前瞻科技與管理　11 卷 2 期，28-47 頁（2023 年 5 月）
Journal of Advanced Technology and Management Vol. 11, No. 2, pp. 28-47 (May, 2023)
DOI:10.6193/JATM.202305_11(2).0002

漫談輕量級密碼學標準發展與實現

劉江龍 *

國防大學理工學院電機電子工程學系教授

摘 要

物聯網（Internet of Things, IoT）的普及為生活帶來便利，但也因為物聯網設備公開部署及無線傳輸的特性，產生了其他安全問題。傳統加密標準難以或不可能在資源受限的設備中實現，即使勉強實現，它們的性能也可能無法被接受，因此發展可在資源受限的設備執行的輕量級密碼學演算法實有其必要性及迫切性。有鑑於此，美國國家標準暨技術研究院（National Institute of Standards and Technology, NIST）於 2018 年正式向世界徵求輕量級密碼學演算法，預期於 2022 年確立輕量級密碼學標準。本文目的在說明傳統密碼演算法在資源受限設備上實現的限制、輕量級密碼學演算法需求的重點及 NIST 目前制定輕量級密碼學標準的發展，並以目前進入最後決選的 ASCON 輕量級加密演算為範例進行實現，說明如何在一個物聯網的應用中實現一個符合輕量級密學標準的演算法，以保護物聯網中訊息傳遞的安全。

關鍵詞：物聯網安全、密碼學標準、輕量級密碼學、物聯網加密、資源受限設備

* 通訊作者：劉江龍
電子郵件：chianglung.liu@gmail.com
（收件日期：2022 年 2 月 10 日；修正日期：2022 年 2 月 28 日；接受日期：2022 年 2 月 28 日）

Journal of Advanced Technology and Management Vol. 11, No. 2, pp. 28-47 (May, 2023)
DOI:10.6193/JATM.202305_11(2).0002

Development and Implementation of Lightweight Cryptography Standards

Chiang-Lung Liu*

Professor, Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University

Abstract

The proliferation of the Internet of Things (IoT) brings convenience to life, but also creates other security concerns due to the nature of open deployment and wireless of IoT devices. Traditional encryption standards are difficult or impossible to implement in resource-constrained devices, and even if they do, their performance may be unacceptable, so the development of lightweight cryptographic algorithms that can be executed in resource-constrained devices is necessary and urgent. The National Institute of Standards and Technology (NIST) of the United States therefore officially solicited lightweight cryptography algorithms in 2018, and is expected to complete the establishment of lightweight cryptography standards in 2022. The purpose of this paper is to explain the limitations of traditional cryptographic algorithms on resource-constrained devices, the focus of lightweight cryptographic algorithm requirements, and the development of NIST's current lightweight cryptography standards. The lightweight encryption algorithm ASCON is implemented as an example to illustrate how to implement an algorithm that conforms to the lightweight cryptography standards in an IoT application to protect the security of message transmission in the IoT.

Keywords: IoT security, cryptography standards, lightweight cryptography, IoT encryption, resource-constrained devices

* Corresponding Author: Chiang-Lung Liu
E-mail: chianglung.liu@gmail.com

壹、簡介

物聯網（Internet of Things）是一個無處不在（Ubiquitous）的網絡，其是由可唯一識別的事物所組成，並透過大量數據傳輸作為智慧決策的基礎，以提供廣泛的服務，如圖 1 所示。小從電子健康，大至核電廠網路系統等，均屬於物聯網服務的範疇。據統計，到 2020 年為止，連接到網際聯網的物件數量已增長到 500 億，還在大幅成長，可謂網際網路的第一次演進（Evans, 2011）。每個物聯網系統目的在提供特定的服務，而每個服務的交付則取決於感知層（Perception Layer）收集的訊息（Message）。感知層是物聯網中的最底層，其是由資源受限的設備或無線感測網路（Wireless Sensor Network）所構成。一般來說，這些設備大多數是公開部署的，並使用無線媒體進行傳輸。公開部署的特性使得這些設備容易受到節點竄改（Node Tampering），而無線傳輸的方式也使得訊息容易遭到攔截。對於關鍵服務（例如核電監控）而言，若訊息遭到攔截或更改，則可能導致生命和金錢的重大損失。Ronen and Shamir（2016）曾利用對物聯網產品攻擊來證明其脆弱性，而聰明的駭客則可以利用這些漏洞來攻擊各種物聯網設備，例如智慧家電及智慧電腦等；Ge, Hong, Alzaid, and Kim（2017）則使用階層式攻擊（Hierarchical Attack）表示模型展示了跨協議（Cross-Protocol）物聯網產品的漏洞；開放網路應用程序安全計畫（Open Web Application Security Project, OWASP）（“OWASP internet of things top 10,” n.d.）也將隱私、身分驗證或授權（Authorization）不足、缺乏傳輸加密和實體層安全性差等同列為物聯網十大漏洞。

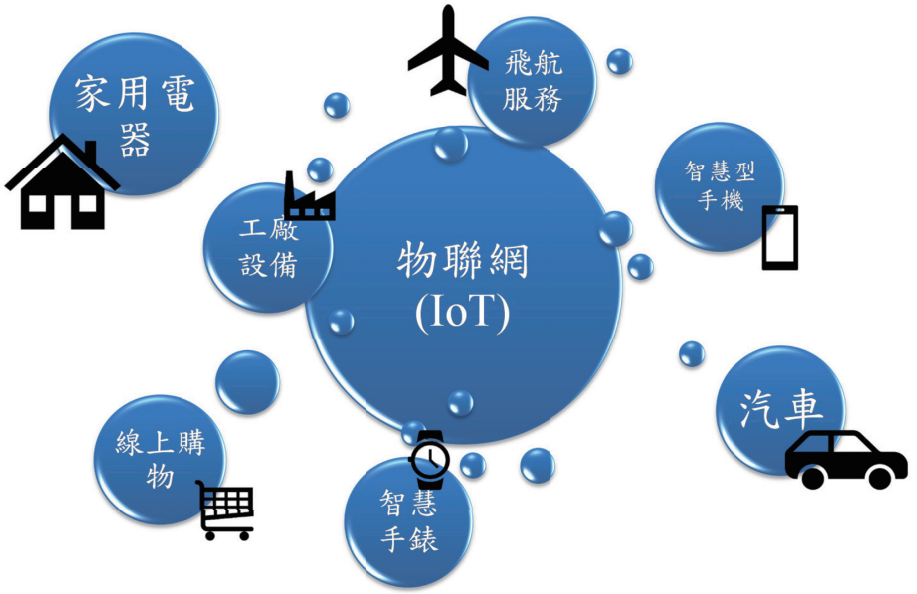


圖 1　物聯網概念圖

資料來源：作者自行研究整理。

根據 Kamal（2017）提出的物聯網參考架構，物聯網安全可分為五個功能組件，包括身

分管理、身分認證（Authentication）、授權、密鑰交換和管理、信任和聲譽等，其也構成聯網安全的六大重點領域，分別為機密性（Confidentiality）、完整性（Integrity）、可用性（Availability）、身分認證、存取控制（Access Control）及不可否認性（Non-Repudiation）等，而一般密碼學元件（Cryptographic Primitives）則可以滿足這些目標的實現，其中機密性和完整性被認為首要的實現目標。但若利用傳統的密碼學元件來實現訊息的機密性和完整性則需大量的資源分配，其對於資源有限的物聯網設備來說，無疑是一種挑戰。一般認為，物聯網設備的特點是有限的計算能力、有限的記憶體（Memory）空間、有限的電源及有限的電池壽命（Singh, Sharma, Moon, and Park, 2017），這些設備通常具有低至 2 KB 和 1 KB 的隨機存取記憶體（RAM）和唯讀記憶體（ROM），成為傳統密碼學元件實作的瓶頸（Vikas, Sagar, and Munjul, 2021; Zhao and Ge, 2013），這也清楚地指出開發輕量級密碼學（Lightweight Cryptography）演算法的必要性。

