Questions

1. Callie wants to send the message M = 13 to Alice. Using Alice's public and private keys, calculate the ciphertext C, and the value for R when Alice recovers the message.

Solution :

1. Callie encrypts message M = 13:

• Alice's public key is (n, e) = (33, 3).

$C=M^e \bmod n = (13)^3 \bmod 33 = 2197 \bmod 33 = 19$

C = 19.

• Callie sends to Alice ciphertext C = 19.

Alice receives and decrypts ciphertext C = 19:

• Alice uses her private key (n, d) = (33, 7).

• When $M = 19^7 \bmod 33$

How to calculate 19^7

$(19)^1 = 19 \bmod 33 = 19$

$(19)^2 = 19^2 \bmod 33 = 361 \bmod 33 = 31$

$(19)^4 = 31^2 \bmod 33 = 961 \bmod 33 = 4$

$(19)^8 = 4^2 \bmod 33 = 16 \bmod 33 = 16$

BINARY OF 7 = 0111

$19^7 \bmod 33 = 19 \text{ X } 31 \text{ X } 4 \text{ MOD } 33$

$= 2356 \text{ MOD } 33 = 13$

the remainder is R = 13.

• R = 13 = M, the original message from Callie!

Example 2 : Dexter wants to set up his own public and private keys.

He chooses p = 23 and q = 19 with e = 283.Find d so that ed has a remainder of 1 when divided by (p − 1)(q − 1).

Solution :

With p = 23, q = 19, we have m = (p − 1)(q − 1) = 22 x 18 = 396.

We want to find d so that ed = 283d has a remainder of 1 when divided by m = 396.

One way to do this is by simple trial and error, increasing the value of d until 283d divided by 396 leaves a remainder of 1.

d	$283d$	Remainder when $283d$ is divided by 396
1	283	283
2	566	170
3	849	57
4	1132	340
5	1415	227
6	1698	114
7	1981	1

We see that d = 7 works; that is ed = 283 × 7 = 1981 leaves a remainder of 1 when divided by 396.

In general, trial and error could take a very long time, as the value of d could be a big number. Instead, an ancient technique called Euclid's Algorithm can be used to find d in the linear Diophantine equation 283d + 396y = 1.

Example 3 :

- Choose p = 3 and q = 11
- Compute n = p * q = 3 * 11 = 33
- Compute φ(n) = (p - 1) * (q - 1) = 2 * 10 = 20
- Choose e such that 1 < e < φ(n) and e and n are coprime. Let e = 7
- Compute a value for d such that (d * e) % φ(n) = 1.

- One solution is d = 3 [(3 * 7) % 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of $m = 2$ is $c = 2^7 \% 33 = 29$
- The decryption of $c = 29$ is $m = 29^3 \% 33 = 2$

Example 4 :

- p = 11, q = 7, n = 77, $\Phi(n) = 60$
- d = 13, e = 37 (ed = 481; ed mod 60 = 1)
- Let M = 15. Then $C \equiv M^e \bmod n$
  - $C \equiv 15^{37} \pmod{77} = 71$
- $M \equiv C^d \bmod n$
  - $M \equiv 71^{13} \pmod{77} = 15$

Example 5 :

1. Let p = 7 and q = 13 be the two primes.

2. n = pq = 91 and $\phi = (p - 1)(q - 1) = 72$.

3. Choose e. Let's look among the primes.

• Try e = 2. gcd(2,72) = 2 (does not work)

• Try e = 3. gcd(3,72) = 3 (does not work)

• Try e = 5. gcd(5,72) = 1 (it works)

We choose e = 5.

$$d = 29.$$

In general, we use $d = x \bmod \phi$.

