Lab Assignment 6

AP21110010302

Krish Srivastava

CSE – E
Network Security – CSE 315L

1.

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
// import java.util.Scanner;
// import javax.xml.bind.DatatypeConverter;

public class genandverifyDSwithRSAandSHA256
{
    private static final String
    SIGNING_ALGORITHM = "SHA256withRSA";
    private static final String RSA = "RSA";
    // private static Scanner sc;
    public static byte[] createDigitalSignature(byte[] input, PrivateKey Key)
throws Exception  {
        Signature sig = Signature.getInstance(SIGNING_ALGORITHM);
        sig.initSign(Key);
        sig.update(input);
        return sig.sign();
    }

    public static KeyPair generateRSAKeyPair() throws Exception  {
        SecureRandom sr = new SecureRandom();
        KeyPairGenerator kpg = KeyPairGenerator.getInstance(RSA);
        kpg.initialize(2048, sr);
        return kpg.generateKeyPair();
    }
    public static boolean verifyDigitalSignature(byte[] input, byte[]
signatureToVerify, PublicKey key) throws Exception  {
        Signature sig = Signature.getInstance(SIGNING_ALGORITHM);
        sig.initVerify(key);
        sig.update(input);
        return sig.verify(signatureToVerify);
```

```
    }

    public static void main(String args[]) throws Exception  {
        String input = "Java is an" + "object-oriented language";
        KeyPair keyPair = generateRSAKeyPair();
        byte[] sig = createDigitalSignature(input.getBytes(),
keyPair.getPrivate());
        System.out.println("Signature Value:\n " + sig);
        System.out.println("Verification: "+
verifyDigitalSignature(input.getBytes(), sig, keyPair.getPublic()));
    }
}
```

As string input is already defined in the code, the corresponding output will be:

```
PS C:\Users\krish>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsIn
a-project\bin' 'genandverifyDSwithRSAandSHA256'
Signature Value:
 [B@3d71d552
Verification: true
PS C:\Users\krish>
```

2

```
┌─[retr0@parrot]─[~]
└──╼ $wget https://github.com/OpenVPN/easy-rsa/archive/3.0.1.tar.gz
--2024-03-31 22:45:40--  https://github.com/OpenVPN/easy-rsa/archive/3.0.1.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/OpenVPN/easy-rsa/tar.gz/refs/tags/3.0.1 [following]
--2024-03-31 22:45:40--  https://codeload.github.com/OpenVPN/easy-rsa/tar.gz/refs/tags/3.0.1
Resolving codeload.github.com (codeload.github.com)... 20.207.73.88
Connecting to codeload.github.com (codeload.github.com)|20.207.73.88|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: '3.0.1.tar.gz'

3.0.1.tar.gz            [  <=>               ]  43.21K   126KB/s    in 0.3s

2024-03-31 22:45:41 (126 KB/s) - '3.0.1.tar.gz' saved [44242]
```

```
┌─[retr0@parrot]─[~]
└──╼ $tar xzvf 3.0.1.tar.gz
easy-rsa-3.0.1/
easy-rsa-3.0.1/COPYING
easy-rsa-3.0.1/ChangeLog
easy-rsa-3.0.1/KNOWN_ISSUES
easy-rsa-3.0.1/Licensing/
easy-rsa-3.0.1/Licensing/gpl-2.0.txt
easy-rsa-3.0.1/README
easy-rsa-3.0.1/README.quickstart.md
easy-rsa-3.0.1/build/
easy-rsa-3.0.1/build/Building.md
easy-rsa-3.0.1/build/build-dist.sh
easy-rsa-3.0.1/distro/
easy-rsa-3.0.1/distro/README
easy-rsa-3.0.1/distro/windows/
easy-rsa-3.0.1/distro/windows/EasyRSA-Start.bat
easy-rsa-3.0.1/distro/windows/Licensing/
easy-rsa-3.0.1/distro/windows/Licensing/mksh-Win32.txt
easy-rsa-3.0.1/distro/windows/README-Windows.txt
easy-rsa-3.0.1/distro/windows/bin/
easy-rsa-3.0.1/distro/windows/bin/easyrsa-shell-init.sh
easy-rsa-3.0.1/doc/
easy-rsa-3.0.1/doc/EasyRSA-Advanced.md
easy-rsa-3.0.1/doc/EasyRSA-Readme.md
easy-rsa-3.0.1/doc/EasyRSA-Upgrade-Notes.md
easy-rsa-3.0.1/doc/Hacking.md
easy-rsa-3.0.1/doc/Intro-To-PKI.md
easy-rsa-3.0.1/doc/TODO
easy-rsa-3.0.1/easyrsa3/
```

```
root@KRISHSRIVASTAVA:~/easy-rsa# ./easyrsa init-pki


WARNING!!!

You are about to remove the EASYRSA_PKI at: /root/easy-rsa/pki
and initialize a fresh PKI here.