有鑑於輕量級密碼學演算法在物聯網中的重要性，美國國家標準暨技術研究院（National Institute of Standards and Technology, NIST）於 2013 年啟動了輕量級密碼學標準計畫，其目的在瞭解輕量級密碼學標準的需求，並針對資源有限的設備制定輕量級密碼學演算法的標準。本文目的在說明傳統密碼演算法在資源受限設備上實現的限制，及輕量級密碼學演算法需求的重點，進而說明 NIST 目前制定輕量級密碼學標準的發展，其中包括對輕量級密碼學演算法的需求項目、規範及評估標準等，最後以目前進入最後決選的 ASCON 輕量級加密演算為範例進行實現，說明如何在一個物聯網的應用中實現一個符合輕量級密學標準的演算法，以保護物聯網中訊息傳遞的安全。

本文其餘各節安排如下：第貳節說明傳統密碼演算法於資源受限設備實現的限制；第參節為輕量級密碼學標準的發展；第肆節為輕量級密碼法 ASCON 之實現；第伍節則為本文之結論。

貳、傳統密碼演算法於資源受限設備實現的限制

一、傳統密碼法概述

為說明輕量級密碼學演算法在資源受限設備實現的限制及輕量級密碼學標準的發展，我們有必要先對傳統密碼演算法的結構及操作方式進行說明。傳統密碼演算法可概分為對稱式金鑰（Symmetric-Key）演算法及非對稱式金鑰（Asymmetric-Key）演算法兩大類，如圖 2 所示。對稱式金鑰演算法的加解密雙方使用同一把金鑰（Key）進行加密與解密，大部分的密碼學演算法屬於對稱式的，有名的資料加密標準（Data Encryption Standard, DES）及先進資料加密標準（Advanced Encryption Standard, AES）為對稱式密碼演算法的代表；非對稱式金鑰演算法的加解密雙方則是使用不同的金鑰進行加解密，這也是「非對稱式」名稱的由來。非對稱式金鑰演算法又稱為公開金鑰（Public-Key）演算法，其是基於數學上難解的問題進行演算法的設計，主要是用來解決對稱式金鑰密碼系統金鑰管理上的問題，在加密及解密的速度上明顯落後對稱式金鑰演算法，不適合進行大量資料的加解密，實務上主要用於金鑰的

傳輸（例如 SSL 協定）或數位簽章（Digital Signature），著名的非對稱式金鑰演算法首推 RSA，而橢圓曲線密碼學（Elliptic Curve Cryptography, ECC）則因其安全性較佳，為非對稱式金鑰密碼系統實現的首選。

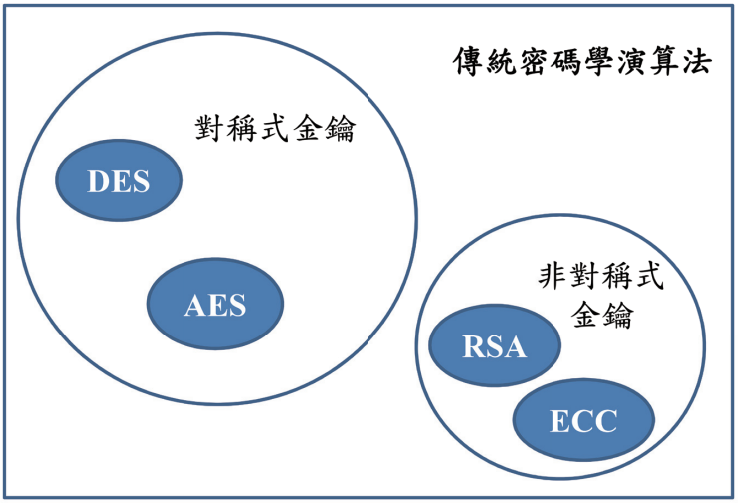


圖 2　傳統密碼演算法概念圖

資料來源：作者自行研究整理。

對稱式演算法依照加密的操作方式的不同，可概分為區塊密碼法（Block Cipher）及串流密碼法（Stream Cipher）兩類。區塊密碼法的操作方式如圖 3 中虛線部分所示，未加密的明文（Plaintext）可視為以位元（Bit）為單位所組成的位元串流（Bit Stream），在此稱為明文串流。區塊密碼法是將明文串流切割為相等位元長度的明文區塊，再依特定模式（圖 3 中為 ECB 模式）利用對稱金鑰將每一個明文區塊加密為密文（Ciphertext）區塊，最後將密文區塊組合成單一的密文。為了抵抗統計式攻擊（Statistical Attack），區塊密碼法採用多回合的替代（Substitution）及排列（Permutation）結構（稱為 S-P Network）；另由於安全需求，區塊密碼法同時透過回合金鑰產生程序為每一回合產生一把不同的子金鑰（Sub-Key），以提供各回合加密使用。事實證明，這種結構具備良好的安全性效果，例如 DES 及 AES 均採用這樣的結構。

串流加密的操作方式如圖 3 的右半部所示，首先將加密金鑰轉換為一長串不具週期且與明文串流同樣長度的金鑰串流（Key Stream），再將此金鑰串流與明文串流透過互斥或（Exclusive or, XOR）運算產生密文。由於 XOR 加密方式簡單，串流加密演算法非常適合需要即時加密的運算，無線網路加密即為一例。

雜湊函數（Hash Function）也常結合密碼法以進行資料完整性的保護，其作用類似資料壓縮方法，可將任意長度的明文輸入處理成固定長度的輸出（稱為雜湊值），MD5 為著名的雜湊函數，其輸出長度為固定的 128 位元。與一般無失真（Lossless）的壓縮方法不同的是，雜湊演算法是單向的，意即無法將雜湊值回復成原始訊息，這個特性有助於抵抗攻擊者任意竄改或偽冒原始訊息。一般而言，雜湊函數運作過程是不需要金鑰參與的，但基於特殊用途，

也有基於金鑰參與運作的雜湊函數，其輸出特別稱為訊息認證碼（Message Authentication Code, MAC），其與一般雜湊（Hash）功能的區別如圖 4 所示。

