ECE/CS 5560

Lab 4: Public Key Encryption and Signature

Covered Topic

- Public-key cryptography
- The RSA algorithm and key generation
- Big number calculation
- Encryption and Decryption using RSA
- Digital signatures
- X.509 certificates

This lab is part of Assignment 4

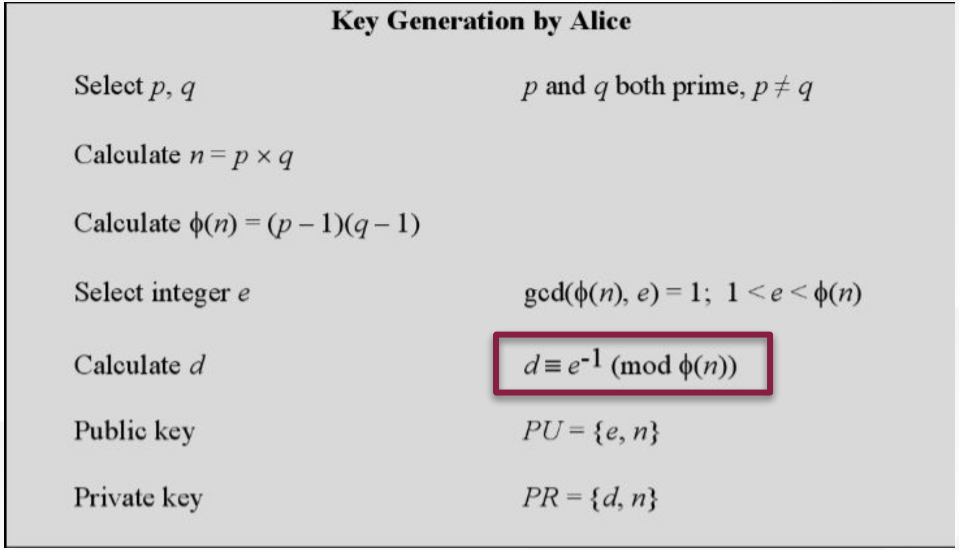
Task 1

- Goal: Deriving the Private Key

```
p:
877120207828103588060123669605304803636762908805750
39025592945358193408249897

q:
102835471351264451708400576484301274347085188629221
996951152314010256656047547

e:
65537

N:
901990700037220684000499639846381245527509884514014
708960288439420664527461699910675170666871590973556
373453958389578171059667963876154490777546489852659
```

Task 1

- Python code for calculate d:
  pow(number**(e)**, power**(-1)**, modulus**(phi)**)



Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$\boxed{d \equiv e^{-1} \pmod{\phi(n)}}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

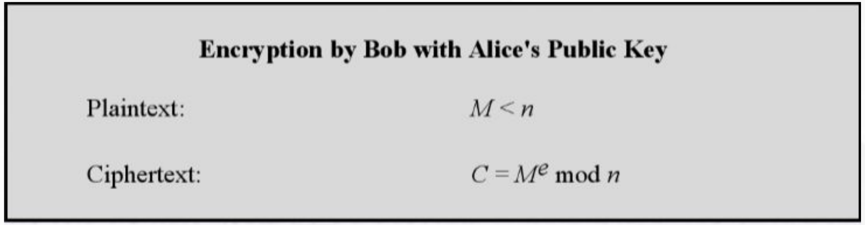
Task 1

- Randomly generate p and q
  number.getPrime(size)
- Run the above code, measure execution time
- Report observation

```
import time

start = time.time()
…
end = time.time()

execution = end - start
```

Task 2

- Goal: Encrypting a Message using public key
- Message: **Hello, this is my first RSA message!**
- Known parameters: **N, e, d, msg (all numbers in HEX)**

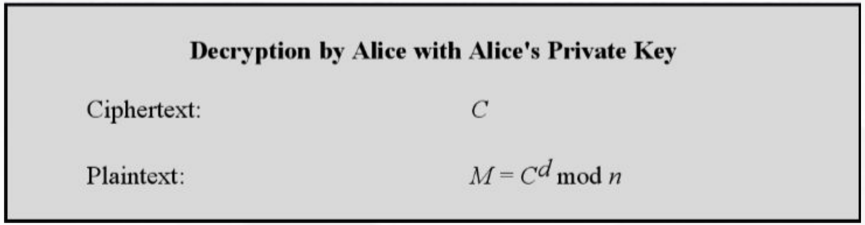


Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Task 3

- Goal: Decrypting a Message
- Known parameters: **N, e, d, ciphertext (all numbers in HEX)**

