資訊安全 HW2

40647027S 陳冠穎

1. (1.) 135 mod 61
   135 = 2*61 + 13
   61 = 4*13 + 9
   13 = 1*9 + 4
   9 = 2*4 + 1

   1 = 9 - 2*4
   1 = 9 - 2(13-1*9) = 3*9 – 2*13
   1 = 3(61 - 4*13) – 2*13 = 3*61 - 14*13
   1 = 3*61 – 14(135 - 2*61) = 31*61 – 14*135
   1 = 31*61 + (-14)*135
   The modular multiplicative inverse of 135 mod 61 = 47 + 61k, k∈ Z.

   (2.) 7465 mod 2464
   7465 = 3*2464 + 73
   2464 = 33*73 + 55
   73 = 1*55 + 18
   55 = 3*18 + 1

   1 = 55 – 3*18
   1 = 55 – 3(73 - 1*55) = 4*55 – 3*73
   1 = 4(2464 – 33*73) – 3*73 = 4*2464 – 135*73
   1 = 4*2464 – 135(7465 - 3*2464) = 409*2464 – 135*7465
   1 = 409*2464 + (-135) * 7465

   The modular multiplicative inverse of 7465 mod 2464 = 2329 + 2464k, k∈Z.

   (3.) 42828 mod 6407
   42828 = 6*6407 + 4386
   6407 = 4386 + 2021
   4386 = 2*2021 + 344
   2021 = 5*344 +301
   344 = 301 +43
   301 = 7*43 + 0
   因為 gcd(42828, 6407) = 43，說明 42828 與 6407 不互質，因此模反元素不存在。

2. 因為 gcd(4, 13) = 1，By Fermat's little theorem $4^{12} \equiv 1\ mod(13)$。

$$4^{255} \equiv 4^{12^{21}} * 4^3 \equiv 1 * 4^3 \equiv 64 \equiv 12\ mod(13)$$

因為 gcd(7, 93) = 1，By Fermat's little theorem $7^{92} \equiv 1\ mod(93)$

$$7^{1013} \equiv 7^{92^{11}} * 7 \equiv 1 * 7 \equiv 7 \text{ mod}(93)$$

3. 令 $\mathrm{P} = p_1 p_2 \dots p_k$

且$P_i = \frac{P}{p_i}, \forall\ i = 1,2, \dots, k$

因為$p_1, \dots, p_k$彼此互質，所以 gcd(P_i, p_i)=1, $\forall \mathrm{i} = 1,2, \dots, \mathrm{k}$

因此P_i在 mod p_i下具有乘法反元素M_i,

即

$$M_i P_i \equiv 1\ (\text{mod}\ p_i), \forall\ i = 1,2, \dots, k$$

,

取

$$\mathrm{n} = \sum_{i=1}^{k} n_i M_i P_i\ (mod\ P)$$

因為 $p_j | P_i, \forall \mathrm{i} \neq \mathrm{j}$ ，所以

$$n_i M_i P_i \equiv 0\ \left(\text{mod}\ p_j\right), \forall \mathrm{i} \neq \mathrm{j} \rightarrow \mathrm{n} \equiv \sum_{i=1}^{k} n_i M_i P_i \equiv n_j M_j P_j \equiv n_j\ \left(mod\ p_j\right), \forall j = 1,2, \dots, k$$

所以 n 為該系統的解。

接著證明其唯一性:

若存在x_1, x_2使得

$$x_1 \equiv x_2 \equiv n_j\ \left(mod\ p_j\right), \forall \mathrm{j} = 1,2, \dots, \mathrm{k} \rightarrow p_j | (x_1 - x_2), \forall \mathrm{j} = 1,2, \dots, \mathrm{k}$$

，因為$\mathrm{P} = p_1 p_{2 \dots} p_k$，所以$\mathrm{P} | (x_{1-} x_2)$，因此$x_1 \equiv x_2 (mod\ P)$。

4. (1.) 令$\overline{L_i}, \overline{R_i}$為以$\overline{L_{i-1}}, \overline{R_{i-1}}$為輸入且使用$\bar{k}$為 key，經過第 i 輪費斯托網路之後的結果。