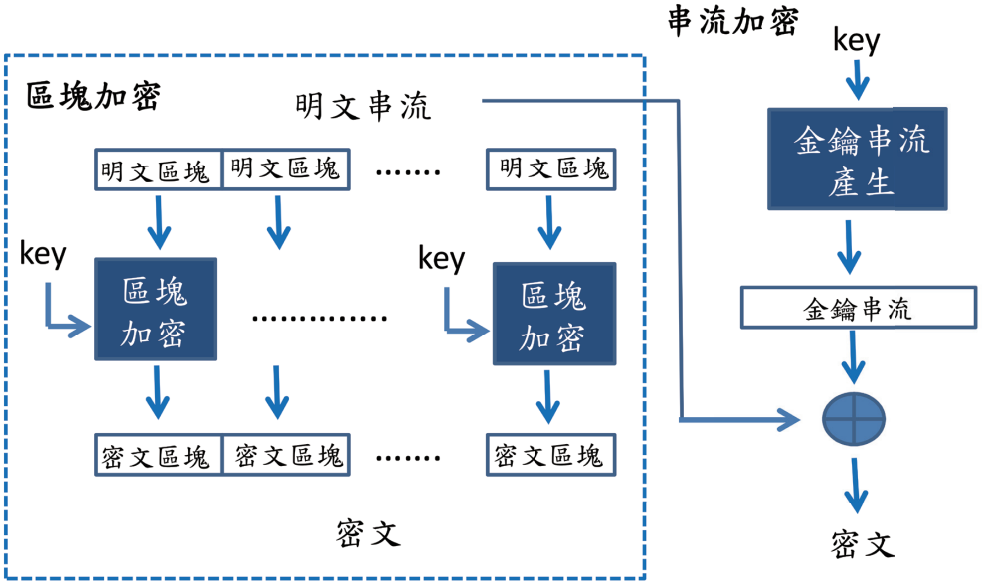


圖 3　區塊加密及串流加密示意圖

資料來源：作者自行研究整理。

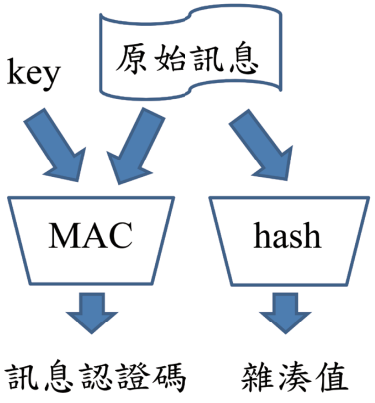


圖 4　MAC 與一般雜湊功能的區別

資料來源：作者自行研究整理。

二、傳統密碼法在資源受限設備之執行效率

為瞭解傳統密碼演算法是否符合輕量級密碼學演算法，必須先對輕量級密碼學演算法加以定義。輕量級密碼學演算法根據其可以實現的硬體環境加以區分，Hatzivallis, Fysarakis,

Papaefstathiou, and Manifavas（2018）依據硬體實現、軟體實現及設備能力將輕量級密碼學演算法分為三類，分別為超輕量級（Ultra-Lightweight）密碼演算法、低成本（Low-Cost）密碼演算法和輕量級密碼學演算法，如表 1 所示。其中硬體實現是以晶片面積（Chip Area）或等效閘（Gate Equivalence, GE）為衡量基準，軟體實現則表現在 ROM 及 RAM 的需求上，而設備能力則是已商業化的等效晶片。以下本文僅就軟體實現的角度說明傳統密碼學演算法在資源受限設備的實現能力。

表 1　輕量級密碼學演算法分類

區分	GE	ROM/RAM 需求	設備能力
超輕量級	超過 1,000	4 KB/256 B	8051, ATtiny 45
低成本	超過 2,000	4 KB/8 KB	ATmega 128
輕量級	超過 3,000	32 KB/8 KB	其他

資料來源：作者自行研究整理。

如表 1 所示，超輕量級及低成本兩類密碼演算法實現所需的 ROM 需求均為 4 KB，如扣掉系統的啟動（Boot）及載入（Loader）程式所占的記憶體空間，對於存放 DES 或 AES 這樣對稱式密碼法的程式來說是相當嚴苛的；若從輕量級密碼學演算法的角度來看，DES 及 AES 可能有機會放得進具有 32KB ROM 的設備，但 DES 及 AES 回合運算中使用許多查表法來加速其運算，而替換盒（S-Box）除了必須占用 ROM 的儲存空間外，在程式執行期間也會占用 RAM，對低容量的 RAM（8 KB）的設備來說，則為另一項考驗。如果捨棄使用查表法的程式實現模式，雖然可以減少記憶體需求，但必然影響其執行效能，這種狀況同樣發生在傳統串流加密演算法中。

傳統的非對稱式密碼法可用來產生數位簽章，以進行資料完整性驗證，其作法如圖 5 所示。訊息發送方首先利用發送者的私密金鑰（Private Key）對原始訊息的雜湊值進行加密，產生其數位簽章，再將數位簽章連同原始訊息送給收方。如果訊息接收方要驗證資料的完整性，則可先利用發送方的公開金鑰將數位簽章解密，取得原始訊息的雜湊值；接著將收到的原始訊息重新製作雜湊值，並與解簽章後所得的雜湊值比對，若相同，則可確定資料的完整性。若使用 RSA 這樣的非對稱式金鑰密碼法，為了符合金鑰的安全性，必須採用數千位元的金鑰進行指數運算，對配備 32 位元以下的 CPU 及有限 RAM 的設備而言，可能有執行上的困難，即使勉強可以執行，其效果也可能不彰。如果改用 ECC，雖然可以採用較短的金鑰達到同樣的安全性，可是運算效能仍然比傳統對稱式加密法低很多。

另外，為了同時維護訊息的機密性及完整性，原始訊息在傳送前必須使用對稱式金鑰加密，其與非對稱式金鑰密碼法搭配使用的場景如圖 6 所示。在這種狀況下，如果同時採用傳統的對稱式及非對稱式金鑰密碼法，其對運算資源的需求必然大於個別執行時的需求。為解決這個問題，許多目前發展的輕量級的密碼演算法已結合 MAC 的概念進行加解密演算法的設計，使其可同時提供加密及完整性認證的功能，目前 NIST 正在發展中的輕量級密碼學標準所徵求的帶有關聯數據的認證加密（Authentication Encryption with Associated Data, AEAD）即是採用這種方式，將在後續介紹輕量級密碼學標準發展時加以說明。

