CPSC 418 / MATH 318 — Introduction to Cryptography
ASSIGNMENT 3

Name: Artem Golovin
Student ID: 30018900

Problem 1 — A modified man-in-the-middle attack on Diffie-Hellman, 12 marks

(a) Let $y_a \equiv (g^a)^q \pmod{p}$, $y_b = (g^b)^q \pmod{p}$ and key K:

  i. *Alice* receives malicious y_a and sends it to *Bob*.
  ii. *Bob* receives malicious y_b and sends it to *Alice*.
  iii. *Alice* computes $K \equiv y_b^a \equiv ((g^b)^q)^a \pmod{p}$
  iv. *Bob* computes $K \equiv y_a^b \equiv ((g^a)^q)^b \pmod{p}$
  v. *Alice* and *Bob* get the same key K, because:

$$y_b^a \equiv ((g^b)^q)^a \equiv g^{bqa} \equiv g^{aqb} \equiv ((g^a)^q)^b \equiv y_b^a \pmod{p}$$

(b) ???

(c) In this version, $Mallory$ does not have to pick a number e, where $1 < e < p$. Therefore, by knowing values $g^a \pmod{p}$ and $g^b \pmod{p}$, $Mallory$ is more likely to compute $g^{abq} \pmod{p}$, which is a private key used by *Alice* and *Bob*.

Problem 2 — RSA and binary exponentiation, 24 marks

(a) i. Given $M = 17$, public key $(e, n) = (11, 77)$:

$$C \equiv M^e \pmod{n}$$
$$C \equiv 17^{11} \pmod{77}$$

Binary exponentiation:

$$e = 11 = 1011_2$$
$$b_0 = 1, b_1 = 0, b_2 = 1, b_3 = 1$$
$$r_0 \equiv 17 \pmod{77}$$
$$r_1 \equiv 17^2 \equiv 58 \pmod{77}$$
$$r_2 \equiv 58^2 \times 17 \equiv 54 \pmod{77}$$
$$r_3 \equiv 54^2 \times 17 \pmod{77} \equiv 61 \pmod{77}$$

Therefore:

$$C \equiv 17^{11} \equiv 61 \pmod{77}$$

ii. Given that $n = pq$ and $\phi(n) = (p-1)(q-1)$ and $n = 77$, we can say that $p = 11$, $q = 7$.
To find d, we need to solve the following congruence:

$$de \equiv 1 \pmod{\phi(n)}$$
$$\text{Where } \phi(n) = (11-1)(7-1) = 60$$
$$d \times 11 \equiv 1 \pmod{60}$$

Solving $gcd(e, \phi(n)) = gcd(11, 60) = 1$ to confirm that inverse of e exists:

$$60 = 11 \times 5 + 5$$
$$11 = 10 \times 1 + 1$$
$$10 = 2 \times 5 + 0$$

Applying Extended Euclidean Algorithm to find d:

$$1 = 11 - 10 = 11 - ((2 \times 5) + 0) = 11 - (2 \times 5) - 0$$
$$= 11 - 2 \times (60 - 11 \times 5) = 11 - 2 \times 60 + 10 \times 11$$
$$= 11 \times 11 + (-2) \times 60$$

Therefore $gcd(60, 11) = 11 \times 11 + (-2) \times 60$ and $d = 11$:

$$11 \times 11 \equiv 1 \pmod{60}$$

iii. Given $C = 21$ and $(d, n) = (11, 77)$:

$$M \equiv C^d \pmod{n}$$
$$M \equiv 32^{11} \pmod{77}$$

Binary exponentiation:

$$11 = 1011_2$$
$$b_0 = 1, b_1 = 0, b_2 = 1, b_3 = 1$$

$$r_0 \equiv 32 \pmod{77}$$
$$r_1 \equiv 32^2 \equiv 23 \pmod{77}$$
$$r_2 \equiv 23^2 \times 32 \equiv 16928 \equiv 65 \pmod{77}$$
$$r_3 \equiv 65^2 \times 32 \equiv 135200 \equiv 65 \pmod{77}$$

Therefore, $M = 65$

(b) i. Given $s_0 = b_0$, $s_{i+1} = 2s_i + b_{i+1}$ for $0 \leq i \leq k-1$

Proof. Base case. Let $i = 0$

$$s_i = \sum_{j=0}^{i} b_j 2^{i-j}$$
$$s_0 = \sum_{j=0}^{0} b_j 2^{0-0}$$
$$= b_0$$

Induction hypothesis. Assume $i = m$, where $m \in \mathbb{Z}$ and $0 \leq m \leq k$:

$$s_m = \sum_{j=0}^{m} b_j 2^{m-j}$$
$$= b_0 + \sum_{j=0}^{m+1} b_j 2^{(m+1)-j}$$

Inductive case. Assume $i = m + 1$:

$$s_{m+1} = \sum_{j=0}^{m+1} b_j 2^{(m+1)-j}$$

Left hand side:

$$s_{m+1} = 2s_m + b_{m+1}$$

Right hand side:

$$\sum_{j=0}^{m+1} b_j 2^{(m+1)-j} = 2 \times \sum_{j=0}^{m} b_j 2^{(m-j)} + b_{m+1}$$
$$= 2s_m + b_{m+1}$$

$$\text{Therefore, } s_i = \sum_{j=0}^{i} b_j 2^{i-j}$$

$\square$

ii. Let r_i, $0 \le i \le k$, $k = \lfloor log_2 n \rfloor$.