5. The encryption function is

$$E(M) = M^e \bmod n = M^5 \bmod 91.$$

The decryption function is

$$D(M) = M^d \bmod n = M^{29} \bmod 91.$$

6. Suppose the message is $M = 10$.

$$E(M) = E(10) = 10^5 \bmod 91 = 82$$

$$D(E(M)) = D(82) = 82^{29} \bmod 91 = 10$$

7. Let's see how to compute efficiently $82^{29} \bmod 91$ using the square-and-multiply algorithm.

$$\begin{aligned}
(82)^1 &\equiv 82 \pmod{91} \\
(82)^2 &\equiv 81 \pmod{91} \\
(82)^4 &\equiv (81)^2 \equiv 9 \pmod{91} \\
(82)^8 &\equiv (9)^2 \equiv 81 \pmod{91} \\
(82)^{16} &\equiv (81)^2 \equiv 9 \pmod{91}
\end{aligned}$$

Since $29 = 16 + 8 + 4 + 1$ (in binary 29 is 11101), we deduce that

$$\begin{aligned}
82^{29} &\equiv (82)^{16}(82)^8(82)^4(82)^1 \pmod{91} \\
&\equiv (9)(81)(9)(82) \pmod{91} \\
&\equiv 10 \pmod{91}
\end{aligned}$$

We conclude that $82^{29} \bmod 91 = 10$.

Example :

Creating a Public and Private Key

- Randomly choose two LARGE distinct primes p and q.
- Let $m=pq$
- Calculate $\phi(m)=(p-1)(q-1)$
- Choose an integer e such that $(e,\phi(m))=1$
- Find an integer d which satisfies $ed \equiv 1 \bmod \phi(m)$

Your public key will be the ordered pair (e,m),
while your private key is the ordered pair (d,m).

In order to avoid confusion with the Greatest Common Divisor function we will explicitly state when $(\#,\#)$ is a public/private key. It is important that the only information made public is your public key. This means p, q, $\phi(m)$ and d must be kept secret because with any of this information, one could break the code.

Encryption

Set Up

To begin, we must associate each letter of the alphabet with a unique number. This will allow us to convert our message into a series of numbers which we can then perform operations on. Let us use the following table for this,

Letter	Number	Letter	Number
A	00	N	13
B	01	O	14
C	02	P	15
D	03	Q	16
E	04	R	17
F	05	S	18
G	06	T	19
H	07	U	20
I	08	V	21
J	09	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25
		"__"	26

Note that instead of letting A=0, we set it equal to 00. This is because once we get up to K we start using double digits. If we have a mix of single digits and double digits it would be impossible to convert back to our original message. Also, it is useful to denote spaces in between words with a number. We will use an underscore between words instead of space to make it clearer.

Lets look at an example, say our plaintext is

message: NUMBER_THEORY

N	U	M	B	E	R	_	T	H	E	O	R	Y
13	20	12	01	04	17	26	19	07	04	14	17	24

So we get,

plaintext: 13201201041726190704141724

The Encryption Algorithm

- Convert plaintext into its numerical equivalent, as shown previously
- Recall the public key you are using is (e,m)
- Separate these numbers into blocks, which are viewed as a single number, and label them P_1, P_2, ..., P_k$, making sure that for each $1 \leq i \leq k$ that $P_i < m$.
- For each $1 \leq i \leq k$, compute $(P_i)_e \bmod m$ and label the result C_i
- The resulting blocks $C_1, C_2, ..., C_k$ are your new cipher text.

You can now transmit $C_1, C_2, ..., C_k$

Example of the Encryption Algorithm

Lets say our old friend Chloe chose $p=31$, and $q=37$ for her prime numbers,

which gives $m=(31)(37)=1147$.

$\phi(m)=\phi(1147)=(31-1)(37-1)=1080.$

Then she must find an integer e which is relatively prime to 1080. She randomly selects $e=17$, and note that $(17,1080)=1$. So Chloe publishes her public key of (17,1147).

Say Jack wants to send Chloe an encrypted message using RSA. All he knows is Chloe's public key (17,1147), so he uses this in the encryption algorithm.

Jack's secret message to Chloe states,

message: WE_LOVE_MATH

After converting to numericals using the table above Jack has,

plaintext : 22042611142104261200 1907

He breaks the numerical form of the message into blocks of 3 making sure each block is less than m.

plaintext in blocks: 220 426 111 421 042 612 001 907

So,

(1)

$$220=P_1$$
$$426=P_2$$
$$111=P_3$$
$$421=P_4$$
$$042=P_5$$
$$612=P_5$$
$$001=P_7$$
$$907=P_8$$

He then computes for each P_i, from $i=1$ to $i=8$, $P_{ei} \bmod m$, with e=17, and m=1147 taken from Chloe's public key. And the result is the ciphertext