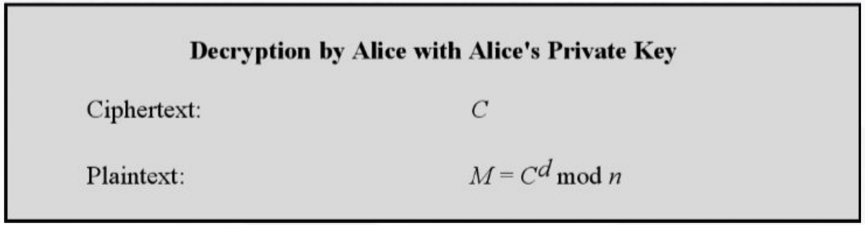


Decryption by Alice with Alice's Private Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Task 4

- Goal: Signing a Message using private key
- Message: **This is a contact for $20,000**
- Same algorithm as Task 3



Decryption by Alice with Alice's Private Key

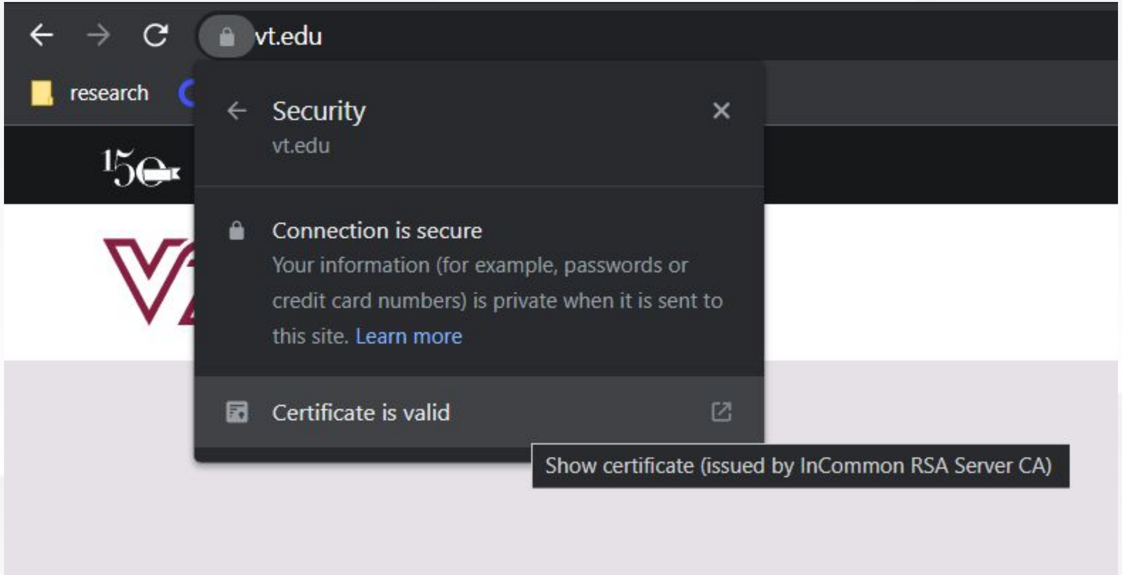
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Task 5

- Goal: Verifying a Message
- Sign a message using private key, decrypt it using public key
- Generate files with size from 1KB, 100KB, 1MB, to 10MB
- Measure the execution time

Task 6 (Extra Credit)

- Using our code to Verify an Web (X.509) Certificate



vt.edu

research

Security

vt.edu

Connection is secure

Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

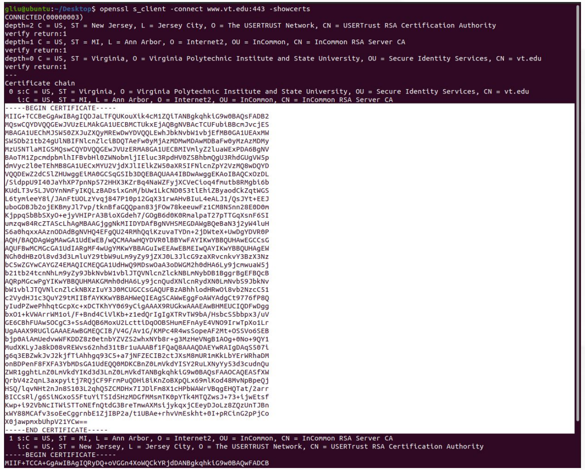
Certificate is valid

Show certificate (issued by InCommon RSA Server CA)

