
Target	呼叫 ct = enc(pt)函式能夠正確加密明文 pt 並產生密文 ct 呼叫 pt = dec(ct)函式能夠正確解密密文 ct 並產生明文 pt IoT Device 呼叫 TP Device 上的/cpabe/enc/data 服務進行加密卸載 IoT Device 呼叫 TP Device 上的/cpabe/dec/data 服務進行解密卸載
Test Object	ct = enc(公開金鑰路徑, BASE64 編碼明文字串, 存取政策) pt = dec(公開金鑰路徑, 私密金鑰路徑, BASE64 編碼密文字串) /cpabe/enc/data /cpabe/dec/data
Severity	Critical
Test Source	TS 1. 1000 KB 的資料/30 個屬性的存取政策 TS 2. 5000KB 的資料/30 個屬性的存取政策 TS 3. 10000 KB 的資料/30 個屬性的存取政策
Instructions	本地解密 Step 1. 在 IoT 裝置上分別對 TS1 至 T3 執行 pt = dec(ct)並記錄時間為本地解密的時間 卸載解密 Step 2. IoT Device 呼叫 TP Device 上的/cpabe/dec/data 並分別對 TS1 至 T3 為參數並記錄時間。 Step 3. 在 TP Device 上分別對 TS1 至 TS3 執行 pt = dec(ct)並記錄時間 Step 4. 在 TP Device 上分別對 TS1 至 TS3 解密後的 pt 執行 ct = enc(pt)並在存取政策中止加入一個屬性並記錄時間。 Step 5.TP Device 將 ct 回傳給確 IoT Device 並記錄時間 Step 6. IoT Device 對 ct 執行 pt = dec(ct)並記錄時間 Step 7. 將 Step 2-6 的時間累計為卸載解密的時間
Test Result	根據實驗結果繪製資料量對於 IoT Device 本地解密與卸載解密時間的關係圖

5. 測試結果與分析

5.1. 功能測試結果與分析

Test Case	Result (Pass / Fail)	Comment
FT01	Pass	無
FT02	Pass	無
FT03	Pass	無
Rate	100%	

5.2. 效能測試結果與分析

Test Case	Expected Result	Test Result	Comment
PT01	N/A	Pass	詳細測試結果請參閱圖 3 IoT Device 與 TP Device 效能差了將近 4 倍
PT02	N/A	Pass	詳細測試結果請參閱圖 4 IoT Device 在屬性數量高時 CPU 使用率過高可能影響其它工作的正常運行
PT03	N/A	Pass	詳細測試結果請參閱圖 5 資料量太大時因為傳輸時間太長導致卸載解密時間反而更長

