前瞻科技與管理　13 卷 1 期，1-20 頁（2024 年 11 月）
Journal of Advanced Technology and Management Vol. 13, No. 1, pp. 1-20 (November, 2024)
DOI:10.6193/JATM.202411_13(1).0001

NAES-512：抗量子計算 512 位元先進密碼技術之設計與 FPGA 實現

劉江龍 [1,*]　張凱博 [2]

[1] 國防大學理工學院電機電子工程學系教授
[2] 國防大學理工學院電機電子工程學系研究生

摘 要

由於量子電腦的快速發展及 Grover 量子演算法的威脅，使用 128 位元金鑰的先進加密標準（Advanced Encryption Standard, AES）技術（AES-128）在具有足夠量子位元的量子電腦問世後，將不再具備計算上的安全性，而使用 256 位元金鑰的 AES-256 則必須提升到 512 位元金鑰才能具備同樣的安全性，然而目前的 AES 標準內並無 AES-512 的規範。本文提出可相容於 AES-256 的 AES-512 加解密技術（以下簡稱為 NAES-512）以解決上述之問題。NAES-512 採用 AES-256 架構，並使用新的金鑰擴展演算法以產生加密及解密所需的回合金鑰；本研究同時利用 Verilog 硬體描述語言實現 NAES-512 相關模組設計。實驗結果證明，NAES-512 採用之金鑰擴展演算法具有良好的效率及安全性，也可有效對 AES-256 產生的密文進行解密；另外，NAES-512 可正確在 FPGA 開發板上進行加密及解密，可作為量子計算威脅下，高安全加密技術設計之參考。

關鍵詞：先進加密標準、抗量子計算、對稱式加密、區塊密碼法、NAES-512

[*] 通訊作者：劉江龍
電子郵件：chianglung.liu@gmail.com
（收件日期：2024 年 6 月 6 日；修正日期：2024 年 7 月 9 日；接受日期：2024 年 7 月 17 日）

Journal of Advanced Technology and Management Vol. 13, No. 1, pp. 1-20 (November, 2024)
DOI:10.6193/JATM.202411_13(1).0001

NAES-512: Design and FPGA Implementation of Quantum-Resistant 512-Bit AES

Chiang-Lung Liu[1,*], Kai-Po Chang[2]

[1]Professor, Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University
[2]Postgraduate, Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University

Abstract

Due to the rapid development of quantum computers and the threat of Grover's quantum algorithm, the AES (Advanced Encryption Standard) encryption technology (AES-128) using a 128-bit key will no longer be computationally efficient after the advent of quantum computers with sufficient qubits. Moreover, AES-256 using 256-bit key must be upgraded to a 512-bit key to maintain the same security. However, there is no AES-512 specification included in AES standard. This article proposes an AES-512 encryption and decryption technology based on AES-256 (called NAES-512) to solve the above problems. NAES-512 adopts the AES-256 architecture and uses a new key expansion algorithm to generate the round keys required for encryption and decryption. This study also uses Verilog to implement NAES-512 related modules. Experimental results prove that the key expansion algorithm used by NAES-512 has good efficiency and security, and can effectively decrypt ciphertext generated by AES-256. Experimental results also show that NAES-512 can correctly perform encryption and decryption on the FPGA development board, and can be used as a reference for design of high-security encryption technology under the threat of quantum computing.

Keywords: Advanced Encryption Standard, quantum-resistant cryptography, symmetric encryption, block cipher, NAES-512

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

壹、前言

由於電腦及網際網路的快速發展，使用網路進行數位資料交流已成為人們生活的日常。在享受科技帶來便利的同時，若有心人士利用此環境進行機密資料的竊取，可能導致難以估計之損失，如何確保資料儲存及傳輸安全已成為資訊時代的重要課題。近代密碼技術可利用數學演算法將電腦的數位資料進行轉換，得到另一筆完全異於原本文件的內容；在接收端則可利用相對演算法及特定金鑰（Key），將原始數位資料還原。對於具有機密性質的資料而言，此為最簡單的保護措施。密碼演算法可概分為對稱式（Symmetric-Key）及非對稱式（Asymmetric-Key）兩類。傳統上認為對於每秒可窮舉 10^{12} 金鑰的暴力攻擊（Brute-Force Attack）而言，具備 128 位元的對稱式密碼系統（例如 128 位元的先進加密標準 [Advanced Encryption Standard, AES]）及同等安全性的非對稱式密碼系統（例如 256 位元的 Elliptic Curve Cryptography [ECC] 及 3,072 位元的 Rivest-Shamir-Adleman [RSA]）被認為具有相當的安全性（Stallings, 2010）；然而由於近年來量子計算技術的進步，在面對量子計算的威脅時，上述密碼系統的安全則受到嚴重的考驗（劉江龍，2019）。舉例來說，目前廣為大眾使用的 AES（National Institute of Standards and Technology, 2023）在 Grover（1997）演算法的威脅下，只剩下一半金鑰長度的安全性（劉江龍，2019）。此意謂當量子電腦普遍後，使用者只能使用 256 位元金鑰的 AES（以下簡稱 AES-256），以維持等同於 AES-128 計算上的安全性。但是對於目前使用 256 位元金鑰的 AES-256 使用者而言，因為在 AES 標準中並無使用 512 位元金鑰的演算法，導致目前使用 AES-256 的系統，在面對量子計算的挑戰時，則無更進一階的加密演算法，以維持原有 256 位元安全性的優勢。面對這個問題，使用者可能使用不同的 2 把 256 位金鑰，並進行 2 次 AES-256 的加密，但由於存在中途相遇攻擊（Meet in the Middle Attack）（Stallings, 2010）的緣故，此舉並無法提高金鑰長度的安全性；而使用 3 把 256 位元金鑰並透過 3 次 AES 加密方式，雖能將 AES 的安全性提高至等同於 512 位元金鑰的安全性，但必須額外增加 2 倍加密及解密時間，對於有即時通訊需求的系統而言，有可能形成通訊上的瓶頸。因此，設計可提供 512 位元金鑰安全性及相容於 AES 標準的加密技術，在量子計算技術快速進步及高通訊吞吐量需求的今天，實有其必要性及急迫性。

針對上述問題，Moh'd, Jararweh, and Tawalbeh（2011）提出利用 AES-128 的加密架構所發展的 512 位元金鑰的密碼技術（以下簡稱 AES-512），其方法是將 AES-128 的明文（Plaintext）輸入的長度改為 512 位元，並將 AES 內部的操作所使用的狀態矩陣（State），由 4 × 4 改為 8 × 8 的位元組矩陣；另依據 AES 的 128 位元的金鑰擴展（Key Expansion）方式，產生 512 位元的 11 把回合金鑰（Round Key），以完成加密所需的 11 次加入回合金鑰（Add Round Key）運算。由於 Moh'd et al. 所提方法是採用 AES-128 架構進行 AES-512 設計，因此在金鑰長度較長狀況下，利用原先AES-128的金鑰擴展方式，存在有安全性上疑慮。除此之外，使用 AES-512 後，使用者將無法對原來利用 AES-256 加密的密文（Ciphertext）進行解密，因此，必須同時維持兩套系統，造成維護上的負擔。為解決上述問題，本文提出一個簡單且相容於 AES-256 的新 AES-512 密碼法（以下簡稱 NAES-512），使其可以利用 AES-256 演算法進行加密及解密，同時具有 512 位元金鑰的安全性，其優點如下：

一、原本使用 AES-256 系統的用戶，可以使用 NAES-512 演算法對舊有 AES-256 所產生的密文進行解密。換言之，使用者可以將舊有的 AES-256 密碼系統直接遷移到 NAES-512 系統，不需要同時維持兩個密碼系統。

二、使用者可以輸入 512 位元金鑰 NAES-512 系統進行加解密，在面對 Grover（1997）演算法的威脅下，仍具有 256 位元的安全性。

三、實作簡單。由於 NAES-512 採用 AES-256 加解密架構，並未增加加密及解密的複雜度，使得 NAES-512 可以在原有的軟硬體上實現，減少密碼系統遷移時的支出。

