Final Exam

111061548 游鎮遠

1、 Discrete logarithm (DL) based identication scheme is better than traditional password-based identication scheme, for reasons like:

security: password-based identication can be easily guessed, and easily attacked by such as brute force, dictionary attacks and phishing. But DL-based identication use mathematical algorithms to ensure identification, it difficult to attack.

Anthentication strength: the passwords that human generated are often weak and vulnerable to social enginering attacks. But DL-based scheme use cryptographic keys and digital signature to provide stronger security.

And about the efficiency in terms of computation and communication complexities,

DL-based identication typically involve more computational than password-based identication, because of Encryption、Decryption and digtial signature operations are computationally intensive tasks, and also DL-based identication may need additional communication overhead than password-based, because of cryptography often involves the exchange of keys and certificates.

2.(a) ElGamal encryption is "not" secure against the CCA.

CCA means that : provide Od (decryption Orcale) to attacker

ElGamal encryption

keyGen()

Attacker,

knows p, g, y, a, b,

and want to compute m.

challenge ciphertext :

$C = (g^k, my^k)$ $\xrightarrow{C=(a,b)}$ $m = ?$

Compute

$b_1 = 2 \cdot b \bmod p$

$\xleftarrow{c'=(a,b')}$

$c' \neq c$,

not the challenge ciphertext,

$(g^k, 2my^k) \neq (g^k, my^k)$ $\xrightarrow{m'}$ $m = \frac{m'}{2}$,

the Attacker can simply compute $\frac{m'}{2}$ to obtains m.

#

2.(b)

If the hash function is applied to the original message, then the signature is hash value, so ElGamal signature scheme is secure from chosen plaintext attack, because find a meaningful message $h(m') = m$ is not easy.

2. (c)

We cannot ask the oracle about the signature of m, but we can design a forger algorithm to query the oracle for any message except m.

① query the oracle m', where $\frac{m}{m'} \bmod p-1 = u$

the oracle return $r = g^k \bmod p$, $s = k^{-1}(m - rx) \bmod p-1$

② compute $s' \bmod p-1 = su$, and $r' \bmod p-1 = ru$,

$r' \bmod p = r$.

③ then check $y^{r'} r^{s'} = y^{ru} r^{su} = (y^r r^s)^u = (g^{m'})^u = g^m$

④ get m, r', s' ✩.

3. (a) They can use Diffie-Hellman key exchange to share the common key to protect their message

Alice send to Bob : $c = g^a$, where $a \in Z_{p-1}$

Bob send to Alice : $d = g^b$, where $b \in Z_{p-1}$,

Alice can compute : $k = d^a = g^{ab}$

Bob can compute : $k = c^b = g^{ab}$.

3. (b) The attacker can do Man-in-the-middle attack.

Alice $\xrightarrow{c = g^a}$ Attacker $\xrightarrow{c' = g^{a'}}$ Bob

Bob $\xrightarrow{d = g^b}$ Attacker $\xrightarrow{d' = g^{b'}}$ Alice

Alice will compute : $k = d'^a = g^{ab'}$

Bob will compute : $k = c'^b = g^{a'b}$.

So the attacker can know the message that they send, and even can change it.

3. (c)

$p = 107$, $g = 2$

ⓐ Alice send to Bob : $4 = 2^2$, where $2 \in \mathbb{Z}_{106}$

Bob send to Alice : $61 = 2^{10}$, where $10 \in \mathbb{Z}_{106}$

Alice can compute : $4^{10} = 61^2 = 83 = 2^{2 \cdot 10}$

Bob can compute : $61^2 = 4^{10} = 83 = 2^{2 \cdot 10}$

ⓑ let $a' = 20$, $b' = 30$

Alice $\xrightarrow{4 = 2^2}$ Attacker $\xrightarrow{83 = 2^{20}}$ Bob

Bob $\xrightarrow{61 = 2^{10}}$ Attacker $\xrightarrow{34 = 2^{30}}$ Alice

Alice will compute : $K = 34^2 = 86$

Bob will compute : $K = 83^{10} = 25$

#

4.

We query the singning orcale the message $m' = m \cdot r^e \bmod N$,

the signig orcale will return $\delta' = m'^d = m^d \cdot r \bmod N$.

then we can compute $\delta = \frac{\delta'}{r} = m^d \bmod N$

#

$m' \bmod N = mr^e$

$\delta' = m'^d \bmod N = m^d r$

$\delta = \frac{\delta'}{r} = m^d$

5.

Instead of trusted T to setup,

1. Each A_i selects x_i and $f_i(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \dots + a_1 x + s$ with $f_i(0) = x_i$ and publishes $h_i = g^{x_i}$

2. The public key is $h = \prod_{i=1}^{N} h_i$;

3. Share the secret key $x = \sum_{i=1}^{n} x_i = f(0)$,

   where $f(x) = \sum_{k=1}^{n} f_k(x)$,

   each A_i sends $s_{ij} = f_i(j)$ to A_j via secure channel.

   each A_j computes it share $s_j = \sum_{k=1}^{n} s_{k,j} = \sum_{k=1}^{n} f_k(j) = f(j)$.

   each A_j should check whether the received share $s_{i,j}$ from A_j is valid.