Task 6 (Extra Credit)

- Choose a domain name with multiple certificates: **vt.edu**
- Use openssl to obtain certificates:

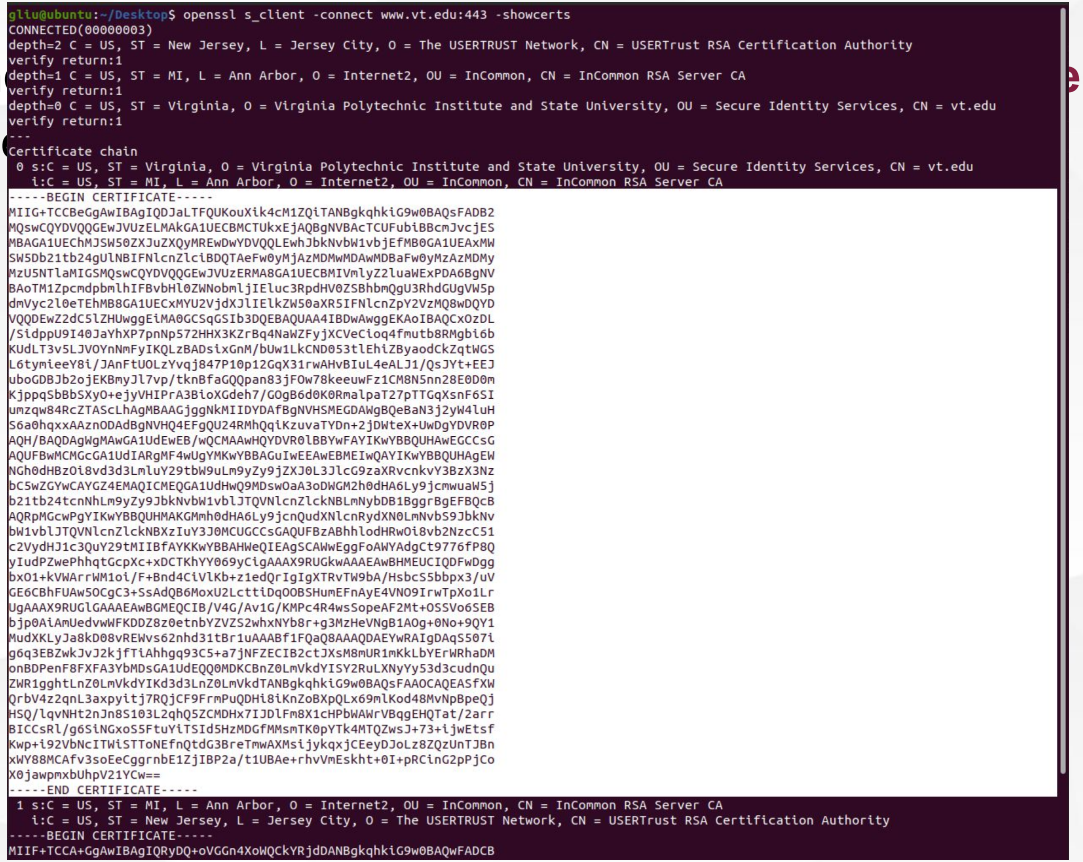
```
$ openssl s_client -connect www.vt.edu:443
-showcerts > openssl_out.txt
```



```
gliu@ubuntu:~/Desktop$ openssl s_client -connect www.vt.edu:443 -showcerts
CONNECTED(00000003)
depth=2 C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority
verify return:1
depth=1 C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
verify return:1
depth=0 C = US, ST = Virginia, O = Virginia Polytechnic Institute and State University, OU = Secure Identity Services, CN = vt.edu
verify return:1
---
Certificate chain
 0 s:C = US, ST = Virginia, O = Virginia Polytechnic Institute and State University, OU = Secure Identity Services, CN = vt.edu
   i:C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
-----BEGIN CERTIFICATE-----
MIIG+TCCBeGgAwIBAgIQDJaLTFQUKouXik4cM1ZQiTANBgkqhkiG9w0BAQsFADB2
MQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUkxEjAQBgNVBAcTCUFubiBBcmJvcjES
MBAGA1UEChMJSW50ZXJuZXQyMREwDwYDVQQLEwhJbkNvbW1vbjEfMB0GA1UEAxMW
SW5Db21tb24gUlNBIFNlcnZlciBDQTAeFw0yMjAzMDMwMDAwMDBaFw0yMzAzMDMy
MzU5NTlaMIGSMQswCQYDVQQGEwJVUzERMA8GA1UECBMIVmlyZ2luaWExPDA6BgNV
BAoTM1ZpcmdpbmlhIFBvbHl0ZWNobmljIEluc3RpdHV0ZSBhbmQgU3RhdGUgVW5p
dmVyc2l0eTEhMB8GA1UECxMYU2VjdXJlIElkZW50aXR5IFNlcnZpY2VzMQ8wDQYD
VQQDEwZ2dC5lZHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCxOzDL
/SidppU9I40JaYhXP7pnNp572HHX3KZrBq4NaWZFyjXCVeCioq4fmutb8RMgbi6b
KUdLT3v5LJVOYnNmFyIKQLzBADsixGnM/bUw1LkCND053tlEhiZByaodCkZqtWGS
L6tymieeY8i/JAnFtUOLzYvqj847P10p12GqX31rwAHvBIuL4eALJ1/QsJYt+EEJ
uboGDBJb2ojEKBmyJl7vp/tknBfaGQQpan83jFOw78keeuwFz1CM8N5nn28E0D0m
KjppqSbBbSXyO+ejyVHIPrA3BioXGdeh7/GOgB6d0K0RmalpaT27pTTGqXsnF6SI
umzqw84RcZTAScLhAgMBAAGjggNkMIIDYDAfBgNVHSMEGDAWgBQeBaN3j2yW4luH