NAES-512 採用新的金鑰擴展演算法，允許輸入 512 位元的主金鑰（Master Key），來產生 AES-256 各回合加密及解密所需要的回合金鑰。因此，在面對 Grover（1997）演算法的攻擊下，具有 512 位元的金鑰空間（亦即具備 2^{512} 種可能金鑰），因此，即使具備 512 個量子位元以上量子電腦發展完成，仍具有 256 位元傳統電腦的安全性，可有效抵抗量子計算的攻擊。為了維持金鑰的安全性，NAES-512 在金鑰擴展的過程中額外進行字組替換運算，以強化金鑰的雪崩效應（Avalanche Effect）（Stallings, 2010）。對於攻擊者而言，將面臨比 AES-256 更複雜的金鑰結構，加深破密的困難度。

為完整說明 NAES-512，本文其餘各章節安排如下：由於 NAES-512 是基於 AES-256 的架構發展，因此在第貳章先說明 AES-256 密碼法的架構及金鑰擴展演算法；第參章說明 NAES-512 所採用的金鑰擴展演算法；第肆章為 NAES-512 在 FPGA 實現的說明及實驗結果；第伍章為 NAES-512 的安全性分析；第陸章為結論。

貳、AES-256 架構及金鑰擴展演算法

AES 是美國國家標準暨技術研究院（National Institute of Standards and Technology, NIST）於 2001 所公布的先進資料加密標準，依據使用金鑰位元長度的不同，區分為使用 128 位元金鑰的 AES-128、使用 192 位元金鑰的 AES-192，及使用 256 位元金鑰的 AES-256 等三種演算法，其中由於 AES-256 的金鑰空間為 2256，因此對暴力攻擊的抵抗能力最佳。

一、AES-256 架構

AES-256 的加密架構如圖 1 所示，使用多回合簡單的替代及排列架構，以達到一個安全密碼演算法所需要的混淆（Confusion）及擴散（Diffusion）效果。AES-256 採用 14 回合（Round 1 ~ Round 14）設計，如圖 1 左半部所示。長度為 16 個位元組的明文先經過加入回合金鑰運算，再進到回合 1（Round 1）至回合 14（Round 14）進行回合運算，最後輸出密文。除了第 14 回合外，AES-256 的每個回合的設計均相同，其架構如圖 2 所示。包括四個運算單元，依序為位元組替換（Substitute Bytes）、列位移（Shift Rows）、行混合（Mix Columns）及加入回合金鑰等，但在第 14 回合少了行混合運算。在四個運算當中，加入回合金鑰運算需要回合金鑰參與，加上 14 個回合運算之前的加入回合金鑰運算，共需要 15 把不同的回合金鑰（Round-Key 0 ~ Round-Key 14），其是透過 AES-256 的金鑰擴展演算法產生，如圖 1 右半部所示，下一節將詳細說明 AES-256 的金鑰擴展演算法。

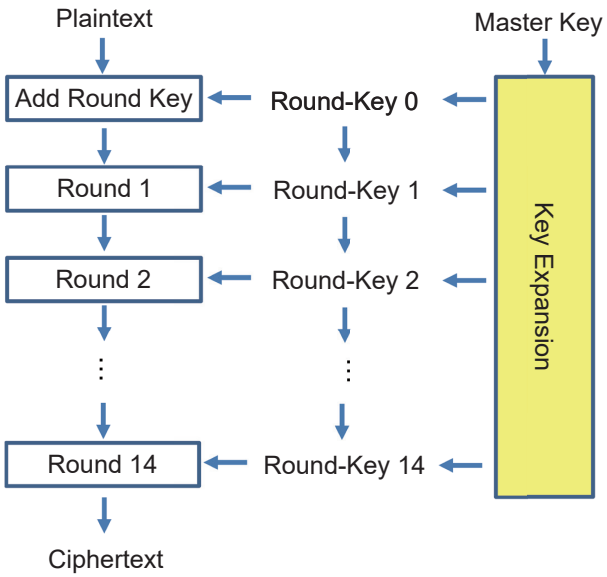

圖 1　AES-256 加密架構

資料來源：作者自行研究整理。

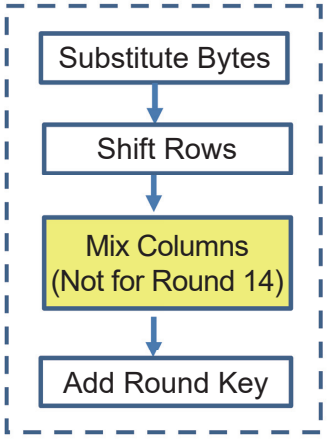

圖 2　AES-256 加密演算法的回合架構

資料來源：作者自行研究整理。

AES-256 解密架構是 AES-256 加密演算法的逆運算，如圖 3 所示。密文經加入回合金鑰運算後，再經過 14 回合逆運算，最後輸出明文。各回合也是由 4 個運算組成（如圖 4 所示），分別為逆列位移（Inverse Sift Rows）、逆位元組替換（Inverse Substitute Bytes）、加入回合金鑰及逆行混合（Inver Mix Columns）等運算，但是第 14 回合少了逆行混合。由於 AES-256 的解密架構是加密架構的逆運算，因此，也包含了 15 個加入回合金鑰運算，但是回合金鑰的供給順序是倒過來的，也就是第 1 個加入回合金鑰運算使用 Round-Key 14，Round 1 的加入回合金鑰運算使用 Round-Key 13，依此類推，最後加入回合金鑰運算使用 Round-Key 0 等。

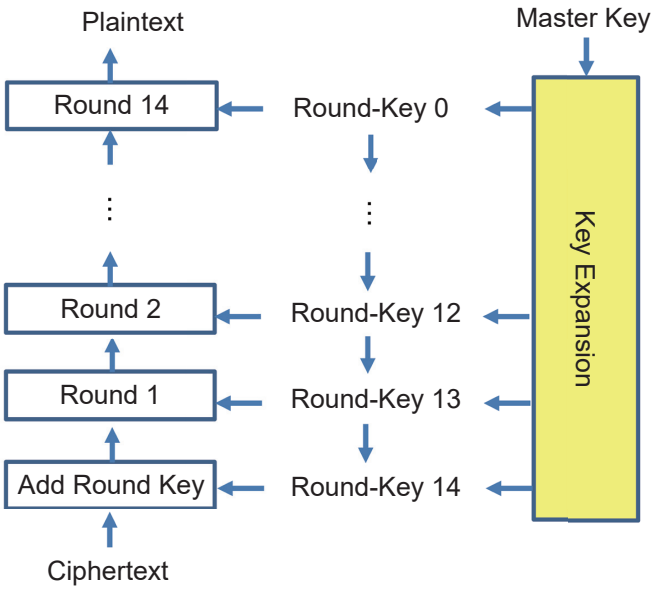

圖 3 AES-256 解密架構

資料來源：作者自行研究整理。

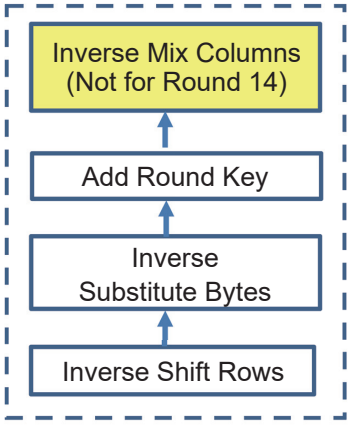

圖 4 AES-256 解密演算法的回合架構

資料來源：作者自行研究整理。

二、AES-256 回合金鑰擴展演算法

AES-256 的加密及解密過程均需要 15 個回合金鑰的參與，如圖 5 所示。為安全的產生這 15 個回合金鑰，AES-256 以 4 個位元組為單位（稱為字組），透過金鑰擴展演算法將 256 位元（共 8 個字組）的主金鑰，擴展成 15 個回合金鑰所需的 60 個字組（$w[0] \sim w[59]$）。其中每 4 個字組（共 128 位元）構成一把回合金鑰，Round-Key 0 使用 $w[0] \sim w[3]$，Round-Key 1 使用 $w[4] \sim w[7]$，依此類推，Round-Key 14 則使用 $w[56] \sim w[59]$。

