Name: Chen Hao Cheng
HW3

Review Question Chapter 2:

2.9) List and briefly define three uses of a public-key cryptosystem.

Public key cryptosystem (public key encryption) is a cryptographic technique that provides confidentiality for data to transmit or store. It is different with symmetric keys (public key and private key).

3 Uses:

Encryption and Decryption:

The sender encrypts the data using the receiver's public key while transmit the data to the receiver. The receiver decrypts the data with receiver's private key.

Digital Signature:

Digital signature is a method that ensures the message for authentication and integrity by doing that the hash message for the digit code and then encrypts using sender's private key and it is added as signature.

Key exchange:

Only the public key can be exchanged by the sender and the receiver. They have their different private key.

2.10) What is the difference between a private key and a secret key?

Private Key	Secret Key
Private Key is used in asymmetric encryption	Secret is used in symmetric encryption
The algorithm in asymmetric encryption create public key and private key	The algorithm in symmetric encryption create secret key
Public key is shared between the sender and receiver. They bother keep their own private key.	The secret key is shared between the sender and the receiver.
One key is to encrypt the message, another key is to decrypt the message	The key is used to encrypt and decrypt

Problems:

2.5)

MAC(Message authentication code)**:** It is an authentication technique and an algorithm that produces MAC based on the message and the secret key.

DS (Digital signature)**:** Digital signature is a method that ensures the message for authentication and integrity by doing that the hash message for the digit code and then encrypts using sender's private key and it is added as signature.

a) (**Message integrity**) Digital signature can protect this **message integrity attack**. The digital code is created using the hash of the message, then encrypted with sender's private key, then add into signature. The message can be assured from the authorized party that declare to be and the message does not get changed. MAC also protects **message integrity attack**. MAC is create based on the message and secret key with the hash function for the message, and then encrypt with sender's private key and added as signature. While transmitting, MAC is with the message. On receiver side, the receiver gets the message with the MAC. Then the message with the secret key is inputted to MAC algorithm that produces MAC. The obtained MAC and received MAC will compare to check for message authentication. If they are the same, then we can be sure the receiver identity.
b) (**Replay**) Since there's no change in the message, then the repeat message cannot be identified by Bob because DS and MAC only assures the message is received from authenticated sender without change.
c) (**Sender authentication with cheating third party**) In this case, Bob can check for the signature of the sender by decrypting the signature for both public key from the two senders that result the hash message using DS. If the hash message matches one of them, then the person is the authenticated user. If Bob wants to use MAC, Bob can ask for the secret key because he has the secret key. Therefore, who has the same secret key, then that person is the authenticated user.
d) (**Authentication with Bob cheating**) In DS, Alice can ask Bob to forward the message with the signature. If the signature does not match, then Alice can prove. If using MAC, however, Alice cannot prove that she didn't send the message. So, Bob can say Alice has created the message and send.

2.6)
We can think the given function H(m) is a collision-resistant hash function that accepts the message that is in variable length and we can get hash output in fixed size. Meaning that this function cannot produce a unique output for each arbitrary input. The reason is that the possible output is only 2^n for an arbitrary input. More than one input can have the same hash value as the input. Even if the hash function is collision-resistant, it is still infeasible to find the value of y where $y \neq x$ with H(y) = H(x). This feature guarantees that an alternate message cannot be finding using the same hash value. It is infeasible to find the pair of message for (x,y) where H(x) = H(y). Thus, the statement is false.

Review Question Chapter 3:
3.1) In general terms, what are four means of authenticating a user's identity?

Something the individual knows: Example: Password, a personal identification number (PIN), or answers to a prearranged set of questions.

Something the individual possesses: Example: Include electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a ***token.***

Something the individual is (static biometrics): Example: Recognition by fingerprint, retina, and face.

Something the individual does (dynamic biometrics): Examples: Include recognition by voice pattern, handwriting characteristics, and typing rhythm.

Problems:

3.1) A strong password contains:

1) A combination of upper case letters, lower case letters, numeric, and special characters
2) Not too short
3) It should not be related to personal information such as birthdate, car number, and such
4) It should not have common names or patterns
5) It should not contain reverse of vocabulary

a) **YK334**, this password has a combination of letter and numbers but it is too short, so it is not suitable.
b) **mfmitm (my favorite movie is tender mercies),** this is suitable because it is not easier to guess even if there's no numeric or special character.
c) **Natalie1**, this is not a suitable password because it contains the common name of a person.
d) **Washington**, this is not a suitable password because it is a command word.
e) **Aristotle,** this is a common vocabulary for a person's name so it is not suitable.
f) **tv9stove,** this password is a combination of letters and numbers and it is not easy to guess, so it is suitable.
g) **12345678,** this only contains sequence numbers, so it is definitely not a suitable password.
h) **dribgib,** if you reverse the password, you will get "bigbird". This is not a suitable password.

3.3)

a) Since it is four-character combination of 26 alphabetic character and the adversary get no feedback from trying each character, it is 26^4 = 456976 possible passwords, and this is the worst case. On average, the adversary probably tries half or more of possible passwords so it is $\frac{456976}{2}$ = 228488 trails. There are 86400 seconds in one day. Then $\frac{456976}{86400}$ = 2.644 days, which is 63.456 hours.
b) This time the adversary gets feedback from trying **each incorrect character**. Thus, it will get 26 * 4 = 104 attempts for the worst case. On average, the adversary tries half or more. Thus, $\frac{104}{2}$ = 52, which is 52 seconds.

3.4)

a) Since each digit contains one of r values so the total number of source elements is r^k, so the probability of selecting a correct source element on one try for the adversary is $P = r^k$.

b) We should minus the probability of selecting a correct target element on one try P is r^p from the probability of selecting a correct source element on one try P is r^k. After that divides the obtained result with the product of the probability of selecting a correct target element on one try P is r^p and the probability of selecting a correct source element on one try P is r^k, therefore,

$$p = \frac{r^k - r^p}{r^k * r^p} = \frac{r^k - r^p}{r^{k+p}}$$

c) Since each digit contains one of r values so the total number of target element is r^p, so the probability of selecting a correct target element on one try $P = r^k$

3.6)

All password is 10 characters long. The password is limited to use 95 ASCII characters. Thus, it has 95^{10}. The cracker can do encryption rate of 6.4 million per second.

$$Time = \frac{95^{10}}{6.4 \frac{encryption}{second}} = \frac{\approx 6 * 10^{19}}{6.4 * 10^6}$$
$$\approx 9.4 * 10^{12} seconds$$

And then we convert the time to year, 1 year is 365 days * 24 hours/day * 60 minutes/hour * 60 seconds/ minute = 31,536,000 seconds.

$$Total\ Time = \frac{9.4 * 10^{12} seconds}{31{,}536{,}000\ seconds/year} = 298072.044647 \approx 300000\ years$$