密碼工程 Midterm

學號:112550090 姓名:曾士珍

Problem 1

In GF(2^4) Compute f(x) + g(x), f(x) − g(x), f(x) × g(x) mod P(x), where P(x) = $x^4 + x + 1$

a) f(x) = $x^2 + 1$, g(x) = $x^3 + x^2 + 1$

f(x) + g(x) = $(x^2 + 1) + (x^2 + 1) + x^3 = \boldsymbol{x^3}$

f(x) − g(x) = $(x^2 + 1)$-$(x^2 + 1)$- $x^3 = \boldsymbol{x^3}$

f(x) × g(x) = $(x^2 + 1)(x^3 + x^2 + 1) = x^5 + x^4 + x^3 + 1$

$x^5 + x^4 + x^3 + 1 \text{mod} P(x) = (x^2 + x) + (x + 1) + (x^3) + 1 = \boldsymbol{x^3 + x^2}$

b) f(x) = $x^2 + 1$, g(x) = $x + 1$

f(x) + g(x) = $x^2 + x + 1 + 1 = \boldsymbol{x^2 + x}$

f(x) − g(x) = $(x^2 + 1)$-$(x + 1) = \boldsymbol{x^2 + x}$

f(x) × g(x) = $(x^2 + 1)(x + 1) = \boldsymbol{x^3 + x^2 + x + 1}$

Problem 2

In GF(2^8), f(x) = $x^7 + x^5 + x^4 + x + 1$ and g(x) = $x^3 + x + 1$

a) Calculate f(x) + g(x), f(x) − g(x), f(x) × g(x) mod m(x), where m(x) = $x^8 + x^4 + x^3 + x + 1$

f(x) + g(x) = $x^7 + x^5 + x^4 + x^3 + x + 1 + x + 1 = \boldsymbol{x^7 + x^5 + x^4 + x^3}$

f(x) − g(x) = $\boldsymbol{x^7 + x^5 + x^4 + x^3}$

f(x) × g(x) = $x^{10} + x^6 + x^3 + x^2 + 1$

$x^{10} + x^6 + x^3 + x^2 + 1 mod\ m(x) = (x^6 + x^5 + x^3 + x^2) + x^6 + x^3 + x^2 + 1$

$= \boldsymbol{x^5 + 1}$

b) Show that f(x) = $x^4 + 1$ is reducible over GF(2^8)

$(x^2+1)^2 = x^4+x^2+x^2+1 = x^4+1$

x^4+1可以拆成$(x^2+1)^2$在 GF(2^8)，故可說明其為 reducible

Problem 3

Consider the field GF(2^4) with the irreducible polynomial P(x) = x^4+x+1. Find:

a) the inverse of f(x) = x

b) the inverse of g(x) = x^2+x by trial and error

x^nmod P(x)	x^3	x^2	x	1
x^0	0	0	0	1
x^1	0	0	1	0
x^2	0	1	0	0
x^3	1	0	0	0
x^4	0	0	1	1
x^5	0	1	1	0
x^6	1	1	0	0
x^7	1	0	1	1

x^nmod P(x)	x^3	x^2	x	1
x^8	0	1	0	1
x^9	1	0	1	0
x^{10}	0	1	1	1
x^{11}	1	1	1	0
x^{12}	1	1	1	1
x^{13}	1	1	0	1
x^{14}	1	0	0	1
x^{15}	0	0	0	1

(x^3到x^4，可以看成先左移一位(乘以 x 的概念)，再 xor(x+1)(把x^4換成$x+1$))

(a) 因為$x^{15} \equiv 1(modP(x))$，因此可得 x 的 inverse 是$x^{14} \equiv \boldsymbol{x^3+1}(modP(x))$

(b) 因為$x^{15} \equiv 1(\text{mod P(x)})$，又$x^2+x \equiv x^5(\text{mod P(x)})$

因此可得x^5的 inverse 是$x^{10} \equiv \boldsymbol{x^2+x+1}(\text{mod P(x)})$

Problem 4

In GF(2^8), find:

