密码学课程设计

华中科技大学网安学院

OJ:HTTP://10.12.162.1:5880/，HTTP://124.71.166.97:5880/

一、原始SPN算法的实现

内容：按照教材例3.1实现SPN加解密函数。

要求：

（1）正确实现算法的加解密过程

（2）快速实现算法

一、原始SPN算法的实现

输入：

第一行，一个正整数n,表示n组数据

接下来n行，每行包含两个用空格分开的16进制字符串。第一个字符串长度为8，表示32比特密钥key，第二个字符串长度为4，表示16比特明文。

输出：

共n行，每行包含两个用空格分开的16进制字符串。第一个字符串表示对应行的明文加密后的密文，第二个字符串表示将密文最后1比特取反，解密后得到的明文。

二、原始SPN的线性分析

内容：根据已知明密文对分析原始SPN的密钥。

要求：

（1）实现教材所给算法。

（2）能根据所给明密文对分析对应位置密钥。

（3）分析出所有32比特密钥。

二、原始SPN的线性分析

输入：

第一行：一个正整数n,表示输入的明密文对数量

第二行：4n个16进制字符，表示n组明文

第三行：4n个16进制字符，表示n组对应密文

输出：

一行，8个16进制字符，表示32比特密钥key

1<=n<=8000

三、原始SPN的差分分析

内容：根据已知明密文对，选择明密文分析原始SPN的密钥。

要求：

（1）实现教材所给算法。

（2）能根据所给明密文对分析对应位置密钥。

（3）分析出所有32比特密钥。

三、原始SPN的差分分析

输入：

一行：4*65536个16进制字符，表示0000-ffff所对应的密文

输出：

一行，8个16进制字符，表示32比特密钥key

四、SPN增强

内容：对原始SPN进行改进。

要求：

（1）选择合适的密钥长度、分组长度、S盒、P置换、轮数。

（2）效率较高

（3）输出达到随机数检测标准

四、SPN增强

输入：

第一行：32个16进制字符，表示128比特密钥

第二行：1e6个16进制字符，表示明文

输出：

一行：利用所设计的算法对明文进行加密，输出16进制密文。密钥长度不足128比特的，选取输入密钥的前面部分作为密钥，密钥长度超过128比特的，全部补0。

五、RSA参数生成

内容：求RSA参数d。

要求：

（1）利用加法、减法、乘法、模运算等基本运算。

（2）自己实现求逆,gcd

（3）简单检查参数的合法性

五、 RSA参数生成

输入：

第一行：1个正整数n，表示问题的个数

接下来n行，大整数e，p，q

输出：

共n行，如果三个参数正确，则输出对应的d,否则输出 ERROR.

六、模重复平方

内容：正确计算a^e(modN)。

要求：

（1）利用加法、减法、乘法、模运算等基本运算。

（2）自己实现expmod(a,e,n)

六、模重复平方

输入：

第一行：1个正整数n，表示问题的个数

接下来n行，包含大整数e，m，p,q，p,q均不超过1024bit

输出：

共n行，输出expmod(m,e,pq)

七、中国剩余定理

内容：正确计算c^d(modpq)。

要求：

（1）利用六中的基本运算和中国剩余定理计算c^d(modpq)。 。

（2）结合五中的求逆运算

七、中国剩余定理

输入：

第一行：1个正整数n，表示问题的个数，大整数e，p,q,其中，p,q不超过1024比特

接下来n行，每一行一个大整数c

输出：

共n行，输出expmod(c,d,pq)

八、Montgomery算法

内容：正确计算a^e(modN)。

要求：

（1）实现Montgomery算法。 。

（2）Montgomery+中国剩余定理。

八、 Montgomery算法

输入：

第一行：1个正整数n，表示问题的个数，大整数e，p,q,其中，p,q不超过1024比特

接下来n行，每一行一个大整数c

输出：

共n行，输出expmod(c,d,pq)

