Solver: Phan Van Dan

Email Address: phan0041@e.ntu.edu.sg

1. (a)

Determine $\text{dlog}_{7,41}38$.

Let $\text{dlog}_{7,41}38 = x$ with x is an integer
Means $7^x = 38 \pmod{41}$
We have table:

i	1	2	3	4	5
7^i	7	49	343	2401	16807
7^i mod 41	7	8	15	23	38

With x = 5 then $7^x = 38 \pmod{41}$
Hence $\text{dlog}_{7,41}38 = 5$

(b)

First:
{13} HEX = {0001 0011} BIN
{CC} HEX = {1100 1100} BIN
Second:
f(x)*g(x) = {13} HEX x {CC} HEX = {0001 0011} BIN x {1100 1100} BIN
= {111100100100 } BIN
Third:
m(x) = x^8 + x^4 + x^3 + x + 1 = {100011010} BIN
Forth:
f(x)*g(x) mod m(x)
= {1000 1100 0100} BIN mod {100011010} BIN = {11010010} BIN
Hence result is: $x^7 + x^6 + x^4 + x$

(c)

Diffusion refers to dissipating the statistical structure of plaintext over cipher text.
Diffusion means that if we change a character of the plaintext, then several characters of the cipher text should change, and similarly, if we change a character of the cipher text, then several characters of the plaintext should change.
The step in AES that involves diffusion is Mix Column.
Explanation: Mix Column transformation combined with the shift row transformation ensures that after a few rounds all output bits depend on all input bit. (Slide 15)

2. (a)

Determine 13^{-1} mod 67

Let $a = 13^{-1}$ mod 67
⇨ 13xa = 1 (mod 67)
⇨ 13xa + 67xb = 1 with a,b are different signed integer
⇨ a = (1 – 67xb)/13

b	1	2	3	4	5	6
a	5.2	10.3	15.5	20.6	25.8	31

With b = 6 then a = 31 ➔ 13x31 + 67x6 = 1 ➔ 13x31 = 1 (mod 67)
Hence 31 = 13^{-1} mod 67

(b)

(i)

Playfair cipher uses polyalphabetic substitution since one character in plaintext might be substituted by several character in cipher text depends on the key.

(ii)

It is important to have strong random generator for driving RSA Algorithm because this algorithm security is based on the infeasible computation to determine d given e and n. The number n is calculated based on two large random prime number p and q. If the generator of these two numbers is not strongly random, attacker might make use of it to guess the key d of algorithm.

(ii)

Fundamental differences of encryption when techniques based on computational hardness (ECC or RSA) are applied instead of confusion and diffusion techniques (AES) is that it uses public key encryption (asymmetric) instead of symmetric.

In AES, both sender and receiver must share the common key in order to start a communication. This key must be keep secret from any other party outside.

In ECC or RSA, only receiver keeps the private key and issues public key. Any party outside can use public key to send message to receiver without prior knowing receiver. The receiver uses private key to decrypt message but it is infeasible for any party to decrypt message without knowing private key. The infeasibility of this algorithm is based on the computational hardness of existing hardware.

In fact, both technologies are currently applied. ECC and RSA used as a secure way to transfer the key between sender and receiver. Then sender and receiver use AES algorithm with this secret key to send and receive message.

3. (a)

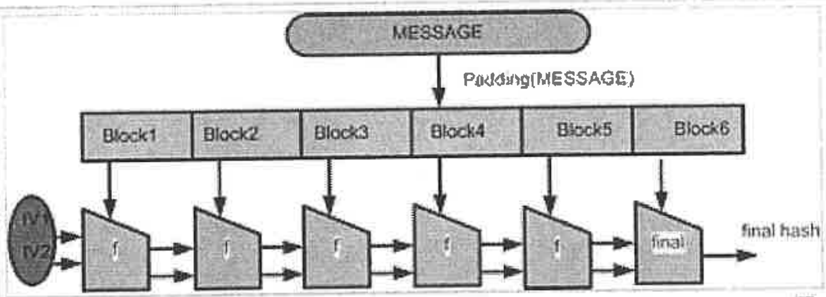
The main differences of cryptographic hash function compared to symmetric encryption is that using the hash function, receiver cannot reverse the encrypted results while using the symmetric encryption, receiver can reverse the encrypted results to see the original message.

As we can observe from password requirement, the system only need to verify whether the password is valid without necessity of knowing the password itself.

Both techniques allow this validation process. However in the hash technique, attacker cannot obtain any password even if the attacker can get full control over the system, the only thing they can do is verify whether a password is valid. Whereas in encrypted technique, if attacker can get control over system, the attacker can use master key and encryption password to find out all the password.

Hence the hash approach is more secured and preferable rather than encrypted approach.

(b)



The Wide pipe hash construction. The intermediate chaining values have been doubled.

Merkle-Damgard Hash Construction

Assumption:
MAC(K,M) = H(K||M)
K: one block b bits
M: blocks b bits
M': consist of x blocks b bits = m1, m2, …, mx
Hash function is available

Attack:
MAC(K,M||M') = H(K||M||M')
Based on the characteristic of Merkle-Damgard hash function which encrypt by block and all K, M, M' are of length multiple of b bits and the hash function is available.
Hence we can calculate H(K||M||M') from H(K||M) by:
Block0 = H(K||M) ← It is MAC(K,M)
Block1 = E(Block0 XOR m1) ← m1 is block 1 of M'
Block2 = E(Block1 XOR m2)
…
Blockx = E(Block(x-1) XOR mx) ← mx is the last block of M'
Hence attacker has MAC(K,M||M') = H(K||M||M') = Blockx
(c)
(i)
Existential Forgery attack is attacker can produce at least on attacked message.
Attack message is (x,s) = (a^e , a) means x = a^e , s = a
When Alice receive this, she will compute s^e = a^e and this exactly equal to x. Hence Alice will accept this message.
(ii)
Attack principle: choose signature → compute message (not vice versa)
Hence Attacker has no control over the message. If there is a message, he doesn't know how to compute the signature.
Modification: Introduce the structure to message, even for better security put in the timestamp.

Explanation: Since now the message must be predetermined by some structure, the attacker are infeasible to choose a signature that can exponentially computed to the predetermined structured message → can not attack.

4.

(a)

The purpose of requiring the uniqueness of the challenges is to protect the man-in-middle attack and subsequent replay attack.

(b)

A protocol is said to have perfect forward secrecy if compromise of long-term keys does not compromise past session keys. (slide 20 – key establishment part 1)

The protocol version is not perfect forward secrecy because if the attacker records the past traffic and gain the access to long term key.

He can use the long term key to decrypt the message to get the session key.

(c)

Attack:

First step: Intruder record the message E(PU_B,k) by:

(1) A → I(B): A, E(PU_B,k)
(2) I(A) → B: A, E(PU_I,k)
(3) B → A: B, E(PU_A,k)

Second step: Intruder send message to B on his behalf, not pretend anyone else:

(4) I → B: I, E(PU_B,k)
(5) B → I: B, E(PU_I,k)

Intruder will use his private key to decrypt E(PU_I,k) to get session key k

Modification, encrypt the identity instead of expose to public.

(1) A → B: E(PU_B, A||k)
(2) B → B: E(PU_B, B||k)

For reporting of errors and errata, please visit pypdiscuss.appspot.com
Thank you and all the best for your exams! ☺