(a) $(x^3+x^2+x)(x^3+x^2)^{-1}$mod (x^8+x^4+1) =

$(x^8+x^4+1) = (x^3+x^2)(x^5-x^4+x^3-x^2)+2x^4+1$

$(x^3+x^2)(x^5-x^4+x^3-x^2) \equiv 1 \bmod (x^8+x^4+1)$

$(x^3+x^2)(x^5+x^4+x^3+x^2) \equiv 1 \bmod (x^8+x^4+1)$

$(x^3+x^2)^{-1} = (x^5+x^4+x^3+x^2)\bmod (x^8+x^4+1)$

$(x^3+x^2+x)(x^3+x^2)^{-1} = (x^3+x^2+x)(x^5+x^4+x^3+x^2)$

$\equiv x^8+x^6+x^5+x^3 \equiv \boldsymbol{x^6+x^5+x^4+x^3+1}\bmod (x^8+x^4+1)$

(b) $(\boldsymbol{x^6 + x^3 + 1})\ (\boldsymbol{x^4 + x^3 + 1})\boldsymbol{mod}(\boldsymbol{x^8 + x^4 + 1})$

$(x^6 + x^3 + 1)(x^4 + x^3 + 1) \equiv x^{10} + x^9 + x^7 + x^4 + 1 mod(x^8 + x^4 + 1)$

$\equiv (x^6 + x^2) + (x^5 + x) + x^7 + x^4 + 1 \equiv \boldsymbol{x^7 + x^6 + x^5 + x^4 + x^2 + x + 1}$

Problem 5

a) Explain why this mix column operation can be implemented with a simple look up table and XOR.

在 AES 演算法中，mix column 是對狀態矩陣乘 pre-defined matrix 進行線性變換。pre-defined matrix 的元素是 GF(2^8)有限域中的係數，其中運算要模一個不可約多項式，假設是 $x^8 + x^4 + x^3 + x + 1$，將其轉成 10 進位表示為：256+16+8+2+1=283=0x11B。

假設狀態矩陣$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0x87 \\ 0x6E \\ 0x46 \\ 0xA6 \end{pmatrix}$， pre-defined matrix $= \begin{pmatrix} 2\ 3\ 1\ 1 \\ 1\ 2\ 3\ 1 \\ 1\ 1\ 2\ 3 \\ 3\ 1\ 1\ 2 \end{pmatrix}$，

在進行矩陣乘法時，會需要將數值相乘及相乘完的結果相加兩個步驟，以計算b_0為例：$b_0 = a_0 \times 2 + a_1 \times 3 + a_2 \times 1 + a_3 \times 1$

兩個數值的加法，相當於兩數的每個位元進行 xor 運算，若結果超過 8 位則需 mod 0x11B，由此可知加法的部分可以用 xor 快速解決。

乘法能預先計算並建表，以上述例子來看表中存狀態矩陣每個字節乘以 2 和 3 的結果，如下表(超過 8 位需 mod 0x11B)：

字節	Mul2	Mul3
0×00	0×00	0×00
0×01	0×02	0×03
0×02	0×04	0×06
…	…	…
0×46	0×8C	0xCA
0×6E	0xDC	0xB2
0×87	0×0E	0×0F
0xA6	0×4C	0xE2
…	…	…
0xFF	0xE5	0xFE

由上述$b_0 = a_0 \times 2 + a_1 \times 3 + a_2 \times 1 + a_3 \times 1$

$= Mul2[0x87] \oplus Mul3[0x6E] \oplus 0x46 \oplus 0xA6 = 0xFA \oplus 0xA6 = 0x5C$

故可說明 mix column operation 可以透過 XOR 跟查表來得到結果

b) Apply the same idea used above, explain why the byte substitution, shift row and mix column can be combined and implemented as a simple look-up table operation. And then the whole AES is implemented by look up table and few XORs.