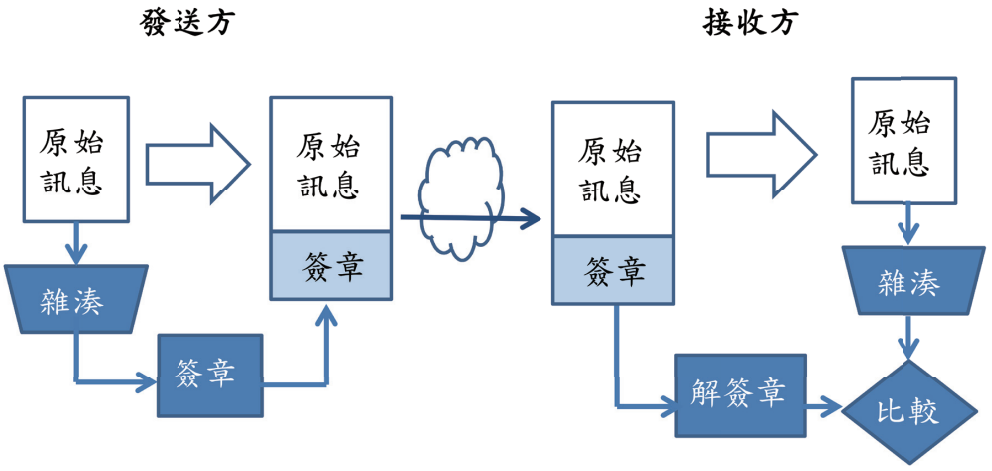


圖 5　使用非稱式金鑰密碼法的完整性保護流程

資料來源：作者自行研究整理。

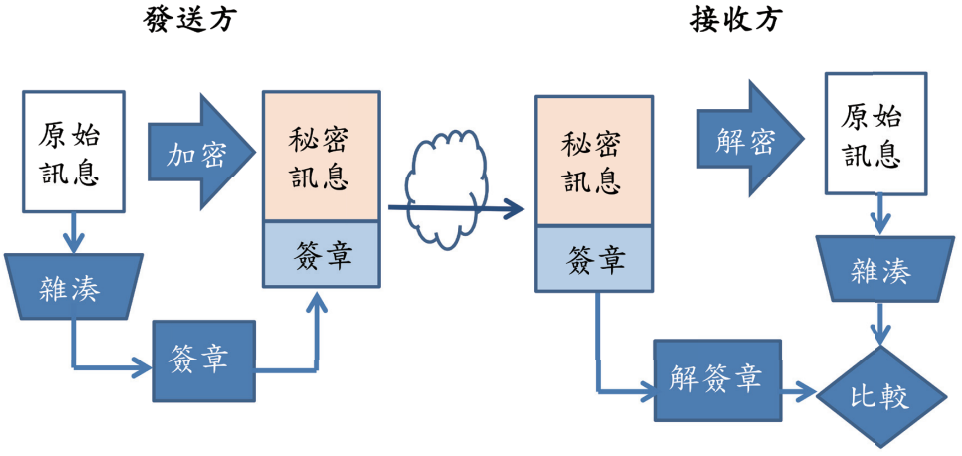


圖 6　結合對稱式及非稱式金鑰密碼法的完整性保護流程

資料來源：作者自行研究整理。

參、輕量級密碼學標準的發展

如前所述，由於 RFID 標籤及智慧卡等小型計算設備的部署變得越來越普遍，許多傳統的加密標準（例如 AES）難以或不可能在資源受限的設備中實現，即使勉強實現，它們的性能也可能無法被接受。有鑑於此，NIST 於 2013 年啟動了輕量級密碼學演算法計畫，其目的在研究目前由該單位認可的密碼技術標準在資源受限設備的效能，並瞭解特定輕量級密碼學標準的需求。經過了五年的準備，於 2018 年正式向世界徵求輕量級密碼學演算法，經過二輪的篩選，確立了 10 個演算法進入最後的決選，預計在 2022 年開完第五次輕量級密碼學研討會後確立輕量級密碼學標準。本節將區分為四小節就輕量密碼學標準制定過程、項目及規範、評估標準與最後進入決選的 10 個輕量級演算法分析進行說明。

一、輕量級密碼學標準制定過程

為了制定輕量級密碼學，NIST 於 2015 年 6 月舉辦了首屆輕量級密碼學研討會（“Lightweight Cryptography Workshop 2015”, 2015），以獲取大眾對目標設備的功能之限制，以及輕量級密碼學實際應用的要求和特點的反饋；2016 年 8 月公布了《輕量級密碼學報告草案》（McKay, Bassham, Turan, and Mouha, 2016），並聽取大眾的回饋；2016 年 10 月舉辦了第二次輕量級密碼學研討會（“Lightweight Cryptography Workshop 2016”, 2016），主要討論演算法提交的配置文件、評估工具和方法，以及密碼分析和輕量級密碼設計實施的最新工作，並持續蒐集大眾的回饋；2017 年 3 月底發行了《輕量級密碼學報告》（McKay, Bassham, Turan, and Mouha, 2017），編號為 NISTIR 8114，其中主要確立了輕量級密碼學標準的需求；2017 年 4 月及 2018 年 5 月，NIST 陸續公布了《輕量級密碼學標準程序草案》（Bassham, Çalık, McKay, Mouha, and Turan, 2017）及《輕量級密碼學標準程序提交需求及評估標準草案》（National Institute of Standards and Technology [NIST], 2018a），並聽取大眾的意見；2018 年 7 月，NIST 正式公布《輕量級密碼學標準程序提交需求及評估標準》（NIST, 2018b），並正式公開向全世界徵求輕量級密碼學演算法。

在輕量級密碼學演算法公開徵求後，NIST 陸續收到大量的提交演算法，經過評選後，於 2019 年公布進入第一輪決選的 56 個演算法（NIST, 2019），經過進一步評選後，於同年公布了進入第二輪決選的 32 個演算法（NIST, 2021a）。之後分別於 2019 年 11 月及 2020 年 10 月，針對這 32 個演算法召開了第三屆及第四屆輕量級密碼學研討會，並於 2021 年 3 月，公布了進入最後決選的 10 個演算法名單（NIST, 2021b），預計於 2022 年 5 月舉辦第五次公布輕量級密碼學研討會，討論這 10 個入圍決賽的演算法在各個方面的表現，並獲得輕量級密碼元件標準化的反饋。

二、輕量級密碼學標準徵求的項目及規範

輕量級密碼學是針對可以在廣泛的硬體和軟體上實現的各種設備，從 8 位元、16 位元和 32 位元等常見的微控制器（Microcontrollers），至超低成本應用的 4 位元微控制器及只能從環境中獲得有限的電量 RFID 標籤等，均屬於其評選範圍。為了滿足廣泛設備的需求，輕量級加密演算法不僅需要使用非常少量的 GE 來實現，還必須滿足嚴格的時序和功率要求，這樣的演算法不是目前傳統密碼學演算法做得到的，這也是 NIST 制定輕量級密碼學標準的目的。

NIST 經過廣泛的意見回饋及討論後，確立目前徵求的輕量級密碼學演算法有兩大項目：AEAD 及雜湊函數。認證加密目的在加密及解密過程中附加訊息認證的功能，以同時提供訊息的機密性及完整性保護，可視為加密函數與 MAC 的結合；雜湊函數的功能則如上一節所述，在提供對明文的單方向的壓縮功能，亦即對不定位元長度的明文輸入產生固定位元長度的輸出。

NIST 沒有要求提交者必須同時提交這兩種演算法，使用者也可以自行定義其演算法的特性及其支援的資源受限設備。NIST 的徵求文件中同時提供資料模板（Template），讓使

用者可以在提交文件中使用，如表 2 所示。儘管如此，NIST 對個別演算法仍有最低的設計規範，說明於以下各小節。

表 2　NIST 徵求文件提供的密碼演算法文件模板

文件名稱（Profile Name）	
功能特性（Functionality）	密碼演算法的目的（例如：認證加密、雜湊、訊息認證等）
設計目標（Design Goals）	列出設計目標
物理特性（Physical Characteristics）	列出物理特性，並提供可接受的範圍（例如：64 到 128 位元組的 RAM）
性能特徵（Performance Characteristics）	列出性能特徵，並提供可接受的範圍（例如：延遲 [Latency] 不超過 5 ns）
安全特性（Security Characteristics）	最低安全強度、相關攻擊模型、旁道 (Side Channel) 抵抗要求等。

資料來源：作者自行研究整理。

（一）AEAD 規範

NIST 所定義的 AEAD 加密演算法是一個具有四個位元組串列（Byte-String）輸入和一個位元組串列輸出的函數。其四個輸入分別為可變長度的明文、可變長度的關聯資料（Associated Data）、固定長度的隨機亂數（Nonce）和固定長度的密鑰；輸出則是可變長度的密文，其應支援認證解密（Authenticated Decryption），也就是使用者可以從密文進行訊息完整性的驗證。如果密文是有效的（亦即通過完整性驗證），則必須可以從輸入的有效參數還原明文；如果是無效的（亦即無法通過完整性驗證），則驗證解密過程不應產出明文。

