UNIVERSITY OF BIRMINGHAM

M.Sc. Computer Science
Computer Systems
Network Security – Additional Exercises [Solutions]

Question #1: Consider a 16-block cipher. How many possible input blocks does this cipher have? How many possible mappings are there? If we view each mapping as a key, then how many possible keys does this cipher have?

First of all, lets talk about permutations: *The number of permutations of n distinct objects is n factorial, usually written as n!, which means the product of all positive integers less than or equal to n.* For example: The set **{1,2,3}** has three values, and it will have 3! = 6 possible permutations i.e. (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), and (3,2,1). More at: https://en.wikipedia.org/wiki/Permutation

Now lets consider that we have a 3-block cipher (in simple terms, having 3-bit inputs), so we can have **2^3** different possible inputs i.e. 000, 001, 010, 011, 100, 101, 110, 111.

Lets consider a few mappings from input message (3-bit number) to ciphertext. In simple terms, given an input (of 3 bits) what ciphertext we can produce for it (also in 3-bits).

Mapping #1 **Message -> Ciphertext**	**Mapping #2** **Message -> Ciphertext**	...	**Mapping #K** **Message -> Ciphertext**
000 -> 001 001 -> 010 010 -> 011 011 -> 100 100 -> 101 101 -> 110 110 -> 111 111 -> 000	000 -> 010 001 -> 011 010 -> 100 011 -> 101 100 -> 110 101 -> 111 110 -> 000 111 -> 001	...	000 -> 100 001 -> 111 010 -> 011 011 -> 010 100 -> 001 101 -> 110 110 -> 000 111 -> 101

Do you see, in how many different ways we can map the given inputs to ciphertext? In other words, the output is just a re-arrangement of input numbers i.e. permutations.

Essentially, as we have 8 possible inputs, we can have **8!** different permutations in this case, which is the same as **2^3!** (where 3 is the input block size). For each mapping we will have 1 key that will generate that mapping from the given inputs, so will have **2^3!** keys.

Now, coming back to the original question, if we have a **16-block cipher** (16-bit inputs) then we will have **2^16** possible input blocks (that's the number of possible binary values in 16 bits). Each mapping from the input to ciphertext will be a permutation of the **2^16** input blocks; so there are **2^16!** possible mappings; so there are **2^16!** possible keys.

UNIVERSITY OF BIRMINGHAM

Question #2: Suppose n = 1000, a = 1017, and b = 1006. Use an identity of modular arithmetic to calculate in your head (a • b) mod n.

We can use the following identity: [(a mod n) * (b mod n)] mod n = (a*b) mod n

Therefore, a mod n = 1017 mod 1000 = 17, b mod n = 1006 mod 1000 = 6
So [(a mod n) * (b mod n)] mod n = [17 * 6] mod 1000 = 102 mod 1000 = 102

Question #3: Show that Trudy's known-plaintext attack, in which she knows the (ciphertext, plaintext) translation pairs for seven letters, reduces the number of possible substitutions to be checked by approximately 10^9.

If Trudy knew that the words "bob" and "alice" appeared in the text, then she would know the ciphertext for b, o, a, l, i, c, e (since "bob" is the only palindrome in the message, and "alice" is the only 5-letter word.
If Trudy knows the ciphertext for 7 of the letters, then she only needs to try 19!, rather than 26!, plaintext-ciphertext pairs. The difference between 19! and 26! is 26*25*24...*20, which is 3315312000, or approximately 10^9.

Question #4: a) Using RSA, choose p = 3 and q = 11, and encode the word "dog" by encrypting each letter separately. Apply the decryption algorithm to the encrypted version to recover the original plaintext message. (b) Repeat part (a) but now encrypt "dog" as one message m.

We are given p = 3 and q = 11 . We thus have n = 33 and q = 11 . Choose e = 9 (which is a good value to choose, since the resulting calculations are less likely to run into numerical stability problems than other choices for e) since 3 and (p − 1) * (q − 1) = 20 have no common factors.
Choose d = 9 also so that e * d = 81 and thus e * d − 1 = 80 is exactly divisible by 20. We can now perform the RSA encryption and decryption using n = 33 , e = 9 and d = 9 .

Letter	m	m^e	Ciphertext = m^e mod n
d	4	4^9 = 262144	= 4^9 mod 33 => 25
o	15	15^9 = 38443359375	= 15^9 mod 33 => 3
g	7	7^9 = 40353607	= 7^9 mod 33 => 19

We can decrypt the above message, as shown below:

Ciphertext	c^d	m = c^d mod n	Letter
25	25^9 = 38146972265625	= 25^9 mod 33 => 4	d
3	3^9 = 19683	= 3^9 mod 33 => 15	o
19	19^9 = 322687697779	= 19^9 mod 33 => 7	g

UNIVERSITY OF BIRMINGHAM

In order to encrypt the “dog” as one message, we will first consider each letter as a 5-bit number: 00100, 01111, 00111 (as there are 26 letters in English alphabet and we need a minimum of 5 bits for each letter).

Now, we concatenate each letter to get 001000111100111 and encrypt the resulting decimal number m=4583. The concatenated decimal number m (= 4583) is larger than current n (= 33). We need m < n. So we use p = 43, q = 107, n = p*q = 4601, z = (p-1)(q-1) = 4452, e = 61, d = 73.

Ciphertext $c = m^e \bmod n = 4583^{61} \bmod 4601$

$4583^{61} =$
213865776018280578040896021565305671886114998690297887338084388043028645956206139567258407209497648456409561187848752467850332361977771297302589617569184002920486328061975277854477915672551018944928209725081857698028817189 83

Ciphertext $c = 4583^{61} \bmod 4601 = 402$

For decryption, we can use the following formula:

Plaintext $m = c^d \bmod n = 402^{73} \bmod 4601$

$402^{73} =$
128381331361977163419571213253979328764353314748253620932840526279302715886101239205328724963357096749312228022145381501293424137054020458145987149793872321410147032277945864998179456333905 92

Plaintext $m = 402^{73} \bmod 4601 = 4583$