九、SM2公钥加密算法（安装）

GmSSL:实现了SM2、 SM3、 SM4、 SM9、ZUC算法，支持国产硬件。 GmSSL项目是OpenSSL项目的分支，并与OpenSSL保持接口兼容。因此GmSSL可以替代应用中的OpenSSL组件，并使应用自动具备基于国密的安全能力。

安装gmssl： http://gmssl.org/

ubuntu 18.04

git clone https://github.com/guanzhi/GmSSL.git

进入GmSSL目录，执行

./config

make

make install

运行gmssl，一般会出错，提示没有找到libssl和libcrypto，此时可通过ldd /usr/local/bin/gmssl查看依赖的路径，并将刚刚生成的两个库文件拷贝过去。

十、SM2公钥加密算法（查看gmssl 支持的曲线）

命令： gmssl ecparam -list_curves

可以看到很多被命名的椭圆曲线，gmssl暂不支持用户自定义椭圆曲线。

```
Oakley-EC2N-4:
      IPSec/IKE/Oakley curve #4 over a 185 bit binary field.
      Not suitable for ECDSA.
      Questionable extension field!
brainpoolP160r1: RFC 5639 curve over a 160 bit prime field
brainpoolP160t1: RFC 5639 curve over a 160 bit prime field
brainpoolP192r1: RFC 5639 curve over a 192 bit prime field
brainpoolP192t1: RFC 5639 curve over a 192 bit prime field
brainpoolP224r1: RFC 5639 curve over a 224 bit prime field
brainpoolP224t1: RFC 5639 curve over a 224 bit prime field
brainpoolP256r1: RFC 5639 curve over a 256 bit prime field
brainpoolP256t1: RFC 5639 curve over a 256 bit prime field
brainpoolP320r1: RFC 5639 curve over a 320 bit prime field
brainpoolP320t1: RFC 5639 curve over a 320 bit prime field
brainpoolP384r1: RFC 5639 curve over a 384 bit prime field
brainpoolP384t1: RFC 5639 curve over a 384 bit prime field
brainpoolP512r1: RFC 5639 curve over a 512 bit prime field
brainpoolP512t1: RFC 5639 curve over a 512 bit prime field
sm2p256v1 : SM2 curve over a 256 bit prime field
wapip192v1: WAPI curve over a 192 bit prime field
sm9bn256v1: SM9 BN curve over a 256 bit prime field
```

十、SM2公钥加密算法（查看SM2曲线）

命令:gmssl ecparam -text -noout -name sm2p256v1 -param_enc explicit

Field Type: prime-field

Prime:

00:ff:ff:ff:fe:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:ff:ff:ff:ff:00:00:00:00:ff:ff:ff:ff:ff:
ff:ff:ff

A:

00:ff:ff:ff:fe:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:ff:ff:ff:ff:00:00:00:00:ff:ff:ff:ff:ff:
ff:ff:fc

B:

28:e9:fa:9e:9d:9f:5e:34:4d:5a:9e:4b:cf:65:09:
a7:f3:97:89:f5:15:ab:8f:92:dd:bc:bd:41:4d:94:
0e:93

Generator (uncompressed):

04:32:c4:ae:2c:1f:19:81:19:5f:99:04:46:6a:39:
c9:94:8f:e3:0b:bf:f2:66:0b:e1:71:5a:45:89:33:
4c:74:c7:bc:37:36:a2:f4:f6:77:9c:59:bd:ce:e3:
6b:69:21:53:d0:a9:87:7c:c6:2a:47:40:02:df:32:
e5:21:39:f0:a0

Order:

00:ff:ff:ff:fe:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:72:03:df:6b:21:c6:05:2b:53:bb:f4:09:39:
d5:41:23

Cofactor: 1 (0x1)