S6a0hqxxAAznODAdBgNVHQ4EFgQU24RMhQqiKzuvaTYDn+2jDWteX+UwDgYDVR0P
AQH/BAQDAgWgMAwGA1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsG
AQUFBwMCMGcGA1UdIARgMF4wUgYMKwYBBAGuIwEEAwEBMEIwQAYIKwYBBQUHAgEW
NGh0dHBzOi8vd3d3LmluY29tbW9uLm9yZy9jZXJ0L3JlcG9zaXRvcnkvY3BzX3Nz
bC5wZGYwCAYGZ4EMAQICMEQGA1UdHwQ9MDswOaA3oDWGM2h0dHA6Ly9jcmwuaW5j
b21tb24tcnNhLm9yZy9JbkNvbW1vblJTQVNlcnZlckNBLmNybDB1BggrBgEFBQcB
AQRpMGcwPgYIKwYBBQUHMAKGMmh0dHA6Ly9jcnQudXNlcnRydXN0LmNvbS9JbkNv
bW1vblJTQVNlcnZlckNBXzIuY3J0MCUGCCsGAQUFBzABhhlodHRwOi8vb2NzcC51
c2VydHJ1c3QuY29tMIIBfAYKKwYBBAHWeQIEAgSCAWwEggFoAWYAdgCt9776fP8Q
yIudPZwePhhqtGcpXc+xDCTKhYY069yCigAAAX9RUGkwAAAEAwBHMEUCIQDFwDgg
bxO1+kVWArrWM1oi/F+Bnd4CiVlKb+z1edQrIgIgXTRvTW9bA/HsbcS5bbpx3/uV
GE6CBhFUAw5OCgC3+SsAdQB6MoxU2LcttiDqOOBSHumEFnAyE4VNO9IrwTpXo1Lr
UgAAAX9RUGlGAAAEAwBGMEQCIB/V4G/Av1G/KMPc4R4wsSopeAF2Mt+OSSVo6SEB
bjp0AiAmUedvwWFKDDZ8z0etnbYZVZS2whxNYb8r+g3MzHeVNgB1AOg+0No+9QY1
MudXKLyJa8kD08vREWvs62nhd31tBr1uAAABf1FQaQ8AAAQDAEYwRAIgDAqS507i
g6q3EBZwkJvJ2kjfTiAhhgq93C5+a7jNFZECIB2ctJXsM8mUR1mKkLbYErWRhaDM
onBDPenF8FXFA3YbMDsGA1UdEQQ0MDKCBnZ0LmVkdYISY2RuLXNyYy53d3cudnQu
ZWR1gghtLnZ0LmVkdYIKd3d3LnZ0LmVkdTANBgkqhkiG9w0BAQsFAAOCAQEASfXW
QrbV4z2qnL3axpyitj7RQjCF9FrmPuQDHi8iKnZoBXpQLx69mlKod48MvNpBpeQj
HSQ/lqvNHt2nJn8S103L2qhQ5ZCMDHx7IJDlFm8X1cHPbWAWrVBqgEHQTat/2arr
BICCsRl/g6SiNGxoS5FtuYiTSId5HzMDGfMMsmTK0pYTk4MTQZwsJ+73+ijwEtsf
Kwp+i92VbNcITWiSTToNEfnQtdG3BreTmwAXMsijykqxjCEeyDJoLz8ZQzUnTJBn
xWY88MCAfv3soEeCggrnbE1ZjIBP2a/t1UBAe+rhvVmEskht+0I+pRCinG2pPjCo
X0jawpmxbUhpV21YCw==
-----END CERTIFICATE-----
 1 s:C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
   i:C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority
-----BEGIN CERTIFICATE-----
MIIF+TCCA+GgAwIBAgIQRyDQ+oVGGn4XoWQCkYRjdDANBgkqhkiG9w0BAQwFADCB
```

