Mike Simister - 10095107
Introduction to Cryptography CPSC 418 Fall 2016
Department of Computer Science
University of Calgary

November 18, 2016

HOME WORK #3

Problem	Marks
1	
2	
3	
4	
5	
6	
7	
Total	

Problem 1. (Flawed hash function based MAC designs, 28 marks)

(a)

(i)

$$\begin{aligned} M_1 &= f(f(IV, K), M_1) \\ &= f(H_1, M_1') \leftarrow \text{ note: } M_1 \text{ is now padded} \\ &= H(M) \\ &= H_L \\ &= MAC_1 \end{aligned}$$

$$\begin{aligned} M_2 &= H(K||M_1'||X) \\ &= f(f(f(IV, K), M_1'), X)) \\ &= f(MAC_1, X) \\ &= MAC_2 \end{aligned}$$

(ii)

$$\begin{aligned} C_k(M) &= H(M'||K) \text{ where M' is M with padding} \\ M_1 &= f(IC, M) \\ &= f(f(IV, M_1'), K) \\ &= H(M) \\ &= H_L \\ &= MAC_1 \end{aligned}$$

$$\begin{aligned} M_2 &= f(IV, M_2) \\ &= f(f(IV, M), K) \leftarrow \text{ since } H(M_2) = H(M_1) \\ &= H(M) \\ &= HL \\ &= MAC_1 \end{aligned}$$

(b)

(i)

M_1 is one block so no padding, $MAC_1 = E_k(M_1)$

M_2 is one block so no padding, $MAC_2 = E_k(M_2)$

M_3 is two blocks i.e. $\begin{cases} \text{Block 1} & = M_1 \\ \text{Block 2} & = \{0\}^n \end{cases}$

$$\begin{aligned} MAC_3 &= E_k(M_1 \oplus \{0\}^n) \\ &= E_k(MAC_1) \end{aligned}$$

(ii) M_1 is one block so no padding, $MAC_1 = E_k(M_1)$

M_2 is one block so no padding, $MAC_2 = E_k(M_2)$

M_3 is two blocks so it will take 2 iterations to compute MAC_3

Iteration 1 $E_k(M_1) = MAC_1$

Iteration 2 $E_k(MAC_1 \oplus X) = MAC_3$

M_4 is two blocks so it will take 2 iterations to compute MAC_4

Iteration 1 $E_k(M_2) = MAC_2$

Iteration 2 $E_k(MAC_1 \oplus X \oplus MAC_2 \oplus MAC_2)$

Iteration 2 $E_k(MAC_1 \oplus X) = MAC_3 \leftarrow$ Since $MAC_2 \oplus MAC_2$ cancel

Problem 2. (A modified man-in-the-middle attack on Diffie-Hellman, 12 marks)

(a)

Alice	Mallory	Bob
chooses a	chooses q	chooses b
computes g^a		computes g^b
sends $g^a \rightarrow$	intercepts g^a sends $g^{aq} \rightarrow$	recieves g^{aq}
recieves $g^{bq} \leftarrow$	$\leftarrow g^{bq}$ sends intercepts g^b	$\leftarrow g^b$ sends
computes $K = g^{abq}$		computes $K = g^{abq}$

(b)

Things we know:

(1) g is a primitive root, or a generator in $\mathbb{Z}_p^*$

(2) the order of g is $|\mathbb{Z}_p^*| = (p-1)$

(3) $p = mq + 1$, $(p-1) = mq$

(4) $g^{(p-1)} \equiv 1 \pmod p$, $g^{mq} \equiv 1 \pmod p$

(5) $g^{2mq} \pmod p \equiv g^{mq} * g^{mq} \pmod p \equiv 1 * 1 \pmod p \equiv 1 \pmod p$

(6) $g^{(k)mq} \pmod p \equiv 1 \pmod p$ where $k \in \mathbb{Z}_{0\geq}$, that is $0 \leq k$

