Suggested Solutions to Homework Assignment #1B

(prepared by Hung-Wei Hsu)

1. Exercise problems from [Stallings 6E]:

4.16 **a**. We want to show that $m > 2r$. This is equivalent to $qn + r > 2r$, which is equivalent to $qn > r$. And since $n > r$, we must have $qn > r$.

b. If you study the pseudocode for Euclid's algorithm in the text, you can see that the relationship defined by Euclid's algorithm can be expressed as $A_i = q_i A_{i+1} + A_{i+2}$. The relationship $A_{i+2} < A_i/2$ follows immediately from **a**.

c. From **b**, we see that $A_3 < 2^{-1}A_1$, that $A_5 < 2^{-1}A_3 < 2^{-2}A_5$, and in general that $A_{2j+1} < 2^{-j}A_1$ for all integers $j, 1 < 2j+1 \leq k+2$, where k is the number of steps in the algorithm. As a consequent, $2^j A_{2j+1} < A_1$.
Now since $1 \leq m, n \leq 2^N$, we have $1 \leq A_1, A_2 \leq 2^N$. That makes $2^j A_{2j+1} < 2^N$. If the ending step number is odd, we take $j = (k+1)/2$ to obtain $(k+1)/2 < N$, and if it is even, we take $j = k/2$ to obtain $k/2 < N$. In either case $k < 2N$.

5.1 We want to show that $d(x) = a(x) \times b(x) \bmod (x^4 + 1) = 1$. Substituting into Equation (5.12) in Appendix 5A, we have:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 0E \\ 09 \\ 0D \\ 0B \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

But this is the same set of equations discussed in the subsection on the MixColumn transformation:

$$\begin{aligned}
&(\{0E\} \cdot \{02\}) \oplus \{0B\} \oplus \{0D\} \oplus (\{09\} \cdot \{03\}) = \{01\} \\
&(\{09\} \cdot \{02\}) \oplus \{0E\} \oplus \{0B\} \oplus (\{0D\} \cdot \{03\}) = \{00\} \\
&(\{0D\} \cdot \{02\}) \oplus \{09\} \oplus \{0E\} \oplus (\{0B\} \cdot \{03\}) = \{00\} \\
&(\{0B\} \cdot \{02\}) \oplus \{0D\} \oplus \{09\} \oplus (\{0E\} \cdot \{03\}) = \{00\}
\end{aligned}$$

The first equation is verified in the text. For the second equation, we have $\{09\} \cdot \{02\} = 00010010$; and $\{0D\} \cdot \{03\} = \{0D\} \oplus (\{0D\} \cdot \{02\}) = 00001101 \oplus 00011010 = 00010111$. Then

$$\begin{aligned}
\{09\} \cdot \{02\} &= 00010010 \\
\{0E\} &= 00001110 \\
\{0B\} &= 00001011 \\
\{0D\} \cdot \{03\} &= \underline{00010111} \\
& 00000000
\end{aligned}$$

For the third equation, we have $\{0D\} \cdot \{02\} = 00011010$; and $\{0B\} \cdot \{03\} = \{0B\} \oplus$

$(\{0B\} \cdot \{02\}) = 00001011 \oplus 00010110 = 00011101$. Then

$$\begin{array}{lcl} \{0D\} \cdot \{02\} & = & 00011010 \\ \{09\} & = & 00001001 \\ \{0E\} & = & 00001110 \\ \{0B\} \cdot \{03\} & = & \underline{00011101} \\ & & 00000000 \end{array}$$

For the fourth equation, we have $\{0B\} \cdot \{02\} = 00010110$; and $\{0E\} \cdot \{03\} = \{0E\} \oplus (\{0E\} \cdot \{02\}) = 00001110 \oplus 00011100 = 00010010$. Then

$$\begin{array}{lcl} \{0B\} \cdot \{02\} & = & 00010110 \\ \{0D\} & = & 00001101 \\ \{09\} & = & 00001001 \\ \{0E\} \cdot \{03\} & = & \underline{00010010} \\ & & 00000000 \end{array}$$

Thus, we found out $d(x) = a(x) \times b(x) \bmod (x^4 + 1) = 1$ by calculating these four equations.

5.4 **a.**

00	04	08	0C
01	05	09	0D
02	06	0A	0E
03	07	0B	0F

b.

01	05	09	0D
00	04	08	0C
03	07	0B	0F
02	06	0A	0E

c.

7C	6B	01	D7
63	F2	30	FE
7B	C5	2B	76
77	6F	67	AB

d.

7C	6B	01	D7
F2	30	FE	63
2B	76	7B	C5
AB	77	6F	67

e.