Type the word 'yes' to continue, or any other input to abort.
  Confirm removal: yes

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /root/easy-rsa/pki


root@KRISHSRIVASTAVA:~/easy-rsa#
```

```
root@KRISHSRIVASTAVA:~/easy-rsa# ./easyrsa build-ca
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:krishhostname

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/root/easy-rsa/pki/ca.crt
```

```
root@KRISHSRIVASTAVA:~/easy-rsa# cd pki
root@KRISHSRIVASTAVA:~/easy-rsa/pki# dir
ca.crt           index.txt       issued               private  reqs     safessl-easyrsa.cnf
certs_by_serial  index.txt.attr  openssl-easyrsa.cnf  renewed  revoked  serial
root@KRISHSRIVASTAVA:~/easy-rsa/pki# cat ca.crt
-----BEGIN CERTIFICATE-----
MIIDUTCCAjmgAwIBAgIUPJXzZ93MMzRyvJ6Dw7CWBOZDqZQwDQYJKoZIhvcNAQEL
BQAwGDEWMBQGA1UEAwwNa3Jpc2hob3N0bmFtZTAeFw0yNDAzMzExNzI2NDhaFw0z
NDAzMjkxNzI2NDhaMBgxFjAUBgNVBAMMDWtyaXNoaG9zdG5hbWUwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDlJ5Sx4qHpfTce47SE6mCJbNutUASp5lPv
dsxQztgN3EiRvmrGakdtST8ohpPNg9eF04ruayFb0aVzVP2pTqOtHiEoDkrARaSc
3HBfIcbIQWe5wLU8Rh6jlcVvwgyMzIHggXQLErzgy5WACPjuoGvsllJZnB2D44Yi
UC/nKxncfOmnn53wVoj4uKAjAtcqiz9VKVWtcx+AFkR2DbfmJoWADu2iIzefTKX3
G8fKiitWfFMC0HP77kCy9GWQjSW429oRuT26vhOSwedfSK65TPqSEajNNAThsfSD
9XO0f4Km/oojjaUgCbpWhfrkNQshIvamrcOzgKS+VKEQ+MlbT39XAgMBAAGjgZIw
gY8wHQYDVR0OBBYEFPFYUBXgWbUF3gDXtlNSmmPgY6KcMFMGA1UdIwRMMEqAFPFY
UBXgWbUF3gDXtlNSmmPgY6KcoRykGjAYMRYwFAYDVQQDDA1rcmlzaGhvc3RuYW1l
ghQ8lfNn3cwzNHK8noPDsJYE5kOplDAMBgNVHRMEBTADAQH/MAsGA1UdDwQEAwIB
BjANBgkqhkiG9w0BAQsFAAOCAQEAYVHT0yPhWxsZzaqjy3CLREcsJV2OnEWRVHIV
QddK7xH4myR/Ck2TOhEZrn94tg10sik3Wb8zhTc4iwJE+7NRG2ZXDUooZQpW5fio
tCS4WbAfAtrWyXa7grjZclLgHXP2v27X95kAAhq4ChahdjL5IuC95Y+b0jgV2/BM
COAXhjjVWxZauSWuaf+ji+pANQ2pd4MxZg2tYu79QO/mD54IxR1go6cha6jiJB+Z
kmbY8lXd2oUF0HxVEPOXY1yV50iSOo8lhvrowgLq7EeSYVGiksShcIJfJiHD0F7N
rC8pZ+tlNLiR58hR1zF2tHP9eKB36W5nRy6gw1nNNyhILI3Qiw==
-----END CERTIFICATE-----
root@KRISHSRIVASTAVA:~/easy-rsa/pki#
```

```
root@KRISHSRIVASTAVA:~/easy-rsa# ./easyrsa gen-req SRMkabachha
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
...+....+.....+.+........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.
.+.+..+............+...++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*...
........+...+.....+....+........+...+.......++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++
....+....+..+....+..............+..+...+....+..+...+...+..........++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++*...+..+....+.........+........+....+.....+...+.....
..+......+...+..+.........+...+.+...................++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++*.+...+.............+......+...+......+..+.........+++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [SRMkabachha]:personalCA'

Keypair and certificate request completed. Your files are:
req: /root/easy-rsa/pki/reqs/SRMkabachha.req
key: /root/easy-rsa/pki/private/SRMkabachha.key
```

```
root@KRISHSRIVASTAVA:~/easy-rsa# ./easyrsa sign-req client SRMkabachha
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 825 days:

subject=
    commonName                = personalCA'


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /root/easy-rsa/pki/easy-rsa-118.8NKeF9/tmp.1mcIBn
Enter pass phrase for /root/easy-rsa/pki/private/ca.key:
40B7013D117F0000:error:0700006C:configuration file routines:NCONF_get_string:no value:../cry
pto/conf/conf_lib.c:315:group=<NULL> name=unique_subject
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'personalCA''
Certificate is to be certified until Jul  4 17:32:19 2026 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /root/easy-rsa/pki/issued/SRMkabachha.crt
```

```
root@KRISHSRIVASTAVA:~/easy-rsa# ./easyrsa revoke SRMkabachha
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)


Please confirm you wish to revoke the certificate with the following subject:

subject=
    commonName                = personalCA'


Type the word 'yes' to continue, or any other input to abort.
  Continue with revocation: yes
Using configuration from /root/easy-rsa/pki/easy-rsa-161.tBtKow/tmp.L4Tcpb
Enter pass phrase for /root/easy-rsa/pki/private/ca.key:
Revoking Certificate FD4D4590C440E04D7A13C611624F3B32.
Data Base Updated

IMPORTANT!!!

Revocation was successful. You must run gen-crl and upload a CRL to your
infrastructure in order to prevent the revoked cert from being accepted.
```