Solutions to HW10

EE 450

Dr. Walker

R8.2)

Any different entities in a given network might need encryption for their communication. A few example are:

1) A laptop and a web server
2) Two routers
3) Two DNS name servers

R8.10)

No. This is because a hash function is a one-way function. That is, given any hash value, the original message cannot be recovered (given h such that h=H(m), one cannot recover m from h). In fact what a hash function guarantees is to generate the same key when it see a file for the second time. You may also note that the amount of information that the amount of information that can fit in a hash generated code is much less that in even a small file.

P8.3)

Every letter in the alphabet appears in the phrase "The quick fox jumps over the lazy brown dog." Given this phrase in a chosen plaintext attack (where the attacker has both the plain text, and the ciphertext), the Caesar cipher would be broken - the intruder would know the ciphertext character for every plaintext character. However, the Vigenere cipher does not alway translate a given plaintext character to the same ciphertext character each time, and hence a Vigenere cipher would not be immediately broken by this chosen plaintext attack.

P8.7)

a) We are given $p = 3$ and $q = 11$ We thus have $n = 3$. Choose $e = 9$ (9 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 = $m^e \ mod \ 33$
d	4	262144	25
o	15	38443359375	3
g	7	40353607	19

ciphertext c	c^d	$m = c^d \ mod \ 33$	letter
25	38146972265625	4	d
3	19683	15	o
19	322687697779	7	g

b)

To do that we first consider each letter as a 5-bit number: 00100, 01111, 00111. 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 and we will have:

ciphertext = $m^e mod \ 4601$

m^e= 213865776018280578040896021565305671886114998690297887338084388043028645956206139567258407209497648456409561187848752467850332361977771297302589617569184002920486328061975277854477915672551018944928209725081857698028817189 83

ciphertext = $m^e \ mod \ 4601 \ = \ 402$

c^d= 128381331361977163419571213253979328764353314748253620932840526279302715886101239205328724963357096749312228022145381501293424137054020458145987149793872321410147032277945864998179456333905 92

message = $c^d \ mod \ 4601 \ = \ 4583$

P8.8)

p = 5, q = 11

a) n = pq = 55, z = (p-1)(q-1) = 40
b) e = 3 is less than n and has no common factors with z.
c) d = 27
d) m = 8, m^e = 512, Ciphertext c= m^e mod n = 17

P8.23)

If Trudy does not bother to change the sequence number, R2 will detect the duplicate when checking the sequence number in the ESP header. If Trudy increments the sequence number, the packet will fail the integrity check at R2.

P8.24)

a) Since IV = 11, the key stream is 111110100000 …….

Given, m = 10100000
Hence, ICV = 1010 XOR 0000 = 1010
The three fields will be:

IV: 11
Encrypted message: 10100000 XOR 11111010 = 01011010
Encrypted ICV: 1010 XOR 0000 = 1010

b) The receiver extracts the IV (11) and generates the key stream 111110100000 …….

XORs the encrypted message with the key stream to recover the original message:
01011010 XOR 11111010 = 10100000

XORs the encrypted ICV with the keystream to recover the original ICV:

1010 XOR 0000 = 1010

The receiver then XORs the first 4 bits of recovered message with its last 4 bits:
1010 XOR 0000 = 1010 (which equals the recovered ICV)

c) Since the ICV is calculated as the XOR of first 4 bits of message with last 4 bits of message, either the 1st bit or the 5th bit of the message has to be flipped for the received packet to pass the ICV check.

d) For part (a), the encrypted message was 01011010
Flipping the 1st bit gives, 11011010
Trudy XORs this message with the keystream:
11011010 XOR 11111010 = 00100000

If Trudy flipped the first bit of the encrypted ICV, the ICV value received by the receiver is 0010
The receiver XORs this value with the keystream to get the ICV:
0010 XOR 0000 = 0010
The receiver now calculates the ICV from the recovered message:
0010 XOR 0000 = 0010 (which equals the recovered ICV and so the received packet passes the ICV check)