Regular Paper

Communications of the CCISA
Vol. 30 No. 3 Aug. 2024

物聯網之智慧應用系統中加密金鑰保護機制的強化與實作

郭崇韋[1*]、魏巍[2]、林駿璋[3]、洪宇義[4]、劉嘉瑞[5]
[1,2,3,4,5]逢甲大學資訊工程學系
[1]cwkuo@mail.fcu.edu.tw、[2] weiking1021@gmail.com、[3]M1205003@o365.fcu.edu.tw、
[4]M1221097@o365.fcu.edu.tw、[5]D0909002@o365.fcu.edu.tw

摘要

隨著第五代行動通訊技術(5G)的穩定發展，物聯網(Internet of Things, IoT)的應用日益廣泛，許多環境中透過物聯網裝置來提升工作效率和生活品質。然而，在公開且未受保護的環境中，這些裝置往往承載著敏感個人資料，並面臨來自旁通道攻擊(Side-channel Attacks, SCA)的嚴重威脅。微控制器在進行加密運算時，可能會無意間釋放出特徵電磁訊號，這些訊號若遭到攻擊者截取並分析，將可能導致加密金鑰的洩漏，進而造成敏感資訊的暴露。為了解決這一問題，本文提出了一種輕量級的 AES-128 加密金鑰保護機制，該機制能有效運行於物聯網微控制器中，並增強其抵抗旁通道攻擊的能力。在本文中，我們利用 Wi-Fi 無線模組和 Arduino UNO 開發了一個具備無線通訊功能的智慧門禁卡管理系統，模擬門禁卡在 RFID 讀取器感應過程中遭到 ID 竊取的情境。通過設計跳動式變更金鑰機制，該方案能夠隨時間動態更新金鑰，有效抵禦以功率分析為基礎的旁通道攻擊，從而保障加密金鑰的安全性。實驗結果證明，本機制能顯著提升物聯網裝置在智慧應用系統中的安全性，減少潛在的資料洩漏風險。

關鍵詞：第五代行動通訊技術(5G)、旁通道攻擊(SCA)、AES-128、Wi-Fi、Arduino UNO、無線射頻辨識(RFID)

* 通訊作者 (Corresponding author.)

Regular Paper

Communications of the CCISA
Vol. 30 No. 3 Aug. 2024

Enhancement and Implementation of Encryption Key Protection Mechanisms in Smart Applications of IoT Systems

Chung-Wei Kuo[1*], Wei Wei[2], Chun-Chang Lin [3], Yu-Yi Hong [4], Jia-Ruei Liu[5]
[1,2,3,4,5]Department of Information Engineering and Computer Science, Feng Chia University
[1]cwkuo@mail.fcu.edu.tw, [2] weiking1021@gmail.com, [3]M1205003@o365.fcu.edu.tw,
[4]M1221097@o365.fcu.edu.tw, [5]D0909002@o365.fcu.edu.tw

Abstract

The advent of fifth-generation mobile communication (5G) technology has facilitated the proliferation of Internet of Things (IoT) applications, which have become pervasive across diverse settings, enhancing efficiency and quality of life. However, in open and unprotected environment, these devices often carry sensitive personal data, rendering them susceptible to significant risks posed by side-channel attacks (SCA). It is possible that microcontrollers which are performing encryption operations may unintentionally emit characteristic electromagnetic signals. Should these signals be intercepted and analyzed by an unauthorized third party, the encryption keys they contain could be compromised, resulting in the leakage of sensitive information. To address this issue, we propose a lightweight AES-128 encryption key protection mechanism that can be effectively implemented on IoT microcontrollers, thereby enhancing their resistance to side-channel attacks. In this study, we developed a smart access control Radio Frequency Identification (RFID) management system equipped with wireless communication capabilities, utilizing a Wi-Fi module and Arduino UNO. This system simulates scenarios in which identity theft occurs during the RFID card sensing process. By designing a dynamic key-hopping mechanism, our solution enables the encryption key to be periodically updated, effectively resisting power analysis-based side-channel attacks and ensuring the security of the encryption key. The experimental results demonstrate that our mechanism significantly improves the security of IoT devices within smart application systems, thereby reducing the risk of potential data leakage.

Keywords: 5G, Internet of Things, Side-channel attacks, AES-128, Wi-Fi, Arduino UNO, Radio Frequency Identification

壹、前言

物聯網(Internet of Things, IoT)以及無線通訊技術的快速發展為資訊和通訊技術(Information and Communication Technology, ICT)帶來巨大的影響，許多場域逐漸開始採用 IoT 裝置來創造更多應用，為人民帶來更便利的生活。智慧都市、智慧工廠以及智慧居家便是使用 IoT 大幅提升人類生活品質的例子。根據 statista 指出[1]，連網 IoT 的數量將在今年達到 18 億，預計 2033 年將會有 39.6 億。家電以及相關裝置透過 Wi-Fi 無線傳輸資訊到智慧聯網中樞，讓使用者能夠透過智慧型裝置來一舉操控家裡的智慧聯網設備，其中卻出現資安的隱憂。

在無線通訊過程中，若未對傳輸的資料進行加密，敏感資訊可能會因為訊息被攔截而被輕易洩漏。因此，使用強度足夠的加密演算法來保護資料已成為必然選擇。現行的進階加密標準(Advanced Encryption Standard, AES)和 RSA 加密演算法均屬於安全性高的加密技術，即便密文遭到攔截，攻擊者仍需經歷高成本且耗時的破解過程。然而，旁通道攻擊(Side-Channel Attack, SCA)卻能繞過這些加密演算法的數學複雜度，利用加解密裝置無意間洩漏出的物理特徵訊號，例如功率消耗[2]、電磁輻射[3]和熱能[4]等，來推測加解密金鑰，進而取得明文。

SCA 的研究主要集中在微控制器層面，通過 masking[5]或 hiding[6]技術來減少物理特徵的可觀察性，從而提高對 SCA 的防護能力。然而，隨著技術的進步，這些傳統的防護措施已經難以應對日益複雜的 SCA 攻擊[7]。尤其是在量子計算技術的發展下，現有的防護機制面臨著更大的挑戰。面對這些威脅，急需設計出新的防護策略，讓硬體安全防護跟上時代的進步。

近期出現一篇探討 Wi-Fi 環境下 SCA 的研究指出[8]，攻擊者無需掌握密碼即可破解使用者輸入的資料，顯示了當前 SCA 對資訊安全的嚴重影響。因此，在物聯網裝置設計中，加強硬體層面的安全性變得愈發重要。由於 IoT 裝置的運算能力與電源通常受到限制，AES-128 因其低運算與功耗需求[9]，成為 IoT 裝置的首選加密演算法。AES-128 不僅能在有限的資源下提供足夠的加密強度[10]，還能降低設備的功耗。然而，單純依靠加密演算法來提升安全性仍然不足，從硬體層面加強對 SCA 的防護至關重要。

如果智慧家庭中的 IoT 裝置遭受 SCA 攻擊[11]，大量敏感資料可能會被不法分子竊取，例如門禁系統或監控設備中的隱私資料。因此，本研究針對 IoT 裝置的加密通訊進行改良，旨在提高這些裝置在公開環境下運行時的安全性。我們基於 Arduino UNO 和 ESP8266 Wi-Fi 模組構建了一個 IoT 傳輸系統[12]，模擬開放環境中的加密通訊行為，並通過探棒擷取訊號軌跡(Trace)，對其進行 SCA 攻擊。在實驗中，我們使用 GPU 平台對擷取到的訊號進行統計分析，並在一定數量的軌跡下成功破解了 AES 加密金鑰。這表明，單一固定金鑰的 AES 加密不適用於公開環境的 IoT 裝置，必須透過機制來動態變更 AES 金鑰以加強安全性。