$L_i = R_{i-1} \rightarrow \overline{L_i} = \overline{R_{i-1}}$

$\because$ 在 $\mathrm{f}(\bar{R}, \bar{k})$中$\bar{R}$跟$\bar{k}$會做 xor $\therefore \mathrm{f}(\bar{R}, \bar{k}) = \mathrm{f}(R, \mathrm{k})$

$R_i = \overline{L_{i-1}} \oplus \mathrm{f}(\overline{R_{i-1}}, \overline{k_i}) = \overline{L_{i-1}} \oplus \mathrm{f}(R_{i-1}, k_i) = \overline{L_{i-1} \oplus \mathrm{f}(R_{i-1}, k_i)}$

所以當使用$\bar{X}$作為輸入$\bar{k}$作為 key，則完成全部的流程後加密的結果為$\bar{Y}$。

(2.) 假設明文 M 是 1111 則可以知道$\bar{M}$是 0000，如果用 K 解密 C 的結果為 0101，則暗示了用$\bar{K}$解$\bar{C}$的結果為 1010，因此只要解出來是 1111 或 0000 其中一個出現則可以知道 key 是 K 或$\bar{K}$，代表嘗試一種 key 就已經嘗試了兩種 key，因此需要嘗試的 key 數目為$\frac{2^{56}}{2} = 2^{55}$。

5.

(1.) $(x^3 + 1) = (\mathrm{x} + 1) * (x^2 + \mathrm{x} + 1)$

$\gcd(x^3 + 1, x^2 + x + 1) = (x^2 + \mathrm{x} + 1)$, where the coefficient is in Z_2.

(2.) $x^3 + x + 1 = x * (x^2 + 1) + 1$

$\gcd(x^3 + x + 1, x^2 + 1) = 1$, where the coefficient is in Z_3.

(3.) $x^4 + 8x^3 + 7x + 8 = (6\mathrm{x} + 10) * (2x^3 + 9x^2 + 10\mathrm{x} + 1) + (4x^2 + 9)$

$(2x^3 + 9x^2 + 10\mathrm{x} + 1) = (6\mathrm{x} + 5) * (4x^2 + 9) + 0$

$\gcd(x^4 + 8x^3 + 7x + 8, 2x^3 + 9x^2 + 10\mathrm{x} + 1) = (4x^2 + 9)$, where the coefficient is in Z_{11}.

6.明文為:

My power flurrie
s through the ai
r into the groun
d. My soul is sp
iraling in froze
n fractals all a
round. And one t
hought crystalli
zes like an icy
blast. I'm never
 going back, the
 past is in the
past.

我是參考 wiki 的做法: Reference
程式碼: 附加檔案 attack.py，使用 python3，額外用到的 Package 為 requests。

主要用到的公式

進行密碼塊連結解密的數學公式為

$$P_i = D_K(C_i) \oplus C_{i-1},$$
$$C_0 = IV.$$

Cipher 的第一組為 IV
因此從第二組開始送，原理為只要自己嘗試的 IV 與 cipher 送過去為合法的 padding 就能夠一個 Byte 一個 Byte 找到 D(C_i)。
一開始先從最後一個 Byte 嘗試，假如 IV 的最後一個 Byte 為 IV_15，Cipher 的最後一個 Byte 為 C_15，如果送到 Server 嘗試成功為合法的 padding，則代表 IV_15 xor D(C_15)為 0x01，則 D(C_15) = IV_15 xor 0x01，可以找到 D(C_15)，再來找倒數第二個 Byte，先將 D(C_15) xor 0x02 則只要嘗試 IV 的倒數第二個 Byte 有 0~255 共 256 種可能，因為 D(C_15) xor 0x02 到 Server 那邊解開一定會是 0x02，就能再找到 D(C_14)，以此類堆。
當解完 16 個 Byte 後，再將 D(C)與前一組 Cipher 的每個 Byte 做 xor 則可以得到前一組 cipher 的明文。
重複做完共 13 組即可得到 13 段明文，組合起來即為要求的訊息。

7. Crypto RSA Lab 因內容較多，固寫在另一份 PDF 檔案，檔名: Lab.HW2.pdf