十、SM2公钥加密算法（配置openssl.cnf）

/usr/local/ssl/openssl.cnf

```
# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = C
countryName_default             = CN
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = ST
stateOrProvinceName_default     = HuBei

localityName                    = LT
localityName_default            =WuHan

0.organizationName              = O
0.organizationName_default      = HUST

# we can do this but it is not needed normal
#1.organizationName             = Second Org
#1.organizationName_default     = World Wide
```

```
####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = .                     # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
unique_subject  = no                    # Set to 'no' to allow creation of
                                        # several certs with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber       = $dir/crlnumber        # the current crl number
                                        # must be commented out to leave a V1 CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem # The private key
RANDFILE        = $dir/private/.rand    # private random number file
```

十、SM2公钥加密算法（生成CA私钥）

命令：gmssl ecparam -genkey -name sm2p256v1 -out private/cakey.pem

利用椭圆曲线sm2p256v1，生成CA私钥cakey.pem，内容如下

-----BEGIN EC PARAMETERS-----

BggqgRzPVQGCLQ==

-----END EC PARAMETERS-----

-----BEGIN EC PRIVATE KEY-----

MHcCAQEEIPbi1wvZ1D8gK4Oi8HJ1r24H7kixgW1JX4GUD+/zU4SBoAoGCCqBHM9V

AYItoUQDQgAEifJptvCSUW2Si6+gwqNfilopqShLIk32TNfWYuWqOwLumC6LMje9

duejFpo8JMF8SXi2tJQn5pBf2Fgl+qhkNA==

-----END EC PRIVATE KEY-----

查看CA私钥

命令： gmssl ec -text -in private/cakey.pem

```
read EC key
Private-Key: (256 bit)
priv:
    f6:e2:d7:0b:d9:d4:3f:20:2b:83:a2:f0:72:75:af:
    6e:07:ee:48:b1:81:6d:49:5f:81:94:0f:ef:f3:53:
    84:81
pub:
    04:89:f2:69:b6:f0:92:51:6d:92:8b:af:a0:c2:a3:
    5f:8a:5a:29:a9:28:4b:22:4d:f6:4c:d7:d6:62:e5:
    aa:3b:02:ee:98:2e:8b:32:37:bd:76:e7:a3:16:9a:
    3c:24:c1:7c:49:78:b6:b4:94:27:e6:90:5f:d8:58:
    25:fa:a8:64:34
ASN1 OID: sm2p256v1
NIST CURVE: SM2
```

十、SM2公钥加密算法（生成CA自签名证书）

生成CA自签名证书

命令： gmssl req -x509 -sm3 -days 3650 -key private/cakey.pem -out cacert.pem

-----BEGIN CERTIFICATE-----
MIIB/zCCAaagAwIBAgIJAKKa0PAt9M1FMAoGCCqBHM9VAYN1MFsxCzAJBgNVBAYT
AkNOMQ4wDAYDVQQIDAVIdUJlaTEOMAwGA1UEBwwFV3VIYW4xDTALBgNVBAoMBEhV
U1QxDDAKBgNVBAsMA0NTRTEPMA0GA1UEAwwGY2Fyb290MB4XDTIwMDkyMDIwNTkx
OVoXDTMwMDkxODIwNTkxOVowWzELMAkGA1UEBhMCQ04xDjAMBgNVBAgMBUh1QmVp
MQ4wDAYDVQQHDAVXdUhhbjENMAsGA1UECgwESFVTVDEMMAoGA1UECwwDQ1NFMQ8w
DQYDVQQDDAZjYXJvb3QwWTATBgcqhkjOPQIBBggqgRzPVQGCLQNCAASJ8mm28JJR
bZKLr6DCo1+KWimpKEsiTfZM19Zi5ao7Au6YLosyN71256MWmjwkwXxJeLa0lCfm
kF/YWCX6qGQ0o1MwUTAdBgNVHQ4EFgQUAL5hW3RUzqvsiTzIc1gUHeK5uzQwHwYD
VR0jBBgwFoAUAL5hW3RUzqvsiTzIc1gUHeK5uzQwDwYDVR0TAQH/BAUwAwEB/zAK
BggqgRzPVQGDdQNHADBEAiAaZMmvE5zzXHx/TBgdUhjtpRH3Jpd6OZ+SOAfMtKxD
LAIgdKq/v2Jkmn37Y9U8FHYDfFqk5I0qlQOAmuvbVUi3yvM=
-----END CERTIFICATE-----

