Crypto Assignment 2

Name: CHE Yulin, Student#: 20292673

Q1 CFB vs CBC

- **Answer**: Actually, they are different in the general sense, if the encryption function and decryption are not exclusive or of keys' t bits. The specific deduction of them are elaborated as follows.
- **Cipher Feedback Mode**: The default initial value is denoted as c_0, the specific cipher is as follows. And it is easy to find that here the decryption function is the same as the encryption function.
  $c_0 = IV$, $c_i = m_i \oplus E_k(c_{i-1})$, $m_i = c_i \oplus E_k(c_{i-1})$
- **Cipher Block Chain Mode**: The default initial value is denoted as c_0, the specific cipher is as follows.
  $c_0 = IV$, $c_i = E_k(m_i \oplus c_{i-1})$, $m_i = D_k(c_i) \oplus c_{i-1}$
- **Conclusion**: from the above cipher detail, it is apparent that these two cipher modes are quite different from each other.

Q2 Key Distribution Protocol Design

- **Answer for protocol design**: the key distribution protocol could be summarized in seven steps, corresponding to seven actions, either for Alice or for Bob. Assume the private key for Alice is k_1, and the private key for Bob is k_2 . The session key is denoted as $k_{session}$

  1. Alice: create the session key, and then encrypt that with E_{k1}, yields the result, $E_{k1}(k_{session})$
  2. Alice: send the result $E_{k1}(k_{session})$ in step 1 to Bob
  3. Bob: encrypt $E_{k1}(k_{session})$ with E_{k2}, yields the result, $E_{k2}(E_{k1}(k_{session}))$, and from the property given in the question, the result is equals to $E_{k1}(E_{k2}(k_{session}))$
  4. Bob: send the result $E_{k1}(E_{k2}(k_{session}))$ in step 3 to Alice
  5. Alice: decrypt the result $E_{k1}(E_{k2}(k_{session}))$ with D_{k1}, and Alice gets $E_{k2}(k_{session})$
  6. Alice: send $E_{k2}(k_{session})$ in step 5 to Bob
  7. Bob: decrypt $E_{k2}(k_{session})$ with D_{k2}, and Bob gets $k_{session}$

- **Answer for the proof of three propertites** the first statement is trivial since there are only communications between Alice and Bob without any others interrupting. The second statement is also trivial since $k_{session}$ generated in step 1 is sent to Bob in step 7. The third security property is satisfied, since the communicated messages $E_{k1}(k_{session})$ in step 2, $E_{k1}(E_{k2}(k_{session}))$ in step 4, $E_{k2}(k_{session})$ in step 6 could not be cracked with the two theorems hold. which are it is computationally infeasible to determine the key k given any message m and its ciphertext $E_k(m)$.

Q3 RSA-Like Encryption Scheme

- **Answer a, for explaining how this scheme works** we need to prove the correctness of decryption here, the deduction is as follows. we could either use $M = C^{P'} \mod Q$ or $M = C^{Q'} \mod P$ to decrypt the message, the following considers the proof for the first one, since they are quite similar in deduction phase.

$$
\begin{aligned}
&(M^{PQ} \mod PQ)^{P'} modQ \\
&= M^{PP'Q} \mod Q \\
&= M^{(u(Q-1)+1)\cdot Q} \mod Q \\
&= M^{[u(Q-1)+1]\cdot(Q-1)+u(Q-1)+1} \mod Q \\
&= [(M^{Q-1})^{u(Q-1)+1} modQ] \cdot [(M^{Q-1})^{u} \mod Q] \cdot [M \mod Q] \\
&= M \mod Q \\
&= M
\end{aligned}
$$

- **Answer b, how does it differ from RSA**: 1) there are two private key pairs here, (P', Q), (Q', P), which could be found in step 5, 2) the public key encryption function is different, since in step 4, $C = M^N \mod N$.
- **Answer c, is there any particular advantage of this scheme over RSA** the advantage I found here is that we have two private key pairs here, we could use both to check the validity of the private key pairs.

Q4 ElGamal-Related

- **Answer**: of course, yes. the random number X_B varies with time, which is elaborated as follows. It is quite obvious that X_B introduces the randomness in the session key.

$$
\begin{aligned}
&K = (Y_A)^{X_B} \mod q \\
&C_1 = \alpha^{X_B} \mod q \\
&C_2 = K \cdot M \mod q \\
&M = (C_2 \cdot K^{-1}) \mod q
\end{aligned}
$$

Q5 RSA-Parameter-Choice

- **Answer**: This question is only relevant if you choose p, q in a non-standard way. The standard way to choose
  is to choose them as two independent random $k/2$ bit numbers. If you do it the standard way, the question is not relevant (the probability that $|p-q|$ is too small is negligible – and is dominated by the chances of other kinds of failures).
- This question would be relevant if you were choosing p, q in some funny way that had an unusually high probability of making
  $|p-q|$ be unusually small. Yes, you can quantify how much easier this makes factoring. For instance, the Fermat factoring method works as follows: for $\lceil \sqrt{n} \rceil$, $\lceil \sqrt{n} + 1 \rceil$, $\ldots$.
- So it is quite easy to factorize the $p \cdot q$ when p, q are too close to each other.