75	87	0F	B2
55	E6	04	22
3E	2E	B8	8C
10	15	58	0A

5.6 **a.** AddRoundKey

b. The MixColumn step, because this is where the different bytes interact with each other.

c. The ByteSub step, because it contributes nonlinearity to AES.

d. The ShiftRow step, because it permutes the bytes.

e. There is no wholesale swapping of rows or columns. AES does not require this step because: The MixColumn step causes every byte in a column to alter every other byte in the column, so there is not need to swap rows; The ShiftRow step moves bytes from one column to another, so there is no need to swap columns.

6.4 **a.** No. For example, suppose C_1 is corrupted. The output block P_3 depends only on the input blocks C_2 and C_3.

b. An error in P_1 affects C_1. But since C_1 is input to the calculation of C_2, C_2 is affected. This effect carries through indefinitely, so that all ciphertext blocks are affected. However, at the receiving end, the decryption algorithm restores the correct plaintext for blocks except the one in error. You can show this by writing out the equations for the decryption. Therefore, the error only effects the corresponding decrypted plaintext block.

6.7 For this padding method, the padding bits can be removed unambiguously, provided the receiver can determine that the message is indeed padded. One way to ensure that the receiver does not mistakenly remove bits from an unpadded message is to require the sender to pad every message, including messages in which the final block is already complete. For such messages, an entire block of padding is appended.

6.8 (Modified) As CFB mode has a step of shift register, if there's a bit error in a segment of ciphertext, in the decryption phase, not only will that problematic ciphertext segment itself fail to be decrypted correctly, it will also be put into the shift register for the follow-up segments' decryption, causing the decryption of other segments to be incorrect as well.

The length the error propagates depends on how long the time that problematic ciphertext segment stays in the shift register. In our context here, with $b = 64, s = 8$, the error segment will stay in the shift register for the next eight rounds of decryption, thus 8 segments the propagation will be. Counting the error segment itself, totally are there 9 segments being affected.

6.10 $O_i = C_i \oplus P_i$

2. Consider pseudorandom number generation based on block ciphers and assume AES-128 is used as the encryption algorithm. What is the expected period of the bit stream with the OFB mode of operation? Please justify your answer. (10 points)

Solution.

To compute the expected period of the bit stream, we shall first compute the expected number of rounds the encryption algorithm should go to generated a repeated block. And multiplying the expected value with the block length, we derive the expected period of the bit stream.

In the following table, the first column is about the number of rounds that the encryption algorithm has been applied. The second column shows the relationship between seeds. The third column shows the probability that the generated bit stream repeats exactly at the corresponding round. Assume that the initial seed value is V_0.

Let's look at the probability of repeat to happen at round 3 for example. The first part $\frac{1}{2^{128}-2}$

is the probability for $V_0 = V_3$, i.e., a repeat firstly occurs given that $V_0 \neq V_1$, $V_0 \neq V_2$. The fraction is reasoned as follows:

The AES encryption algorithm is invertible, a one-to-one mapping. There's no way that a new generated block (at here, it's V_3) equals to any previous generated block (V_1, V_2) as they've been mapped onto once. So the number of possibilities of this generated block (the denominator part of the fraction) should be $2^{128} - 2$ (without V_1 and V_2). And the numerator part being one is trivial, for the only way that a repeat can occur, we need V_3 to equal V_0.

The rest of the line represents the probability of having $V_0 \neq V_1$, $V_0 \neq V_2$ (with $V_1 \neq V_2$ for sure).

rounds	seed values transition	probability that the generated bit stream repeat
1	$V_0 \to V_1$	$\frac{1}{2^{128}}$
2	$V_0 \to V_1 \to V_2$	$\frac{1}{2^{128}-1} \times \frac{2^{128}-1}{2^{128}} = \frac{1}{2^{128}}$
3	$V_0 \to V_1 \to V_2 \to V_3$	$\frac{1}{2^{128}-2} \times \frac{2^{128}-1}{2^{128}} \times \frac{2^{128}-2}{2^{128}-1} = \frac{1}{2^{128}}$
$\vdots$	$\vdots$	$\vdots$
2^{128}	$V_0 \to V_1 \to \cdots \to V_{2^{128}}$	$\frac{1}{2^{128}}$

So with the generalization up to round 2^{128}, the expected number of rounds to get a repeated bit stream is:

$1 \times \frac{1}{2^{128}} + 2 \times \frac{1}{2^{128}} + \cdots + 2^{128} \times \frac{1}{2^{128}}$
$= (1 + 2 + \cdots + 2^{128}) \times \frac{1}{2^{128}}$
$= \frac{(2^{128}+1) \times 2^{128}}{2} \times \frac{1}{2^{128}}$
$= \frac{(2^{128}+1)}{2}$

As a result, the expected period of the bit stream is $128 \times \frac{(2^{128}+1)}{2} = 64 \times (2^{128} + 1)$.

$\square$