查看CA自签名证书

命令： gmssl x509 --in cacert.pem -text

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            a2:9a:d0:f0:2d:f4:cd:45
    Signature Algorithm: sm2sign-with-sm3
        Issuer: C = CN, ST = HuBei, L = WuHan, O = HUST, OU = CSE, CN = caroot
        Validity
            Not Before: Sep 20 20:59:19 2020 GMT
            Not After : Sep 18 20:59:19 2030 GMT
        Subject: C = CN, ST = HuBei, L = WuHan, O = HUST, OU = CSE, CN = caroot
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:89:f2:69:b6:f0:92:51:6d:92:8b:af:a0:c2:a3:
                    5f:8a:5a:29:a9:28:4b:22:4d:f6:4c:d7:d6:62:e5:
                    aa:3b:02:ee:98:2e:8b:32:37:bd:76:e7:a3:16:9a:
                    3c:24:c1:7c:49:78:b6:b4:94:27:e6:90:5f:d8:58:
                    25:fa:a8:64:34
                ASN1 OID: sm2p256v1
                NIST CURVE: SM2
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            00:BE:61:5B:74:54:CE:AB:EC:89:3C:C8:73:58:14:1D:E2:B9:BB:34
        X509v3 Authority Key Identifier:
            keyid:00:BE:61:5B:74:54:CE:AB:EC:89:3C:C8:73:58:14:1D:E2:B9:BB:34

        X509v3 Basic Constraints: critical
            CA:TRUE
    Signature Algorithm: sm2sign-with-sm3
        30:44:02:20:1a:64:c9:af:13:9c:f3:5c:7c:7f:4c:18:1d:52:
        18:ed:a5:11:f7:26:97:7a:39:9f:92:38:07:cc:b4:ac:43:2c:
        02:20:74:aa:bf:bf:62:64:9a:7d:fb:63:d5:3c:14:76:03:7c:
        5a:a4:e4:8d:2a:95:03:80:9a:eb:db:55:48:b7:ca:f3
```

十、SM2公钥加密算法（生成用户A私钥）

生成A的私钥

命令： gmssl ecparam -genkey -name sm2p256v1 –out private/UserA.key

```
-----BEGIN EC PARAMETERS-----
BggqgRzPVQGCLQ==
-----END EC PARAMETERS-----
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIErrb3NV+HqWKfoINmBAFyNAe/knxHHXt5tXi2YCqgmtoAoGCCqBHM9V
AYItoUQDQgAEe9p8OgKCf6aXB8yvz2Q+UL3pexIkAK6fTE0sQU6dvjqRlJ4pR9G0
tRyCkOiLA5VXsuKKV57CAlCooWLm/LvAgA==
-----END EC PRIVATE KEY-----
```

十、SM2公钥加密算法（生成用户A证书请求）

生成A的证书请求

命令： gmssl req -new -key private/UserA.key -out UserA.csr

-----BEGIN CERTIFICATE REQUEST-----

MIIBFjCBvAIBADBaMQswCQYDVQQGEwJDTjEOMAwGA1UECAwFSHVCZWkxDjAMBgNV

BAcMBVd1SGFuMQ0wCwYDVQQKDARIVVNUMQwwCgYDVQQLDANDU0UxDjAMBgNVBAMM

BVVzZXJBMFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEe9p8OgKCf6aXB8yvz2Q+

UL3pexlkAK6fTE0sQU6dvjqRlJ4pR9G0tRyCkOiLA5VXsuKKV57CAlCooWLm/LvA

gKAAMAoGCCqBHM9VAYN1A0kAMEYCIQC4F2nQnYxilB/2Z2KWffma6AOZ0aTriMxi

bw0ojShFpgIhAO7TPMR0EeQYu5w92CWKSp2INXFnxbS9IKGI5xM9LoP6

-----END CERTIFICATE REQUEST-----

查看A的证书请求

命令：gmssl req -in UserA.csr -text