So, if $ab \geq m$, we can say that $abq = ((k)m + r)q$ where $0 \leq r < m$, $1 \leq k$

and $g^{(k)mq} \equiv 1 \pmod p$

So, when $ab = (k)m$, $g^{abq} \equiv g^{(k)mq} \equiv 1 \pmod p$

Otherwise, $g^{abq} = g^{((k)m+r)q} \equiv g^{(k)mq} * g^{rq} \equiv 1 * g^{rq} \pmod p$

And since, g is a generator, and q is a constant the values obtained from calculating g^0 $\pmod p$ to $g^{qr} \pmod p$ are unique values and since $0 \leq r < m$ there are m possible values

(c)

The advantage of this variation is that once the attacker, Mallory, has intercepted and modified both g^{aq} and g^{bq} there is no need to do any further work.
That is to say, Alice and Bob can continue communicating completely unaware that Mallory has tampered with their communication. Mallory can easily compute the key and decrypt/encrypt any/all messages she chooses. In the version of discussed in class, after Mallory intercepts g^a and g^b sends g^{ae} or g^{be} Mallory would need to continuously intercept and encrypt/decrypt ALL messages. Otherwise, if even one message is not intercepted by Mallory, the decryption by the intended recipient will fail, and alert the recipient that the system has been compromised.

Problem 3. (Binary Exponentiaion, 12 marks)

(a) Define $s_0 = b_0$ and $s_{i+1} = 2_{Si} + b_{i+1}$

$$
\begin{aligned}
\text{Suppose that: } S_i &= \sum_{n=0}^{i} b_n 2^{i-n} \\
\text{We want to show that: } S_{i+1} &= \sum_{n=0}^{i+1} b_n 2^{i+1-n} \\
\text{Base Case: } &i = 0, \\
S_0 &= \sum_{n=0}^{0} b_n 2^0 = b_0 2^0 = b_0 \\
\text{So, } s_{i+1} &= 2_{Si} + b_{i+1} \\
&= 2\left(\sum_{n=0}^{i} b_n 2^{i-n}\right) + b_{i+1} \\
&= 2\left(b_0 s^{i-0}\right) + 2\left(b_i 2^{i-1}\right) + ... + 2\left(b_i 2^{i-i}\right) + b_{i+1} \\
&= b_0 2^{i+1} + b_1 2i + ... + b_i 2^1 + b(i+1)(2^0) \\
S_{i+1} &= \sum_{n=0}^{i+1} b_n 2^{i+1-n} \quad \square
\end{aligned}
$$

(b)

$$\text{Suppose } r_i \equiv a^{S_i} \pmod{m}$$

$$\begin{aligned}\text{Base Case } r_0 &\equiv a^1 \pmod{m}\\ r_0 &\equiv a \pmod{m} \text{ base case proved } \checkmark\end{aligned}$$

$$\text{W.T.S. } r_{i+1} \equiv a^{S_{i+1}} \pmod{m} \text{ for } 0 \leq i \leq K$$

by the algorithm definition, we are given:

$$r_{i+1} = \begin{cases} r_i^2 \pmod{m} \text{ if } b_{i+1} = 0, \\ r_i^2 a \pmod{m} \text{ if } b_{i+1} = 1 \end{cases}$$

So, in case 1 where $b_{i+1} = 0$,

$$\begin{aligned} r_{i+1} &= r_i^2 \\ r_{i+1} &= r_i * r_i \\ &= a^{S_i} * a^{S_i} \textbf{ By the I.H.} \\ &= a^{2S_i} \\ \text{since } b_{i+1} = 0 \text{ we can write: } &a^{2S_i} \text{ as } a^{2S_i + b_{i+1}} \\ \text{and by defn in part (a) above: } &2S_i + b_{i+1} = S_{i+1} \\ \text{so, } a^{2S_i} &= a^{S_{i+1}} = r_{i+1} \end{aligned}$$

$$\begin{aligned} \text{in case 2, where } b_{i+1} = 1 \\ r_{i+1} &= r_i^2 * a \\ r_{i+1} &= r_1 * r_i * a \\ &= a^{S_i} * a^{S_i} * a \textbf{ by I.H.} \\ &= a^{2S_i + 1} \\ \text{since } b_{i+1} &= 1, \\ a^{2S_i+1} &= a^{2Si + b_{i+1}} \\ \text{and by the same defn in part (a), we can say that:} \\ a^{2Si + b_{i+1}} &= a^{S_{i+1}} = r_{i+1} \quad \square \end{aligned}$$

