EE 595 (PMP) Introduction to Security and Privacy
Homework 2 - Solutions

Assigned: Thursday, January 26, 2017, **Due: Sunday, February 12, 2017**

Instructor: Tamara Bonaci
Department of Electrical Engineering
University of Washington, Seattle

Problem 1

Suppose that $m > 2$ users want to communicate securely and confidentially. Suppose further that each of the m users wants to be able to communicate with every other user without the remaining $m-2$ users being able to listen on their conversation. How many distinct keys are needed if we are using:

- A **symmetric key cryptosystem**, where two users use a shared secret key to communicate,
- A **public key cryptosystem**, where every user has a public key, K_E and a private (secret) key, K_D.

How many keys are needed for each type of cryptosystems if $m = 1000$?

Solution:
Case 1: Classical Cryptosystem
In classical cryptosystems, every user has to posses $m-1$ distinct encryption/decryption keys to be able to communicate with every other user. Since two communicating users share a common key, the total number of cryptographic keys is equal to: $N_1 = \frac{m(m-1)}{2}$. Therefore, for $m = 1000$, $N_1 = \frac{999 \cdot 10^3}{2}$ distinct keys are needed when classical cryptosystem is used.
Case 2: Public Cryptosystem
If m users are using a public key cryptosystem, then in total $N_2 = 2m$ distinct cryptographic keys are needed to make communication secure since every user is assigned one encryption key and one decryption key. Therefore, the total number of public key cryptographic keys $K = (K_E, K_D)$ is equal to:

$$N_2 = 2m \sim \mathcal{O}(m) \tag{1}$$

For $m = 1000$, $N_2 = 2000$ distinct keys are needed when public key cryptosystem is used.

Problem 2

Solve the following system of congruences:

$$13x \equiv 4 \pmod{99}$$
$$15x \equiv 56 \pmod{101}$$

Solution:
The given system of congruences can be solved in two steps:

- Find modular multiplicative inverses of 13 (mod 99) and 15 (mod 101), to get rid of those scaling factors, and
- Apply the Chinese remainder theorem to solve the given system of congruences

Let's start by finding the multiplicative inverses of $x_1 = 13 \pmod{99}$ and $x_2 = 15 \pmod{101}$ as follows:

$$13x_1 \equiv 1 \pmod{99} \rightarrow 13x_1 = 99\lambda + 1 = 91\lambda + (8\lambda + 1) \tag{2}$$

From equation (2) it follows that $\lambda = 8$. Therefore, we can write:

$$13x_1 = 728 + 65 \rightarrow x_1 = 61 \tag{3}$$

Similarly, we can write:

$$15x_2 \equiv 1 \pmod{101} \rightarrow 15x_2 = 101\mu + 1 = 90\mu + (11\mu + 1) \tag{4}$$

From equation (4) it follows that $\mu = 4$. Therefore, we can write:

$$15x_2 = 360 + 45 \rightarrow x_2 = 27 \tag{5}$$

Combining equations (3) and (5), we can redefine the system of congruences (2) as follows:

$$\begin{aligned} x &\equiv 244 \pmod{99} \rightarrow x \equiv 46 \pmod{99} \\ x &\equiv 1512 \pmod{101} \rightarrow x \equiv 98 \pmod{101} \end{aligned} \tag{6}$$

System of congruences (6) can be solved using the *Chinese reminder theorem*, where:

$$\begin{aligned} r &= 2, \\ a_1 &= 46,\ a_2 = 98, \\ m_1 &= 99,\ m_2 = 101, \\ M &= 9999,\ M_1 = 101,\ M_2 = 99 \end{aligned} \tag{7}$$

In order to find a unique solution of the system of congruences, X, we solve the following equations:

$$y_1 M_1 \equiv 1 \pmod{99} \rightarrow 101y_1 \equiv 1 \pmod{99} \rightarrow 2y_1 = 99\lambda + 1 = 98\lambda + (1 + \lambda) \tag{8}$$

From equation (8) it follows that $\lambda = 1$. Therefore $y_1 = 50$. Similarly, for y_2 we can write:

$$99y_2 \equiv 1 \pmod{101} \rightarrow 99y_2 = 101\mu + 1 \rightarrow 99y_2 = 99\mu + (2\mu + 1) \tag{9}$$

From equation (9) it follows that $\mu = 49$. Therefore $y_2 = 50$. Finally, we can compute the solution of the system of congruences X as follows:

$$\begin{aligned} X &= a_1 M_1 y_1 + a_2 M_2 y_2 \pmod{M} \\ &= 46 \cdot 101 \cdot 50 + 98 \cdot 99 \cdot 50 \pmod{9999} \\ &= 232300 + 485100 \pmod{9999} = 7471 \pmod{9999} \end{aligned} \tag{10}$$

Problem 3

In the RSA cryptosystem, a user's public key is given as $e = 31, n = 3599$. Please find the user's private key, and explain your procedure.