```
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = CN, ST = HuBei, L = WuHan, O = HUST, OU = CSE, CN = UserA
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:7b:da:7c:3a:02:82:7f:a6:97:07:cc:af:cf:64:
                    3e:50:bd:e9:7b:12:24:00:ae:9f:4c:4d:2c:41:4e:
                    9d:be:3a:91:94:9e:29:47:d1:b4:b5:1c:82:90:e8:
                    8b:03:95:57:b2:e2:8a:57:9e:c2:02:50:a8:a1:62:
                    e6:fc:bb:c0:80
                ASN1 OID: sm2p256v1
                NIST CURVE: SM2
        Attributes:
            a0:00
    Signature Algorithm: sm2sign-with-sm3
         30:46:02:21:00:b8:17:69:d0:9d:8c:62:94:1f:f6:67:62:96:
         7d:f9:9a:e8:03:99:d1:a4:eb:88:cc:62:6f:0d:28:8d:28:45:
         a6:02:21:00:ee:d3:3c:c4:74:11:e4:18:bb:9c:3d:d8:25:8a:
         4a:9d:88:35:71:67:c5:b4:bd:20:a1:88:e7:13:3d:2e:83:fa
```

十、SM2公钥加密算法（生成用户A证书）

生成A的证书

命令： gmssl ca -days 3650 -in UserA.csr -out certs/UserA.crt

-----BEGIN CERTIFICATE-----
MIICGzCCAcKgAwIBAgIBAjAKBggqgRzPVQGDdTBbMQswCQYDVQQGEwJDTjEOMAwG
A1UECAwFSHVCZWkxDjAMBgNVBAcMBVd1SGFuMQ0wCwYDVQQKDARIVVNUMQwwCgYD
VQQLDANDU0UxDzANBgNVBAMMBmNhcm9vdDAeFw0yMDA5MjAyMTEyMTZaFw0zMDA5
MTgyMTEyMTZaMEoxCzAJBgNVBAYTAkNOMQ4wDAYDVQQIDAVIdUJlaTENMAsGA1UE
CgwESFVTVDEMMAoGA1UECwwDQ1NFMQ4wDAYDVQQDDAVVc2VyQTBZMBMGByqGSM49
AgEGCCqBHM9VAYItA0IABHvafDoCgn+mlwfMr89kPlC96XsSJACun0xNLEFOnb46
kZSeKUfRtLUcgpDoiwOVV7LiileewgJQqKFi5vy7wICjgYcwgYQwCQYDVR0TBAIw
ADALBgNVHQ8EBAMCBSAwKgYJYIZIAYb4QgENBB0WG0dtU1NMIEdlbmVyYXRlZCBD
ZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUIePApg8IOc4nvX/kMH0Cja0Mf4gwHwYDVR0j
BBgwFoAUAL5hW3RUzqvsiTzIc1gUHeK5uzQwCgYIKoEcz1UBg3UDRwAwRAIgI3Q0
qndxJPTgvC8sPNarf7pJwdsIrU0Ajmlv4PdatU4ClHU8Pyt6rT9BaHrf8ppJduPe
GWCy2i4+1au/zyp42b0r
-----END CERTIFICATE-----

查看A的证书

命令：gmssl x509 -in certs/UserA.crt -text

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sm2sign-with-sm3
        Issuer: C = CN, ST = HuBei, L = WuHan, O = HUST, OU = CSE, CN = caroot
        Validity
            Not Before: Sep 20 21:12:16 2020 GMT
            Not After : Sep 18 21:12:16 2030 GMT
        Subject: C = CN, ST = HuBei, O = HUST, OU = CSE, CN = UserA
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:7b:da:7c:3a:02:82:7f:a6:97:07:cc:af:cf:64:
                    3e:50:bd:e9:7b:12:24:00:ae:9f:4c:4d:2c:41:4e:
                    9d:be:3a:91:94:9e:29:47:d1:b4:b5:1c:82:90:e8:
                    8b:03:95:57:b2:e2:8a:57:9e:c2:02:50:a8:a1:62:
                    e6:fc:bb:c0:80
                ASN1 OID: sm2p256v1
                NIST CURVE: SM2
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Key Encipherment
        Netscape Comment:
            GmSSL Generated Certificate
        X509v3 Subject Key Identifier:
            21:E3:C0:A6:0F:08:39:CE:27:BD:7F:E4:30:7D:02:8D:AD:0C:7F:88
        X509v3 Authority Key Identifier:
            keyid:00:BE:61:5B:74:54:CE:AB:EC:89:3C:C8:73:58:14:1D:E2:B9:BB:34

    Signature Algorithm: sm2sign-with-sm3
        30:44:02:20:23:74:34:aa:77:71:24:f4:e0:bc:2f:2c:3c:d6:
        ab:7f:ba:49:c1:db:08:ad:4d:00:8e:69:6f:e0:f7:5a:b5:4e:
        02:20:75:3c:3f:2b:7a:ad:3f:41:68:7a:df:f2:9a:49:76:e3:
        de:19:60:b2:da:2e:3e:d5:ab:bf:cf:2a:78:d9:bd:2b
-----BEGIN CERTIFICATE-----
```

