IT SECURITY

IS 364: ASSIGNMENT 02

NAME: SIMON ELIZABETH E.

REG NO: 2017-04-07373.

1. Answer

SSL stands for secure sockets layer. Protocol for secure web browsers and servers communication, allows for the authentication, encryption and decryption of data sent over the internet. It provides a way for enterprises to encrypt data before sending it to users, preventing third parties from reading it while it’s in transit.

How SSL works

1. A user connects to an SSL-enabled service such as a website.
2. The user’s application requests the server’s public key in exchange for its own public key. This public key exchange provides ways for both parties to encrypt messages that only the other party can read.
3. When the user sends a message to the server, the application uses the server’s public key to encrypt the message.
4. The server receives the user’s message and decrypts it using its private key. Messages sent back to the browser are encrypted in a similar way using a public key generated by the user’s application.

In SSL protocol offer SSL certificate which is digital certificate issued by third part and verifies the identity of the web server and its public key, but once connection established symmetric key algorithm may use to encrypt and decrypt all traffics between web browsers and servers.

The following below is how SSL certificate functions

1. A browser or server attempts to connect to a website secured with SSL. The web browser requests that the web server identify itself.
2. The web server sends the web browser a copy of its SSL certificate.
3. The web browser checks to see whether or not it trusts the SSL certificate. If so, it sends a message to the web server.
4. The web server sends back a digitally signed acknowledgement to start an SSL encrypted session.
5. Encrypted data is shared between the web browser and the web server

IMPROVEMETS MADE FROM SSL1 TO SSL3

SSL 1.0: Developed by Netscape, version 1.0 wasn’t released.

SSL 2.0:  Another version of SSL after version 1.0 had a number of security flaws.

SSL 3.0: This protocol was released in 1996, the number of security flaws leading to the release of SSL 3.0

Some major improvements of SSL 3.0 are:

1. Use of a full 128 bits of keying material even when using the Export cipher
2. Ability of the client and server to send chains of certificates, thus allowing organizations to use certificate hierarchy which is more than two certificates deep.
3. Implementing a generalized key exchange protocol, allowing Diffie-Hellman and Fortezza key exchanges as well as non-RSA certificates.
4. Allowing for record compression and decompression
5. Use of a full 128 bits of keying material even when using the Export cipher
6. Ability to fall back to SSL 2.0 when a 2.0 client is encountered

2) Answer

The SSL protocol was originally developed at Netscape to enable ecommerce transaction security on the web, which required encryption to protect customer’s personal data, as well as authentication and integrity guarantees to ensure a safe transaction. To achieve this, the SSL protocol was implemented at the application layer, directly on top of TCP, enabling protocols above it (HTTP, email, instant messaging, and many others) to operate unchanged while providing communication security when communicating across the network

When SSL is used correctly, a third-party observer can only infer the connection endpoints, type of encryption, as well as the frequency and an approximate amount of data sent, but cannot read or modify any of the actual data.

How TLS does works

The **SSL/TLS Handshake** is where it all starts. The SSL/TLS handshake involves a series of steps through which both the parties – client and server, validate each other and start communicating through the **secure SSL/TLS tunnel.**

TLS uses a combination of symmetric and asymmetric cryptography, as this provides a good compromise between performance and security when transmitting data securely.

With symmetric cryptography, data is encrypted and decrypted with a secret key known to both sender and recipient; typically 128 but preferably 256 bits in length (anything less than 80 bits is now considered insecure). Symmetric cryptography is efficient in terms of computation, but having a common secret key means it needs to be shared in a secure manner.

Asymmetric cryptography uses key pairs – a public key, and a private key. The public key is mathematically related to the private key, but given sufficient key length, it is computationally impractical to derive the private key from the public key. This allows the public key of the recipient to be used by the sender to encrypt the data they wish to send to them, but that data can only be decrypted with the private key of the recipient.

The advantage of asymmetric cryptography is that the process of sharing encryption keys does not have to be secure, but the mathematical relationship between public and private keys means that much larger key sizes are required,

For this reason, TLS uses asymmetric cryptography for securely generating and exchanging a session key. The session key is then used for encrypting the data transmitted by one party, and for decrypting the data received at the other end. Once the session is over, the session key is discarded.

A variety of different key generation and exchange methods can be used, including RSA, Diffie-Hellman (DH), Ephemeral Diffie-Hellman (DHE), Elliptic Curve Diffie-Hellman (ECDH) and Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). DHE and ECDHE also offer forward secrecy whereby a session key will not be compromised if one of the private keys is obtained in future, although weak random number generation and/or usage of a limited range of prime numbers has been postulated to allow the cracking of even 1024-bit DH keys given state-level computing resources. However, these may be considered implementation rather than protocol issues, and there are tools available to test for weaker cipher suites.

