Using OpenSSL Tools to Conduct RSA Operations

This Video Covers:

- Generating RSA keys
- Extracting the public key
- Encryption and decryption

OpenSSL Tools: Generating RSA Keys

Example: generate a 1024-bit public/private key pair

- Use **`openssl genrsa`** to generate a file, **`private.pem`**
- private.pem is a Base64 encoding of DER generated binary output

OpenSSL Tools: Generating RSA Keys

Example: generate a 1024-bit public/private key pair

- Use **`openssl genrsa`** to generate a file, **`private.pem`**
- private.pem is a Base64 encoding of DER generated binary output

```
$ openssl genrsa -aes128 -out private.pem 1024 # passphrase csci476
```

OpenSSL Tools: Generating RSA Keys

Example: generate a 1024-bit public/private key pair

- Use **`openssl genrsa`** to generate a file, **`private.pem`**
- private.pem is a Base64 encoding of DER generated binary output

```
$ openssl genrsa -aes128 -out private.pem 1024 # passphrase csci476
$ more private.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,C30BF6EB3FD6BA9A81CCB9202B95EC1A

sLIQ7Fs5j5zOexdWkZUoiv2W82g03gNERmfG+fwnVnbsIZAuW8E9wiB7tqz8rEL+
xfL+U20lyQNxpmOTUeKlN3qCcJROcGYSNd1BeNpgLWV1bN5FPYce9GRb4tFr4bhK
...
RPtJNKUryhVnAC4a3gp0gcXk1IQLeHeyKQCPQ1SckQRdrBzHjjCNN42NlCVEpcsF
WJ8ikqDd9FslGHc1PT6ktW5oV9cB8G2wfo7D85n91SQfSzuwAcyx7Ecir1o4PfKG
-----END RSA PRIVATE KEY——
```

OpenSSL Tools: Generating RSA Keys *(cont.)*

The ***actual*** content of **`private.pem`**:

```
$ openssl rsa -in private.pem -noout -text
```

OpenSSL Tools: Generating RSA Keys *(cont.)*

The ***actual*** content of **`private.pem`**:

```
$ openssl rsa -in private.pem -noout -text
Enter pass phrase for private.pem: csci476
Private-Key: (1024 bit)
modulus:
    00:b8:52:5c:25:cc:7c:f2:ef:a6:35:9d:de:3d:5d: ...
publicExponent: 65537 (0x10001)
privateExponent:
    4b:0d:ce:53:dd:e6:6b:0d:c6:82:42:9c:42:24:a7: ...
prime1:
    00:ef:14:46:57:9c:d0:4c:98:de:c3:0b:aa:d8:72: ...
prime2:
    00:c5:5d:f8:0b:f9:75:dc:88:ea:d4:d0:56:ee:f9: ...
exponent1:
    00:e6:49:9a:44:14:19:94:5e:7f:dc:52:65:bb:5d: ...
exponent2:
    7c:ad:77:dc:58:a2:13:c6:8a:52:15:aa:55:1c:22: ...
coefficient:
    3a:7c:b9:a0:12:e8:fa:88:b8:6f:38:4a:ed:bc:17: ...
```

OpenSSL Tools: Generating RSA Keys *(cont.)*

The ***actual*** content of **`private.pem`**:

```
$ openssl rsa -in private.pem -noout -text
Enter pass phrase for private.pem: csci476
Private-Key: (1024 bit)
modulus:
    00:b8:52:5c:25:cc:7c:f2:ef:a6:35:9d:de:3d:5d: ...
publicExponent: 65537 (0x10001)
privateExponent:
    4b:0d:ce:53:dd:e6:6b:0d:c6:82:42:9c:42:24:a7: ...
prime1:
    00:ef:14:46:57:9c:d0:4c:98:de:c3:0b:aa:d8:72: ...
prime2:
    00:c5:5d:f8:0b:f9:75:dc:88:ea:d4:d0:56:ee:f9: ...
exponent1:
    00:e6:49:9a:44:14:19:94:5e:7f:dc:52:65:bb:5d: ...
exponent2:
    7c:ad:77:dc:58:a2:13:c6:8a:52:15:aa:55:1c:22: ...
coefficient:
    3a:7c:b9:a0:12:e8:fa:88:b8:6f:38:4a:ed:bc:17: ...
```

n

e

d

p

q

Chinese remainder theorem

OpenSSL Tools: Extracting the Public Key

The ***actual*** content of **`public.pem`**:

```
$ openssl rsa -in private.pem -pubout > public.pem
Enter pass phrase for private.pem: csci476
writing RSA key
$ more public.pem
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4UlwlzHzy76Y1nd49XakNUwqJ
Ud3ph0uBWWfnLnjIYgQL/spg9WE+1Q1YPp2t3FBFljhGHdWMA8abfNXG4jmpD+uq
Ix0WVyXgl2WWi1kY2/vs8xI1K+PumWTtq8R8ueAq7RzETc3873DO1vjMxXWqau7k
zIkUuJ/JCjzjYfbsDQIDAQAB
-----END PUBLIC KEY-----
```

OpenSSL Tools: Extracting the Public Key

The ***actual*** content of **`public.pem`**:

```
$ openssl rsa -in private.pem -pubout > public.pem
Enter pass phrase for private.pem: csci476
writing RSA key
$ more public.pem
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4UlwlzHzy76Y1nd49XakNUwqJ
Ud3ph0uBWWfnLnjIYgQL/spg9WE+1Q1YPp2t3FBFljhGHdWMA8abfNXG4jmpD+uq
Ix0WVyXg12WWi1kY2/vs8xI1K+PumWTtq8R8ueAq7RzETc3873DO1vjMxXWqau7k
zIkUuJ/JCjzjYfbsDQIDAQAB
-----END PUBLIC KEY-----
```

```
$ openssl rsa -in public.pem -pubin -text -noout
Public-Key: (1024 bit)
Modulus:
    00:b8:52:5c:25:cc:7c:f2:ef:a6:35:9d:de:3d:5d: ...
Exponent: 65537 (0x10001)
```

n

e

↪ (e,n) = public key!

OpenSSL Tools: Encryption and Decryption

- Create a plaintext message:

```
$ echo "This is a secret." > msg.txt
```

- Encrypt the plaintext:

```
$ openssl rsautl -encrypt -inkey public.pem -pubin -in msg.txt -out msg.enc
```

OpenSSL Tools: Encryption and Decryption

- Create a plaintext message:

```
$ echo "This is a secret." > msg.txt
```

- Encrypt the plaintext:

```
$ openssl rsautl -encrypt -inkey public.pem -pubin -in msg.txt -out msg.enc
```

- Decrypt the ciphertext:

```
$ openssl rsautl -decrypt -inkey private.pem -in msg.enc
Enter pass phrase for private.pem: csci476
This is a secret.
```