查看B的证书

命令：gmssl x509 -in certs/UserB.crt -text

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
    Signature Algorithm: sm2sign-with-sm3
        Issuer: C = CN, ST = HuBei, L = WuHan, O = HUST, OU = CSE, CN = caroot
        Validity
            Not Before: Sep 20 21:21:03 2020 GMT
            Not After : Sep 18 21:21:03 2030 GMT
        Subject: C = CN, ST = HuBei, O = HUST, OU = CSE, CN = UserB
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:1f:ba:cb:2e:21:40:49:ed:d2:59:2b:06:6f:11:
                    54:b5:f3:d8:ff:fa:57:a8:bb:e0:33:a4:4e:ca:26:
                    21:2e:43:f1:12:e9:70:ad:37:bc:92:fe:51:f3:f3:
                    40:ee:44:92:bd:cb:06:21:0f:44:3d:68:6b:e8:79:
                    c2:50:ba:64:97
                ASN1 OID: sm2p256v1
                NIST CURVE: SM2
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Key Encipherment
        Netscape Comment:
            GmSSL Generated Certificate
        X509v3 Subject Key Identifier:
            02:1A:F9:AE:4D:B6:BE:7F:EF:10:00:81:A0:C6:0D:D4:53:0F:98:60
        X509v3 Authority Key Identifier:
            keyid:00:BE:61:5B:74:54:CE:AB:EC:89:3C:C8:73:58:14:1D:E2:B9:BB:34