Solution:
To find the private key of the given user, e, we use the simple trick, saying that the most plausible choice for primes $p, q, p \cdot q = n$ is:

$$p \sim q \sim \sqrt{n} \tag{11}$$

Using equation (11), we observe that a good guess for a pair of primes (p, q) would be $(p = 59, q = 61)$, as $\sqrt{n} = 3599 = 59.9917$. We therefore find:

$$\phi(n) = \phi(p) \cdot \phi(q) = 58 \cdot 60 = 3480 \tag{12}$$

Given a public cryptographic key $K_E = (b, n) = (31, 3599)$, we observe that $\gcd(b, \phi(n)) = \gcd(31, 3480) = 1$. Therefore, there exist a unique multiplicative inverse of $b \pmod{\phi(n)}$, and using the key generation rules of the RSA cryptosystems:

$$ab = 1 \pmod{\phi(n)}$$

we know that that modular multiplicative inverse is exactly equal to the private cryptographic key $K_D = (a)$. We find such a multiplicative inverse using *Extended Euclidean Algorithm*:

$$\begin{aligned}
3480 &= 112(31) + 8 \rightarrow 8 = 3480 - 112(31) \\
31 &= 3(8) + 7 \rightarrow 7 = 31 - 3(8) \\
8 &= 1(7) + 1 \rightarrow 1 = 8 - 1(7) \\
1 &= 0(7) + 1 \\
1 &= 8 - 31 + 3(8) = 4(8) - 31 = 4(3480) - 448(31) - 31 = 4(3480) - 49(31) \qquad (13)
\end{aligned}$$

From equation (13), we read of the private cryptographic key $K_D = (a, n)$ as $K_D = (-449, 3599)$.

Problem 4)

Prove that the *RSA Cryptosystem* is insecure against a chosen ciphertext attack. In particular, given a ciphertext y, describe how to choose a ciphertext $\hat{y} \neq y$, such that knowledge of the plaintext $\hat{x} = d_K(\hat{y})$ allows $x = d_K(y)$ to be computed.

Hint: Use the multiplicative property of the RSA Cryptosystem, i.e., that:

$$e_K(x_1)e_K(x_2) \bmod n = e_k(x_1 x_2) \bmod n$$

Solution:
Given a ciphertext y, encrypted using the *RSA Cryptosystem*, which has the following *multiplicative property*:

$$e_K(x_1)e_K(x_2) \pmod{n} = e_K(x_1x_2) \pmod{n} \tag{14}$$

an attacker can choose a ciphertext $\hat{y}$ as a multiplicative inverse of the original ciphertext y under modulo n as his chosen ciphertext:

$$y \cdot \hat{y} = e_K(x)e_K(\hat{x}) \pmod{n} = 1 \tag{15}$$

We note that such a multiplicative inverse exists if $\gcd(\hat{y}, n) = 1$. If, however, $\gcd(\hat{y}, n) \neq 1$, then the following cases are possible:

1. $\gcd(\hat{y}, n) = p$,
2. $\gcd(\hat{y}, n) = q$

Both cases are useful to an attacker as the knowledge of either p or q enables him/her to factor n, and hence to find the decryption (private) key (a, n). We therefore only consider the case when $\gcd(\hat{y}, n) = 1$, i.e., a multiplicative inverse of $\hat{y} \pmod{n}$ exist.
Using a multiplicative inverse of $y \pmod{n}$ as his/her chosen ciphertext, an attacker can write:

$$y \cdot \hat{y} = e_K(x) \cdot e_K(\hat{x}) \pmod{n} = e_K(x \cdot \hat{x} \pmod{n}) = 1 \tag{16}$$

Given the encryption rule of the *RSA Cryptosystem*: $e_K(x) = x^b \pmod{n}$, we can rewrite equation (16) as follows:

$$(x \cdot \hat{x})^b \equiv 1 \pmod{n} \tag{17}$$

Based on the fact that $1^b = 1 \pmod{n}$, and that $x \neq 0$, from equation (17) it follows:

$$x \cdot \hat{x} \equiv 1 \pmod{n} \tag{18}$$

Equation (18) represents a congruence equation modulo n. Since n is a product of two primes, $\gcd(\hat{x}, n) = \hat{x}$ or 1. In case when $\gcd(\hat{x}, n) = \hat{x}$, we know that $\hat{x} = \{p, q\}$, which again enables us to factor n and then to find x. In case when $\gcd(\hat{x}, n) = 1$, there exist a unique multiplicative inverse $\hat{x}^{-1} \pmod{n} = x$, which shows that *RSA Cryptosystem* is insecure against chosen ciphertext attack.

Problem 5

This exercise exhibits what is called a *protocol failure*. It provides an example where ciphertext can be decrypted by an opponent without determining the key, if a cryptosystem is used in a careless way. The moral is that it is not sufficient to use a "secure" cryptosystem in order to guarantee "secure" communication.

Suppose Bob has an *RSA Cryptosystem* with a large modulus n for which the factorization cannot be found in a reasonable amount of time. Suppose Alice sends a message to Bob by representing each alphabetic character as an integer between 0 and 25 (i.e., A $\leftrightarrow$ 0, B $\leftrightarrow$ 1, etc.), and then encrpyting each residue modulo 26 as a separate plaintext character.