本研究藉由不斷更換加密演算金鑰的概念，也就是在特定時機將演算法使用的金鑰

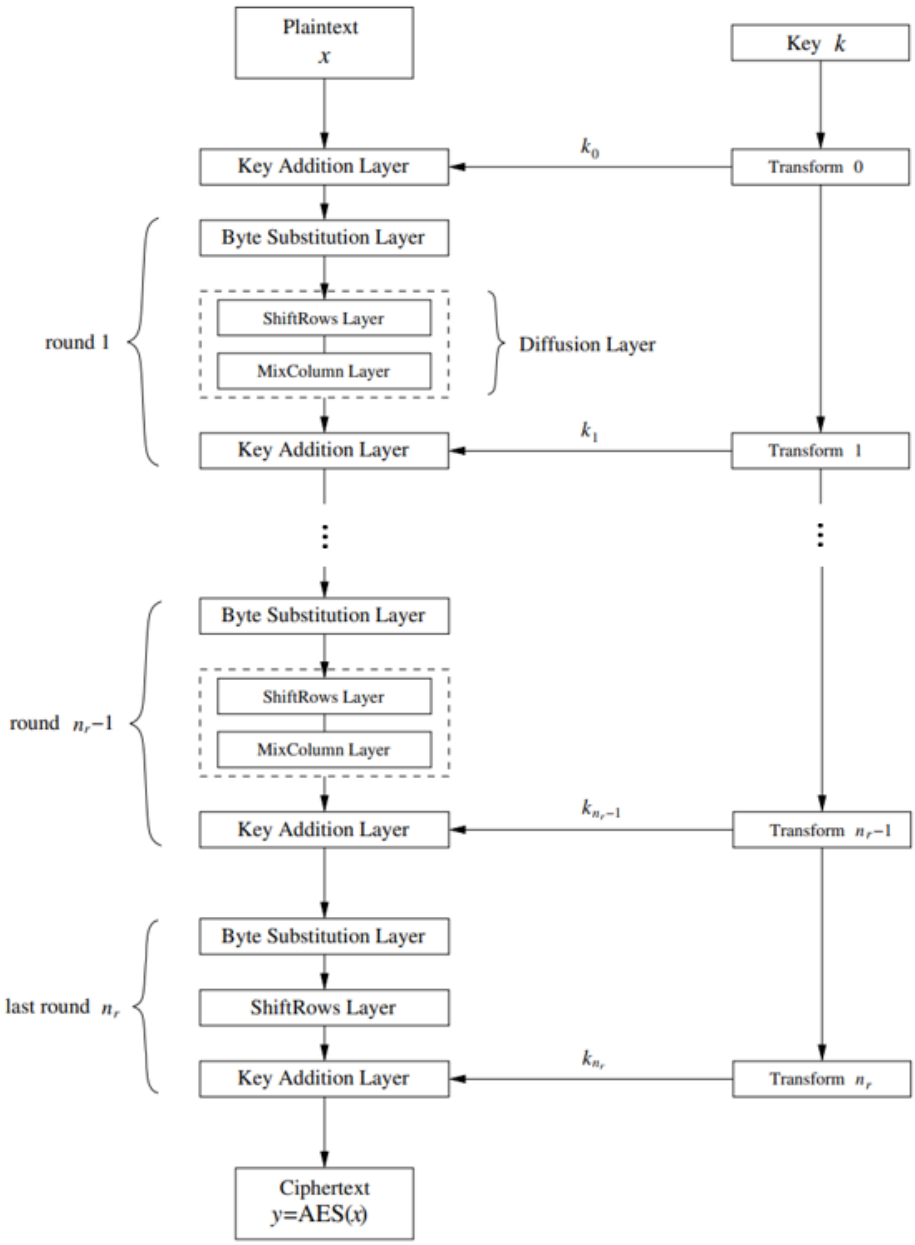
進行替換，並結合迪菲-赫爾曼金鑰交換協定（Diffie-Hellman key exchange, D-H）進行金鑰替換。此機制能在 SCA 成功破解前動態更換 AES 金鑰，顯著提升加密通訊的安全性。

貳、文獻探討

2.1 進階加密標準(Advanced Encryption Standard, AES)

進階加密標準(Advanced Encryption Standard, AES)於 1998 年首次發表[13]，並在 2001 年成為對稱加密演算法的標準，取代了傳統的資料加密標準(Data Encryption Standard, DES)，相較於 AES 大量使用的 128 位元金鑰，後者僅支持 56 位元的金鑰在安全上略顯不足，因此 AES 運算效率高且安全性強，成為 IoT 領域中最常使用的加解密演算法。AES 的加密過程包括以下步驟，首先是 AddRoundKey()，將明文和原始金鑰進行 XOR 運算後，接著進行 SubBytes()，將 AddRoundKey()的結果當作索引值，使用 S-box 進行查表並且一對一替換訊息，再來使用 ShiftRows()以及 MixColumns()將訊息以行或列進行位移轉換，藉此增加加密複雜度，流程如圖一所示。這一連串的步驟將會依照金鑰位元數的不同進行 N 回合，分別是 AES-128 的 10 回合、AES-192 的 12 回合以及 AES-256 的 14 回合[14]，最後一回合會省略 MixColumns()。這三者的差別在於加密時使用金鑰長度的位元數導致加密回合數的不同，藉此將訊息達到足夠的混淆程度。其中由於運算效能以及 IoT 裝置功耗限制上的關係，使用 128 bits 長度金鑰的 AES-128 最常在許多輕量級系統或 IoT 裝置上看到。

以 AES-128 為例，加密的金鑰長度為 128bits，也就是 16bytes。因此金鑰會被分段為 16 個字節，每個字結長度為 1byte (8 bits)的金鑰段並由 16 進制表示成 2 位數值，範圍是 00 到 FF，總共有 256 種可能。首先明文在傳入加密系統時會先依照不同情形轉為對應的 16 進制值，在 AddRoundKey()過程中，4 階矩陣的金鑰矩陣與 128bits 組成的 4 階矩陣明文進行互斥或運算，得到擴展金鑰。SubByte()將擴展金鑰的每個矩陣值對照 S-box 的表格進行替換，ShiftRow()將替換完的矩陣進行行位移，位移方式如下，第 0 列不位移、第一列往左循環位移一位、第二列往左循環位移二位以及第三列往左循環位移三位。MixColumn()將移位後的矩陣與特定矩陣進行相乘，並得到混淆後的矩陣，其中矩陣相乘是基於有限域 GF(2^8)的二元運算，再將二元運算結果進行互斥或運算，因此數值會永遠在 0 到 255 之間。



圖一：AES 加密流程圖

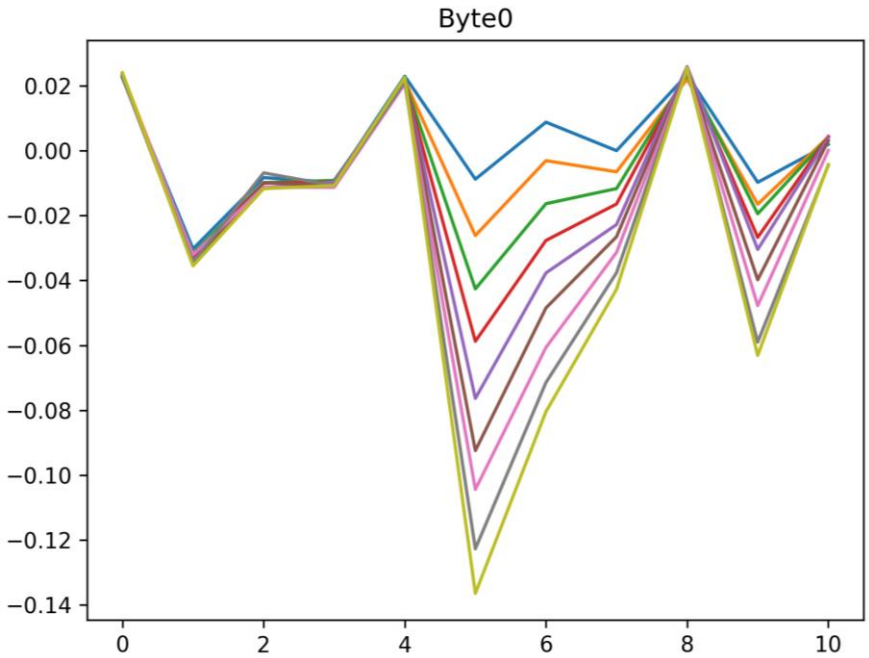
雖然在 2023 年底 NIST 選擇 ASCON 作為新一代的輕量級密碼標準[15]，成為下一個世代的 IoT 加密標準。然而，對於同期的加密演算法效能比較，ASCON 並非最快，但當時的比較基準包含演算法的安全性、在低功耗 FPGA 或微控制器的效能以及對 SCA 攻擊的抵抗程度，基於這些評判標準，ASCON 才得以在眾多演算法當中被選為標準。ASCON 包含 AEAD 以及 SHA-256，以低功耗的優勢保護微型裝置的短時效性資料。但其防護強度並不優於 AES-128，ASCON 明確表示保護對象為短時效性資料，與本研究所要保護的資料形式並不相同，市面上大多數微控制器已使用配置完成的 AES、DES、