Task 6 (Extra Credit)

- Ch edu
- Us



```
gliu@ubuntu:~/Desktop$ openssl s_client -connect www.vt.edu:443 -showcerts
CONNECTED(00000003)
depth=2 C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority
verify return:1
depth=1 C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
verify return:1
depth=0 C = US, ST = Virginia, O = Virginia Polytechnic Institute and State University, OU = Secure Identity Services, CN = vt.edu
verify return:1
---
Certificate chain
 0 s:C = US, ST = Virginia, O = Virginia Polytechnic Institute and State University, OU = Secure Identity Services, CN = vt.edu
   i:C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
-----BEGIN CERTIFICATE-----
MIIG+TCCBeGgAwIBAgIQDJaLTFQUKouXik4cM1ZQiTANBgkqhkiG9w0BAQsFADB2
MQswCQYDVQQGEwJVUzELMAkGA1UECBMCTUkxEjAQBgNVBAcTCUFubiBBcmJvcjES
MBAGA1UEChMJSW50ZXJuZXQyMREwDwYDVQQLEwhJbkNvbW1vbjEfMB0GA1UEAxMW
SW5Db21tb24gUlNBIFNlcnZlciBDQTAeFw0yMjAzMDMwMDAwMDBaFw0yMzAzMDMy
MzU5NTlaMIGSMQswCQYDVQQGEwJVUzERMA8GA1UECBMIVmlyZ2luaWExPDA6BgNV
BAoTM1ZpcmdpbmlhIFBvbHl0ZWNobmljIEluc3RpdHV0ZSBhbmQgU3RhdGUgVW5p
dmVyc2l0eTEhMB8GA1UECxMYU2VjdXJlIElkZW50aXR5IFNlcnZpY2VzMQ8wDQYD
VQQDEwZ2dC5lZHUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCxOzDL
/SidppU9I40JaYhXP7pnNp572HHX3KZrBq4NaWZFyjXCVeCioq4fmutb8RMgbi6b
KUdLT3v5LJVOYnNmFyIKQLzBADsixGnM/bUw1LkCND053tlEhiZByaodCkZqtWGS
L6tymieeY8i/JAnFtUOLzYvqj847P10p12GqX31rwAHvBIuL4eALJ1/QsJYt+EEJ
uboGDBJb2ojEKBmyJl7vp/tknBfaGQQpan83jFOw78keeuwFz1CM8N5nn28E0D0m
KjppqSbBbSXyO+ejyVHIPrA3BioXGdeh7/GOgB6d0K0RmalpaT27pTTGqXsnF6SI
umzqw84RcZTAScLhAgMBAAGjggNkMIIDYDAfBgNVHSMEGDAWgBQeBaN3j2yW4luH
S6a0hqxxAAznODAdBgNVHQ4EFgQU24RMhQqiKzuvaTYDn+2jDWteX+UwDgYDVR0P
AQH/BAQDAgWgMAwGA1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsG
AQUFBwMCMGcGA1UdIARgMF4wUgYMKwYBBAGuIwEEAwEBMEIwQAYIKwYBBQUHAgEW
NGh0dHBzOi8vd3d3LmluY29tbW9uLm9yZy9jZXJ0L3JlcG9zaXRvcnkvY3BzX3Nz
bC5wZGYwCAYGZ4EMAQICMEQGA1UdHwQ9MDswOaA3oDWGM2h0dHA6Ly9jcmwuaW5j
b21tb24tcnNhLm9yZy9JbkNvbW1vblJTQVNlcnZlckNBLmNybDB1BggrBgEFBQcB
AQRpMGcwPgYIKwYBBQUHMAKGMmh0dHA6Ly9jcnQudXNlcnRydXN0LmNvbS9JbkNv
bW1vblJTQVNlcnZlckNBXzIuY3J0MCUGCCsGAQUFBzABhhlodHRwOi8vb2NzcC51