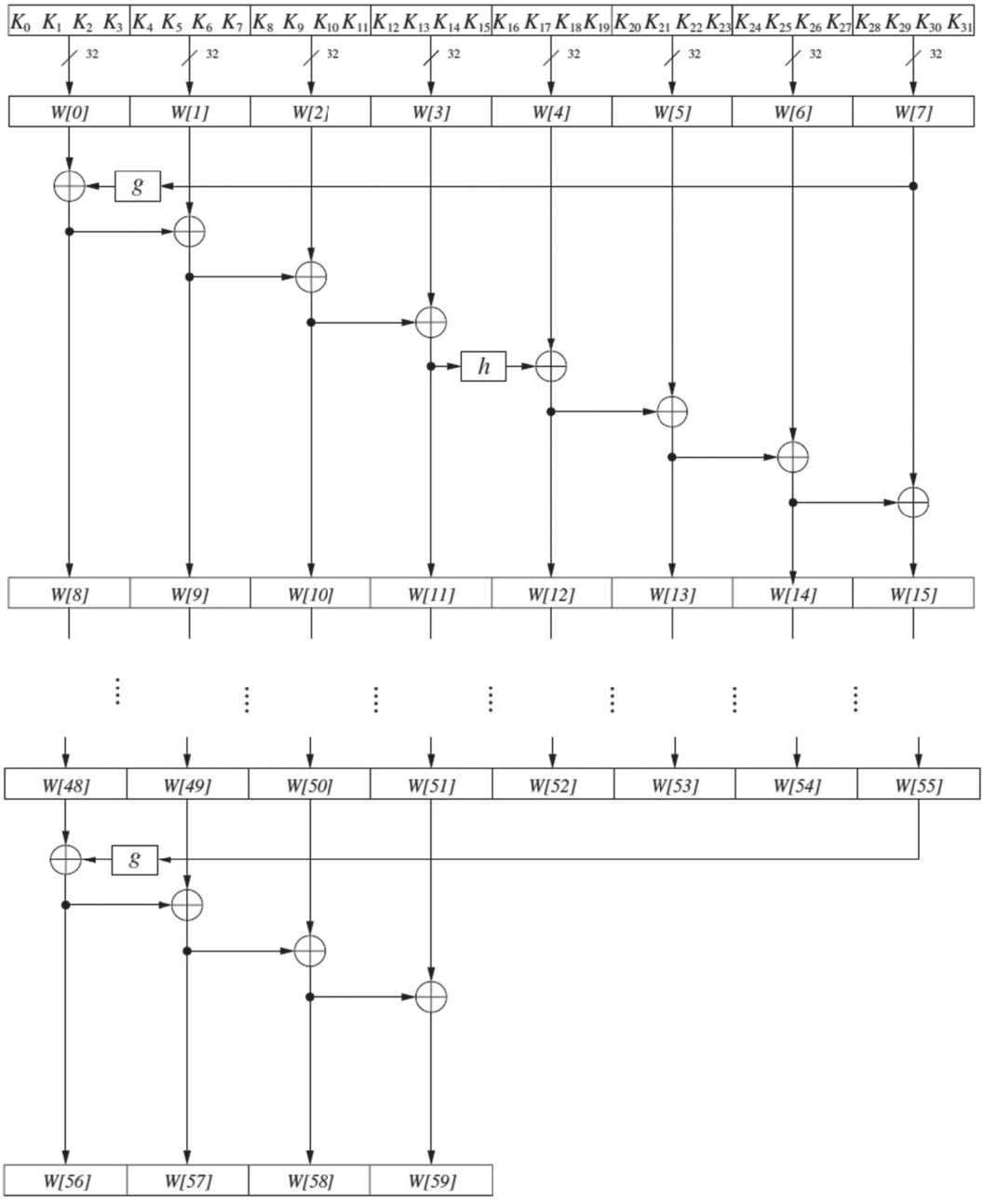

圖 5　AES-256 金鑰擴展示意圖

資料來源：作者自行研究整理。

AES-256 的主金鑰共有 32 個位元組，可表示成 8 個字組（$w[0] \sim w[7]$），其構成前 2 把回合金鑰，接著就以 $w[0] \sim w[7]$ 為基礎，經過 7 個回合金鑰擴展，每回合擴展 8 個字組，逐步擴展生成加密及解密所需要的所有回合金鑰。首先利用 $w[0] \sim w[7]$ 擴展成 $w[8] \sim w[15]$，再利用 $w[8] \sim w[15]$ 擴展成 $w[16] \sim w[23]$，依此類推，直到完成加密所需的 15 個回合金鑰為止。

從主金鑰複製過來的 8 個字組（$w[0] \sim w[7]$），可以提供前兩次加入回合金鑰運算使用。為了產生下一組 8 個字組（$w[8] \sim w[15]$），金鑰擴展演算法則是透過一個 g 函數（如圖 6 所示），先將 $w[7]$ 以位元組為單位往左位移 1 個位元組，再利用 AES 的替換盒（S-Box），將這 4 個位元組進行位元組替換；而完成替換後的最左邊的位元組則根據擴展回合數 j，與不同數值的擴展回合常數（如圖 7 中的 16 進位數值 RC[j]）進行 Exclusive OR（XOR）運算；運算後的 4 個位元組則再與 $w[0]$ 進行 XOR 運算，產生 $w[8]$，接著利用 XOR 運算依序產生出 $w[9] \sim w[11]$。$w[11]$ 中的 4 個位元組再經過 h 函數進行位元組替換，其結果再與 $w[4]$ 進行 XOR 運算以產生 $w[12]$；接著利用 XOR 運算依序產生出 $w[13] \sim w[15]$。依此方式，再利用 $w[8] \sim w[15]$ 產生 $w[16] \sim w[23]$，依此類推，直到完成 $w[56] \sim w[59]$ 之擴展為止。

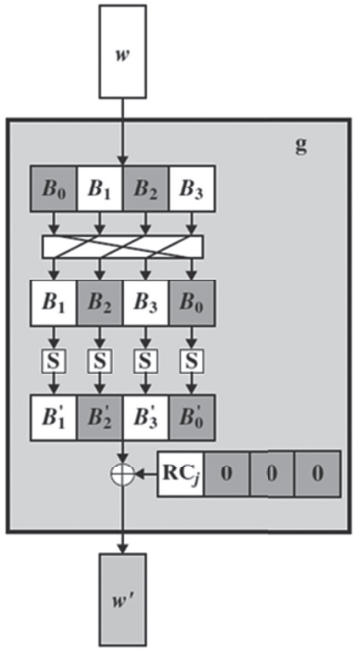

圖 6　AES-256 使用之 *g* 函數示意圖

資料來源：Stallings（2010）。

j	1	2	3	4	5	6	7
RC[j]	h01	h02	h04	h08	h10	h20	h40

圖 7　AES-256 金鑰擴展所使用之 7 個回合常數

資料來源：作者自行研究整理。

在 AES-256 的金鑰擴展演算法中，g 函數及 h 函數使用了互斥或運算及非線性（Nonlinear）的替換盒，具有密碼學上良好的混淆及擴散效果，可以有效抵抗目前已知的密碼分析技術。基於此原因，本研究提出的 NAES-512 採用其為元素，加上額外的字組交換運算，以設計可以使用 512 位元金鑰的 AES 加密及解密技術，同時還可以為 AES-256 加密所