3DES 以及 RSA 加密演算法，其中又以 AES 為大宗。因此本研究著重於 AES-128 加密演算法的問題探討以及改良，藉此符合現實情境。

2.2 旁通道攻擊

SCA 是針對加密演算法的非侵入式攻擊技術，通過分析加解密過程中洩漏的物理特徵來推測金鑰。常見的 SCA 技術包括簡單功率分析(Simple Power Analysis, SPA)[16]、差分功率分析(Differential Power Analysis, DPA)[16]和相關功率分析(Correlation Power Analysis, CPA)[17]。本研究採用 CPA 作為主要的攻擊手段，因為它能有效分析加密裝置的功率消耗與金鑰之間的相關性。

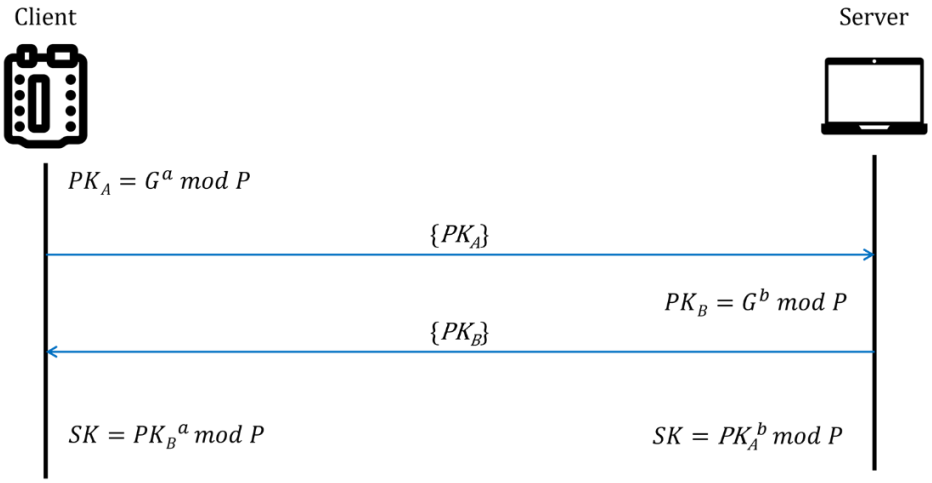
在 AES 加密過程中，攻擊者通常選擇第一輪加密中的 S-box 輸出作為攻擊點，因為此處的訊號僅經過 AddRoundKey()和 SubBytes()運算。攻擊者會將加密過程中的電磁訊號錄製成二進制數值的 Trace，並通過統計方法進行後續分析。CPA 通過計算假設金鑰與實驗測得 Trace 之間的相關性來推測正確的金鑰，由於電磁訊號與功率消耗呈現相關性，因此 CPA 依照功率消耗將 Trace 進行分群動作。用來分群的功率模型是漢明權重(Hamming Weight, HW)[18]，HW 依據每條 Trace 的二進制值中 1 的數量來進行分群，AES-128 的 HW 將會分為 0 到 8，總共 9 群，分群結果如圖二所示結果。依照上述方式先取得隨機明文與所有金鑰的中間值 x (第一輪 S-box 的輸出)，再將加密過程錄製的 Trace y 進行公式 1 計算相關係數 r，就可以依照相關係數的結果來猜測待測物(Device Under Test, DUT)的金鑰。



圖二：Hamming Weight 分群結果

2.3 迪菲-赫爾曼金鑰交換 (D-H)機制

D-H 是一項發表於 1976 年的安全協定[19]，他是現代密碼學的一個重要里程碑，因為 D-H 是公開密鑰的先驅，使得通訊雙方不用預先共享密鑰，也能透過公開通道傳送相關資訊協商作為安全通道之對稱金鑰使用，許多通訊協定在最初都會使用到 D-H 在公開通道交換訊息以此建立安全通道。D-H 有其他多種變種如橢圓曲線迪菲-赫爾曼密鑰交換 (Elliptic Curve Diffie-Hellman Key Exchange)，考量 IoT 情境硬體效能有限條件下，選用的 D-H 不能佔用太多功耗，因此本研究使用簡易版本進行實作。簡易版本的 D-H 使用離散對數$f(x) = G^x \bmod P$問題，首先定義基數 G 以及模數 P，IoT 裝置端及 Server 端分別選定私鑰 a 及 b 後，透過計算$PK_A = G^a \bmod P$及 $PK_B = G^b \bmod P$產生出公鑰PK_A及PK_B後在非安全通道上進行傳輸，再將彼此收到的公鑰計算$SK = {PK_B}^a \bmod P$及 $SK = {PK_A}^b \bmod P$，最後求得共同的對稱金鑰 SK。如圖三所示，若要透過公開內容得出最後的對稱金鑰 SK，則必須先解出密鑰 a 或 b 。D-H 通過雙方選擇私鑰並生成對應的公鑰進行交換，最終在非安全通道上計算出共同的對稱金鑰，攻擊者未能解出私鑰，如此提供了強而有力的安全保障，這種機制在本研究中被用於實現於動態金鑰交換以抵禦 SCA。



圖三：簡易 D-H 協定

2.4 物聯網常用的無線通訊技術

在 IoT 環境中，常見的無線通訊技術包括 Wi-Fi[20]、Bluetooth[21]、Zigbee[22]和 LoRa[23]等。這些技術各自具有不同的特性和優勢，根據應用場景和需求的不同進行選擇，以下將一一介紹不同無線傳輸技術的特性。

Wi-Fi：Wi-Fi 是一種高頻寬、高速率的無線通訊技術，通常運行於 2.4 GHz 或 5 GHz

頻段。其優勢在於能夠支持高吞吐量的資料傳輸，且具備廣泛的基礎設施支援，使其在智慧居家、智慧辦公室等需要穩定且高速數據連接的 IoT 場景中廣泛應用。然而，Wi-Fi 的能耗相對較高，且在開放環境下容易受到旁通道攻擊，對資料傳輸的安全性提出了挑戰。

Bluetooth：Bluetooth 是一種低功耗、短距離的無線通訊技術，適用於可穿戴設備、智慧家電等需要低能耗的 IoT 場景。Bluetooth 的資料傳輸速率較低，但其功耗表現優異，特別適合需要長期運行的小型設備。然而，由於傳輸距離的限制，Bluetooth 不適合需要大範圍覆蓋的應用場景。

Zigbee：Zigbee 是一種低功耗、低數據速率的無線通訊技術，專為低功耗設備設計，通常運行於 2.4 GHz 頻段。Zigbee 的主要優勢在於其網狀網絡(Mesh Network)能力，使其非常適合大規模、多節點的 IoT 網絡，如智慧建築和工業自動化。然而，Zigbee 的數據傳輸速率較低，不適合需要高速數據傳輸的應用場景。

LoRa：LoRa (Long Range)是一種長距離、低功耗的無線通訊技術，適用於需要覆蓋大範圍但資料傳輸需求不高的 IoT 應用，如智慧農業或遠端監控。LoRa 的主要優勢在於其能夠覆蓋數公里範圍，且能耗極低，但資料傳輸速率相對較慢，不適合需要即時傳輸大量資料的應用。