(a) Describe how Eve can easily decrypt a message encrypted in this way.

(b) Illustrate this attack by decrypting the following ciphertext (which was encrypted using an *RSA Cryptosystem* with $n = 18721$ and $b = 25$) without factoring the modulus:

$$365, 0, 4845, 14930, 2608, 2608, 0. \tag{19}$$

Solution:
(a) If Alice, given a plaintext $x = x_1, x_2, x_2, \ldots, x_2$, takes each letter x_i, $1 \le i \le n$, converts it to an integer $z_i \in \mathbb{Z}_{26}$:

$$x \to z, z = z_1, z_2, z_2, \ldots, z_n, \;\; z_i \in \mathbb{Z}_{26}$$

and then encrypts every letter separately, using *RSA Cryptosystem*:

$$e_K(z_i) = z_i^b \pmod{n} = (x_i \pmod{26})^n \pmod{n}$$

she actually limits the plaintext space to $\mathbb{Z}_{26}$, cardinality of which is 26. She also limits the ciphertext space to $\mathbb{Z}_{26}$, i.e., the set of the same cardinality, since Bob, as a valid receiver, has to be able to uniquely decrypt every letter of the ciphertext.
Knowing the public key in this case is, however, sufficient for an attacker to compute a table, representing one-to-one correspondence between the plaintext and the ciphertext. Computed table enables him/her to decrypt any ciphertext, encrypted using *RSA Cryptosystem* in such a way.
(b) In order to decrypt the ciphertext $y = [365, 0, 4845, 14930, 2608, 2608, 0]$, we construct the decryption table 1. By inspection, we can read off letter by letter of the plaintext from the table: $d_K(365) = v$, $d_K(0) = a$, $d_K(4845) = n$, $d_K(14930) = i$, $d_K(2608) = l$. The plaintext is **vanilla**. The code that decrypts the given ciphertext is listed below.

Table 1: Decryption table

x	a	b	c	d	e	f	g	h	i	j	k	l	m
y	0	1	6400	18718	17173	1759	18242	12359	14930	9	6279	2608	4644
x	n	o	p	q	r	s	t	u	v	w	x	y	z
y	4845	1375	13444	16	13663	1437	2940	10334	365	10789	8945	11373	5116

```
function [decryption_table, plaintext] = RSA_decryption_table(b, n, ciphertext)
%RSA_decryption_table - function takes public

%INPUTS:
    %1. (b,n) - public key of the RSA cryptosystem
    %2. ciphertext - given ciphertext
%OUTPUT:
    %1. decryption_table - corresponding decryption table
    %2. plaintext - decrypted plaintext

%% Decryption table construction
for i = 0:1:25
    decryption_table(i + 1) = square_and_multiply(i, b, n);
end

%% Decrpytion
for i = 1:1:length(ciphertext)
    plaintext_aux(i) = find(decryption_table == ciphertext(i)) - 1;
end

plaintext = num2str(plaintext_aux);
```

Problem 6

Suppose that Alice and Bob communicate using *ElGamal Cryptosystem* and that, to save time, Bob uses the same number k each time he encrypts a plaintext message (i.e., k is a fixed secret of Bob, and it is not randomly generated each time encryption is performed). Show how an adversary who possesses a (plaintext, ciphertext) pair x, (Y_1, Y_2) can decrypt any other ciphertext (Y_1', Y_2').

Alice		Bob
Key Generation:		
Generate prime p, integer α		
Generate a,		
compute $\beta = \alpha^a \bmod p$		
$PK_A = (\alpha, \beta, p)$, $SK_A = a$		
Alice publishes $PK_A = (\alpha, \beta, p)$		**Bob knows:** $PK_A = (\alpha, \beta, p)$
		Encryption:
		Generate random integer k
		Compute ciphertext (Y_1, Y_2):
		$Y_1 = \alpha^k \bmod p$,
	$Y_1 = \alpha^k \bmod p$	$Y_2 = x\beta^k \bmod p$
	$Y_2 = x\beta^k \bmod p$	
Decryption:	$\longleftarrow$	
$x = Y_2(Y_1^a)^{-1} \bmod p$		

Figure 1: Schematic illustration of ElGamal key generation, encryption, and decryption.

Solution: A schematic illustration of ElGamal cryptosystem is given in Figure 1 above. If Bob reuses k for each encryption operation, then the ciphertext for some message x will be

$$\begin{aligned} Y_1 &= \alpha^k \bmod p \\ Y_2 &= x\beta^k \bmod p \end{aligned}$$

Given some (plaintext, ciphertext) pair $(x, (Y_1, Y_2))$, Eve can compute β^k as:

$$\beta^k = x^{-1} Y_2 \bmod p \tag{20}$$

Using the knowledge of β^k, for any given ciphertext (Y_1', Y_2'), the plaintext can be computed as:

$$x' = Y_2' \left(\beta^k\right)^{-1} \bmod p. \tag{21}$$

Thus knowing β^k is sufficient to allow us to decrypt any ElGamal-encrypted ciphertext without knowing the secret key, a.