產生的密文進行解密。下一章將詳細說明本研究提出的 NAES-512，包括其運作方式及其採用的金鑰擴展演算法。

參、抗量子計算的 NAES-512

一、NAES-512 架構

為讓以往利用 AES-256 加密後的密文可以在 NAES-512 中順利解密，NAES-512 是採用與 AES-256 相同的加密及解密架構，並採用 512 位元的金鑰作為主金鑰，以抵抗量子演算法的暴力攻擊。圖 8 為 NAES-512 的加密架構，左半部為加密程序，與 AES-256 相同，其中所需要的 15 個回合金鑰是透過 NAES-512 金鑰擴展演算法所產生，每回合金鑰以 4 × 4 位元組矩陣表示。在解密部分，同樣採用 AES-256 的解密程序，NAES-512 則專注於利用其金鑰擴展演算法（如圖 8 右半部黃色部分所示），將 512 位元的主金鑰依序產生與加密相同的 15 把回合金鑰。下一節將詳述 NAES-512 的金鑰擴展演算法。

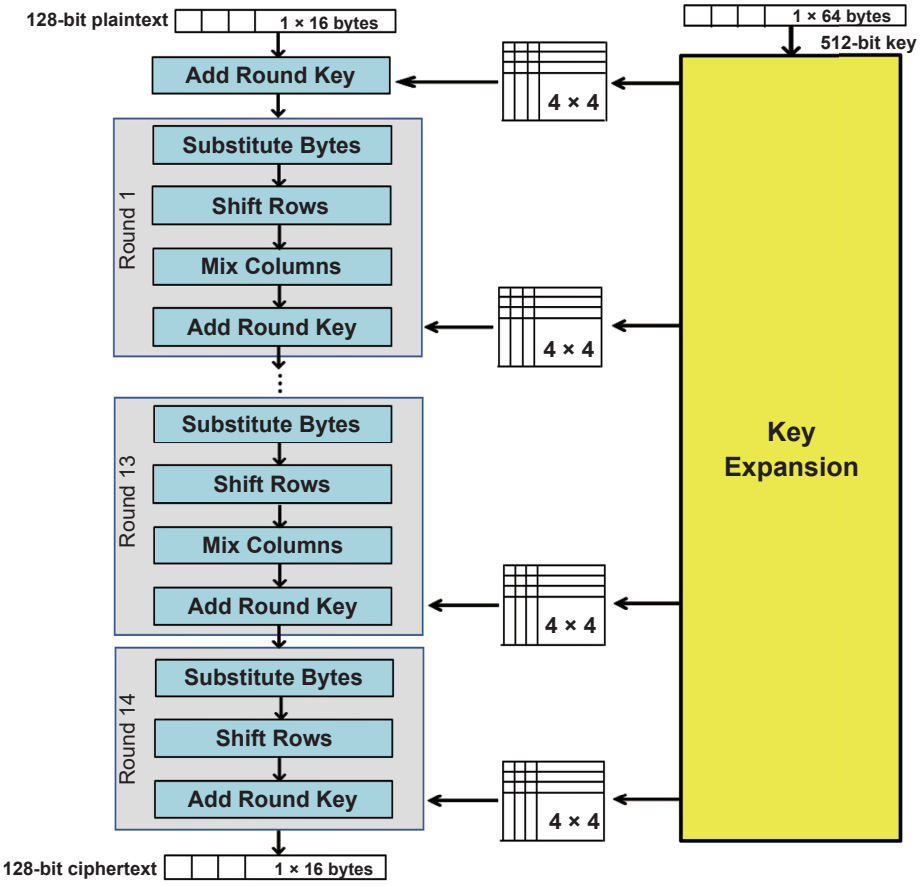

圖 8　NAES-512 的加密架構

資料來源：作者自行研究整理。

二、NAES-512 的金鑰擴展演算法

AES-256 的金鑰擴展演算法使用了替代—置換網路（Substitution-Permutation Network, SP-Network）（Schneier, 1996）設計，使其具備良好的密碼學上的混淆及擴散效果。而 NAES-512 為了同時達到使用 512 位元主金鑰及相容於 AES-256 的目的，其金鑰擴展演算法採用了一對平行的 AES-256 金鑰擴展程序，並在金鑰擴展過程中，額外加入了一個字組交換步驟，目的在使 512 位元的金鑰可以在擴展過程中充分混合，以具備 512 位元金鑰的安全性，其架構如圖 9 所示。

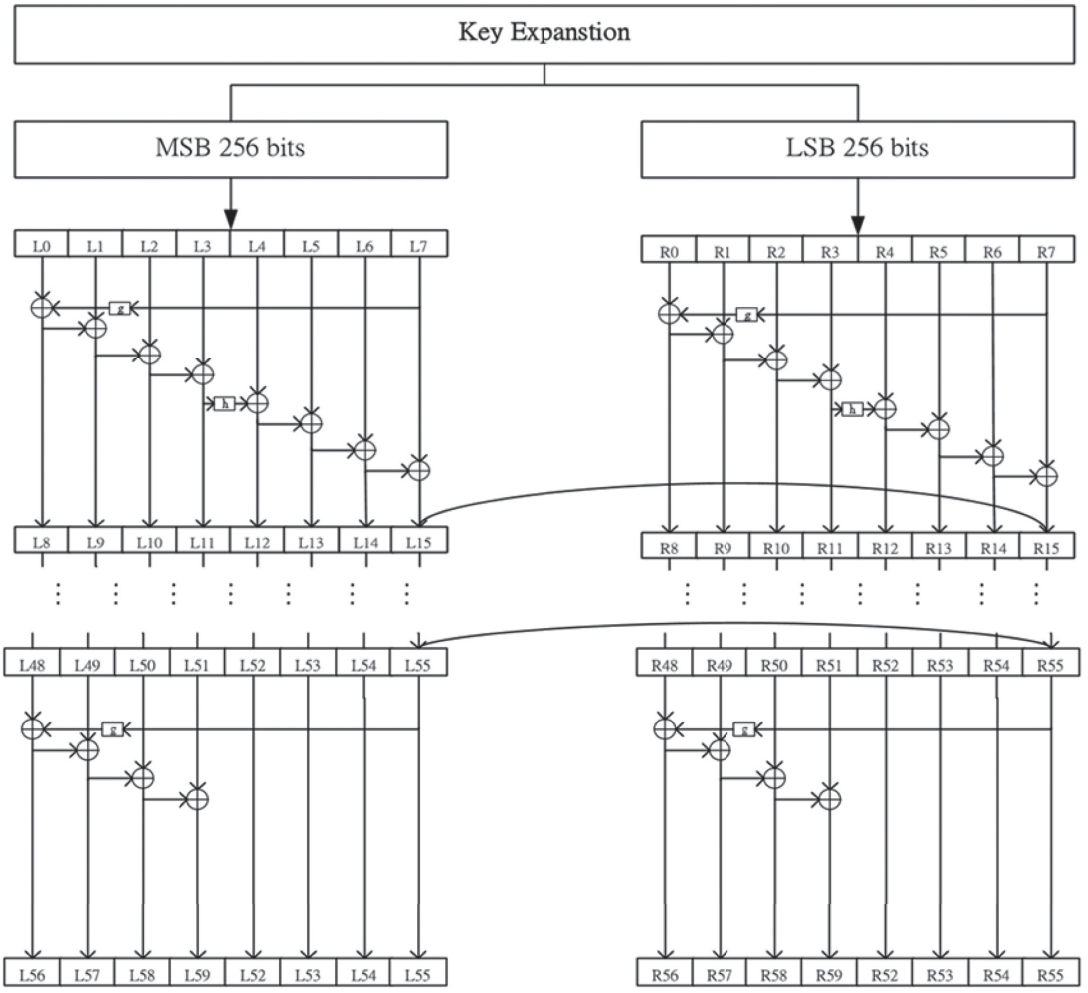

圖 9　NAES-512 的金鑰擴展架構

資料來源：作者自行研究整理。

NAES-512 的金鑰擴展演算法首先將 512 位元的金鑰分成 256 位元的高位元（Most Signiflcant Bits, MSB）及 256 位元的低位元（Least Signiflcant Bits, LSB）兩個部分，這 2 組 256 位元金鑰再分別進入左右兩個平行的 AES-256 的金鑰擴展程序，以進行 7 個回合的字組