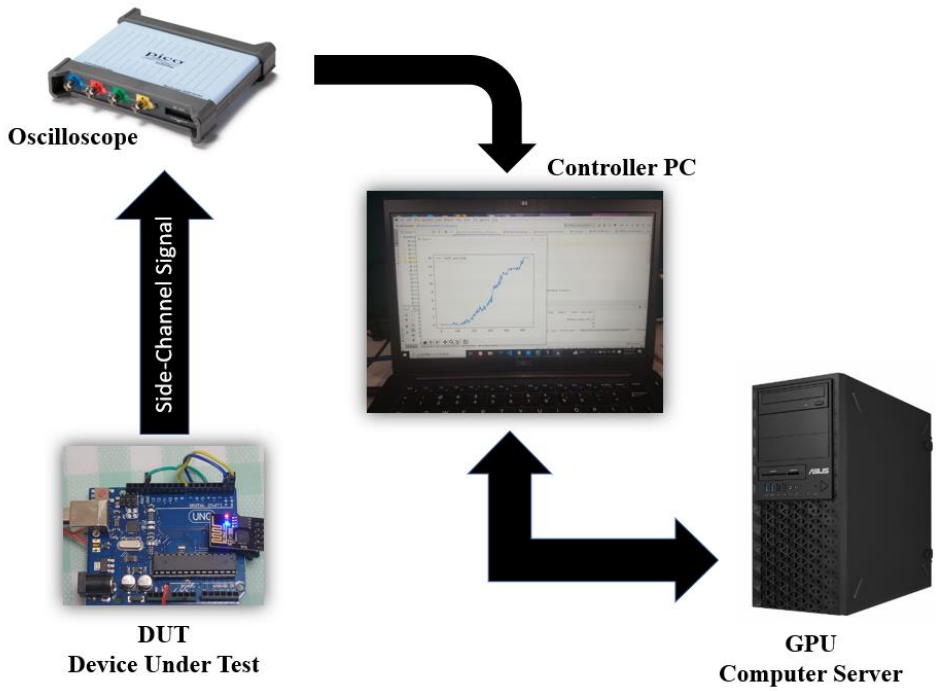
上述的各種通訊技術，以 Wi-Fi 最為普及、穩定和相對較高的數據傳輸速率常用於智慧家庭之中，Wi-Fi 已經成為最常見的無線通訊技術，並且許多現有設備和基礎設施已經針對 Wi-Fi 進行了優化和整合。此外，智慧家庭環境通常涉及大量的即時資料傳輸和多媒體應用，這些都要求通訊技術具有較高的帶寬和穩定性。雖然 Wi-Fi 在開放環境下面臨較多的網路攻擊的情境，尤其是在處理個人隱私的敏感資料時，但這也正是本研究探討的重點之一，通過改進加密金鑰保護機制來增強 Wi-Fi 通訊過程的資料安全性，抵禦 SCA 的威脅。

參、研究方法

3.1 實驗架構

網路環境中進行傳輸時，為了避免機密資訊遭到竊取，傳送端需要將機密資訊進行加密後再傳輸，在硬體條件有限的物聯網之中，對稱式的 AES 加密廣為採用，然而，AES 雖具有良好的安全加密機制，但原始的加密運算，是透過設備中的微控制器，因此，加密過程會洩露出跟運算行為相關的物理特徵。這時候對微控制器進行攻擊，就有可能從中取得加密金鑰，進而破解加密訊息。為了防止 SCA 在 IoT 裝置的攻擊行為，我們提出了基於動態金鑰替換之 AES 加密演算法，在 SCA 尚未攻擊成功時，將用來加密的初始金鑰進行變換。為了驗證防禦效果，我們實作了一個智慧門禁系統的無線通訊應用

平台，其中包含門禁卡感應系統(Arduino UNO)、Wi-Fi 模組(ESP8266)以及門禁控制軟體。其中利用筆記型電腦裝載門禁蒐集與控制軟體，實作物聯網環境中的 Client 以及 Server 進行通訊。搭建完成的智慧應用系統，以實際的 RFID 標籤，透過接在 Arduino UNO 上的 RFID Rader 感應，將讀取到的標籤進行 AES 加密，加密完成後的資料，透過 Wi-Fi 模組送至門禁控制軟體接收並解密。完成智慧應用系統的門禁資料加解密驗證，我們對該系統實施 SCA，透過示波器擷取 Arduino UNO 執行時散發出的電磁輻射獲得加密時的功率消耗，示波器接著將功率消耗以電磁感應探棒，將擷取到的 Trace 送至主控電腦，由主控電腦將 Trace 傳入運算伺服器進行 CPA，並回傳金鑰攻擊的結果，檢視子金鑰攻擊成功的數量，實驗配置如圖四。當我們確定物聯網傳輸情境以及 SCA 的攻擊設置完成之後，就可以開始將 AES 加密改為使用動態金鑰變換 AES，為了得知金鑰更換的效率以及效果，我們分別在完成不同 Trace 數的錄製之後進行金鑰更換，藉此觀察金鑰變換頻率的限制。Client、Server 端的設置以及動態金鑰變換 AES 的設計方式將會在後續詳細解說。

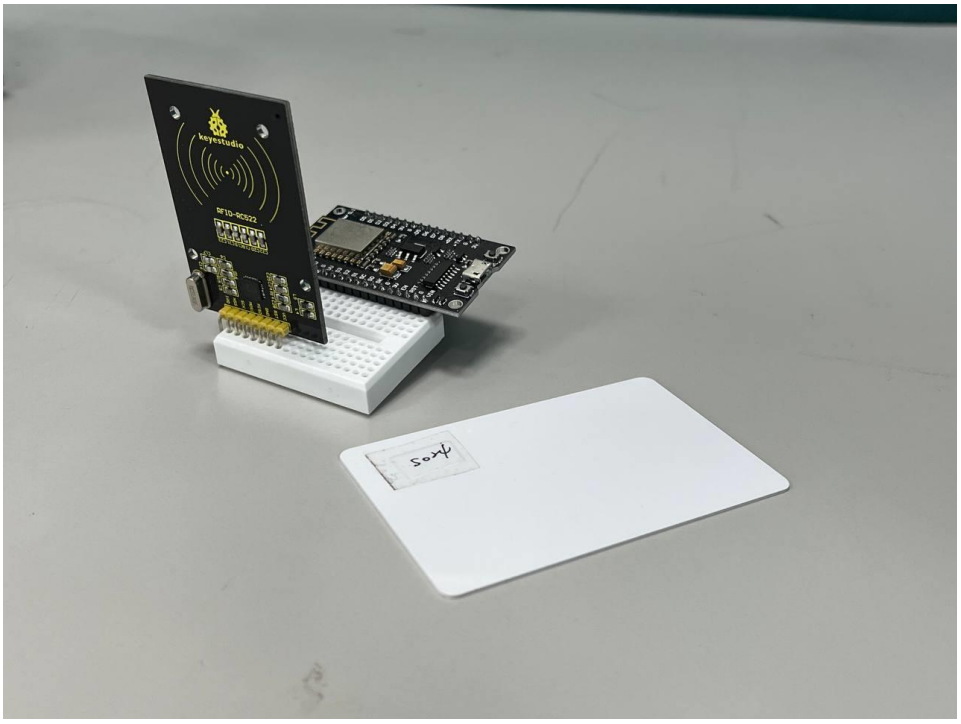


圖四：SCA 情境示意圖[14]

3.2 Client 端

我們將通訊情境建立在使用 Wi-Fi 進行門禁卡感應資訊的無線傳輸，透過 Arduino UNO 開發板 NodeMCU v3、ESP8266 Wi-Fi 模組以及 MFRC552 RFID 感應模組建立簡易的物聯網門禁感應裝置。Client 端的建立與及運作如下，首先載入標頭檔

ESP8266WiFi.h、MFRC522.h 以及 config.h，前兩項分別是 Wi-Fi 模組以及 RFID 感應模組的函式庫，config.h 裡面定義了用於連接的 Wi-Fi SSID、密碼、預設的 AES-128 金鑰、Server IP、Port 以及用於 D-H 的基數及模數數值，再來使用函式 setup()初始化 MFRC522 以及連接上指定 Wi-Fi 後連線到 Server 程式，接著使用函式 loop()監聽 MFRC522 感測器讀取到的門禁卡 UID，並記錄發送的資料筆數，最後將讀取到的門禁卡 UID 字串經過自定義的 AES-128 函式 aes128()加密後發送至 Server 端，此時將會把 Client 端記錄的資料筆數以及 Server 端的接收資料筆數進行對照，以確保傳輸的正確性。在一定筆數的資料發送後，使用 D-H 與 Server 進行金鑰協商，產生下一組 AES 加密用的金鑰，實際配置如圖五所示。