(c)

$$a_n \equiv r_k \pmod{m}, \text{ let } r_k = r_i$$
$$a_n \equiv r_i \equiv a^{S_i} \pmod{m}$$

where:

$$n = b_0 2^k + b_1 2^{k-1} + ... + b_{k-1} 2 + b_0 \textbf{ by defn}$$
$$S_i = b_0 2^{i-0} + b_i 2^{i-1} + ... + b_i 2^{i-i} \textbf{ by unwrapping the } \sum \textbf{ in part (a)}$$

so,

$$n = S_i$$

so,

$$a^n \equiv a^{S_i} \equiv r_i \pmod{m} \quad \square$$

Problem 4. (RSA toy example for practicing binary exponentiaion)

(a)
Using the binary exponentiation algorithm where, $a = 17, n = 11, m = 77$ to calculate $a^n \pmod{m}$ since:

$$\begin{aligned} C &\equiv M^e \pmod{n} \\ &\equiv 17^{11} \pmod{77} \end{aligned}$$

so $n = 1011$

$$\begin{aligned} r_0 &= a \pmod{n} \\ &= 17 \pmod{77} \end{aligned}$$

$$\begin{aligned} r_1 &= (r_0)^2 \pmod{77} \\ &= 17^2 \pmod{77} \\ &= 58 \pmod{77} \end{aligned}$$

$$\begin{aligned} r_2 &= (r_1)^2 * 17 \pmod{77} \\ &= 58^2 * 17 \pmod{77} \\ &= 54 \pmod{77} \end{aligned}$$

$$\begin{aligned} r_3 &= (r_2)^2 * 17 \pmod{77} \\ &= 54^2 * 17 \pmod{77} \\ &= 61 \pmod{77} \end{aligned}$$

so, $C = 61$

(b)

$$\begin{aligned}
n &= pq \text{ where } p, q \text{ are prime}\\
77 &= pq\\
&= 7 * 11\\
&\text{so}\\
\phi(n) &= 6 * 10\\
&= 60\\
&\text{we want } d \text{ such that } de \equiv 1 \pmod{\phi(n)}\\
&\text{or } (d)11 \equiv 1 \pmod{60}
\end{aligned}$$

Extended Euclidean Algorithm:

GCD(60,11) =

A	B	Q	R	Factors	
60	11	5	5	60 =	11(5) + 5
11	5	2	1	11 =	5(2) + 1
5	1	5	0	5 =	5(1) + 0

$$\begin{aligned}
1 &= 11 - 2(5)\\
&= 11 - 2(60 - 5(11))\\
&= 11(11) - 2(60)
\end{aligned}$$

so, $d = 11$, check:
$11 * 11 \equiv 1 \pmod{60}$
$121 \equiv 1 \pmod{60}$

(c)

Decrypt $M \equiv C^d \pmod{77}$
let $a = C = 32$, $n = d = 11$, $m = 77$
$n = 1011$

$$\begin{aligned} r_0 &= a \pmod{77} \\ &= 32 \pmod{77} \end{aligned}$$

$$\begin{aligned} r_1 &= (r_0)^2 \pmod{77} \\ &= 32^2 \pmod{77} \\ &= 23 \pmod{77} \end{aligned}$$

$$\begin{aligned} r_2 &= (r_1)^2 * 32 \pmod{77} \\ &= 23^2 * 32 \pmod{77} \\ &= 65 \pmod{77} \end{aligned}$$

$$\begin{aligned} r_3 &= (r_2)^2 * 32 \pmod{77} \\ &= 65^2 * 32 \pmod{77} \\ &= 65 \pmod{77} \end{aligned}$$

so, $M = 65$

Submitted by Mike Simister - 10095107 on November 18, 2016.