(2)

$$(220)_{17} \bmod 1147$$

$$(426)_{17} \bmod 1147$$

$$(111)_{17} \bmod 1147$$

$$(421)_{17} \bmod 1147$$

$$(042)_{17} \bmod 1147$$

$$(612)_{17} \bmod 1147$$

$$(001)_{17} \bmod 1147$$

$$(907)_{17} \bmod 1147$$

$$\equiv 611 \equiv 1145 \equiv 851 \equiv 510 \equiv 96 \equiv 246 \equiv 1 \equiv 405$$

So his ciphertext becomes,

ciphertext: 611 1145 851 510 96 246 1 405

And Jack sends "611 1145 851 510 96 246 1 405" to Chloe.

The Decryption Algorithm

- Let $C_1, C_2, ..., C_k$ be your ciphertext blocks, and (d, m) be your private key
- For each $1 \leq i \leq k$, compute $(C_i)^d \bmod m$ and the result will be P_i where $0 \leq P_i < m$
- Then, $P_1, P_2, ..., P_k$ is your plaintext in numerical form
- Finally, convert the pairs of two-digit numbers back to its alphabetical equivalent using our table.

Your message should now be comprehensible.

Example of the Decryption Algorithm

Again, returning to our Jack and Chloe example, Jack was sending a message to Chloe using her public key. Now for Chloe to decrypt a message sent to her, she must use her private key. We will re-use Chloe's key information in this example.

- Chloe (and ONLY Chloe) knows m=1147=(31)(37), so $\phi(1147) = 1080$
- Recall she uses e=17
- The next step is solving for d, which is the multiplicative inverse of $e \bmod 1080$

So **Chloe** knows her **private key is (953, 1147)**.

Now lets see how Chloe uses this info to decrypt a message. Say for example, Jack uses Chloe's public key and sends her another message using a block size of 3. This is what Chloe recieves,

ciphertext: 1 41 203 744 472 947 423 968 718

Chloe has all the information she needs to go through the decryption process to see what Jack sent her. Taking each ciphertext block $C_1 = 1, C_2 = 41, ..., C_9 = 560$, and knowing d=953, and m=1147, she computes $C_{953i} \bmod 1147$. The resulting integers are the plaintext $P_1, P_2, ..., P_9$. She gets,

(6)

$$1_{953} 41_{953} 203_{953} 744_{953} 472_{953} 947_{953} 423_{953} 968_{953} 718_{953} \bmod 1147 \equiv 1 = P_1 \bmod 1147 \equiv 324 = P_2 \bmod 1147 \equiv 261 = P_3 \bmod 1147 \equiv 620 = P_4 \bmod 1147 \equiv 41 = P_5 \bmod 1147 \equiv 819 = P_6 \bmod 1147 \equiv 81 = P_7 \bmod 1147 \equiv 413 = P_8 \bmod 1147 \equiv 180 = P_9$$

resulting #'s: 1 324 261 620 41 819 81 413 180

So the resulting numbers should be the plaintext in number form. Since Chloe knows that Jack used a block size of 3, she knows to add extra zeros to P_1=1, P_5=41, and P_7=81. So P_1 becomes 001, P_5 becomes 041, and P_7 becomes 081. So Chloe has the plaintext:

plaintext: 001 324 261 620 041 819 081 413 180

Finally she regroups into pairs of two so she can covert each pair back into its alphabetical equivalent.

00	13	24	26	16	20	04	18	19	08	14	13	18	0
A	N	Y	_	Q	U	E	S	T	I	O	N	S	

Why is there a loner "0" at the end of out message, and how do we convert it back to alphabetical form? Since Chloe knows there is no letter of the alphabet paired up with the single digit "0" in their conversion table, she concludes this extra "'0" must be a **filler space**. Jack simply added a meaningless bit of information to his plaintext in order to divide the plaintext evenly into groups of three. It is important to see here that the original message did not divide evenly into groups of three. If we try and do this, we will be left with the last block of "18", which clearly is not 3 digits as specified. So a filler space is needed, and "0" was randomly chosen to fill in. So finally, we have the message Jack sent to Chloe is,

message: ANY_QUESTIONS