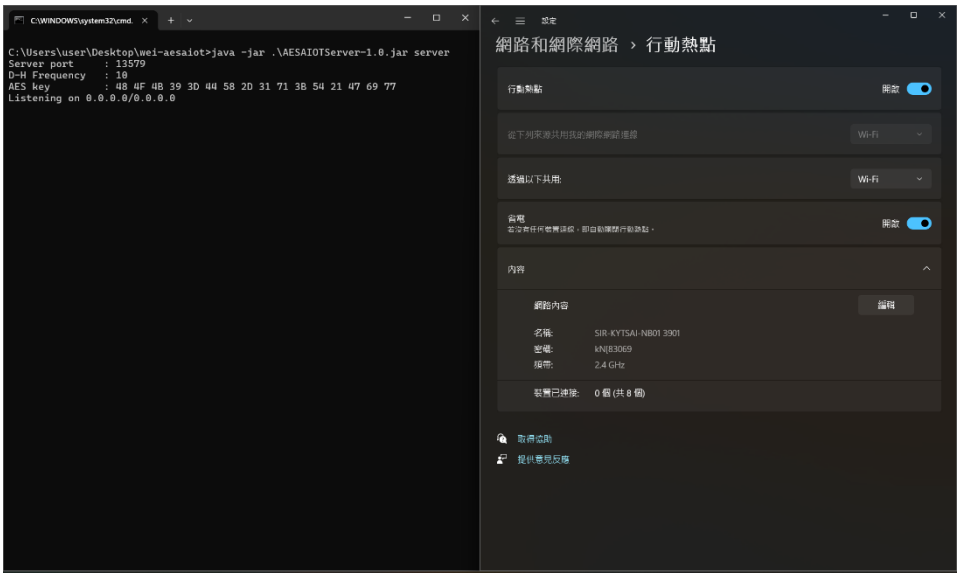


圖五：Client 端實際配置圖

3.3 Server 端

在筆記型電腦上使用 Java 撰寫智慧門禁系統的接收端程式如圖六所示，其中使用套件 info.picocli:picocli、javax.crypto 以及 java.net。透過命令列給定 port 參數啟動 Server 後，使用 java.net.ServerSocket 監聽指定的端口，在通訊端口建立完成後連線並且收送資料。建立連線後，使用初始金鑰進行 D-H 產生下一組金鑰。將來自 Client 的訊息利用 Cipher.getInstance("AES/ECB/NoPadding")進行 AES 解密，若訊息格式為 RFID 格式，則使用訊息內的 UID 在系統中進行搜索，並執行相對應的動作，例如：發送特定的 HTTP 請求、紀錄感應時間進入資料庫。每收到一定筆數的資料，將再與 Client 端進行一次 D-

H 的金鑰協商，產生下一組 AES-128 解密用的金鑰。

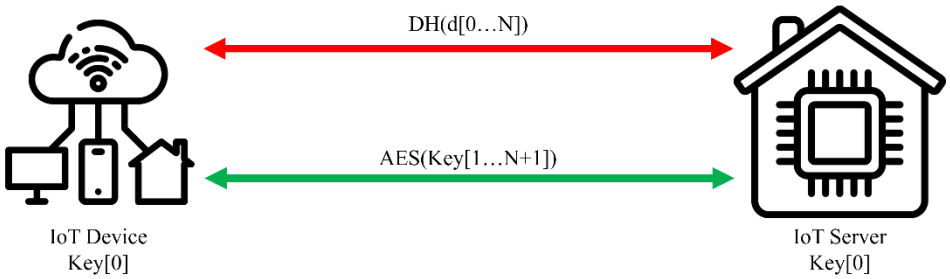


圖六：Server 端配置畫面圖

3.4 動態金鑰變換 AES 的設計

雖然本研究使用的 AES-128 使用的金鑰僅有 128bits，但主要研究對象為旁通道攻擊對 AES 重複使用相同金鑰而洩漏的電磁訊號，因此在實驗中使用更高位元的金鑰進行加解密並沒有實質上的效益。若直接透過 D-H 協商出用於加密之 128bits 金鑰，考量到現今電腦運算能力及 D-H 問題求解可預先被建表，因此 128bits 所產生之金鑰有安全性上的疑慮；若使用 1024bits 以上之 D-H，雖然安全性隨之提高，但計算成本將對 IoT 裝置造成過多負擔，因此僅使用 D-H 來協商出跳動金鑰所需要的 32bits 運算子。

圖七中所示 key[0]為兩端系統內部預先設定的初始金鑰，在建立連線後會依據初始金鑰進行一次 D-H 以協商出第一個 32bits 跳動運算子 d[0]。此處進行 D-H 的資料內容不可透過 key[0]進行 AES 建立安全通道傳輸，系統受內部或外部因素而多次進行重啟時，如果利用此 key[0]進行加密之動作恐因重複執行而洩漏出特徵信息，進而受到 SCA。透過公開通道協商出跳動運算子 d[0]後，即可使用對初始金鑰 key[0]使用湊雜函數 hash(key[0], d[0])求得 key[1]。



圖七：動態金鑰變換 AES 加密演算法之金鑰生成概念

AES-128 的 SCA 針對裝置晶片使用同樣 AES 金鑰所產生出相似的電磁波波形進行統計學習，因此重複使用同一把金鑰進行加密是不安全的。於是我們在每傳送M筆資料後會固定進行 D-H 以協商出下一把跳動運算子以產生出新的金鑰，如圖八所示。在進行 D-H 的時候應使用亂數作為私鑰，對於無法連網的 IoT 裝置，可以使用 Arduino UNO 函式 analogRead()讀取多組 PIN 腳上的類比數值，作為亂數種子使用，以取代預設的時間亂數種子。

```
[11-27 12:45:22] Message number         : 2098
[11-27 12:45:22] Incoming message       : 993B7BC6B34C9DFA48E82502132063A8
[11-27 12:45:22] Current AES key        : 65 CD F3 C2 10 C6 E0 D6 1C F3 83 AF 0C C5 D1 8C
[11-27 12:45:22] Plain text             : Hello world

[11-27 12:45:22] Message number         : 2099
[11-27 12:45:22] Incoming message       : 993B7BC6B34C9DFA48E82502132063A8
[11-27 12:45:22] Current AES key        : 65 CD F3 C2 10 C6 E0 D6 1C F3 83 AF 0C C5 D1 8C
[11-27 12:45:22] Plain text             : Hello world
[11-27 12:45:22] Executing D-H...
[11-27 12:45:22] Generate private key   : 715652219
[11-27 12:45:22] Generate public key    : 241877510
[11-27 12:45:22] Received public key    : 1632961243
[11-27 12:45:22] Shared key             : 1511346633

[11-27 12:45:22] Message number         : 2100
[11-27 12:45:22] Incoming message       : 96EC9F094DA5706B477931B19832120E
[11-27 12:45:22] Current AES key        : AC D1 A2 97 D9 DA B1 83 D5 EF D2 FA C5 D9 80 D9
[11-27 12:45:22] Plain text             : Hello world
```

圖八：D-H 金鑰變換

3.5 簡化版 D-H 實作

以基數 $G = 37$ 及模數 $P = 2147483647$ 為例，IoT 裝置端及 Server 端分別選定私鑰 a 及 b 後，透過計算$PK_A = 37^a \ mod \ 2147483647$及 $PK_B = 37^b \ mod \ 2147483647$產生出公鑰$PK_A$及$PK_B$後在公開通道上進行傳輸，再將彼此收到的公鑰計算$SK =$

$PK_B{}^a \bmod 2147483647$ 及 $SK = PK_A{}^b \bmod 2147483647$，最後求得共同的對稱金鑰 SK，運作流程如表一所示。

