

Ex.No.11  
29/09/2024

## ANALYSIS OF SECURITY PROTOCOLS AT TRANSPORT AND APPLICATION LAYER

### AIM:

To study the working principle of security protocols like SSL/TLS using Wireshark and PGP using chrome plugin – Mailvelope.

### THEORY:

#### Wireshark:

Wireshark is a free and open-source packet analyser. Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.

#### SSL/TLS:

1. SSL: - Secure Socket Layer (SSL) is a cryptographic protocol designed to provide secure communication over a computer network. It was developed by Netscape in the 1990s to establish an encrypted link between the web server and a web browser. SSL operates by using encryption to secure the transmission of data ensuring that sensitive information such as credit card details and personal data remains confidential.
2. TLS: - The Transport Layer Security (TLS) is the successor to SSL and is designed to provide improved security and efficiency. TLS was developed as an enhancement of SSL to address various vulnerabilities and to incorporate modern cryptographic techniques. The first version, TLS 1.0 was based on SSL 3.0 but included significant improvements. TLS continues to evolve with the newer versions offering enhanced security features.

#### PGP:

Pretty Good Privacy (PGP) is an encryption program that provides cryptographic privacy and authentication for data communication. PGP is used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. Phil Zimmermann developed PGP in 1991.

## WORKSHEET (using sample.pcap)

1. What version of SSL is supported by the client?

The version of SSL Supported by client is **SSL 2.0 (0x0002)**

2. List the cryptographic algorithms supported by the client in Client Hello message.

```

▼ Cipher Suites (20 suites)
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
  Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)

```

3. List the various parameters present in the public key certificate of the server.

```

Certificate      a814d7366b78535271ccf60d293e60004b54c40bb16ce5a1d3f9174397
                  82fc06

```

4. Identify the public key of the server? Can you trust the same?

```

Public Key      4e6feeee7e3558966013f9e64027f156218b5c918f1ebe00742a0e9041cf
                  e233

```

Yes, we can trust this because it has been issued by a certificate authority.

5. Identify the length of key exchanged by the Client?

The length of key exchanged by client is 128bits.

```

Handshake Protocol: Server Key Exchange
Handshake Type: Server Key Exchange (12)
Length: 361
▼ EC Diffie-Hellman Server Params
  Curve Type: named_curve (0x03)
  Named Curve: secp384r1 (0x0018)
  Pubkey Length: 97
  Pubkey: 04fd3897955947a7b28a8aa87f78183838384ae32c96a5e6e1063
  Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
  Signature Length: 256
  Signature [...]: 60640f9c4634e092a752f6b4eada6cea6dadeb07abf7c5

```

6. What algorithm is used for encrypting the session key?

```

Random: 670272ee922eda5c418995282fe9c94509abac2bd67c0ce7c589c62e6238890d
Session ID Length: 32
Session ID: 730f000048f7af7b630d400b93a3e72b36057e342b59379b31e240e1d6999755

```

7. List the various parameters specified in the Encrypted handshake message.

```
TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 1300
Encrypted Application Data [...]: ef751b68ef1ba1465140f4003f56f6f2dc2d81e6ee5
[Application Data Protocol: Hypertext Transfer Protocol]
```

8. Calculate the time taken for completion of the entire handshake protocol.

The time taken for completion of the entire handshake protocol is 2.808775 seconds