NIST 允許提交者可以因外部參數（例如密鑰長度或隨機亂數長度）或內部參數（例如回合數）的不同，提交一系列 AEAD 演算法（稱為家族），唯不得超過 10 個家庭成員。AEAD 演算法應接受所有滿足輸入長度要求的位元組串列輸入，如果有所限制，提交的內容內則應包括任何長度限制的理由。

在安全性規範方面，NIST 要求 AEAD 演算法至少要有一家族成員的金鑰長度在 128 位元以上，隨機亂數長度在 96 位元以上，標籤（Tag）長度在 64 位元以上。對 AEAD 演算法的密碼分析攻擊（Cryptanalysis）方面，至少需要在傳統電腦上進行至少 2,112 次運算以破解 128 位元金鑰，而且必須能抵抗自適應選擇明文攻擊（Adaptive Chosen-Plaintext Attacks），以確保機密性；密文必須能抵抗自適應偽造嘗試（Adaptive Forgery Attempts）攻擊，以確保完整性。另外，NIST 對 AEAD 的隨機亂數也有以下之要求：

1. 只要隨機亂數不在同一密鑰下重複使用，AEAD 演算法必須具備安全性；
2. 當隨機亂數必須重複使用時，必須說明其可能造成的安全損失；
3. 如果隨機亂數不會因重複使用而失去所有安全性，則提交的演算法可以將其宣告為其演算法的特徵。

（二）雜湊函數規範

雜湊函數是具有一個位元組串列輸入和一個位元組串列輸出的函數。其輸入是可變長度的訊息，而輸出則是一個固定長度的雜湊值。NIST 要求雜湊函數應接受所有滿足指定最大訊息長度的位元組串列輸入，如果有所限制，則提交的內容應包括任何長度限制的理由。

NIST 允許提交者可以因外部參數（例如最大訊息長度及輸出大小）或內部參數（例如回合數）的不同，提交一系列雜湊函數，唯該系列的家族成員不得超過 10 個，另該家族應包含一個具有 256 位元組以上輸出的主要成員，而該成員的訊息長度輸入限制不得小於 250-1 位元組。

在安全性規範方面，NIST 要求雜湊函數不得指定小於 256 位元的輸出，而對雜湊函數的密碼分析攻擊至少需要在傳統電腦上進行 2,112 次運算。此外，若要找到此雜湊函數的碰撞（Collision），在計算上應是不可行的。NIST 還要求雜湊函數應該能夠抵抗長度擴展攻擊（Length Extension Attacks），亦即攻擊者應無法創造一個內含其他訊息雜湊值的新的訊息雜湊值。在一些實際應用中，雜湊函數可能還需要滿足其他安全屬性，應在提交雜湊函數時對所有安全屬性進行描述。

（三）設計規範

在設計方面，NIST 要求提交的內容應說明 AEAD 和雜湊演算法有哪些共同的設計組件，並解釋這些共同組件如何降低實現成本；AEAD 演算法和雜湊函數演算法必須在具有低容量的 RAM 和 ROM 硬體以及嵌入式軟體中實現，並且能支援各種實現策略（包括低能耗 [Energy Consumption]、低功耗 [Power Consumption]、低延遲）及 8 位元、16 位元和 32 位元的微控制器架構。為了滿足某些資源嚴格受限的環境，提交的演算法可能發生無法滿足上述的所有性能要求狀況，NIST 允許設計者在各種性能要求之間進行權衡，並在提交文件中加以解釋已識別的瓶頸及採取的權衡策略。

NIST 要求每個提交都應附有可移植的 C 語言參考軟體的實現，以支援大眾對演算法的理解、密碼分析及實現驗證等。此參考實現應易於理解，並且不應包含僅用於優化某些平臺性能的程式碼，其正確性也應在 NIST 測試平臺上進行驗證。

三、評估標準

評估的目的是在驗證提交的項目是否符合上一節所描述的最低可接受度規範。在安全的評估方面，將對提交項目進行所有已知統計式攻擊及旁道攻擊的評估；在成本評估方面，將對提交的項目酌情依據各種成本指標進行評估，例如晶片面積、記憶體大小及能耗等；在性能評估方面，則將對提交的項目酌情依據各種性能指標進行評估，例如延遲、吞吐量（Throughput）、功耗等。

NIST 同時在徵求文件中同時提到，具有以下條件的提交項目將被優先選擇成為標準：（一）具有重要第三方（Third-Party）分析或利用現有標準組件；（二）同時在硬體及軟體雙方面實現均有優異表現；（三）在資源高度受限制的硬體實現表現出色。

在評估程序方面，NIST 會組成一個由 NIST 研究人員所組成的內部遴選小組來分析所

有提交的內容。另外，NIST 也會公開所有提交內容及其內部評估結果，並強烈鼓勵大眾對提交文件進行評估並公布結果。NIST 將考慮其自己的分析以及收到的公眾意見做出最後的決定。目前 NIST 已將前二輪評選結果以 NISTIR 8268（Turan, McKay, Çalık, Chang, and Bassham, 2019）及 NISTIR 8369（Turan et al., 2021）文件發布。

四、最終回合演算法分析

NIST 根據其內部的評選及外部的回饋意見，從第二輪 32 個入圍的演算法中評選出 10 個演算法進入最後決選，分別是 ASCON、Elephant、GIFT-COFB、Grain-128AEAD、ISAP、PHOTON-Beetle、Romulus、SPARKLE、TinyJAMBU 和 Xoodyak 等。其中有四個提交項目同時提供 AEAD 及雜湊函數演算法，分別為 ASCON、PHOTON-Beetle、SPARKLE 及 Xoodyak 等，其餘六個提交項目只提交 AEAD 演算法，如表 3 所示。

表 3　最終回合入選演算法的提交功能

分類	演算法
同時提供 AEAD 及雜湊函數	ASCON、PHOTON-Beetle、SPARKLE、Xoodyak
只提供 AEAD	Elephant、GIFT-COFB、Grain-128AEAD、ISAP、Romulus、TinyJAMBU

資料來源：作者自行研究整理。

另依操作模式不同，可將上述 10 個演算法區分為海綿結構（Sponge）、區塊加密、加密後訊息認證（Encrypt-then-MAC）及串流加密等四類，各演算法的歸屬如表 4 所示。因篇幅的關係，有興趣的讀者可參閱第二輪評選結果文件 NISTIR 8369 以獲得更細部的區分方式、安全性分析及效能分析。

表 4　最終回入選演算法操作模式

分類	演算法
海綿結構	ASCON、PHOTON-Beetle、SPARKLE、TinyJAMBU、Xoodyak
區塊加密	GIFT-COFB、Romulus
加密後訊息認證	Elephant、ISAP
串流加密	Grain-128AEAD

資料來源：作者自行研究整理。

肆、輕量級密碼法 ASCON 之實現

在前面的內容中，我們分別介紹了物聯網對輕量級密碼學之需求以及 NIST 對輕量級密

碼學演標準徵求之項目、規範及評估標準。在本節中，我們將以最後決選的 10 個演算法中的 ASCON 演算法作為範例進行實現，以說明如何在一個物聯網的應用中實現一個符合輕量級密碼學標準的演算法，以保護物聯網中訊息傳遞的安全。