在 AES 中，Byte Substitution、Shift Rows 和 Mix Columns 操作可以結合一起建表來完成，這些操作雖然在 AES 中是分開進行的，但對於每個輸入字節的最後結果可先計算好並存在表中。

Byte Substitution 是利用 S-box 進行非線性字節替換， Shift Rows 是將狀態矩陣每一行的字節進行移位。SubBytes 和 ShiftRows 結合可以建表成一個操作。每個字節(0x00 到 0xFF)在 SubBytes 和 ShiftRows 之後的狀態可以透過查表對應到一個新的字節。

接著再預先建表 Mix Columns operation，如 (a) 所述。Mix Columns 的結果取決於 SubBytes 和 ShiftRows 之後的字節的新位置和值。

由上所述，在 AES 的每一輪操作中，不再是逐步執行 SubBytes、ShiftRows 和 Mix Columns，而是直接查詢預先建好的表來獲取每個狀態字節的轉換值，再使用必要的 XOR 操作進行結合（AddRoundKey 操作）。

Problem 6

In order to make AES encryption and decryption more similar in structure, the MixColumns operation is missing in the last round.

Explain how to take advantage of this property to share some of the code (for software implementation) or chip area (for hardware implementation) for AES encryption and decryption.

在解密過程中，需要進行逆操作，即 InvSubBytes、InvShiftRows 和 InvMixColumns。加解密能否共用程式碼或硬體，需要先看這些操作的順序是否可以互換。

1. InvSubBytes 只對單個字節進行操作，而 InvShiftRows 只改變字節的位置，因此這兩個操作是獨立的，順序無關。即：
   $InvSubBytes(InvShiftRows(S)) = InvShiftRows(InvSubBytes(S))$
2. AddRoundKey 是將狀態矩陣與輪密鑰進行 XOR 操作，而 InvMixColumns 是 MixColumns 的逆操作。需要調整輪密鑰，才能使在 AddRoundKey 和 InvMixColumns 操作互換順序得到的結果相同。

   調整輪密鑰 K使其成為K'，使得：

   $AddRoundKey(S, K) = AddRoundKey(InvMixColumns(S), K')$

   即滿足$S \bigoplus K = InvMixColumns(S \bigoplus K')$

   可得$K = InvMixColumns(K') \Longrightarrow K' = MixColumns(K)$

   若將每一輪的密鑰轉成 K'，就可以使 AddRoundKey 和 InvMixColumns 操作互換順序。

通過共用 SubBytes/InvSubBytes 和 ShiftRows/InvShiftRows 的操作，以及調整 AddRoundKey 和 InvMixColumns 的順序，可以實現加密和解密過程中程式碼和硬體 module 的共用，從而提高效率並節省資源。

Problem 7

Under what circumstances could you choose 3DES over AES? Also, what are the advantages of choosing AES over 3DES? Is 3DES susceptible to Meet-in-the-Middle Attacks like 2DES? Please explain.

選擇 3DES 而非 AES 的情況

1. 系統中有舊的設備或軟體仍在使用 DES，為了保持相容性，會選擇 3DES。
2. 某些嵌入式系統，可能存在硬體上更好支援 3DES 的情況。
3. 現有的加密基礎設施大量使用 3DES，轉換到 AES 的成本和時間較高。

選擇 AES 的優點

1. 安全性：AES 的密鑰長度可以是 128 位、192 位或 256 位，而 3DES 實際上的安全性相當於 112 位密鑰（因為 3DES 使用三個 56 位的 DES 密

鑰）。

2. 效能：AES 的設計更加現代化，在大多數硬體和軟體實現中，AES 的運行速度遠快於 3DES。
3. 簡潔性：AES 的算法設計較為簡單，而 3DES 則需要三次 DES 運算，計算複雜度和實現難度較高。
4. 未來保障：AES 是目前的加密標準，被全球廣泛使用，並且經過了大量的安全分析。選擇 AES 可以保證未來的安全需求。