Proof. Base case. Let $i = 0$:

$$\begin{aligned} r_i &\equiv a^{s_i} \pmod{m} \\ r_0 &\equiv a^{s_0} \pmod{m} \\ &\equiv a^{b_0} \equiv a \pmod{m} \end{aligned}$$

Where $s_0 = b_0$ is shown in part (i)
Induction hypothesis. Let $p = i$, $p \in \mathbb{Z}$, $0 \le p \le k$,

$$r_p \equiv a^{s_p} \pmod{m}$$

Inductive case. Show $r_{p+1} \equiv a^{s_{p+1}} \pmod{m}$.

$$\begin{aligned} &\textbf{Case } b_{i+1} = 0 \\ r_{p+1} &\equiv r_p^2 \pmod{m} \\ &\text{Therefore:} \\ a^{s_p+1} &\equiv a^{2s_p+b_{p+1}} \\ &\equiv a^{2s_p} \\ &\equiv a^{s_p} \times a^{sp} \\ &\equiv r_p \times r_p \\ &\equiv r_p^2 \pmod{m} \end{aligned}$$

$$\begin{aligned} &\textbf{Case } b_{i+1} = 1 \\ a^{s_p+1} &\equiv a^{2s_p+b_{p+1}} \\ &\equiv a^{s_p} \times a^{s_p} \times a^{b_{p+1}} \\ &\equiv r_p^2 \times a^{b_{p+1}} \\ &\equiv r_p^2 \times a \pmod{m} \end{aligned}$$

Therefore, $r_{p+1} \equiv a^{p+1} \pmod{m}$ and $r_i \equiv a^i \pmod{m}$ $\square$

iii. *Proof.* Given proof of (ii), we can say that $a^n \equiv r_k \pmod{m}$, where $n = s_k$. Therefore,

$$\begin{aligned} a^{s_k} &\equiv a^{2s_{k-1}+b_k} \\ &\equiv (a^{s_{k-1}})^2 \times a^{b_k} \\ &\equiv (r_{k-1})^2 \times a^{b_k} \\ &\equiv r_k \pmod{m} \end{aligned}$$

$\square$

Problem 3 — Fast RSA decryption using Chinese remaindering, 8 marks

(a) ok

Problem 4 — The ElGamal public key cryptosystem is not semantically secure, 10 marks

Proof. By definition, a PKC is polynomially secure if no passive attacker can in expected polynomial time select two plaintexts M_1 and M_2 and then correctly distinguish between $E(M_1)$ and $E(M_2)$, where $E(M_1)$ and $E(M_2)$ are encryptions of M_1 and M_2 respectively with probability $p > \frac{1}{2}$.

However, it is given that $Mallory$ can assert whether $C = E(M_1)$ or $C = E(M_2)$ in polynomial time using modular exponentiation by Euler's Criterion with probability $p' = 1$, $p' > p$. It contradicts the definition of polynomially secure PKC, and therefore shows that ElGamal is not semantically secure.

To further prove that statement, using the fact that $y \equiv g^x \pmod{p}$ and g is a primitive root of p, we can show that:

$$\begin{aligned}
\left(\frac{y}{p}\right) &\equiv (y)^{\frac{p-1}{2}} \equiv (g^x)^{\frac{p-1}{2}} \\
&\equiv (g^{x(p-1)})^{\frac{1}{2}} \\
&\equiv ((g^{(p-1)})^x)^{\frac{1}{2}} \\
&\equiv 1 \pmod{p}
\end{aligned}$$

We can see that $Mallory$ only nees to compute $\left(\frac{y}{p}\right)$ and $\left(\frac{C_2}{p}\right)$ to find $E(M_1)$ and $E(M_2)$. $\square$

Problem 5 — An IND-CPA, but not IND-CCA secure version of RSA, 10 marks

Proof. Given encryption of message M, $C = (s||t)$, where $s \equiv r^e \pmod{n}$, $t = H(r) \oplus M$ and $H : \{0,1\}^k \mapsto \{0,1\}^m$, and decryption of C, $M \equiv H(s^d \pmod{n}) \oplus t$, we can consider two plaintexts M_1 and M_2 with following enryption process: $C = (s||t) = (r^e(\pmod{n}))||H(r) \oplus M_i)$, where $i = 1$ or 2.

We can mount CCA using $C' = (s||t \oplus M_1)$:

$$\begin{aligned}
C' &= (s||t \oplus M_1) \\
&= (r^e(\pmod{n}))||H(r) \oplus M_i \oplus M_1)
\end{aligned}$$

Decryption of M_i:

$$\begin{aligned}
M_i &\equiv H(s^d(\pmod{n}))) \oplus t \\
&\equiv H(r^{ed}(\pmod{n}))) \oplus H(r) \oplus M_i \oplus M_1 \\
&\equiv H(r \pmod{n})) \oplus H(r) \oplus M_i \oplus M_1 \\
&= M_i \oplus M_1
\end{aligned}$$

Where $ed \equiv 1 \pmod{\phi(n)}$

Therefore, $M_i = 0$, C is an encryption of M_1, because $M_1 \oplus M_1 = 0$, otherwise $M_i = M_2$ $\square$