Practical Cryptography Systems
Weekly Assignment 1
Ujjawal Sharma

1. **What is the key space (total number of possible key settings) of a four-rotor Naval Enigma (M4)?**

Answer:

Key Space of a 4 rotor Naval Enigma (M4) depends on following factors:

(i) Reflector Selection:

There are 2 possible reflectors, "Thin B" or "Thin C".

(ii) Rotor Selection:

There were two possible rotors for the thin left-hand position ("Beta and Gamma"). The three other rotors where chosen, in any order, from a possible eight supplied by the machine (numbered from I to VIII). This gives a total of 2*8*7*6 = 672 positions.

(iii) Ring Settings:

Each of the four rotors could have its outer ring rotated through 26 positions to change the alignment of the internal wiring. This gives, 26*26*26*26=456976 possibilities.

(iv) Plug Board Connections:

For the 13 jumpers, we can have,

j	C(26,2j)	(2j-1)!!	Result
0	1	1	1
1	325	1	325
2	14,950	3	44,850
3	230,230	15	3,453,450
4	1,562,275	105	164,038,875
5	5,311,735	945	5,019,589,575
6	9,657,700	10,395	100,391,791,500
7	9,657,700	135,135	1,305,093,289,500
8	5,311,735	2,027,025	10,767,019,638,375
9	1,562,275	34,459,425	53,835,098,191,875
10	230,230	654,729,075	150,738,274,937,250
11	14,950	13,749,310,575	205,552,193,096,250
12	325	316,234,143,225	102,776,096,548,125
13	1	7,905,853,580,625	7,905,853,580,625
Total possibilities			532,985,208,200,576

(v) Indicator Settings:

Only the two right hand rotors contribute to the key length, giving 26*26 possibilities.

Effective Key Length is calculated by multiplying all the above factors as they all have been calculated independently.

So, total number of possible keys:

$$2 * 672 * 456{,}976 * 532{,}985{,}208{,}200{,}576 * 676 = 221{,}286{,}292{,}668{,}406{,}558{,}235{,}295{,}744$$

2. The Lucky13 attack relies on changing the size of the input to HMAC so that it crosses a 64-byte boundary. Why?

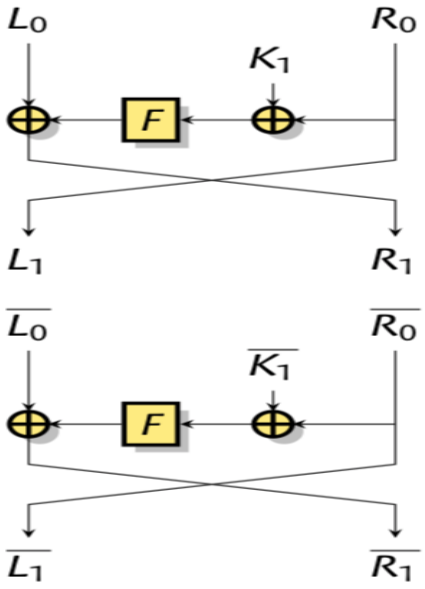
Answer: TLS uses HMAC with either MD5, SHA1 or SHA256 as the hash function. All these processes the message taking 64 byte messages at a time. So, here the hashing time will become the function of number of blocks and not on the number of bytes the message contains. If we will increase the length of the message to reach 65 bytes (outside the boundary), the HMAC process will require another block to process it, thus, spending some extra time. The Lucky13 attack takes a message greater than 55 bytes (including the TLS padding). The same message with padding removed, will fall below the limit.
When the attacker will try to play with the message, the decryption process will take a longer time. By repeating the process a large number of times, it is possible to prove the success of decryption. After this, the attacker can then launch a padding oracle attack.