一、ASCON 家族與實現標的

ASCON 在 2014 至 2019 年 CAESAR 競賽中被評選為輕量級應用（Lightweight Applications）的首選加密項目（CAESAR, 2019），這也是本文選擇其作為範例進行實現的原因之一。ASCON 代表的是一輕量級密碼學套件（Suite），如圖 7 所示，主要包含 AEAD 及雜湊函數兩類，其中 AEAD 又區分為 Ascon-128、Ascon-128a 及抗量子金鑰搜尋的 Ascon-80pq 三種版本，而雜湊函數則有 Ascon-Hash 跟 Ascon-Xof 等兩類。ASCON 最後提交到 NIST 的標準文件中的版本為 Ascon v12，主要包括 Ascon-Hash128 及 Ascon-Hash128a 的 AEAD 及分別搭配其使用的雜湊函數 Ascon-Hash128 及 Ascon-Hash128a。本文以 Ascon-128 為範例進行實現。

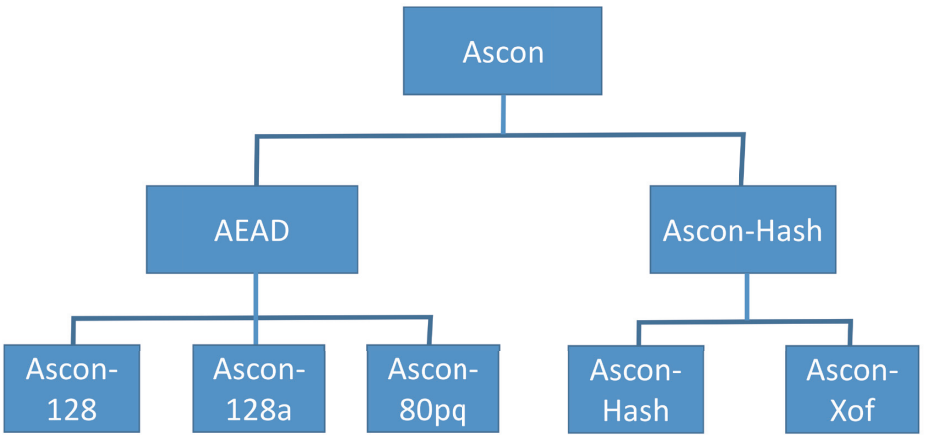


圖 7　ASCON 家族

資料來源：作者自行研究整理。

Asocn-128 的 AEAD 加密函數共有 4 個輸入及 2 個輸出，其方塊圖如圖 8 所示。其中輸入參數 K、N、A、P 分別表示 128 位元秘密金鑰、128 位元隨機亂數、任意長度關聯資料、任意長度明文資料，而輸出參數 C 跟 T 則分別表示輸出之密文及標籤。AEAD 的解密函數共有 5 個輸入及 1 個輸出，其方塊圖如圖 9 所示。其中 K、N、A、C 及 T 為輸入資料，其表示意義跟加密函數中使用的參數相同；而輸出則有兩種可能，如果是明文 P，表示密文為合格的密文，否則輸出錯誤訊息「⊥」。

圖 8　Ascon-128 加密函數方塊圖

資料來源：作者自行研究整理。

圖 9　Ascon-128 解密函數方塊圖

資料來源：作者自行研究整理。

二、物聯網應用架構

圖 10 為本文實現的物聯網應用架構（以下簡稱本實現）。本實現採用以 ESP8266 為核心的 WEMOS D1 mini 作為資源受限的設備，其具備 32 位元 CPU、64 KB 啟動唯讀記憶體（Boot ROM）、96 KB 的 RAM，並提供 4 MB 外部快閃記憶體（Flash ROM），可以儲存作業系統與應用程式，是具備較多資源的微控制器。本實現之所以採用 D1 mini 作為資源受限的設備有兩大原因，第一是其具備 WiFi 的無線上網功能，另外還支援 Python 程式執行的作業系統 MicroPython，有利於物聯網應用的快速開發。

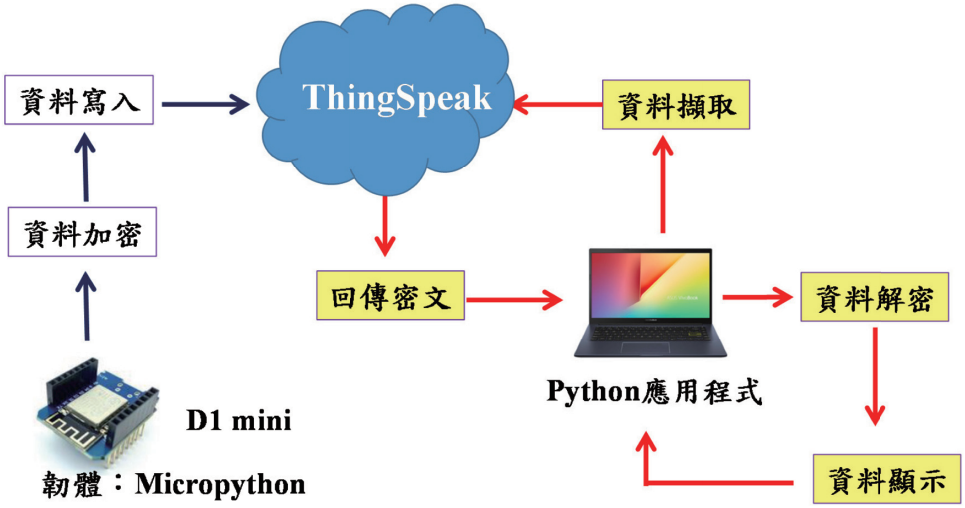


圖 10　本文所實現的物聯網加密架構

資料來源：作者自行研究整理。

本實現假設在 D1 mini 上的 Python 應用程式每 20 秒會接收來自溫度感測器的數據，並利用在 D1 mini 上實現的 Ascon-128 加密函數將數據資料進行加密，並上傳至 ThingSpeak 雲端資料庫中。而使用者可利用本實現所開發的視窗版 Python 應用程式，從 ThingSpeak 雲端資料庫擷取加密後的資料，並透過應用程式中的 Ascon-128 解密功能進行解密，將結果呈現在視窗畫面上。

三、Ascon-128 加解密函數實現

Ascon-128 加密程序使用的是海綿結構，區分成四個階段，如圖 11 所示。在初始階段（Initialization），首先將金鑰 *K*、亂數 *N*，及初始值 IV 初始化成 320 位元，接著在關聯資料處理階段將關聯資料 *A* 區分為若干 64 位元區塊，每個區塊再透過排列運算進行混合，稱為吸收（Absorb）；接著在明文處理階段陸續擠出（Squeeze）64 位元資料，並與每一個 64 位元明文區塊進行 XOR 運算，將其加密成密文區塊；在最後階段（Finalization）加入金鑰 *K* 以產出標籤 *T*。