產生，如圖 9 所示。與 AES-256 不同的是，NAES-512 在產生新一回合的 8 個字組後，會將所屬 8 個字組的最後一個字組進行交換。在第 1 回合中，512 位元主金鑰的 256 位元 MSB 可以區分為 L0 ~ L7 等 8 個字組，AES-256 金鑰擴展程序會以這 8 個字組為基礎，產生第 2 回合的 8 個字組（L8 ~ L15）；同樣的，右半部的 256 位元 LSB 可以區分為 R0 ~ R7 等 8 個字組，AES-256 金鑰擴展程序會以這 8 個字組為基礎，產生第 2 回合的 8 個字組（R8 ~ R15），最後再將 L15 跟 R15 交換；同理，在後續的第 3 ~ 6 回合中，L23 會跟 R23 交換，L31 會跟 R31 交換，L39 會跟 R39 交換，L47 會跟 R47 交換，L55 會跟 R55 交換；由於第 7 回合只需要產生前 4 個字組即可，因此不需要進行字組交換。NAES-512 加密及解密所需要的 15 把回合金鑰則採用左半部金鑰擴展程序所產生 60 個字組（L0 ~ L59）。Round-Key 0 使用 L0 ~ L3，Round-Key 1 使用 L4 ~ L7，依此類推，Round-Key 14 則使用 L56 ~ L59。

由於左半部的 L15 是第 2 回合中最後產生的字組，其混合了 L0 ~ L7 等 8 個字組中的部分位元；同理，右半部的 R15 是第 2 回合中最後產生的字組，其混合了 R0 ~ R7 等 8 個字組中的部分位元。因此，在產生第 3 回合的 8 個字組前，若將 L15 跟 R15 交換，則可以在第 3 回合字組產生時，將前一回合的左半部及右半部字組充分混合。如此，在 7 個回合的金鑰擴展過程中，總共經過 5 個回合的字組交換步驟，可以加強了 512 位元金鑰的混淆及擴展效果，此為 NAES-512 金鑰擴展演算法額外加入字組交換步驟設計的原理。

若要利用 NAES-512 對 AES-256 產生的密文進行解密，依據上述 NAES-512 的金鑰產生程序，會將 256 位元金鑰作為 MSB，並複製一份作為 LSB，此時左右兩組金鑰相同。在第 1 回合時，由於 L7 與 R7 相同（無交換），第 2 回合產生的 L15 與 R15 也會相同；當 L15 與 R15 交換後，由於兩個數值不變，導致 L23 與 R23 也會相同；同理，後續的 L31 與 R31、L39 與 R39、L47 與 R47 及 L55 與 R55 均會相同。換言之，由左半部金鑰產生程序所產生的 60 個字組（L0 ~ L59）將與 AES-256 金鑰產生程序所產生的 60 個字組相同，因此，具有相同的 15 個回合金鑰，這是為何 NAES-512 可為 AES-256 產生的密文進行解密的原因。

肆、NAES-512 之效能驗證

本章針對 NAES-512 在硬體上進行功能性驗證，並與 AES-256 進行相容性測試。第一節說明 NAES-512 在 ModelSim 模擬及 FPGA 實現結果，第二節為 AES-256 之相容性測試，第三節則為 NAES-512 與 AES-256 及 AES-512 效能比較。

一、NAES-512 硬體效能驗證

本研究使用硬體描述語言 Verilog 完成 NAES-512 加密及解密模組設計，並利用 ModelSim 模擬工具（Intel Corporation, 2018）進行功能性之驗證，最後利用 Quartus Prime 整合開發工具（Intel Corporation, 2020）完成電路分析及合成，並將完成合成之電路燒錄在 DE2-115 FPGA 開發板（Terasic Inc., 2021），同時搭配自行開發的 DE2-115 的 LCD 顯示控制電路，驗證 NAES-512 在 FPGA 執行之成效。實驗所使用之工具說明如下：

（一）開發工具：Quartus Prime，其為 Intel 提供之 FPGA 整合開發工具，本實驗利用其進

行 NAES-512 相關模組之 Verilog 程式編輯、電路分析及合成、FPGA 腳位設定及電路燒錄等。

（二）模擬工具： ModelSim-Intel FPGA 10.5b，其為附屬在 Quartus Prime 16.1 版上之功能模擬軟體，本實驗利用其作為 NAES-512 相關模組之功能模擬。

（三）開發板：本實驗使用 Altera DE2-115 FPGA 開發平臺進行 NAES-512 硬體功能驗證，如圖 10 所示。其具有 114,480 邏輯單元（Logic Elements, LE）、432 M9K 記憶體模組、3,888 K 位元嵌入式儲存器及 4 個鎖相迴路（Phase-Locked Loops, PLL）。DE2-115 並提供 18 個滑動開關（Slide Switches, SW）、4 個按鈕及 16 × 2 LCD 顯示模組等，可提供輸入及顯示功能，方便測試資料輸入及執行結果驗證。

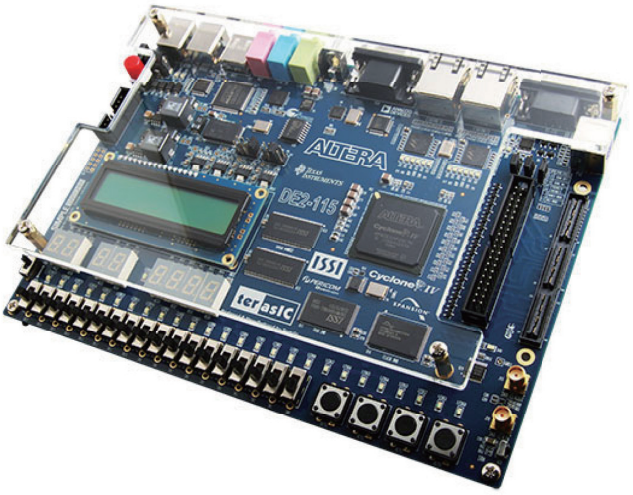

圖 10　DE2-115 FPGA 開發平臺

資料來源：Terasic Inc.（2021）。

NAES-512 的 Verilog 實作模組架構如圖 11 所示，包含加密（Encrypt）、解密（Decrypt）、金鑰擴展（Keygen）三個主要模組。在加密模組部分，則包含了回合加密（Round）及加入回合金鑰（Add-Rnkey）等模組，在回合加密模組部分，除了加入金鑰模組外，又實作了位元組取代（Sub-Byte）、列位移（Sif-Row）及行混合（Mix-Colm）等模組；在解密模組部分，則包含了回合解密（I-Round）及加入回合金鑰等模組，在回合解密模組部分，除了加入回合金鑰模組外，又實作了反位元組取代（Inv-Sbyte）、反列位移（Inv-Sifrow）及反行混合（Inv-Mixclm）等模組；在金鑰擴展模組方面，則實作了 g 函數（G-Function）及 h 函數（H-Function）兩個模組。上述各模組均為本研究依據實驗需求，利用 Verilog 硬體描述語言所撰寫而成。

NAES-512 功能驗證的主要目的在於驗證其加解密程式之正確性，因此，在實驗設計上，先將 512 位元金鑰輸入到 NAES-512，對 128 位元明文加密，以產生 128 位元密文；接著將密文及相同的金鑰輸入到 NAES-512 進行解密，以還原明文。以 16 進位表示之 128 位元明文及 512 位元金鑰如表 1 所示。

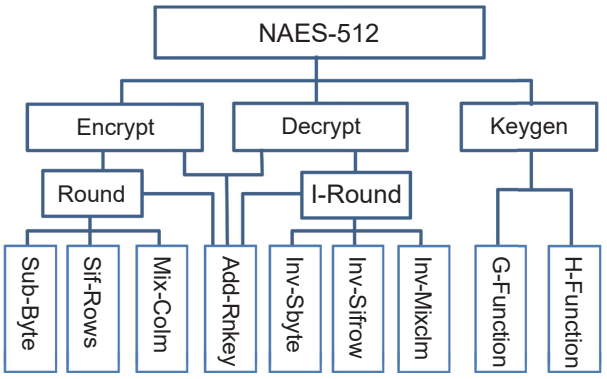

圖 11　NAES-512 實作模組架構

資料來源：作者自行研究整理。

表 1　NAES-512 模組驗證使用之明文及金鑰

項目	NAES-512 模組驗證
明文	00112233445566778899aabbccddeeff
金鑰	000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f10111213141516171 8191a1b1c1d1e1f000102030405060708090a0b0c0d0e0f