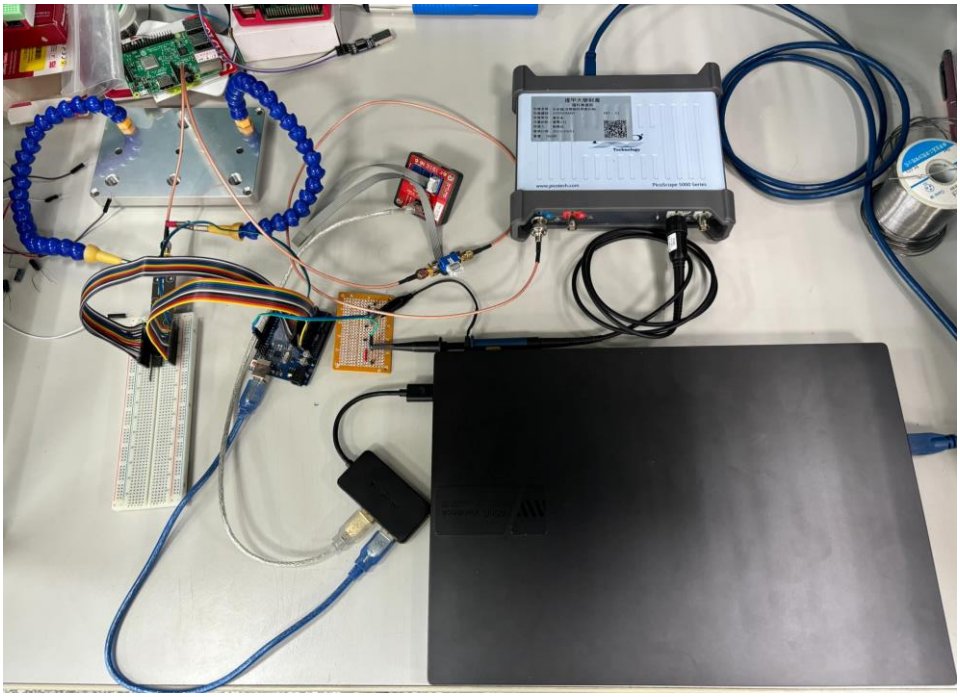
表一：D-H 機制的運作流程與相關參數可視狀態

IoT 裝置端（A）	公開訊息	Server 端（B）
（選定私鑰） $a = 777$	$G = 37$ $P = 2147483647$	（選定私鑰） $b = 888$
（計算公鑰PK_A） $PK_A = G^a \bmod P$ $= 37^{777} mod\ 2147483647$ $= 1348037377$		（計算公鑰PK_B） $PK_B = G^b \bmod P$ $= 37^{888} mod\ 2147483647$ $= 37387895$
（公鑰交換）	$PK_A = 1348037377$ $PK_B = 37387895$	（公鑰交換）
（計算金鑰SK） $SK = PK_B{}^a mod\ P$ $= 1142936476$	（需要解出私鑰a, b） 1348037377 $= 37^a\ mod\ 2147483647$ 或 37387895 $= 37^b\ mod\ 2147483647$	（計算金鑰SK） $SK = PK_A{}^b\ mod\ P$ $= 1348037377$

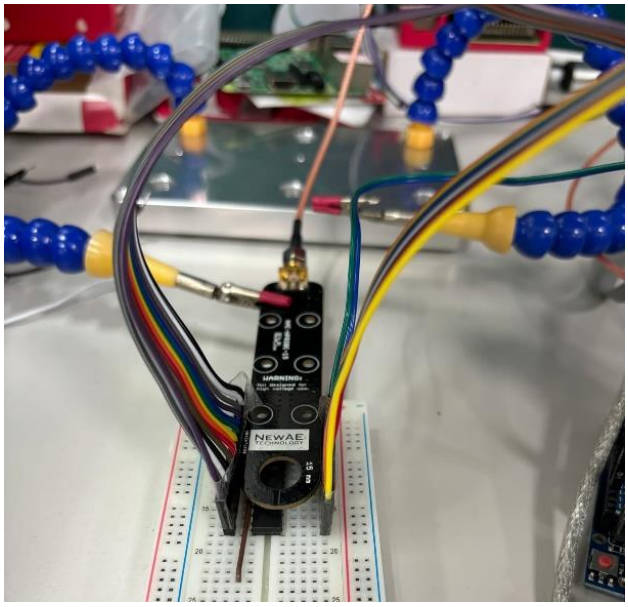
肆、實驗結果

4.1 實驗環境配置

本研究實驗環境配置如圖九，第三章提及的 Arduino UNO Client 端為資料傳輸端以及 SCA 目標，Java Server 端為資料接收端，用以驗證資料傳輸的正確性。攻擊平台的建立方式參考文獻[24]的實驗設置，使用 Chipwhisperer H-Field 探棒將 Client 端的電磁輻射錄製成 Trace，如圖十所示，連接到示波器 PicoScope 5244B 進行訊號處理後傳入主控電腦，再由主控電腦將 Trace 進行格式處理後傳入運算伺服器進行 CPA 運算，伺服器端的相關程式使用與 Peng 等人一樣的[25]，並回傳結果。



圖九：實驗環境示意圖

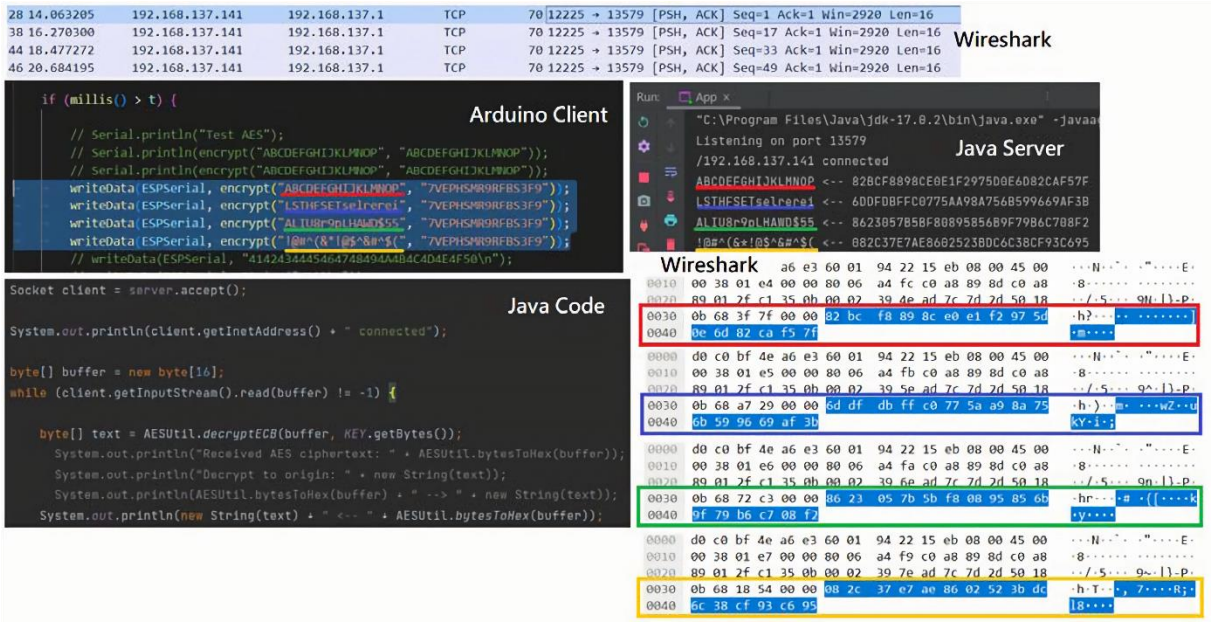


圖十：探棒擷取訊號實體圖

4.2 AES 加解密驗證

首先進行傳輸通道的確認與建立，在 Arduino UNO 與主控電腦之間透過簡易 D-H 使用 key[0]建立 AES 安全通道，接著加密測試訊息並傳輸至 Server 端，從 Server 端透過 Wireshark 觀察封包的接收與加密封包情形，再由 Server 端進行解密確認整段傳輸的