3DES 是否易受 Meet-in-the-Middle Attack

Meet-in-the-Middle Attack 是針對多重加密算法的一種已知攻擊方式，這種攻擊試圖通過在中間找到相同的中間結果來減少總的加密強度。

在 2DES 中，即使用兩次 DES 加密，由於 Meet-in-the-Middle Attack 的存在，安全性只相當於單次 DES，而非期望的兩倍（112 位密鑰）。使用此攻擊可將 2DES 的攻擊複雜度從2^{112}簡化到$2^{56} + 2^{56} = 2^{57}$，這樣 2DES 的安全性並不顯著高於單次 DES。

3DES 的設計目的是為了增加安全性並抵抗 Meet-in-the-Middle Attack。雖然理論上也可以針對 3DES 進行此攻擊，但其複雜度非常高，達到2^{112}次計算，使得攻擊幾乎不可能成功。

Problem 8

If a company has encrypted its most sensitive data with a key held by the chief technology officer and that person was fired, the company would want to change its encryption key. Describe what would be necessary to revoke the old key and deploy a new one.

1. 確認密鑰撤銷的範圍和影響

要確定舊密鑰被用於哪些數據的加密，以及在哪些系統中使用，評估更換密鑰對業務運營的影響。

2. 通知所有使用密鑰的實體

撤銷的是對稱密鑰，所有共享該密鑰的單位都要被通知。

撤銷的是非對稱密鑰對，需要撤銷包含 CTO 私鑰相關公鑰的證書，並使用

CRL 或 OCSP 通知所有依賴方。

通知內容都應該包括密鑰識別資訊、撤銷時間、撤銷原因。

3. 生成新密鑰

使用符合 NIST 標準的安全方法生成新的加密密鑰。

4. 安全分發新密鑰

若是對稱密鑰，使用安全的通信通道或密鑰管理系統將新密鑰分發給所有需要的單位。

若是非對稱密鑰對，則通過受信任的通道分發新密鑰對，並在使用 PKI 時獲取新證書。

5. 更新密鑰管理系統

在密鑰管理系統中更新新密鑰的狀態，確保新密鑰被標記為有效，並且舊密鑰被標記為撤銷。考慮將撤銷的密鑰列入黑名單，以便進行審計和管理，而不僅僅是刪除它。

6. 若使用配對密鑰

使用的是配對密鑰，直接通知另一單位密鑰已被撤銷。如果密鑰是註冊在某個基礎設施中，需通知該基礎設施密鑰已被撤銷。

7. 審計和記錄

保留密鑰更換過程的詳細記錄，包括生成新密鑰、數據重新加密、舊密鑰銷毀等步驟。定期審核密鑰管理流程和記錄，確保符合安全政策和標準。

Problem 9

a) Explain how Alice's laziness might get her in trouble.

1. Alice 的公鑰模數 n 在每個月都保持不變，而公鑰指數 e 按照質數序列遞增，因此攻擊者可以預測她未來的 e。
2. 攻擊者可以對這個固定的 n 進行因數分解。成功後，攻擊者就可以使用 Alice 過去和未來的所有 e 來解密她的所有通訊，因為私鑰是 n 的質因數。
3. 對於較小的質數（例如 3, 5, 7 等），存在 Hastad's Broadcast Attack。

如果同一消息被多個小質數公鑰加密，攻擊者可以通過數學方法得到明文。

b) Explain how Alice's scheme could be broken.

1. 相鄰質數之間的差距通常不大。因此新的 n 值與之前的 n 值之間的差值也不大。有相似性則可以通過 common modulus attack 或中國剩餘定理來攻擊。

2. 質數序列可使因數分解變得更容易。若攻擊者發現 Alice 用的初始質數，並猜測到她使用的質數序列模式，因數分解攻擊的難度可能會降低，因為可以針對特定的質數範圍進行優化。

Problem 10

a) Describe how the matrix multiplication in the Inverse MixColumns step is performed within AES decryption. Your answer should include a brief explanation of the arithmetic used in the finite field GF(2^8) and how it differs from standard matrix multiplication.

