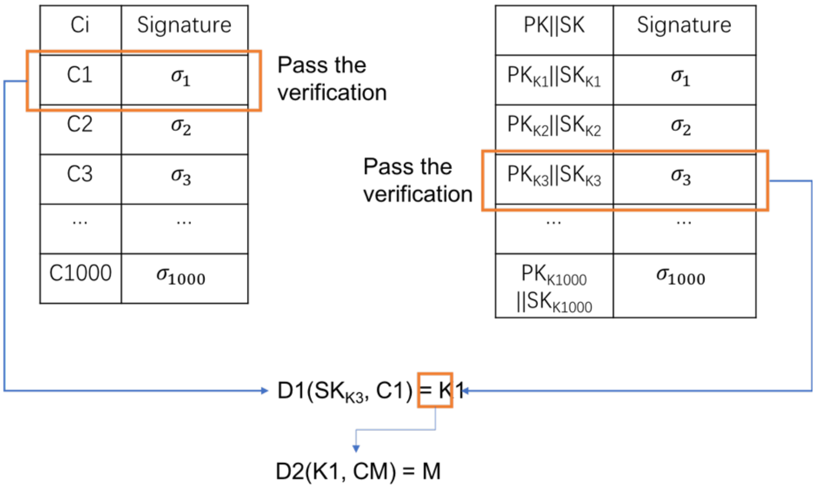
Programming Assignment 3

Teacher holds a CA key pairs PK_C and SK_C. PK_C is shown below while SK_C is kept secret by the teacher. One of the 10^3 ciphertext Ci is signed by SK_C, and rest of the Ci are signed by other random keys, which are not shown here. Also we have 10^3 public key and secret key pairs PK_{Ki} and SK_{Ki}, and one of them is signed by SK_C as well, while other key pairs are signed by other random keys. You are required to use PK_C to verify the signature and pick out the right c_i as well as the PK_{Ki} and SK_{Ki}. Then you use SK_{Ki} to decrypt c_i to derive k_i, which is the secret key that can decrypt another ciphertext CM to recover the final message M. The flow is shown in the following figure.



Algorithms

ECDSA is used for all the signature scheme where SHA256 is used to hash the message .
D1 is the RSA-OAEP encryption algorithm, and D2 is the RC4 algorithm.

Example:

Signature

The public key for the ECDSA is as follows. X and Y are the coordinates of the public key point, and Gx and Gy are the coordinates of the base point. The curve we use is P-224.

X	= "A374F7774070CD8BECCDD5B01450FFBC0033EE5FFBFEC7829C10DDD1 "
Y	= "F484F9AF3B9DD12C93F5DA7971333C3A0DCDBD20865CCE59145D9180 "
Gx	= "B70E0CBD6BB4BF7F321390B94A03C1D356C21122343280D6115C1D21"
Gy	= "BD376388B5F723FB4C22DFE6CD4375A05A07476444D5819985007E34"

D1

RSA key (PK||SK) is constructed in the form "N,e,d,p,q" . For example, the following is one RSA key with signature (r, s) appended at the end:

9338d7e99c1e2be468b02142510b205d5852f6c0873fe51a521f4353743a6032f415dcac0ca6
5f6e271dfbc66bddf17d1adb3cc9c30ebc7b84b0922aca360c29b7dda3a7a104bcca2c68f8af3
e2ee85bc946c0fe9ad82dc56b954e476c83e7d9f57e18f3e2c6330b4590ff445864e168fc53a4

```
edf6672f742f52162a206aa6bf7d092c356e1b445825d211760df7e958b84f13753bde4e0bf4e
9e22b90496aa3d95bc57fc0ce00ac7bd22e9c0da2f89cda233be53ca028f6e4c3d472d63b4d0
6ffd86046c46f674e0cd0c27878649631b0be08ad5dc05efe741055edd69b07780406fca2b18
279c86a2033fed8b9c1223af4aee0545f3ecf59af6d389695e389,10001,2ee676f23707ed97b0
3a1bfe526f26fa55bee858e13d5bb10ce464c05b509580e5fd68f56e7a3a008f799fd1d05f3e2
54abef918958946465b391cf780bfc3142f3cee7be271edeac24716a24d3f084ae54e23017936
d12e095de13823bbc9cac7c84a0eb7f9c81d19265e0bd9d5197226a046f48000d33f422bf65f
a94010aa886a29920da5aa521918b5e8e89887fd448839dcb2fcd061023456c5cc9e6f48939d
d07c5f015a7d7999a553e4ae4fb5314be270b82656700961d1d1f79ca259f2149596422245e5
5e00f8800588173694d943903b33b603bf24cb601ad6400957e29738c6df9d7ad7211520824
0790cf2ae6da5bcfc2bae2c6468fbeb6a9401,c07f69e06dea921c01173ff1309f3d82f02346e48
a159e2b18667f329ddfb246eab029ab3cb057ca62b71fd37c7156bed36889a4409e8f8d3c564
f5b9ade0cd0864fff608404712e4934691e378bd9c2a1e905d9651868b9cd8b314e4806b5b59
f0c7dfc3cc5184f3703c82d49cf3858f0274e02cff1c011029d6c9752c372c9,c3c9dd0f56dc046a
736a5e88897567cf40965938600da79774a3d7f8d4eed5a1af58987e20555bcbc1915f339010
33bd1a81a8762b2096f4aef3eb034094937cbaa2cb5863166138e6df0beb81513c2b077a1900
dd439f2ae944264dea76db25010bde4658f975745627eb64f9e659704ccbda98f771951483b8
66ca5c7c8ac1,744b681a5e2fe816672642b26adcc3e59b288bfd4031b5d531b2dce8,58bb3d1
cdbbf6e86e9f8fc15d3930feda249d0c98e9b791a0b728a39
```

"Tiny Bitcoin Mining":

After final message M is found, now you become a "miner" like in Bitcoin. You have to do a following quiz to compete with other groups:

1. Find a number r
2. Do the process: SHA256(M||r) to get a hashed value (r is hex encoded)
3. See if the most significant 24 bits are 0^{24}. If not, repeat process 1,2 and try another number.

Keep in mind that r is also what you need to submit! So don' t forget to record your r number. Every group needs to find a different r! So remember to claim the r you find to your classmates and tell them this r belongs to your group and cannot be used by other group.

You need to submit the following things for this task:

1. All of your code files.
2. A document including:

   The M and r you find.

   A explanation on your code design.

***Added task:**

Implement the following two algorithms: Elgamal signature scheme and Elgamal encryption scheme to perform the following operation.

1. Concatenate all the names in your group to form a string s.
2. Generate a random key string k with the same length.
3. Xor the k with s to derive C_s.
4. Encrypt k using Elgamal encryption scheme to get ciphertext C_k.

5. Digitally sign C_k using Elgamal signature scheme.
6. Receiver after receiving C_s, C_k, and signature should be able to decrypt C_k, verify the signature, and derive the string s.

Choose 2048-bit group size for both encryption and signature schemes. All the key materials should be hard-coded in the program for the ease of verification.

Elgamal signature scheme:

Key generation

1. Randomly choose a secret key x with 1 < x < p − 2.
2. Compute y = g^x mod p.
3. The public key is y.
4. The secret key is x.

Signature generation:

1. Choose a random k such that 1 < k < p − 1 and gcd(k, p − 1) = 1.
2. Compute $r \equiv g^k \, mod \, p$ **alice 产生签名**
3. Compute $s \equiv (H(m) - xr)k^{-1} \, mod \, (p-1)$
4. If s=0, start over again **hash值为十六进制, 所以可以进制减法**

(r, s) is the signature pair

Verification

1. 0<r<p and 0<s<p-1.
2. $g^{H(m)} \equiv y^r r^s (mod \, p)$

Elgamal Encryption scheme: Please refer to the slide

You need to submit the following things for this task:

1. Source code
    a) Source code should have user-friendly interface, so that it is easy for anyone to run and verify every functionality easily.
2. Document that explain the code as well as how to run the program for each of the steps.