c2VydHJ1c3QuY29tMIIBfAYKKwYBBAHWeQIEAgSCAWwEggFoAWYAdgCt9776fP8Q
yIudPZwePhhqtGcpXc+xDCTKhYY069yCigAAAX9RUGkwAAAEAwBHMEUCIQDFwDgg
bxO1+kVWArrWM1oi/F+Bnd4CiVlKb+z1edQrIgIgXTRvTW9bA/HsbcS5bbpx3/uV
GE6CBhFUAw5OCgC3+SsAdQB6MoxU2LcttiDqOOBSHumEFnAyE4VNO9IrwTpXo1Lr
UgAAAX9RUGlGAAAEAwBGMEQCIB/V4G/Av1G/KMPc4R4wsSopeAF2Mt+OSSVo6SEB
bjp0AiAmUedvwWFKDDZ8z0etnbYZVZS2whxNYb8r+g3MzHeVNgB1AOg+0No+9QY1
MudXKLyJa8kD08vREWvs62nhd31tBr1uAAABf1FQaQ8AAAQDAEYwRAIgDAqS507i
g6q3EBZwkJvJ2kjfTiAhhgq93C5+a7jNFZECIB2ctJXsM8mUR1mKkLbYErWRhaDM
onBDPenF8FXFA3YbMDsGA1UdEQQ0MDKCBnZ0LmVkdYISY2RuLXNyYy53d3cudnQu
ZWR1gghtLnZ0LmVkdYIKd3d3LnZ0LmVkdTANBgkqhkiG9w0BAQsFAAOCAQEASfXW
QrbV4z2qnL3axpyitj7RQjCF9FrmPuQDHi8iKnZoBXpQLx69mlKod48MvNpBpeQj
HSQ/lqvNHt2nJn8S103L2qhQ5ZCMDHx7IJDlFm8X1cHPbWAWrVBqgEHQTat/2arr
BICCsRl/g6SiNGxoS5FtuYiTSId5HzMDGfMMsmTK0pYTk4MTQZwsJ+73+ijwEtsf
Kwp+i92VbNcITWiSTToNEfnQtdG3BreTmwAXMsijykqxjCEeyDJoLz8ZQzUnTJBn
xWY88MCAfv3soEeCggrnbE1ZjIBP2a/t1UBAe+rhvVmEskht+0I+pRCinG2pPjCo
X0jawpmxbUhpV21YCw==
-----END CERTIFICATE-----
 1 s:C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, CN = InCommon RSA Server CA
   i:C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority
-----BEGIN CERTIFICATE-----
MIIF+TCCA+GgAwIBAgIQRyDQ+oVGGn4XoWQCkYRjdDANBgkqhkiG9w0BAQwFADCB
```

Task 6 (Extra Credit)

- Extract public key (**issuer: c1**):

For modulus (n):

```
$ openssl x509 -in c1.pem -noout -modulus
```

Print out all the fields, find the exponent (e):

```
$ openssl x509 -in c1.pem -text -noout
```

- Extract signature (**server: c0**):

```
$ openssl x509 -in c0.pem -text -noout
```

(look for Signature Algorithm)

```
$ cat signature | tr -d ’[:space:]:’
```

Task 6 (Extra Credit)

- Extract the body of the certificate

```
$ openssl asn1parse -i -in c0.pem -strparse 4 -out c0_body.bin  -noout
$ sha256sum c0_body.bin
```

- Verify the signature
  - Decrypt signature using public key (e,n), **code in previous tasks**
  - Check if the decrypted value is partially the same as sha256sum of the certificate body

Questions?

Yousef Akbar

akbary@vt.edu

Office Hours: Mon. 1:30 - 3:30pm, Thu. 1:30 - 3:30pm

Zoom: https://virginiatech.zoom.us/j/6931202457