The set of algorithms that cipher suites usually contain include:

A [key exchange algorithm](https://en.wikipedia.org/wiki/Key_exchange), a [bulk encryption algorithm](https://en.wikipedia.org/wiki/Link_encryption), and a [message authentication code](https://en.wikipedia.org/wiki/Message_authentication_code) (MAC) algorithm.

IMPROVEMETS MADE FROM SSL1 TO SSL3

TLS 1: This protocol was defined in RFC 4346 in April of 2006.

1. The Implicit Initialization Vector (IV) is replaced with an explicit IV to protect against Cipher block chaining (CBC) attacks.
2. Handling of padded errors is changed to use the bad record Mac alert rather than the decryption failed alert to protect against CBC attacks.
3. Premature closes no longer cause a session to be non-resumable.

TLS 2: This protocol was defined in RFC 5246 in August of 2008, contains improved flexibility

1. The MD5/SHA-1 combination in the pseudorandom function (PRF) was replaced with cipher-suite-specified PRFs.
2. The MD5/SHA-1 combination in the digitally-signed element was replaced with a single hash. Signed elements include a field explicitly specifying the hash algorithm used.
3. TLS Extensions definition and AES Cipher Suites were merged in.

TLS 3: This protocol is currently being revised, the major differences from TLS 1.2 include:

1. The list of supported symmetric algorithms has been pruned of all legacy algorithms. The remaining algorithms all use Authenticated Encryption with Associated Data (AEAD) algorithms.
2. All handshake messages after the Server Hello are now encrypted.
3. TLS 1.2 version negotiation verification mechanism was deprecated in favor of a version list in an extension.
4. Session resumption with and without server-side state and the PSK-based cipher suites of earlier versions of TLS have been replaced by a single new PSK exchange

3) Answer

Public key infrastructure (PKI) is a catch-all term for everything used to establish and manage public key encryption, one of the most common forms of internet encryption. It is baked into every web browser in use today to secure traffic across the public internet, but organizations can also deploy it to secure their internal communications and access to connected devices.

PKI it very important it’s because this combination of encryption and authentication makes trustworthy online communication possible.

PKI works on keys and certificates*.*A key, as already noted, is a long string of bits a number, in other words that's used to encrypt data. For instance, if you used the ancient and simple [Caesar cipher](http://practicalcryptography.com/ciphers/caesar-cipher/) with a cryptographic key of 3, that would mean that every letter in your message is replaced by one three letters later in the alphabet A becomes D, B becomes E, and so forth. To decode its message, your recipient need know not only that you were using the Caesar cipher but that your key was 3.

Obviously the mathematics behind modern encryption is muchmore complicated than this. One of the ways it's different gets around a somewhat obvious problem with the Caesar cipher, you have to somehow let your recipient know the key used to encode the encrypted message. PKI gets its name because each participant in a secured communications channel has two keys. There's a publickey, which you can tell to anyone who asks and is used to encode a message sent to you, and a private key, which you keep secret and use to decrypt the message when you receive it. The two keys are related by a complex mathematical formula that would be difficult to derive from brute force*.*

So that covers how data is encrypted within a public key infrastructure. But remember, PKI is widely used because, in addition to encrypting messages, it also lets you know that the person with whom you're exchanging encrypted messages is who they say they are. That's where certificatescome in*.*

With these concepts under our belt, these are the elements that go into PKI:

* A certificate authority, which issues digital certificates, signs them with its own public key, and stores them for reference.
* A registration authority*,*which verifies the identities of those requesting digital certificates. A CA can act as its own registration authority or can use a third party to do so.
* A certificate database that stores both the certificates and metadata about them most importantly, the period of time for which the certificate is valid.
* A certificate policyoutlining the PKI's procedures, which allows outsiders to judge how trustworthy the PKI is.

What makes PKI secure

PKI is great for securing email for the same reason that it's great for securing web traffic: because data flowing over the open internet can be easily intercepted and read if it isn't encrypted, and because it can be difficult to trust that a sender is who they claim to be if there isn't some way to authenticate their identity. As we've seen, establishing near-universal PKI for web traffic has been relatively easy because most of the necessary infrastructure is built into web browsers and servers. Email is accessed through more heterogonous clients, which makes things a bit trickier.

One of the oldest and best-established PKI systems for securing email is S/MIME; there's also PGP (Pretty Good Privacy), Support for these kinds of email protections are built into clients like Microsoft outlook.