AES 操作於有限域 GF(2^8)，其中每個元素是一個 8 位元的字節。在這個域中的乘法是 mod 一個不可約多項式進行的，通常是$x^8 + x^4 + x^3 + x + 1$。與標準矩陣乘法不同，在 GF(2^8)中的運算使用多項式算術，係數是二進制值。

b) Provide the mathematical expressions that represent multiplications.

在 GF(2^8) 中的乘 2 相當於左移 1 位和加法相當於 XOR。

$x \cdot 9 = x \cdot (2^3 + 1) = (((x \cdot 2) \cdot 2) \cdot 2) \oplus x$

$x \cdot 11 = x \cdot (2^3 + 2 + 1) = (((x \cdot 2) \cdot 2) \cdot 2) \oplus (x \cdot 2) \oplus x$

$x \cdot 13 = x \cdot (2^3 + 2^2 + 1) = (((x \cdot 2) \cdot 2) \cdot 2) \oplus ((x \cdot 2) \cdot 2) \oplus x$

$x \cdot 14 = x \cdot (2^3 + 2^2 + 2) = (((x \cdot 2) \cdot 2) \cdot 2) \oplus ((x \cdot 2) \cdot 2) \oplus (x \cdot 2)$

c) What are the advantages and potential drawbacks of using LUTs for this purpose?

優點：

1. 每次查表的時間複雜度是 O(1)，通過用簡單的內存訪問替代計算，減少了運行時的計算開銷。
2. 簡化了 AES 解密的實現，因為複雜的操作被簡單的查找操作所取代。

缺點：

1. 查表需要額外的內存
2. 查表易受到側信道攻擊，如緩存定時攻擊，會洩漏訪問的內存位置資訊。

Problem 11

PBC mode is not secure. Show how an attacker who knows IV, C_0, C_1, C_2 (which are public) and also knows that $m_1 = m_2 = x$ (for a known x) can easily compute m_0.

$C_0 = E(m_0) \oplus IV, C_1 = E(m_1) \oplus m_0, C_2 = E(m_2) \oplus m_1$

$m_1 = m_2 = x$，可得 $C_2 = E(m_1) \oplus m_1$

$C_2 \oplus m_1 = E(m_2) = E(m_1)$，可得$m_0 = C_1 \oplus E(m_1) = C_1 \oplus C_2 \oplus m_1$

攻擊者可以利用已知的密文和明文之間的關係，以及加密過程中的特性，輕鬆計算出未公開的明文塊。這說明 PBC 模式存在嚴重的安全漏洞，因此在實際應用中不應使用該模式。

Problem 12

Describe how to produce an existential forgery against this MAC scheme.

$M_1 = m_{10}||m_{11}|| \dots ||m_{1n}, M_2 = m_{20}||m_{21}|| \dots ||m_{2n}$

$T_1 = E_k(E_k(\dots E_k(IV_1 \oplus m_{10}) \oplus m_{11}) \dots \oplus m_{1n})$

$T_2 = E_k(E_k(\dots E_k(IV_2 \oplus m_{20}) \oplus m_{21}) \dots \oplus m_{2n})$

讓$M_3 = M_1||M_2$，M_3的前半部與M_1計算的結果相同，因此過程會產生T_1。以T_1作為新的 IV，對M_3的後半部進行 CBC 加密，會與M_2有一樣的過程，得到的最終 tag 會是T_2，因此便構造出 MAC 值為(IV_1, T_2)的M_3。

上述例子說明 MAC 結構存在安全漏洞，容易被偽造攻擊。

Extra Credit 1

安全目標

1. 確保只有授權用戶才能訪問敏感資訊
2. 防止未經授權的修改，確保存儲數據的準確性和完整性
3. 減少停機時間，使在攻擊期間也能運行
4. 跟蹤和記錄用戶操作，以檢測和響應安全漏洞或政策違規行為