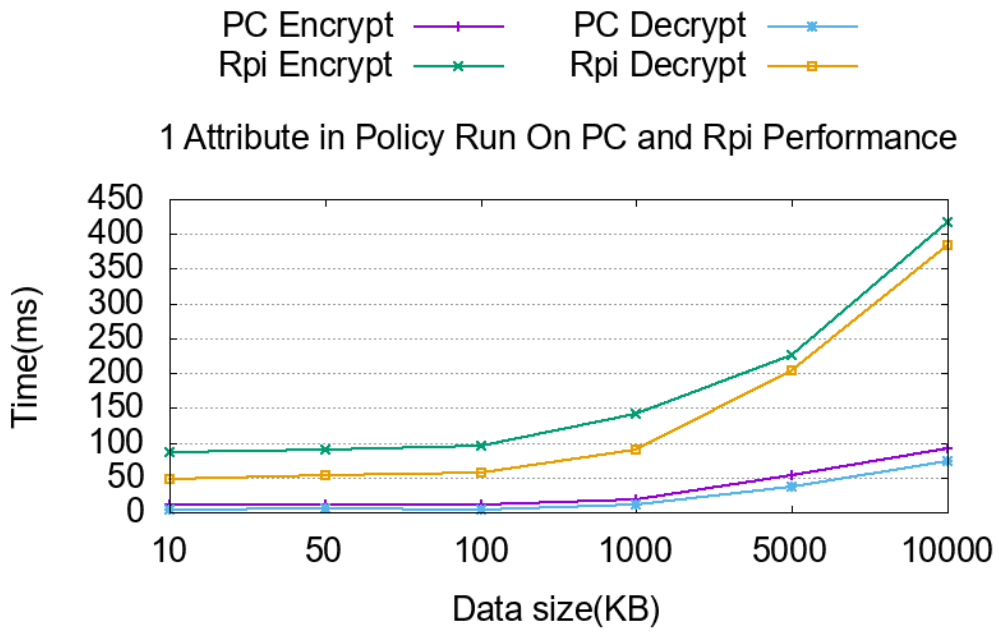

圖 3. 資料量與加解密時間的關係圖

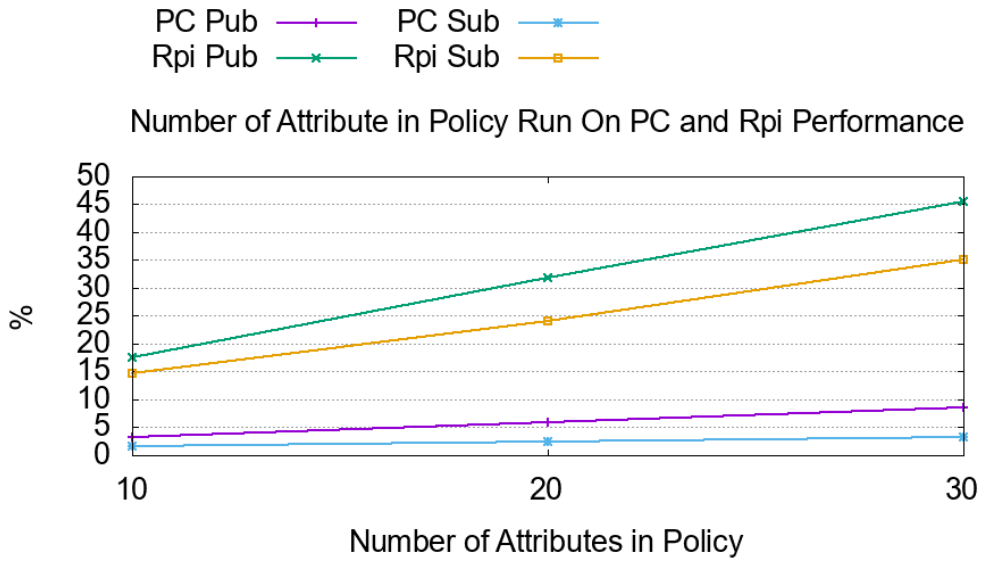

圖 4. 存取政策中屬性故數與加解密時 CPU 使用率關係圖

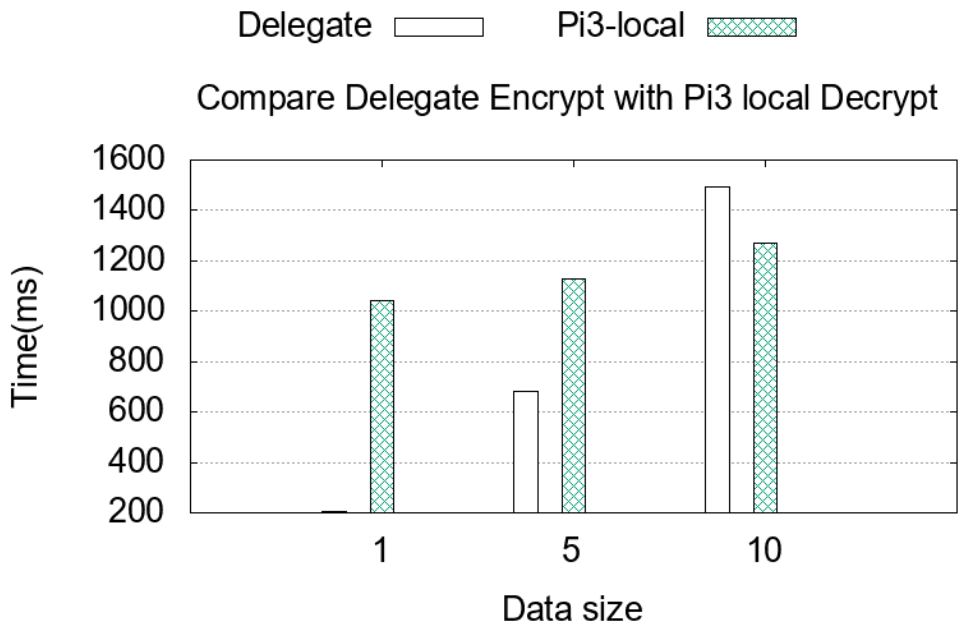

圖 5. 資料量與 IoT Device 本地解密與卸載解密時間的關係圖

附錄 A: 追溯矩陣

表格 9. 系統模組與測試案例的追溯矩陣

Test Cases / Module	FT01	FT02	FT03	PT01	PT02	PT03
2.1 Security Measure Offloading	◎	◎	◎	◎	◎	◎
2.2 Statistical Device-Behavior Trust Evaluation			◎			◎