SSL Content Types

- Handshake Protocol – 22 (0x16)
  - –responsible for authentication and session key setup
- ChangeCipherSpec Protocol - 20 (0x14)
  - –Notify start of encryption
- Alert Protocol – 21 (0x15)
  - –Reporting of warnings and fatal errors
- Application Protocol – 23 (0x17)
  - –Actual encryption and transport of data

TLS is encryption for data in transit, not data at rest.

The end host or recipient in a TLS connection must be able to decrypt the encrypted traffic sent to it in order to be processed and/or displayed in the web browser, mail app or SIP Message.

Cryptography Algorithms

- Symmetric algorithms like AES and 3DES use a single key for encryption and decryption
  - –the same key that encrypts data is used to decrypt it
- Asymmetric algorithms like RSA use two separate keys
  - –one for encryption and the other for decryption
- Symmetric ciphers are computationally much faster than asymmetric ciphers
  - -symmetric ciphers are preferred over asymmetric ciphers when encrypting/decrypting large amounts of data

Cryptography Algorithms

- The method most often used is a combination of both asymmetric and symmetric ciphers

- Asymmetric ciphers are used to either exchange or generate the key material from which symmetric "session" keys are derived

- Symmetric keys are known as session keys because they are used for a single session and then discarded

- They are the shared secrets used for symmetric encryption/decryption of the bulk of the data

Sharing Session Keys

- Session keys are computed on each side of the connection

- The components used to generate the session keys are passed back and forth in one of two ways
  - referred to as "key exchange" and "key generation"

- The **key exchange** method uses asymmetric encryption to send a pre-master secret securely to the server.

- The **key generation** method is used to exchange unencrypted components, which both sides will use to derive symmetric session keys

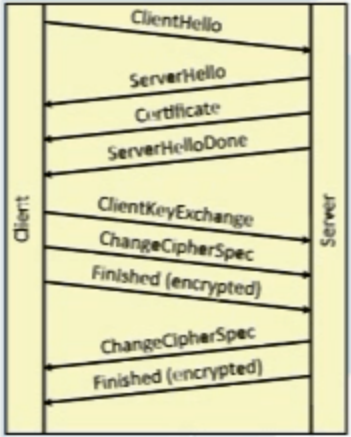
Key Exchange Method

- The RSA and DSA algorithms are commonly used with the key exchange method

- The client generates a pre-master secret
  - both sides will generate the actual cryptographic keys used to encrypt/decrypt the data transferred over this session

- This pre-master secret must remain private to protect the confidentiality of the session.
  - It is the seed from which the final keys will be derived

TLS Handshake



Key Generation Method

- Another algorithm used for the key generation method is Diffie-Hellman (DH)

- The client and server independently generate a master secret after an initial exchange of components that are required for that process, all of which can be public and therefore do not require encryption

- Each side adds components that must remain private to protect the confidentiality of the process
  - these components are not transmitted across the network but remain private to each side

Key Generation Method

• The important advantage of **key generation** (DH) over **key exchange** (RSA) is:

- if the traffic is intercepted by an attacker during this handshake the attacker does not have enough information to compute the master secret and derive the cryptographic keys
- even if the attacker were to be in possession of the server's private key

• Using the key exchange method (RSA), an attacker could independently derive the cryptographic keys and decrypt the exchanged data if they had access to the server's private key.

Ephemeral RSA or Diffie-Hellman Handshake

Diffie Hellman:

https://www.youtube.com/watch?v=YEBfamv-_do

1. both agree : g^pk mod p (generator & prime modulus)
2. side1: g^pk1 mod p = r1 . r1 sent publically to side2
3. side2: g^pk2 mod p = r2 . r2 sent publically to side1
4. side1: r2^pk1 mod p = r
5. side2: r1^pk2 mod p = r
6. r on both sides will be same.

MITM can only intercept r1 and r2 and cant figure out r.