MAS433 Cryptography:
Tutorial 5
Key Establishment and Management
03.12.2010

Problem 1. Randomness in key generation. Suppose that the key generation process is biased. Each key bit is independently generated, but the value of each key bit is '1' with probability 0.95. Suppose that the key size is 64 bits.

1.1 What is the entropy of each key bit?

$$-0.95 \times \log_2 0.95 - 0.05 \times \log_2 0.05$$

$$= 0.2864$$

1.2 What is the entropy of the key?

$$0.2864 \times 64 = 18.33$$

1.3 What is the probability that 64 bits in a key with values '1'?

$$0.95^{64} = 0.03752$$

1.4 What is the probability that at least 63 bits in a key with values '1'?

$$\sum_{i=63}^{64} \binom{64}{i} \times 0.95^{i} \times 0.05^{64-i}$$

$$= 0.1639$$

1.5 What is the probability that at least 62 bits in a key with values '1'?

$$\sum_{i=62}^{64} \binom{64}{i} \times 0.95^{i} \times 0.05^{64-i}$$

$$= 0.3735$$

1.6 What is the probability that at least 61 bits in a key with values '1'?

$$\sum_{i=61}^{64} \binom{64}{i} \times 0.95^{i} \times 0.05^{64-i}$$

$$= 0.6014$$

1.7 In Problem 1.6, how many keys have at least 61 bits with values '1'? Write your result in the form of $2^{X.X}$.

$$\sum_{i=61}^{64} \binom{64}{i} = 2^{15.42}$$

1.8 Compare the results of Problem 1.2 and 1.7. Explain how the entropy of a key is related to the strength of the key.

From 1.6 & 1.7, after trying $2^{15.42}$ keys, a secret key can be found with probability 0.5014.

The entropy of a secret key is 15.33, quite close to $\log_2 2^{15.42}$. It indicates that the entropy of a secret key is closely related to the complexity of brute force attack.

Problem 2. Public key certificate.

2.1 In secure shell (SSH), the public key certificate is not used. What is the risk?

The public key may be modified during the transmission.

2.2 When you connect to a website secured by TLS/SSL, if you notice that the public key certificate of that website is invalid (a warning window would appear if the certificate is invalid), and you continue to access that website, what is the risk?

If an attacker is ~~modify~~ modifying the network traffic and launching the man-in-the middle attack, then the communication between the client and the server may be known to the attacker.

Problem 3. Secret Sharing.
Alice is using the RSA digital signature scheme. Her public key is (n, e), and her private key is d. Alice uses the following scheme to protect her private key: generate three integers d_1, d_2, d_3 in $[0, n-1]$ so that $d_1 + d_2 + d_3 = d$. After storing each d_i on a trusted server S_i $(1 \leq i \leq 3)$, Alice deletes her private key x.

3.1 To sign a message m, Alice sends $H(m)$ (the message digest of m) to each server S_i ($1 \leq i \leq 3$), and each server S_i performs a local computation and sends T_i to Alice. Given T_1, T_2, T_3, Alice generates the signature without reconstructing her private key. Explain how server S_i computes T_i and how Alice generates the signature.

Solution.

S_1 : $T_1 = (H(m))^{d_1} \bmod n$

S_2 : $T_2 = (H(m))^{d_2} \bmod n$

S_3 : $T_3 = (H(m))^{d_3} \bmod n$

Then S_i sends T_i to Alice

Alice : $S = (T_1 \times T_2 \times T_3) \bmod n$

3.2 To provide fault tolerance, Alice shares her private key d among the three servers so that any two of the three servers can be used to sign messages without reconstructing her private key d, but the signature cannot be generated with only one server. Explain how to share the private key d, and how to sign a message.

Solution: Alice chooses three random numbers d_1, d_2, d_3 so that $d_1 + d_2 + d_3 = d$.

Alice gives d_1, d_2 to S_1

gives d_2, d_3 to S_2

gives d_3, d_1 to S_3

1) when S_1 & S_2 are available,

S_1 computes $T_1 = H(m)^{d_1+d_2} \bmod n$

S_2 computes $T_2 = H(m)^{d_3} \bmod n$

S_1 sends T_1 to Alice, S_2 sends T_2 to Alice.

Alice computes $S = (T_1 \times T_2) \bmod n$,

2) when S_2 & S_3 are available,

S_2 computes $T_2 = H(m)^{d_2+d_3} \bmod n$

S_3 computes $T_3 = H(m)^{d_1} \bmod n$

S_2 sends T_2 to Alice; S_3 sends T_3 to Alice,

Alice computes $S = (T_2 \times T_3) \bmod n$,

3) when S_1 & S_3 are available,

S_1 computes $T_1 = H(m)^{d_1+d_2} \bmod n$

S_3 computes $T_3 = H(m)^{d_3} \bmod n$

Alice computes $S = (T_1 \times T_3) \bmod n$;