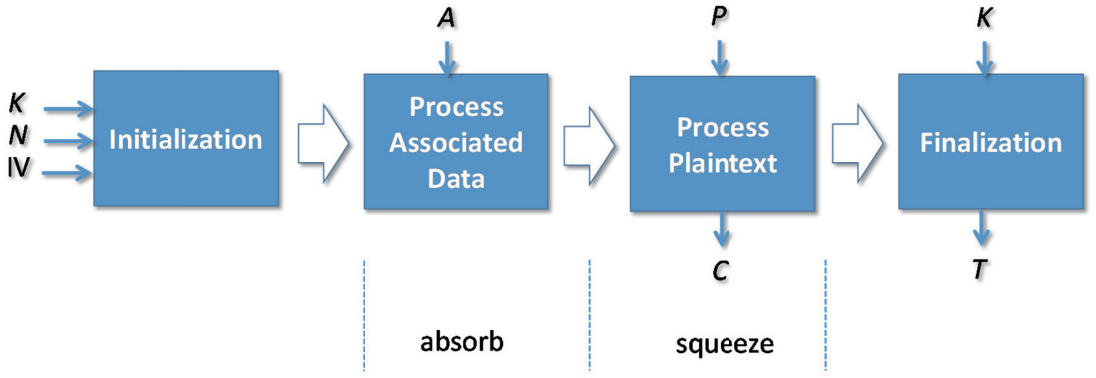


圖 11　Ascon-128 加密程序示意圖

資料來源：作者自行研究整理。

以下 Python 程式為在 D1 mini 上實現的 Ascon-128 加密函數 ascon_encrypt，其輸入（key, nonce, associateddata, plaintext）分別表示金鑰、隨機亂數、關聯資料及明文等四個位元組串列。程式在指定資料的初始狀態（ascon_initialize）後，陸續進行 Ascon-128 加密程序中的四個階段，其對應函數分別為 ascon_initialize、ascon_process_associated_data、ascon_process_plaintext 及 ascon_finalize，而最後的輸出則是結合密文及標籤的唯一位元組串列輸出，與 NIST 對 AEAD 加密函數的規範相符。

```
def ascon_encrypt(key, nonce, associateddata, plaintext):
    S = [0, 0, 0, 0, 0] # initial state
    ascon_initialize(S, key, nonce)
    ascon_process_associated_data(S, associateddata)
    ciphertext = ascon_process_plaintext(S, plaintext)
    tag = ascon_finalize(S, key)
    return ciphertext + tag
```

Ascon-128 的解密程序如圖 12 所示，除了第三個階段（Process Ciphertext）將密文解出明文外，其餘部分均相同。

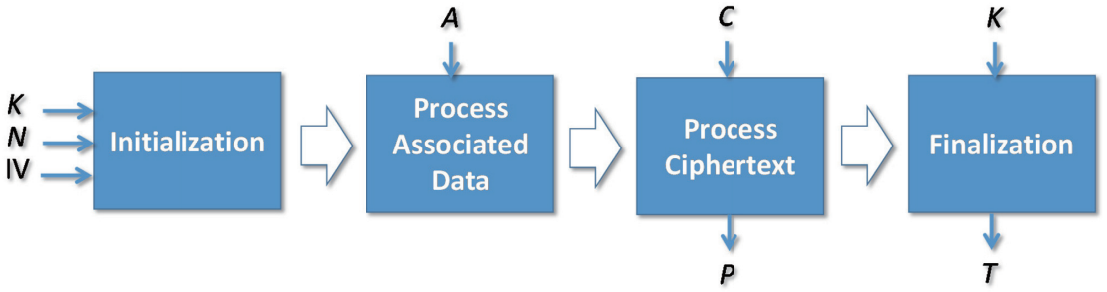


圖 12　Ascon-128 解密程序示意圖

資料來源：作者自行研究整理。

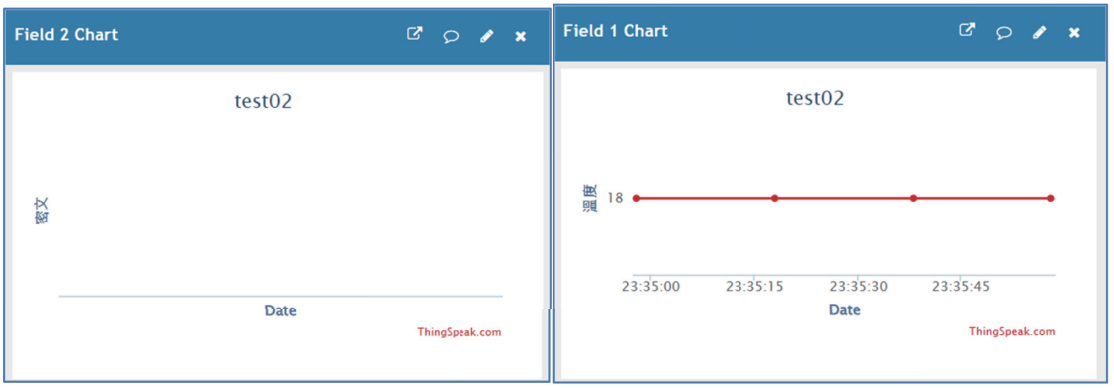
以下為使用者端以 Python 實現的 Ascon-128 解密函數 ascon_decrypt。AEAD 解密函數共有四個輸入，其中 ciphertext 為包含實際的密文及標籤的位元組串列。在指定資料的初始狀態（ascon_initialize）後，陸續進行 Ascon-128 解密程序中的四個階段，前三個階段的對應函數分別為 ascon_process_associated_data、ascon_process_ciphertext 及 ascon_finalize，最後階段則將 ascon_finalize 函數的輸出 tag 與輸入密文的標籤部分 ciphertext[-16:] 進行比對，如果相符，表示密文驗證成功，並輸出明文，否則無明文輸出，與 NIST 對 AEAD 的解密函數規範相符。

```
def ascon_decrypt(key, nonce, associateddata, ciphertext):
    S = [0, 0, 0, 0, 0] #initial state
    ascon_initialize(S, key, nonce)
    ascon_process_associated_data(S, associateddata)
    plaintext = ascon_process_ciphertext(S, ciphertext[:-16])
    tag = ascon_finalize(S, key)
    if tag == ciphertext[-16:]:
        return plaintext
    else:
        return None
```

四、實驗結果

由於 ThingSpeak 免費版本限制上傳資料間隔不得小於 15 秒，因此本實驗規劃在 D1 mini 上的 Python 應用程式每 20 秒接收溫度感測器的數據，並利用在 D1 mini 上實現的 Ascon-128 加密函數將數據資料進行加密，並上傳至 ThingSpeak 雲端資料庫中。ThingSpeak

提供網頁式介面，可提供使用者在 ThingSpeak 頻道中以圖形方式顯示上傳的數據資料。由於我們上傳的是加密的資料，無法以視覺化方式呈現，因此，本實驗在上傳加密資料時會另外上傳未加密的資料，以提供解密端進行解密結果正確性的比對。圖 13(a) 及圖 13(b) 分別顯示已上傳至 ThingSpeak 的三筆加密及未加密資料，其中圖 13(a) 因是加密後的資料，無法正常呈現數值圖形，而圖 13(b) 則是其相對之未加密資料，可以明確顯示其上傳的數值，其表示在 D1 mini 上實現的加密及上傳的功能正常。此時使用者可以利用本實現所完成的視窗版 Python 應用程式即時從 ThingSpeak 雲端資料庫擷取密文及解密，如圖 14 所示，其顯示成功所擷取的密文資料及解密之結果。



(a) (b)