9. Examine the certificates of gmail server, any bank server. Identify the Certification Authority, Hierarchy of Certification Authority, Validity date, crypto algorithms used for signing etc.

```

Certificates (829 bytes)
Certificate Length: 826
Certificate [...]: 308203363082029fa003020102020101300d06092a864886f70d01010405003081a9310b3009060355040613025859311530130603550408130c536e616b6520446573657274...
  signedCertificate
    version: v3 (2)
    serialNumber: 0x01
    signature (md5WithRSAEncryption)
      Algorithm Id: 1.2.840.113549.1.1.4 (md5WithRSAEncryption)
    issuer: rdnSequence (0)
      [...]rdnSequence: 7 items (pkcs-9-at-emailAddress=ca@snakeoil.dom,id-at-commonName=Snake Oil CA,id-at-organizationalUnitName=Certificate Authority,id-at-...
        RDNSSequence item: 1 item (id-at-countryName=XY)
        RDNSSequence item: 1 item (id-at-stateOrProvinceName=Snake Desert)
        RDNSSequence item: 1 item (id-at-localityName=Snake Town)
        RDNSSequence item: 1 item (id-at-organizationName=Snake Oil, Ltd)
        RDNSSequence item: 1 item (id-at-organizationalUnitName=Certificate Authority)
        RDNSSequence item: 1 item (id-at-commonName=Snake Oil CA)
        RDNSSequence item: 1 item (pkcs-9-at-emailAddress=ca@snakeoil.dom)
    validity
      notBefore: utcTime (0)
      notAfter: utcTime (0)
    subject: rdnSequence (0)
      [...]rdnSequence: 7 items (pkcs-9-at-emailAddress=www@snakeoil.dom,id-at-commonName=www.snakeoil.dom,id-at-organizationalUnitName=Webserver Team,id-at-...
        RDNSSequence item: 1 item (id-at-countryName=XY)
        RDNSSequence item: 1 item (id-at-stateOrProvinceName=Snake Desert)
        RDNSSequence item: 1 item (id-at-localityName=Snake Town)
        RDNSSequence item: 1 item (id-at-organizationName=Snake Oil, Ltd)
        RDNSSequence item: 1 item (id-at-organizationalUnitName=Webserver Team)
        RDNSSequence item: 1 item (id-at-commonName=www.snakeoil.dom)
        RDNSSequence item: 1 item (pkcs-9-at-emailAddress=www@snakeoil.dom)

```

10. Enlist the differences between HTTP and HTTPS.

HTTP	HTTPS
HTTP uses port number 80 for communication.	HTTPS uses 443 port number for communication.
In HTTP, encryption is absent.	In HTTPS encryption is present.
HTTP does not require any certificates.	HTTPS requires SSL certificate.
HTTP works at Application Layer.	HTTPS works at Transport as well as Application Layer.

## WORKSHEET (For packet captured for real time data)

1. What version of TLS is supported by the client?

Client Supports TLS Version 1.0 (0x0301)

```

Transport Layer Security
  TLSv1.3 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 1780
    Handshake Protocol: Client Hello

```

2. List the cryptographic algorithms supported by the client in Client Hello message.

```

Cipher Suites (16 suites)
  Cipher Suite: Reserved (GREASE) (0x8a8a)
  Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
  Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
  Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
  Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
  Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)

```

3. List the various parameters present in the public key certificate of the server.

```

Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 4052
    Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 4048
      Certificates Length: 4045
      Certificates (4045 bytes)
        Certificate Length: 1778
        Certificate [...]
          signedCertificate
          algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted [...]
          Certificate Length: 1344
        Certificate [...]
          signedCertificate
          algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted [...]
          Certificate Length: 914
        Certificate [...]
          signedCertificate
          algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted [...]

```

4. Identify the public key of the server? Can you trust the same?

```

Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 333
    Handshake Protocol: Server Key Exchange
      Handshake Type: Server Key Exchange (12)
      Length: 329
      EC Diffie-Hellman Server Params
        Curve Type: named_curve (0x03)
        Named Curve: secp256r1 (0x0017)
        Pubkey Length: 65
        Pubkey: 0465d0c431465b7ca0a1a1009e33f3744aad6e7435c7227e896fea2d110164991641686cd92f25d70e99e3fce487e0bb7a8c31b3ca7242fb35f4a8045cde80ea
      Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
        Signature Hash Algorithm Hash: Unknown (8)
        Signature Hash Algorithm Signature: Unknown (4)
        Signature Length: 256
        Signature [...]

```

Yes, we can trust it because it is being signed by the certificate authority.

5. Identify the length of key exchanged by the Client?

```

▼ Handshake Protocol: Client Key Exchange
  Handshake Type: Client Key Exchange (16)
  Length: 66

```

6. What algorithm is used for encrypting the session key?

The algorithm used to encrypt session id is RSA.

```

▼ Random: 127b41eab53322affc8ecadb8970dbd31baa47b26e010dcc21e61ec7b22cc8f1
  GMT Unix Time: Oct 29, 1979 23:24:50.000000000 India Standard Time
  Random Bytes: b53322affc8ecadb8970dbd31baa47b26e010dcc21e61ec7b22cc8f1
  Session ID Length: 32
  Session ID: 8b33a9598c9eabeb5ba79adee21cbe86576f16ec717c25bd6616fb39b284f7c6
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)

```

7. List the various parameters specified in the Encrypted handshake message.

```

▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 40
  Handshake Protocol: Encrypted Handshake Message

```

8. Calculate the time taken for completion of the entire handshake protocol.

The time taken is 0.003117 seconds.