角色與其功能

Site Administrators

職責：維護服務器和數據庫、執行安全政策

權限：能使用所有系統資源、可修改系統配置、可查閱審計記錄、能刪除用戶帳戶、組織所有者

Organization Owners

職責：管理其組織內的資料、添加或刪除組織成員、配置資料設置、為協作者打開使用權限

權限：完全控制組織內的資料、能夠邀請或刪除成員、修改庫設置和權限、資料管理員

Repository Administrators

職責：控制資料的訪問、合併請求和管理分支、配置資料設置

權限：資料設置、能夠添加或刪除協作者、管理請求和分支、協作者

Collaborators

職責：推送和拉取庫的變更、創建和審查請求、評論問題和請求

權限：讀寫指定資料的訪問權、能夠創建分支和請求

Users

職責：創建和管理個人資料、對公共資料進行貢獻、參與問題的討論

權限：完全控制個人資料、公共資料的讀取權限、作為協作者對資料有有

限的修改權限

Visitors

職能：查看公共資料內容、閱讀公共資料中的問題和討論。

權限：公共資料的讀取權

安全政策

1. 實施角色訪問控制（RBAC），確保用戶擁有其角色所需的最低權限。
2. 使用行業標準加密協議加密靜態和傳輸中的數據並定期備份數據。
3. 定期培訓員工事件響應流程並進行演練。
4. 保持詳細的用戶操作和系統變更日誌，定期審查日誌和審計記錄以檢測和響應潛在的安全問題。
5. 為用戶提供定期的安全培訓和更新，強調保護其帳戶和數據的最佳做法。

Extra Credit 2

a) Here are two 8-character English words encrypted with the same "one-time pad".

兩組密文使用相同的 pad，因此可以通過對應字節的 XOR 運算來找到明文之間的關係。

e9^f4=1d, 3a^3a=00, e9^fe=17, c5^c7=02

fc^e1=1d, 73^68=1b, 55^4a=1f, d5^df=0a

結果為 1d 00 17 02 1d 1b 1f 0a

由於單詞長度相同且加密方式是對應字節 XOR，所以可以推測原文是 8 字符英文單詞 XOR 結果。密文為 ASCII 碼範圍內，可以寫程式來窮舉。

```python
def xor_bytes(byte1, byte2):
    return [b1 ^ b2 for b1, b2 in zip(byte1, byte2)]
def bytes_to_string(byte_list):
    return ''.join(chr(b) for b in byte_list)
cipher1_hex = "e9 3a e9 c5 fc 73 55 d5"
cipher2_hex = "f4 3a fe c7 e1 68 4a df"
cipher1 = bytes.fromhex(cipher1_hex)
cipher2 = bytes.fromhex(cipher2_hex)
xor_result = xor_bytes(cipher1, cipher2)
print("XOR結果: ", xor_result)
# 正確應該要用較多單字的檔案來取代
simple_words = [
    "feedback", "security", "identity", "function", "analysis", "computer"
    "hardware", "software", "database", "networks"
]
# 過濾掉不是8個字的單字
words = [word.lower() for word in simple_words if len(word) == 8]
# 嘗試找到兩個異或結果為xor_result的單詞
for word1 in words:
    word1_bytes = [ord(c) for c in word1]
    potential_word2_bytes = xor_bytes(word1_bytes, xor_result)
    potential_word2 = bytes_to_string(potential_word2_bytes)
    if potential_word2 in words:
        pad = xor_bytes(word1_bytes, cipher1)
        print(f"word1 = {word1}, word2 = {potential_word2}, pad = {pad}")
```

```
XOR結果:  [29, 0, 23, 2, 29, 27, 31, 10]
word1 = security, word2 = networks, pad = [154, 95, 138, 176, 142, 26, 33, 172]
word1 = networks, word2 = security, pad = [135, 95, 157, 178, 147, 1, 62, 166]
```

窮舉的結果為：("networks", "security")

Extra Credit 3