圖 13　在 ThingSpeak 中呈現的 (a) 加密資料及 (b) 未加密資料

資料來源：作者自行研究整理。

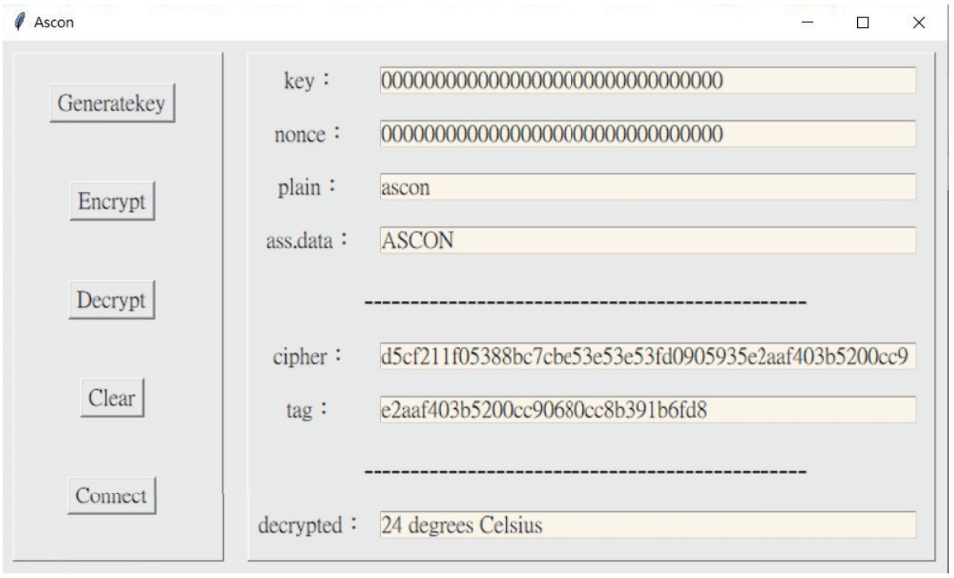


圖 14　包含 Ascon-128 的加密與解密程式的視窗畫面

資料來源：作者自行研究整理。

伍、結論

物聯網的普及為生活帶來便利，但也因為物聯網設備公開部署及無線傳輸的特性，使得在物聯網中所傳輸的資料容易受到攔截及節點竄改，進而導致生命和金錢的重大損失。為了加強物聯網資料傳遞的安全性，利用密碼學技術對傳輸資料進行加密及識別則為可行的解決方法。然而對於 RFID 標籤或智慧卡等資源高度受限的設備而言，AES 這樣的傳統加密標準難以或不可能在這樣的設備中實現，即使勉強實現，它們的性能也可能無法被接受，因此發展可在資源受限的設備順利執行的輕量級密碼學演算法實有其必要及迫切性。

有鑑於輕量級密碼學演算法在物聯網中的重要性，NIST 於 2013 年即啟動了輕量級密碼學標準計畫，其目的在瞭解輕量級密碼學標準的需求，並針對資源有限的設備制定輕量級密碼學演算法的標準，於 2018 年正式向世界徵求輕量級密碼學演算法，經過二輪的篩選，確立了 10 個演算法進入最後的決選，預期於 2022 年在開完第五次輕量級密碼學研討會後確立輕量級密碼學標準。

在本文中，我們首先說明傳統密碼演算法在資源受限設備上實現的限制，及輕量級密碼學演算法需求的重點，以此為基礎，說明 NIST 目前制定輕量級密碼學標準的發展，其中包括對輕量級密碼學演算法的需求項目、規範及評估標準等，最後以目前進入最後決選的 ASCON 輕量級加密演算為範例進行實現，說明如何在一個物聯網的應用中實現一個符合輕量級密碼學標準的演算法，以保護物聯網中訊息傳遞的安全。期望讀者在閱讀完本文後，對輕量級密碼學演算法的需求、標準制定過程和規範及未來在物聯網上應用的瞭解有所幫助。

參考文獻

Bassham L., Çalık C., McKay K., Mouha N., and Turan M. S., 2017, “Profiles for the Lightweight Cryptography Standardization Process (Draft White Paper),” *NIST*, https://csrc.nist.gov/CSRC/media/Publications/white-paper/2017/04/26/profiles-for-lightweight-cryptography-standardization-process/draft/documents/profiles-lwc-std-proc-draft.pdf (accessed February 10, 2022).

CAESAR, 2019, “Cryptographic Competitions,” https://competitions.cr.yp.to/caesar-submissions.html (accessed February 10, 2022).

Evans D., 2011/4, “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything,” *Cisco*, https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf (accessed February 10, 2022).

Ge M., Hong J. B., Alzaid H., and Kim D. S., 2017, “Security modeling and analysis of cross-protocol IoT devices,” paper presented at *the IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (TrustCom), Sydney, Australia. doi:10.1109/Trustcom/BigDataSE/ICESS.2017.350

Hatzivasilis G., Fysarakis K., Papaefstathiou I., and Manifavas C., 2018, “A Review of Lightweight

Block Ciphers," *Journal of Cryptographic Engineering*, 8(2), 141-184. doi:10.1007/s13389-017-0160-y

Kamal R., 2017, *Internet of Things: Architecture and Design Principles*, Chennai, India: McGraw Hill Education.

"Lightweight Cryptography Workshop 2015," 2015, https://csrc.nist.gov/events/2015/lightweight-cryptography-workshop-2015 (accessed February 10, 2022).

"Lightweight Cryptography Workshop 2016," 2016, https://csrc.nist.gov/events/2016/lightweight-cryptography-workshop-2016 (accessed February 10, 2022).

McKay K. A., Bassham L., Turan M. S., and Mouha N., 2016, *Report on Lightweight Cryptography* (DRAFT NISTIR 8114), https://csrc.nist.gov/CSRC/media/Publications/nistir/8114/draft/documents/nistir_8114_draft.pdf (accessed February 10, 2022).

McKay K. A., Bassham L., Turan M. S., and Mouha N., 2017, *Report on Lightweight Cryptography* (NISTIR 8114), https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf (accessed February 10, 2022).

National Institute of Standards and Technology, 2018a, "Draft Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process," https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/Draft-LWC-Submission-Requirements-April2018.pdf (accessed February 10, 2022).

National Institute of Standards and Technology, 2018b, "Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process," https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf (accessed February 10, 2022).

National Institute of Standards and Technology, 2019, "Lightweight Cryptography Round 1 Candidates," https://csrc.nist.gov/Projects/lightweight-cryptography/round-1-candidates (accessed February 10, 2022).

National Institute of Standards and Technology, 2021a, "Lightweight Cryptography Round 2 Candidates," https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates (accessed February 10, 2022).

National Institute of Standards and Technology, 2021b, "Lightweight Cryptography Finalists," https://csrc.nist.gov/Projects/lightweight-cryptography/finalists (accessed February 10, 2022).

"OWASP Internet of Things Top 10," n.d., https://owasp.org/www-project-internet-of-things-top-10 (accessed February 10, 2022).

Ronen, E., and Shamir, A., 2016, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," paper presented at *the IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbruecken, Germany. doi:10.1109/EuroSP.2016.13

Singh S., Sharma P. K., Moon S. Y., and Park J. H., 2017, "Advanced Lightweight Encryption Algo-

rithms for IoT Devices: Survey, Challenges and Solutions," *Journal of Ambient Intelligence & Humanized Computing*. doi:10.1007/s12652-017-0494-4

Turan M. S., McKay K. A., Çalık Ç., Chang D., and Bassham L, 2019, *Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process* (NISTIR 8268). doi:10.6028/NIST.IR.8268

Turan M. S., McKay K., Chang D., Çalık Ç., Bassham L., Kang J., and Kelsey J., 2021, *Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process* (NISTIR 8369). doi:10.6028/NIST.IR.8369

Vikas, Sagar B. B., and Munjul M., 2021, "Security Issues in Wireless Sensor network—A survey," *Journal of Discrete Mathematical Sciences and Cryptography*, 24(5), 1415-1427. doi:10.1080/09720529.2021.1932937

Zhao K., and Ge L., 2013, "A survey on the internet of things security," paper presented at *the Ninth International Conference on Computational Intelligence and Security*, Emeishan, China. doi:10.1109/CIS.2013.145