## PRETTY GOOD PRIVACY - ANALYSIS

### 1. Installation of Mailvelope

#### Setup

This keyring does not yet contain a key pair.  
A key pair is required to encrypt and decrypt messages, as well as to invite your contacts to end-to-end encrypted communication.

#### Generate key

If you're using this extension for the first time and if you do not have a key pair yet, please generate one now.

Generate key

#### Import Key

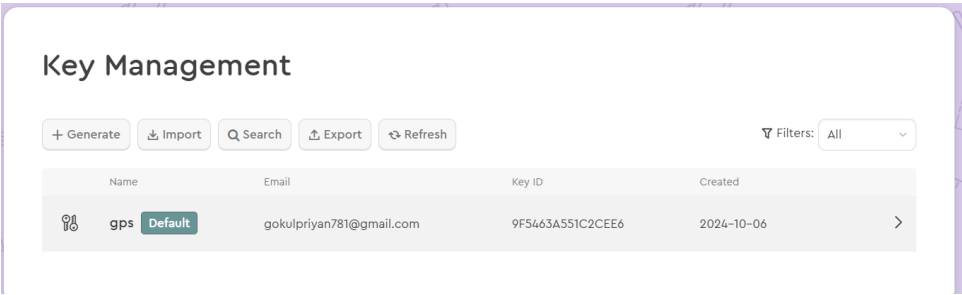
Do you already have a key pair on another device? You can import your existing keys. Just export the key pair from the other device and then import them here.

Import Key

#### GnuPG connection

See available settings at: [OpenPGP Preferences](#)

### 2. Generation of public-private key pair



3. Import Public Key

Import Keys

You can add keys either as file or as text from the clipboard.

Select files

Drag file to this window or Add file

4. Export Public Key

Export key

Which key would you like to export?

Public

Private

All

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: Mailvelope v5.2.0
Comment: https://mailvelope.com

xsFNBGcCdZoBEADBAOtPXoRcxwck1zgb8SMTbIGJTV/IMXZklUtt
jXtuYm
ShilLelBaDym32rUftqTwpIjm7t0WoKDXR0mP7ePq3H16mrGKuyW
4+boNim
oUaxOwO3e0D+k4JbEKmRhLup5Av2GvDWjCjYwMGtNfBbmEvys
MHJzeCEKLI
taOPk1/irUzBYpR+xoC7sNRd2EeazfenKx/zgzKScsmNF+7eN5uZL
/ro8p
8hT8zCqObv+AAUyW5jH3xXHigsxqCwKVUEmFK/DR9lI+q1YWCFz
cvaPIIVXv1
```

5. Encrypt and send a email

< Encrypt data

Encryption successful

Encrypted for           gokulpriyan781@gmail.com  
Signed by               gps (gokulpriyan781@gmail.com)

Encrypted files

ASC   text.txt

```
-----BEGIN PGP MESSAGE-----
Version: Mailvelope v5.2.0
Comment: https://mailvelope.com

wcFMA1yn2M7Lb0odAQ//ZWxsLFwhiLbmtog4IS2nisq6Lq3pqk/vS6PbqLsV
z3o3hBmU6EnXCLb5ndlj0ah4gm+2LEej8+8XppHDA+05YNdfhidX6bztv6GY
HRnsf0xx0nxwSXhiM+nG9G2hOax1NRGNRIJRSkktHvG5hznzKwsHK9dfzH/1u17
```

**RESULT:**

Thus, we have studied the working principle of security protocols like SSL/TLS using Wireshark and PGP using chrome plugin – Mailvelope.

**Evaluation**

Parameter	Max Marks	Marks Obtained
Uniqueness of the Work	15	
Completion of experiment on time	10	
Documentation	5	
Total	30	
Signature of the faculty with Date		