CS 5770: Software Vulnerabilities and Security

Summer 2016

Lab 1 – Symmetric Key and Public Key Cryptosystems

Assigned: Thursday, June 9, 2016, Due: Sunday, June 19, 2016

Instructor: Tamara Bonaci
College of Computer and Information Science
Northeastern University – Seattle

Your lab reports are due through the course dropbox by 11:59pm on **Sunday, June 19, 2016**. When submitting your report, please include a written component, answering the posed questions and your source code. When appropriate, please include figures and tables to support your conclusions.

1 Background

1.1 The Shift Cipher

The **shift cipher** is one of the oldest known cryptosystems, often attributed to Julius Caesar. The idea used in this cryptosystem is to replace each letter in an alphabet by another letter at a distance K from it.

Formally, let's associate each letter $A, B, ..., Z$ with an integer $0, \ldots, 25$. If we allow the key K to be any integer with $0 \leq K \leq 25$, the *shift cipher* can be defined as:

$$\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}.$$
For $0 \leq K \leq 25$,

$$y = e_K(x) = (x + K) \bmod 26, \tag{1}$$
$$x = d_K(y) = (y - K) \bmod 26. \tag{2}$$

Example: Let $K = 3$ and let the plaintext be *shift*. Assume each letter is shifted right (or left) by 3 places. We then get *VKLIW* as the cipher for the right shift, ir *PEFCQ*, for the left shift.

1.2 The Affine Cipher

The idea of the **affine cipher** is to first **scale** and then shift, which is known as the **affine transformation**.

$$y = e_K(x) = (ax + b) \mod 26, \tag{3}$$
$$d_K(y) = a^{-1}(y - b) \mod 26. \tag{4}$$

In this scheme, the pair (a, b) denotes the cryptographic key K used for encryption/decryption. Here we need to know which pairs (a, b) are valid keys that yield an injective encryption function. Note that we need to know a^{-1} for decryption. Also if $a = 1$, the affine cipher becomes identical to the shift cipher.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Table 1. Mapping of alphabets to numerals

Example: Let $a = 9$ and $b = 3$. Let the plaintext be d that corresponds to the numerical value 3, based on Table 1.

$$e_K(d) = (9 \times 3 + 3) \bmod 26 = 4. \tag{5}$$

For the decryption part,

$$d_K(4) = a^{-1}(4 - b) \bmod 26 = 9^{-1}(4 - 3) \bmod 26 = 9^{-1} \pmod{26} = 3, \tag{6}$$

which is the multiplicative inverse of 9 (mod 26), i.e. $9 \times 3 \equiv 1 \pmod{26}$.

1.3 RSA Cryptosystem

Let's recall that the security of RSA is based on the **difficulty of factoring products of large prime numbers**. This cryptosystem consists of three basic components, namely, **key generation**, **encryption**, and **decryption**, as described below:

Key Generation RSA key generation consists of the following steps:

1. Generation of two distinct large prime numbers p and q (an integer p is prime if its only factors are 1 and p).
2. Computation of $n = pq$ and the *Euler totient function* $\phi(n) = (p - 1)(q - 1)$.
3. Picking a random integer e satisfying $1 < e \leq \phi(n)$, $\gcd(e, n) = 1$, and $\gcd(e, \phi(n)) = 1$[1]
4. Choosing d satisfying $ed \equiv 1 \bmod \phi(n)$.
5. The **public key** is defined by $PK = (e, n)$ and the **private key** is defined by $SK = (d, n)$.

Each communicating party publishes his/her public key $PK = (e, n)$ to the world, and keeps his/her secret key $SK = (d, n)$ private. Anyone can use PK to encrypt, while only the person who possesses the secret key SK can decrypt.

Encryption After Alice's public key PK_A has been generated and published, any user Bob can encrypt messages to send to Alice as follows. The set of possible plaintext messages is equal to the set of integers modulo n, $\mathbb{Z}_n$. For a plaintext x, the ciphertext is given by

$$Y = x^e \pmod{n}.$$

Decryption Upon receiving an encrypted message from Bob, Alice decrypts by computing

$$x = Y^d \pmod{n}.$$

[1] Here, $\gcd(e, n)$ denotes the greatest common divisor of e and n, i.e., the largest integer than divides both e and n. $\gcd(e, n) = 1$ implies that e and n do not share any common factors.

2 Assignments

Problem 1 The following ciphertext was encrypted by a *shift cipher*:

`ycvejqwvhqtdtwvwu`

Please decrypt.

Problem 2 The following ciphertext was encrypted by an *affine cipher*:

`edsgickxhuklzveqzvkxwkzukcvuh`

The first two letter of the plaintext are if. Please decrypt.

Problem 3 In this problem you are to get your hands dirty doing some programming. Let's assume we have a new, restricted alphabet A, C, G, T. For example, this alphabet could correspond to the four nucleotides adenine, cytosine, guanine and thymine, which are the basic building blocks of DNA and RNA codes. Associate the letters A, C, G, T with the numbers 0, 1, 2, 3, respectively.

(a) Using the *shift cipher* with a shift of 1, encrypt the following sequence of nucleotides which is taken from the beginning of the thirteenth human chromosome:

`GAATTCGCGGCCGCAATTAACCCTCACTAAAGGGATCT`

(b) Write a program that performs *affine cipher* encryption on the nucleotide alphabet. What restrictions are there on the affine cipher?

Problem 4 In an RSA cryptosystem, suppose you know n = 718548065973745507, e = 3449 and d = 543546506135745129. Factor n.