資料來源：作者自行研究整理。

金鑰擴展模組為NAES-512 金鑰擴展演算法的實現，其ModelSim模擬結果如圖 12所示。金鑰擴展模組主要由 g 函數及 h 函數構成，其中完成 g 函數可擴展一把回合金鑰，耗費 3 個時脈；另完成 h 函數會產出另一把回合金鑰，耗費 2 個時脈，因此，每擴展一把回合金鑰不會超過 3 個時脈。從圖 12 可以看出，前兩把回合金鑰來自於原始主金鑰，並不消耗時間；4.5 個時脈（45 ns）後，分別產生第 3 及第 4 把回合金鑰（Key 0 及 Key 1）；過了 5 個時脈（50 ns），分別產生第 5 及第 6 把回合金鑰（Key 2 及 Key 3）；依此類推，到了第 7 回合，因為只需做 g 函數，產生第 15 把回合金鑰（Key 13），其金鑰擴展之總時脈數為 31.5 個時脈。為驗證金鑰擴展模組產出的正確性，上述回合金鑰模擬結果與自行利用 Python 開發的 NAES-512 軟體執行結果（如圖 13 右半部所示）進行比較，結果完全相相符，證明金鑰擴展模組設計的正確性。

NAES-512 加密共需要 1 個加入回合金鑰模組、13 個完整回合模組及 1 個不完整回合模組（缺行混合模組），各回合加密之 ModelSim 模擬結果如圖 14 所示。其顯示完成加入回合金鑰及第 1 次回合加密運算共花費 2.5 個時脈，這兩部分會消耗 2 把回合金鑰，來源為原始主金鑰，並不會耽誤運算進行；第 2 回合加密開始，每回合需要消耗 3 個時脈，均不會超過回合金鑰產生時間，不需要額外的時脈延遲，因此，從第 2 回合至第 13 回合，加密共花費 36 個時脈；第 14 回合為不完整的回合加密，需花費 2 個時脈，因此，完成加密一個 128 位元區塊明文共需 40.5 個時脈的時間。圖 15 為加密模組的 ModelSim 模擬結果，與自行利用 Python 開發的 NAES-512 軟體執行結果（如圖 13 中「cipher」欄所示）進行比較，結果完全相符，證明加密模組設計的正確性。

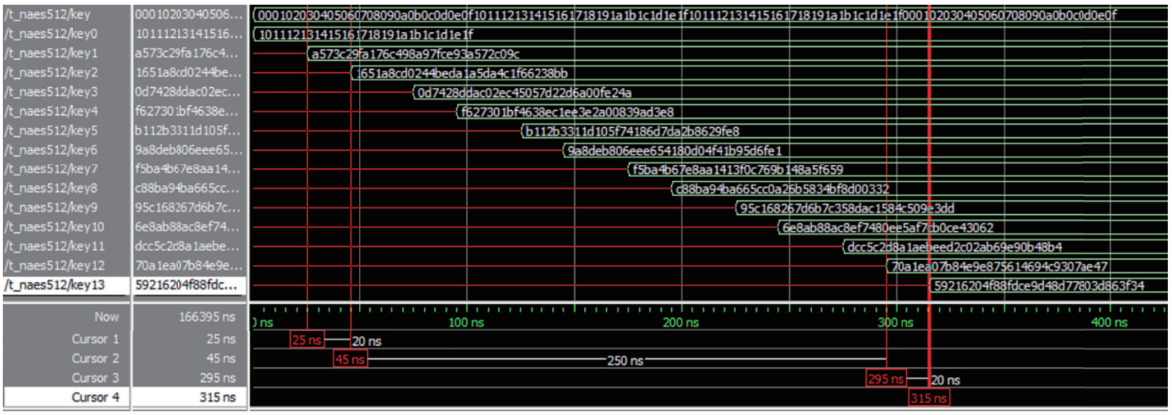

圖 12　金鑰擴展模組之 ModelSim 模擬結果

資料來源：作者自行研究整理。

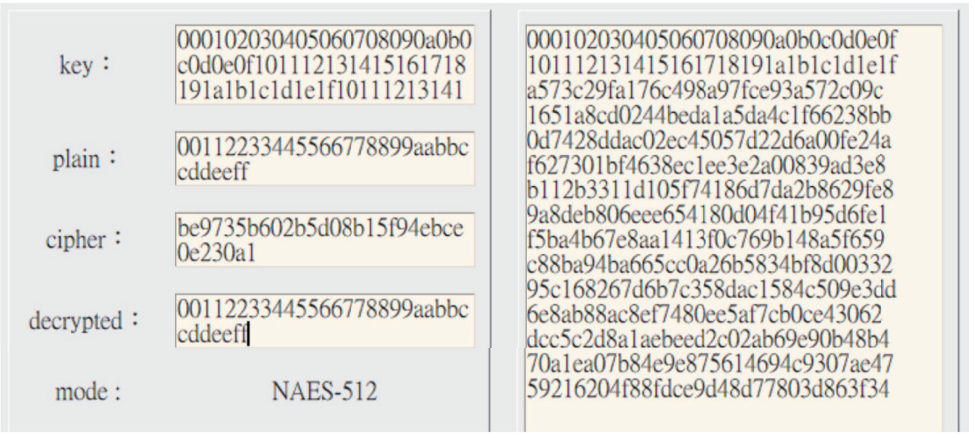

圖 13　軟體執行 NAES-512 金鑰擴展之結果

資料來源：作者自行研究整理。

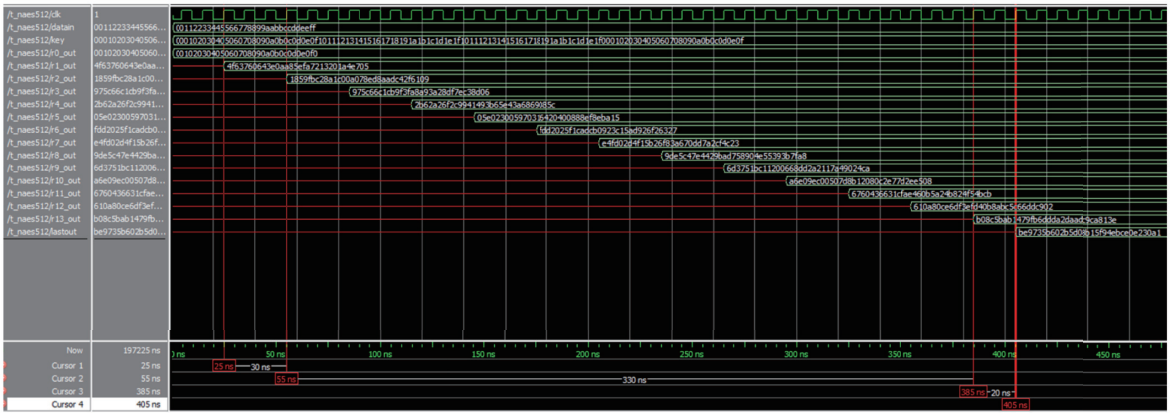

圖 14　各回合加密之 ModelSim 模擬結果

資料來源：作者自行研究整理。

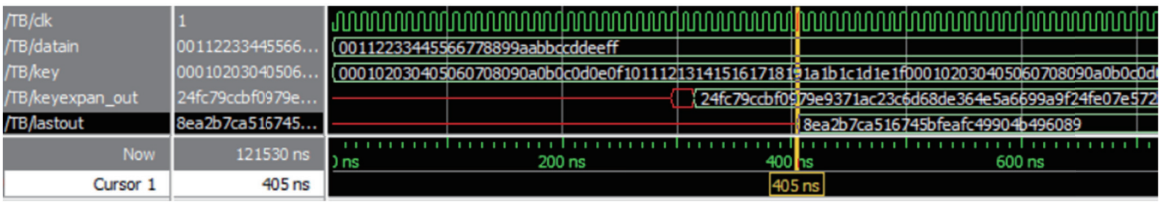

圖 15　加密模組之 ModelSim 模擬結果

資料來源：作者自行研究整理。

為驗證 NAES-512 在硬體上之效能，本實驗首先將模擬完成之 NAES-512 模組燒錄於 DE2-115 FPGA 開發板中，並利用其上的指撥開關輸入明文及金鑰，加密之結果則顯示於板子上之液晶顯示器（Liquid-Crystal Display, LCD）。為了跟 ModelSim 上之加密模擬相互驗證，本實驗採用與表 1 相同的明文及金鑰，經加密後的明文則顯示於圖 16 的 LCD 上，其結果與 ModelSim 模擬之加密結果相符，其說明 NAES-512 硬體實現的可行性。