完整性，如圖十一、圖十二所示。



圖十一：Server 端建置安全通道與實際傳送的加密訊息封包

```
[11-27 12:45:22] Message number         : 2098
[11-27 12:45:22] Incoming message       : 993B7BC6B34C9DFA48E82502132063A8
[11-27 12:45:22] Current AES key        : 65 CD F3 C2 10 C6 E0 D6 1C F3 83 AF 0C C5 D1 8C
[11-27 12:45:22] Plain text             : Hello world

[11-27 12:45:22] Message number         : 2099
[11-27 12:45:22] Incoming message       : 993B7BC6B34C9DFA48E82502132063A8
[11-27 12:45:22] Current AES key        : 65 CD F3 C2 10 C6 E0 D6 1C F3 83 AF 0C C5 D1 8C
[11-27 12:45:22] Plain text             : Hello world
[11-27 12:45:22] Executing D-H...
[11-27 12:45:22] Generate private key   : 715652219
[11-27 12:45:22] Generate public key    : 241877510
[11-27 12:45:22] Received public key    : 1632961243
[11-27 12:45:22] Shared key             : 1511346633

[11-27 12:45:22] Message number         : 2100
[11-27 12:45:22] Incoming message       : 96EC9F094DA5706B477931B19832120E
[11-27 12:45:22] Current AES key        : AC D1 A2 97 D9 DA B1 83 D5 EF D2 FA C5 D9 80 D9
[11-27 12:45:22] Plain text             : Hello world
```

圖十二：Server 端接收加密資料並進行解密

4.3 動態金鑰更換分析

由於物聯網裝置中的微控制器計算能力有限，因此，還需要對此金鑰更換機制進行評估，我們在 Arduino UNO 開發板實際傳送明文訊息，使用不同頻率變換金鑰，並傳送 3,000 筆將明文加密成 AES-128 封包的資料，藉此記錄實際發送所需時間。表二紀錄了

10、30 以及 50 筆不同頻率變換金鑰，每筆資料的平均發送時間，對比目前的市面上的門禁刷卡系統，刷卡前後所花費的時間介於 1 秒到 2 秒之間，本研究所設計的機制，能有效應用於實際的智慧門禁系統。

表二：不同金鑰更換頻率對應發送 3,000 筆訊息之時間對照表

D-H 頻率	3,000 筆加密資料所需時間（ms）	平均發送時間 (ms)
每 10 筆	69,011	≈23
每 30 筆	45,415	≈15
每 50 筆	36,969	≈12

4.4 SCA 結果

在本研究中，我們針對 Arduino UNO 微控制器進行了無線旁通道攻擊（SCA），並在圖九所示的實驗環境下收集了電磁訊號。通過相關係數分析，我們評估了攻擊成功破解加密金鑰的程度。

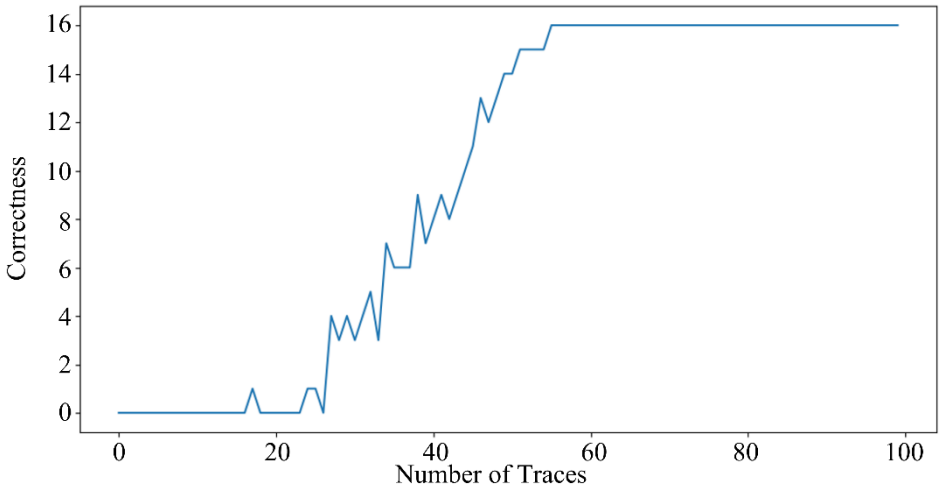
圖十三展示了微控制器 AES 加密演算在未加保護的情況下對金鑰進行攻擊的結果。圖中橫軸代表攻擊所需的數據軌跡（Trace）數量，縱軸則顯示出成功猜測金鑰的正確位元組個數。實驗結果顯示，在未保護的狀況下，僅需約 55 條 Trace 即可完全破解 AES-128 的 16 個子金鑰。接下來，我們在金鑰變換機制下進行了不同次數的加密後進行攻擊，結果如圖十四至圖十六所示。具體來說：

圖十四：在每 50 次加密後變換金鑰的情況下，攻擊最多成功破解 12 個子金鑰。

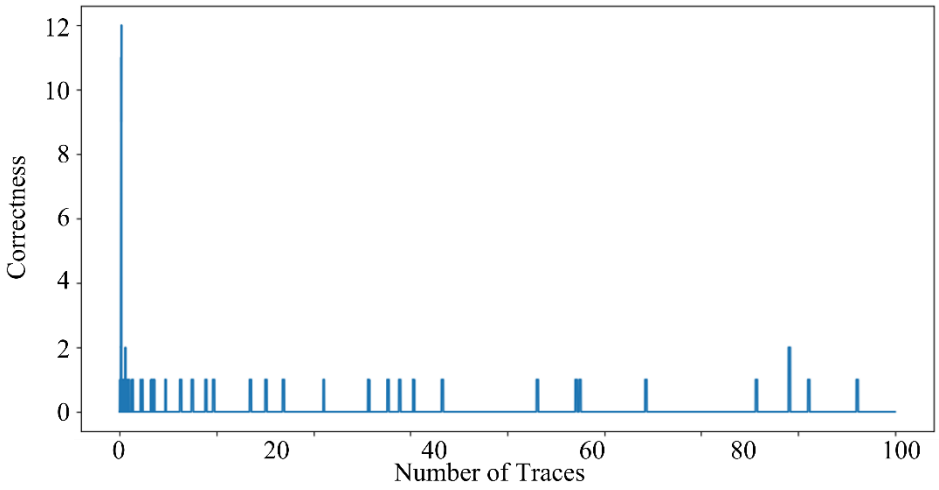
圖十五：在每 30 次加密後變換金鑰的情況下，攻擊最多成功破解 8 個子金鑰。

圖十六：在每 10 次加密後變換金鑰的情況下，攻擊最多僅成功破解 2 個子金鑰。

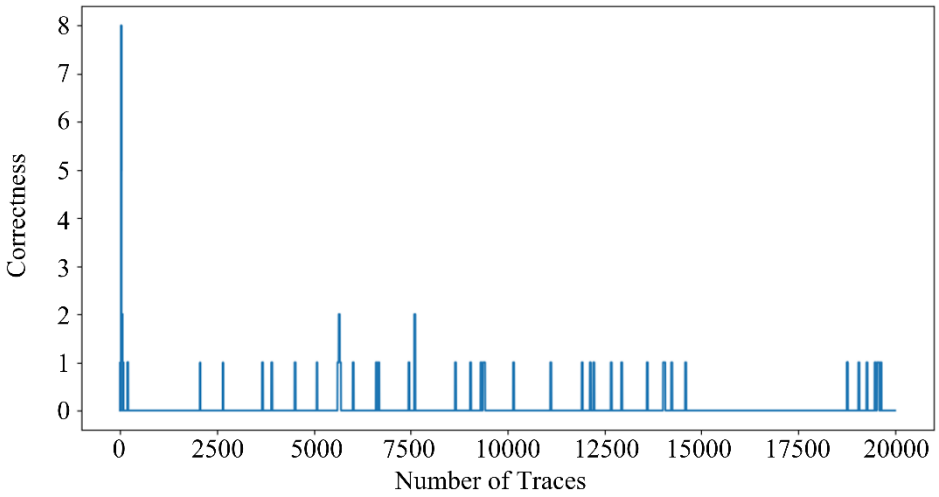
這些結果清楚地表明，隨著金鑰變換頻率的提高，旁通道攻擊的成功率顯著下降，證明了動態金鑰變換機制在增強物聯網裝置安全性方面的有效性。