3. Describe the complementation property of the DES cipher.

Answer: The Complementation property of DES cipher explains that if all the key bits are flipped, then all the sub keys are flipped as well. Also, if we flip the bits of the plaintext, the computing functions will receive an original input, and the cipher text is also negated, i.e.,

$$DES_{\overline{K}}(\overline{P}) = \overline{(DES_K(P))}.$$

The DES's Complimentary property can be demonstrated by the following figure:



Here, L_0, L_1 …. L_{16} are the left part of the input text and R_0,R_1 … R_{16} will be the right part of the input text. The above scheme runs 16 rounds (not shown in the figure) . We can also conclude that for any stage of the Fiestel Network :

$$L_i' = c(L) \text{ and } R_i' = c(R)$$

4. **What is the danger of using a two-time pad. More concretely, what can happen if I produce two equal-length cipher-text $C_1 = M_1 + K$ and $C_2 = M_2 + K$ where M_1, M_2 are different messages but K is identical for both cipher texts. Why is this bad?**

Answer: The danger of two messages encrypted with a same key is that, the below equation will hold true:

$$\mathbf{C_1 \oplus C_2 = (K \oplus M_1) \oplus (K \oplus M_2) = M_1 \oplus M_2}$$

This will mean that, under a chosen plaintext attack, if the attacker want to know a message which is currently in an encrypted form, he will create a message of his own and will send it to the encryption oracle, upon getting the cipher text of his own message he can xor it with the other cipher and can get the message out of it. So, two-time pad is an unsecure implementation.

5. **The KRACK paper relied on forcing nonce re-use for an encryption scheme. Explain the possible implications of this for security. How does this relate to your previous answer?**

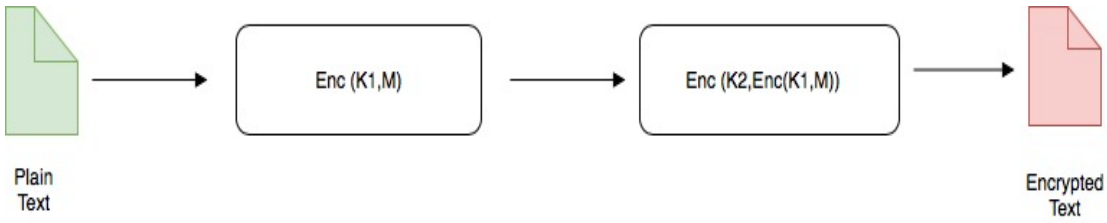
Answer: By forcing nonce reuse, we can compromise the confidentiality of WPA2 system. We can perform packet replays; we can decrypt and forge the traffic. Using this we can also attack the group key, Peer Key and fast BSS transition handshake. The KRACK allows a malicious user to decrypt a TCP packet, learn the sequence number and hijack the TCP stream and inject malicious code. The attacker can also replay broadcast and multicast frames in this attack. If this replay gets executed for a longer duration, the victim can get stuck at some time forever.

This situation is exactly the same as was discussed in the previous question. If messages are getting encrypted using same key, then deciphering them becomes easy as by just initiating a chosen plaintext attack. By reinstalling the same key, it's encrypted his own message with the key, which was used for encrypting someone else's message too. This helps him decrypt other user's conversations.

Long Answer:

Double Decipherment:

The Double Decipherment explained in the question is also termed as Double DES in many places, and is written as shown below:



Just by looking at the equation, we can say that with the addition of another layer of Encryption, that too with a different key, the overall security of the system would have increased. The security definitely increases, but not as much as it was expected to be.

Double DES has a 112-bit key and enciphers blocks of 64 bits. The security of DES depends largely on the size of key space. Although the key is 64 bits long, it only has 56 bits effective, other 8 bits are parity bits. So, we can say that the key space is 2^{56}. We may say that by applying double encryption, the size of key space should double. But, this does not happen. To demonstrate this, lets discuss the Meet-in-the-middle attack.

Meet in the Middle Attack:
We can firstly select any message "M" and can send it to the DES encryption engine, which will encrypt it with the key k_1 to produce a cipher text C. We can encrypt the message M with all the 2^{56} possible keys and will store the result (We have a huge disk with us!!!).

$$M \rightarrow E(k_1, M) \rightarrow E(k_2, E(k_1, M)) = C$$

The stored result is a set of all possible encryptions. Now, we will decrypt the cipher text with using all 2^{56} possible keys.

$$D(k_2, C) = D(k_2, E(k_2, E(k_1, M))) \rightarrow E(k_1, M)$$

After decrypting with each key, we will find the matches with the 2^{56} possible encryptions. After we have found few matches, we could try each possible pair of keys. If more than one plaintext/cipher text correspondence is known (for the key pair), then other correspondences could be used to check which of the keys is correct.

So, effectively it only takes $2^* 2^{56} = 2^{57}$ to break the security of a Super DES, which is not much than the DES which has requires 2^{56} tries to break it.