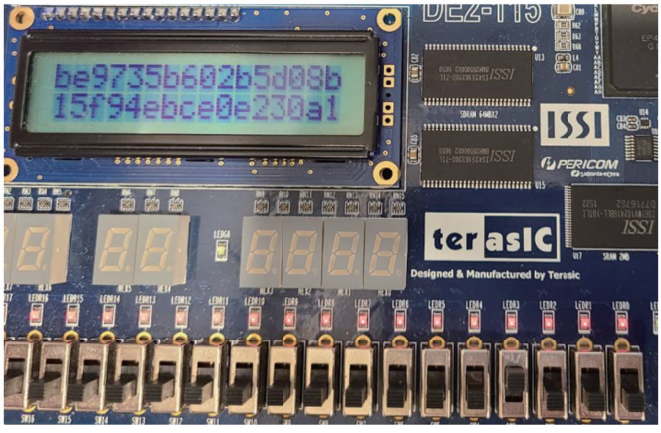

圖 16　NAES-512 於 FPGA 開發板加密後之顯示結果

資料來源：作者自行研究整理。

二、AES-256 相容性驗證

AES-256 相容性驗證的目的在驗證 AES-256 產生的密文是否可以在 NAES-512 中進行解密。在說明 AES-256 相容性之前，先說明 NAES-512 之解密模組的效能。由於解密的回合金鑰是反向輸入的，因此必須等待金鑰擴展結束才能進行解密。如圖 17 所示，解密模組共花費 42 個時脈，加上等待金鑰擴展所需的 31.5 個時脈，NAES-512 解密總共需要花費 73.5 個時脈。圖 17 顯示，當輸入與上一節所示的相同密文及 512 位元金鑰後，由於每個時脈週期為 10 ns，經過 735 ns（73.5 時脈）後完成解密，其解密結果與表 1 上之明文相符，解密正確。

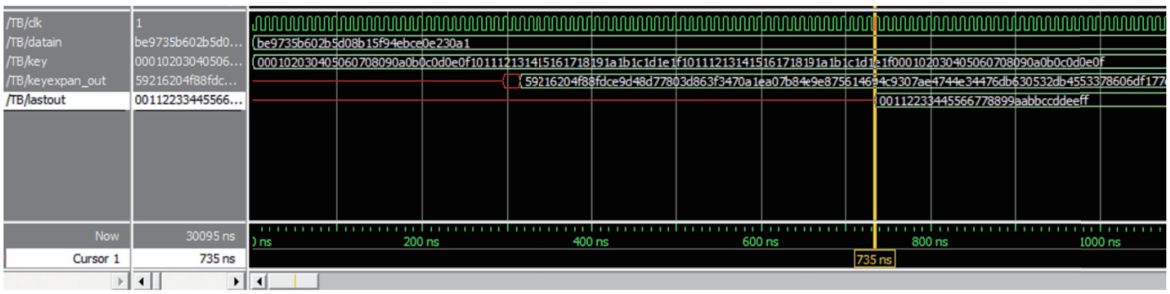

圖 17　解密模組之 ModelSim 模擬結果

資料來源：作者自行研究整理。

如第參章所述，NAES-512 金鑰擴展設計是採用平行的 256 位元金鑰擴展模式，過程中加入特定的字組交換，並由左半部的金鑰擴展演算法輸出回合金鑰。如果輸入 256 位元主金鑰，由於右半部金鑰是複製左半部主金鑰，造成金鑰擴展過程中任兩半部的相對字組值均相同，因此不管如何交換，產生出來的回合金鑰均與單一 AES-256 金鑰擴展的結果相同，因此可以為 AES-256 創造的密文進行解密。表 2 為 AES-256 加密使用之 128 位元明文、256 位元金鑰及最後產生之 128 位元密文，而圖 18 為將上述 256 位元金鑰輸入 NAES-512，並對密文解密的模擬結果，其解密結果與原始明文相同，證明 NAES-512 可對 AES-256 產生之密文正確解密。

表 2　AES-256 加密使用之明文、金鑰及產生之密文

項目	AES-256 加密
明文	00112233445566778899aabbccddeeff
金鑰	000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
密文	8ea2b7ca516745bfeafc49904b496089

資料來源：作者自行研究整理。

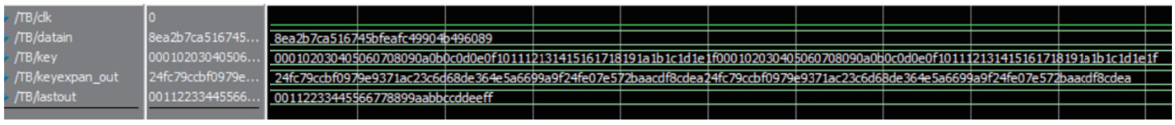

圖 18　利用 NAES-512 對 AES-256 產生的密文進行解密的模擬結果

資料來源：作者自行研究整理。

三、NAES-512 與 AES-256 及 AES-512 效能之比較

本節分別從實作消耗資源及加密與解密速度兩方面，說明 NAES-512 實務上之優勢。NAES-512 與 AES-256 及 AES-512 效能之比較如表 3 所示。在實作消耗資源方面，實作 NAES-512 所使用之邏輯單元數量為 12,985，比實作 AES-256 只多使用了 5,060 個邏輯單元，

但其安全性則可從 256 位元金鑰提升到 512 位元金鑰；另與同樣使用 512 位元金鑰的 AES-512 比較，其消耗邏輯單元數量則遠小於實作 AES-512 所需的 30,867 個邏輯單元。在加密與解密速度方面，由於 NAES-512 是採用 AES-256 的加密與解密架構，且其金鑰擴展演算法採用創新的平行 AES-256 金鑰擴展演算法，因此，加密與解密一個資料區塊的時脈數與 AES-256 是相等的；另與同樣使用 512 位元金鑰的 AES-512 比較，雖然 NAES-512 使用較為複雜的金鑰擴展演算法與較多的回合數，可是加密及解密一個資料區塊的時脈數均不大於兩倍的 AES-512 的加密及解密時脈數。

表 3　NAES-512 與 AES-256 及 AES-512 效能比較

加密演算法	加密時脈數量	解密時脈數量	消耗邏輯單元數量
AES-256	40.5	73.5	7,925
NAES-512	40.5	73.5	12,985
AES-512	28.5	49.5	30,867
QNAES-512	40.5	73.5	39,180

資料來源：作者自行研究整理。

另外，由於 NAES-512 單次加密區塊為 128 位元，而 AES-512 單次加密區塊為 512 位元，因此，利用 4 個 NAES-512 模組一次加密 4 個 128 位元區塊明文，即可提供與 AES-512 同樣的加密位元數。基於這個原因，本研究另實作包含 4 個 NAES-512 模組的模組（以下簡稱 QNAES-512），其消耗邏輯單元數只比 AES-512 多出 8,313 邏輯單元，但由於平行運作的關係，其加密與解密時間均與單一 NAES-512 模組相同。