    Signature Algorithm: sm2sign-with-sm3
        30:45:02:21:00:d9:18:de:85:9d:80:6f:23:ae:22:57:b9:18:
        57:e6:57:b8:42:31:35:f4:49:ca:07:8a:d6:09:07:0a:bc:e4:
        5f:02:20:11:e2:19:cb:61:3f:9d:5e:01:42:c5:58:0f:92:31:
        8e:3c:35:e3:f6:65:c9:93:a7:2b:be:14:51:c5:97:d6:36
```

验证证书的合法性：gmssl verify -verbose -x509_strict -CAfile cacert.pem certs/UserA.crt

十、SM2公钥加密算法（用户A加密发送给B）

输入：依次按X.509格式给定cacert.pem,UserB.crt,UserB.key,UserA.key,cipher（用户A发送给用户B的密文）

输出：如果cipher无法通过验证，那么输出ERROR

否则，输出明文。

十一、基于国密算法的PKCS#7

已知：

（1）可信的CA公钥证书

（2）接收者B的私钥

输入：PKCS#7形式的报文

输出：如果报文正确，输入解密后的消息，否则输出ERROR

检查点：

（1）PKCS#7规范和国密算法的使用

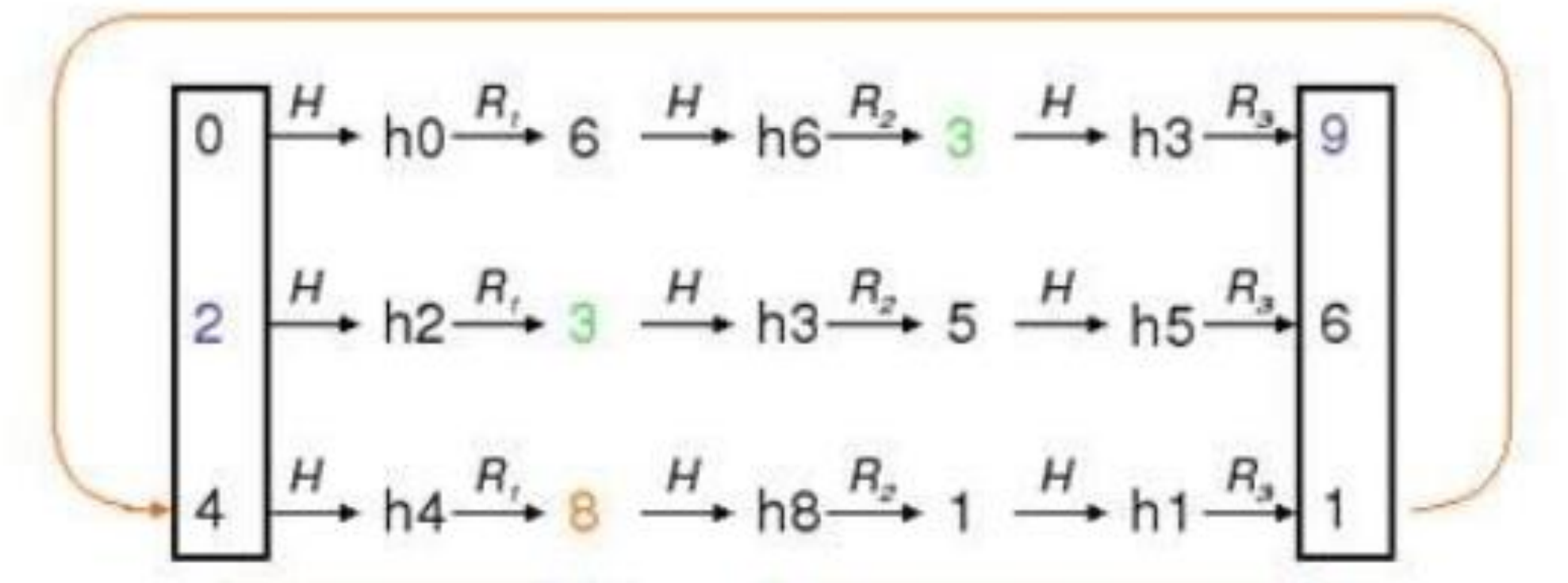
（2）接收者私钥的正确性

（3）发送者身份的合法性，证书链

（4）发送者证书用途（加密/签名）

（5）数据的完整性校验

十二、彩虹表的生成



十二、彩虹表的生成

输入：

(1)a0开头的8位数字+小写字母的彩虹表(SHA1)

(2)m个HASH值

输出：

对于每一个HASH值如果能通过彩虹表找到，输出对应口令，否则输出ERROR

检查点：

（1）如何处理多个R函数

（2）如何快速查找彩虹表