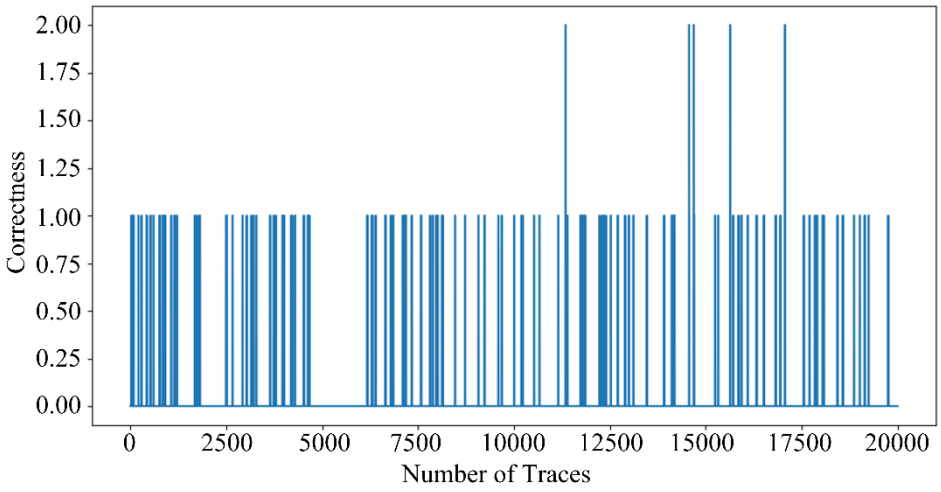
圖十三：未受保護之 AES 加密攻擊結果



圖十四：AES 加密 50 次後更換金鑰之攻擊結果



圖十五：AES 加密 30 次後更換金鑰之攻擊結果



圖十六：AES 加密 10 次後更換金鑰之攻擊結果

伍、結論與未來建議發展

在物聯網 IoT 環境中，隨著設備持續暴露於網際網路，除了採用加密演算法來保護傳輸中的機密資料之外，抵禦 SCA 也同樣重要。若未能有效防範這類攻擊，將可能導致機密資料的洩漏，進而帶來嚴重的安全風險。由於物聯網設備的微控制器通常受限於其尺寸和成本，使得硬體防護措施難以實現，因此，軟體層面的 SCA 防護設計必須在兼顧運算效能與功耗的前提下進行。本研究提出了一種基於動態金鑰變換的 AES 加密機制，該機制通過在旁通道攻擊成功之前變更加密金鑰，有效提升了系統的安全性。我們基於 D-H 密鑰交換原理，設計了一種輕量級且易於實現的跳動金鑰機制，此

機制能夠適用於資源受限的微控制器環境，不僅限於 AES 加密，也適用於其他需要金鑰協商的加密機制。未來的研究可以進一步擴展該機制的應用範圍，探索其在各種物聯網應用場景中的效能與安全性。此外，隨著半導體技術的進化，未來的微控制器可具備更高的計算能力和性能，這將提供更複雜的防禦策略與新技術的實現，進而全面提升物聯網設備的整體安全性。

參考文獻

[1] Statista, Number of IoT connections worldwide 2022-2033, with forecasts from 2024 to 2033,https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (2024/6/12).

[2] S. Moini, S. Tian, D. Holcomb, J. Szefer, and R. Tessier, "Power Side-Channel Attacks on BNN Accelerators in Remote FPGAs", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 357-370, 2021.

[3] B. B. Yilmaz, M. Prvulovic, and A. ZajićA, "Electromagnetic Side Channel Information Leakage Created by Execution of Series of Instructions in a Computer Processor", *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 776-789, 2019.

[4] A. Aljuffri, M. Zwalua, C. Rodolfo W. Reinbrecht, S. Hamdioui, and M. Taouil, "Applying Thermal Side-Channel Attacks on Asymmetric Cryptography", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.29, no. 11, pp. 1930-1942, 2021.

[5] P. Schaumont, K. Tiri, "Masking and Dual-Rail Logic Don't Add Up", *Cryptographic Hardware and Embedded Systems - CHES 2007*, pp. 95-106.

[6] K. Schramm, C. Paar, "Higher Order Masking of the AES", *Topics in Cryptology – CT-RSA 2006*, pp. 208-255.

[7] Y. Baseri, V. Chouhan, and A. Ghorbani, "Cybersecurity in the Quantum Era: Assessing the Impact of Quantum Computing on Infrastructure", *arXiv Cryptography and Security*, https://arxiv.org/abs/2404.10659 (2024/4/16).

[8] S. Chng, H. Y. Lu, A. Kumar, D. Yau c, "Hacker types, motivations and strategies: A comprehensive framework", *Computers in Human Behavior Reports*, vol. 5, pp. 100167, 2022.

[9] P. S. MUNOz, N. Tran, B. Craig, B. Dezfouli, and Y. Liu, "Analyzing the Resource Utilization of AES Encryption on IoT Devices", *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1200-1207.

[10] S. Fatima, T. Rehman, M. Fatima, S. Khan, and M. A. Ali, "Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing", *The 7th International*

Electrical Engineering Conference, 2022, https://www.mdpi.com/2673-4591/20/1/14 (2022/7/29).

[11] M. Devi, A. Majumder, "Side-Channel Attack in Internet of Things: A Survey", *Applications of Internet of Things. Lecture Notes in Networks and Systems*, pp. 213-222.

[12] J. Rawat, I. Kumar, N. Mohd, K. K. S. Rana, N. Pathak, and R. K. Gupta, "IoT-Based Home Automation System Using ESP8266", *International Conference on Innovative Computing and Communications*, pp. 695-707.

[13] J. Daemen and V. Rijmen, "The Design of Rijndael The Advanced Encryption Standard (AES)", Springer, 2020.

[14] C.-W. Kuo, K.-Y. Tsai, W.-M. Weng, C.-C. Lin, Y.-Y. Hong, G.-L. Wang, "Implementation and Analysis of Side-Channel Attack Mitigation Based on Autoencoder," *Communications of the CCISA*, vol. 29, no. 4 , pp. 1-18, 2023.

[15] NIST, "Lightweight Cryptography Standardization Process: NIST Selects Ascon", https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon (2023/7/23).

[16] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *Advances in Cryptology — CRYPTO' 99*, pp. 388-397.

[17] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model", *Cryptographic Hardware and Embedded Systems - CHES 2004*, pp. 16-29.

[18] T. S. Messerges, "Using Second-Order Power Analysis to Attack DPA Resistant Software", *Cryptographic Hardware and Embedded Systems — CHES 2000*, pp. 238-251.

[19] W. Diffie, M. E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, 1976.

[20] F. K. Santoso and N. C. H. Vun, "Securing IoT for smart home system", *2015 International Symposium on Consumer Electronics (ISCE)*, https://ieeexplore.ieee.org/document/7177843 (2015/8/6).

[21] M. Collotta and G. Pau, "A Novel Energy Management Approach for Smart Homes Using Bluetooth Low Energy", *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2988-2996, 2015.

[22] I. A. Zualkernan, A. R. Al-Ali, M. A. Jabbar, I. Zabalawi, and A. Wasfy, "InfoPods: Zigbee-Based Remote Information Monitoring Devices for Smart-Homes", *IEEE Transactions on Consumer Electronics*, vol. 55, pp. 1221-1226, 2009.

[23] K.-L. Tsai, F.-Y. Leu, I. You, S.-W. Chang, S.-J. Hu, and H. Park, Low-Power AES Data Encryption Architecture for a LoRaWAN, *IEEE Access*, pp. 146348-146357.

[24] C.-W. Kuo, C.-C. Lin, Y.-Y. Hong, J.-R. Liu, C.-H. Yeh, and K.-Y. Tsai, "Research and Analysis of the Effects of Different Shielding Materials on Resisting Side-Channel Attacks on IoT Device Microcontroller", *2024 8th International Conference on Cryptography,*

Security and Privacy (CSP), pp. 84-88.

[25] S.-Y. Peng, W.-C. Hong, J.-T. Li, and S.-J. Huang, "Framework for efficient SCA resistance verification of IoT devices", *2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 355-366.