上述效能比較說明，NAES-512 只需花費較 AES-512 稍微多一點資源，即可提供與 AES-512 相當之安全性及加密及解密效能，但是 NAES-512 卻可與 AES-256 相容，這點是 AES-512 無法提供的，也是本研究主要貢獻之所在。

伍、NAES-512 安全性分析

NAES-512 金鑰擴展的原理是利用平行的兩個 AES-256 金鑰擴展架構（如圖 9 所示），並將 512 位元輸入之金鑰區分為左右兩個 256 位元輸入，同時進行 7 個回合金鑰擴展。在擴展過程中，每回合擴展完的結果的最後一個字組（4 個位元組）會進行交換，最後利用左半部金鑰擴展所產生的 60 個字組作為加密及解密使用的 15 個回合金鑰。上述特定字組交換的目的，是讓左右金鑰可以在金鑰擴過程中進行充分混合，這種多回合字組交換程序具有密碼學上的替換及移位效果，讓金鑰擴展結果產生所謂的混淆及擴展的效果。為了說明 NAES-512 金鑰擴展的效果，本研究利用表 2 所示的 256 金鑰作為左半部金鑰擴展的輸入，並改變金鑰的最後字組的其中 1 個位元作為右半部金鑰擴展的輸入，同時記錄 15 個回合金鑰輸出，並將輸出的 15 個回合金鑰與 AES-256 金鑰擴展產生的 15 個回合金鑰進行相對位元差異性比較，以測試金鑰擴展的雪崩效應，其結果如表 4 所示。

表 4 NAES-512 雪崩效應實驗結果

回合金鑰	實驗 1	實驗 2	實驗 3	實驗 4	實驗 5	實驗 6	實驗 7	實驗 8	平均
Key 4	36	28	32	44	28	44	40	40	37
Key 5	19	32	27	19	45	19	37	33	29
Key 6	18	34	24	22	26	22	24	24	24
Key 7	18	34	29	22	37	22	37	32	29
Key 8	66	76	76	54	44	54	62	62	62
Key 9	61	54	79	73	71	73	82	64	70
Key 10	49	65	58	49	61	49	58	66	57
Key 11	46	58	61	68	59	68	63	61	61[a]
Key 12	66	62	62	61	62	61	60	61	62[a]
Key 13	65	62	66	71	62	71	63	65	67[a]
Key 14	72	55	58	63	72	63	66	51	63[a]
密文	65[b]	61[b]	75[b]	64[b]	56[b]	64[b]	60[b]	67[b]	64[a,b]

註：[a] 表良好的雪崩效應。
[b] 表密文的雪崩效應。
資料來源：作者自行研究整理。

本實驗共進行 8 次，由於主金鑰的 256 位元作為前 2 個回合金鑰（Key 0 及 Key 1），因此並無差異性，也由於主金鑰的最後字組並沒有交換，第 3 及第 4 回合金鑰（Key 2 及 Key 3）也沒有差異性。因此，回合金鑰的差異性則從第 5 個回合金鑰（Key 4）開始，如表 4 所示。每個回合金鑰的 8 次實驗的平均差異值則顯示在表 4 的最右一欄。表 4 顯示，從 Key 4 開始，平均位元差異數逐漸增加，而最後 4 個回合金鑰的平均位元差異數則接近一半的位元數（64 位元），在密碼學上為最理想的雪崩效應結果，此證明 NAES-512 金鑰擴展具有良好的雪崩效應。換句話說，NAES-512 金鑰擴展運算具有良好的安全性。表 4 的最後一列則顯示 NAES-512 的密文與 AES-256 產出的密文的位元差異數，最後一欄則為其 8 次實驗的平均值。其顯示，平均位元差異數為 64，此表示 NAES-512 的回合加密結果具有良好的雪崩效應，安全性得以驗證。

陸、結論

本文提出一個簡單且相容於 AES-256 的新 AES-512 密碼法（NAES-512），使其在面對量子計算挑戰時，仍然具有 256 位元的計算上安全性；另外，還可以利用其對 AES-256 產出的密文進行解密。NAES-512 使用 AES-256 的加密及解密架構，並使用 512 位元的金鑰擴展演算法以產生加密及解密所需要的 15 個回合金鑰。

NAES-512 的金鑰擴展是利用平行的兩個 AES-256 金鑰擴展程序，將 512 位元金鑰區分為左右兩個 256 位元輸入，再進行 7 個回合的金鑰擴展。當完成每回合金鑰擴展後，兩組金

鑰擴展所產出的最後一個字組（4 個位元組）會進行交換，其目的在使左右兩半部的金鑰可以充分混合，以達到密碼學上的混淆及擴散效果。左半部最後產生的 60 個字組則組成加密及解密使用的 15 個回合金鑰。當輸入的金鑰為 256 位元時，NAES-512 除了將其作為左半部金鑰擴展的輸入外，還會將其複製一份，作為右半部金鑰擴展的輸入。因此，在兩半部金鑰擴展的過程中，不管字組如何交換，最後產出的回合金鑰會與 AES-256 金鑰擴展過程所產生的金鑰相同，因此可以為 AES-256 產生的密文進行解密。

為驗證 NAES-512 在硬體實現的可行性，本研究同時在 FPGA 實現 NAES-512 系統，主要包括加密模組、解密模組及金鑰擴展模組等模組的實現。軟體模擬結果顯示，完成金鑰擴展之總時脈數為 31.5 個時脈，而完成一個回合 NAES-512 加密需要消耗 3 個時脈，不會超過回合金鑰產生時間，因此，可以在不需要額外的時脈延遲下，順利在 40.5 個時脈的時間內完成一個 128 位元明文區塊加密。實驗結果同時證明，NAES-512 加密及解密演算法可以在 FPGA 實驗板上順利執行；另外，當輸入 256 位元金鑰對 AES-256 產生之密文進行解密時，其解密結果與原始明文相同，證明 NAES-512 可正確對 AES-256 產生之密文進行解密。實驗結果同時顯示，NAES-512 金鑰擴展演算法具有良好的雪崩效應，具備高度的安全性。因此，NAES-512 為一兼具安全性及便利性的 512 位元先進加密及解密系統，可作為在量子計算威脅下，進行高安全加密技術設計之參考。

參考文獻

劉江龍，2019，〈量子計算對當代密碼系統之威脅及對策〉，《前瞻科技與管理》，9（1&2），頁 4-21。doi:10.6193/JATM.201911_9(1_2).0002

Grover L. K., 1997, “Quantum Mechanics Helps in Searching for a Needle in a Haystack,” *Physical Review Letters*, 79(2), 325-328. doi:10.1103/PhysRevLett.79.325

Intel Corporation, 2018/9/23, “ModelSim-Intel® FPGAs Standard Edition Software Version 18.1,” https://www.intel.com/content/www/us/en/software-kit/750368/modelsim-intel-fpgas-standard-edition-software-version-18-1.html (accessed May 19, 2024).

Intel Corporation, 2020/11/22, “Intel® Quartus® Prime Lite Edition Design Software Version 20.1.1 for Windows,” https://www.intel.com/content/www/us/en/software-kit/660907/intel-quartus-prime-lite-edition-design-software-version-20-1-1-for-windows.html (accessed May 19, 2024).

Moh’d A., Jararweh Y., and Tawalbeh L., 2011, “AES-512: 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation,” in *2011 7th International Conference on Information Assurance and Security (IAS)*, Los Alamitos, CA: IEEE Computer Society Press, 292-297. doi:10.1109/ISIAS.2011.6122835

National Institute of Standards and Technology, 2023/5/9, *Advanced Encryption Standard (AES)*, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf (accessed May 19, 2024).

Schneier B, 1996, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed., New York, NY: Wiley.

Stallings W, 2010, *Cryptography and Network Security: Principles and Practice*, 5th ed., Boston, MA: Prentice Hall.

Terasic Inc., 2021, "Altera DE2-115 Development and Education Board," https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=139&No=502#contents1 (accessed May 19, 2024).