08. Security in Computer Networks

Missing 8.8

8.1 What is network security?

Network security encompasses the following aspects in an attempt to keep the Internet safe:

1. Confidentiality: only the sender and intended receiver should be able to read message contents. This is done with encryption.
2. Authentication: sender and receiver want to confirm each other's identity.
3. Message integrity: sender and receiver want to ensure the message was not altered.
4. Access and availability: services must be accessible and available to users.

If these principles are not followed, possible attackers may eavesdrop and intercept messages, modify the messages, spoof and impersonate others, hijack the session or perform denial of service attacks.

8.2 Principles of Cryptography

There are two main key cryptography techniques: symmetric-key cryptography and public-key cryptography:

- Symmetric-key cryptography is encryption where both parties share the same key. This is difficult on the internet because it takes for granted that both parties know the key (which they couldn't securely exchange in the first place).
- Public-key cryptography is encryption where both parties have a publicly available key and a secret key. Communication happens when the sender uses the receiver's public key to encrypt the message. The receiver can use their own secret key to decrypt the message.

Block cipher algorithms split the message up into blocks and encrypt these separately. This method has a weakness because duplicate segments will always encrypt to the same thing. To resolve this, cipher block chaining is used which will send a random first message which is used to encrypt the blocks in a chain.

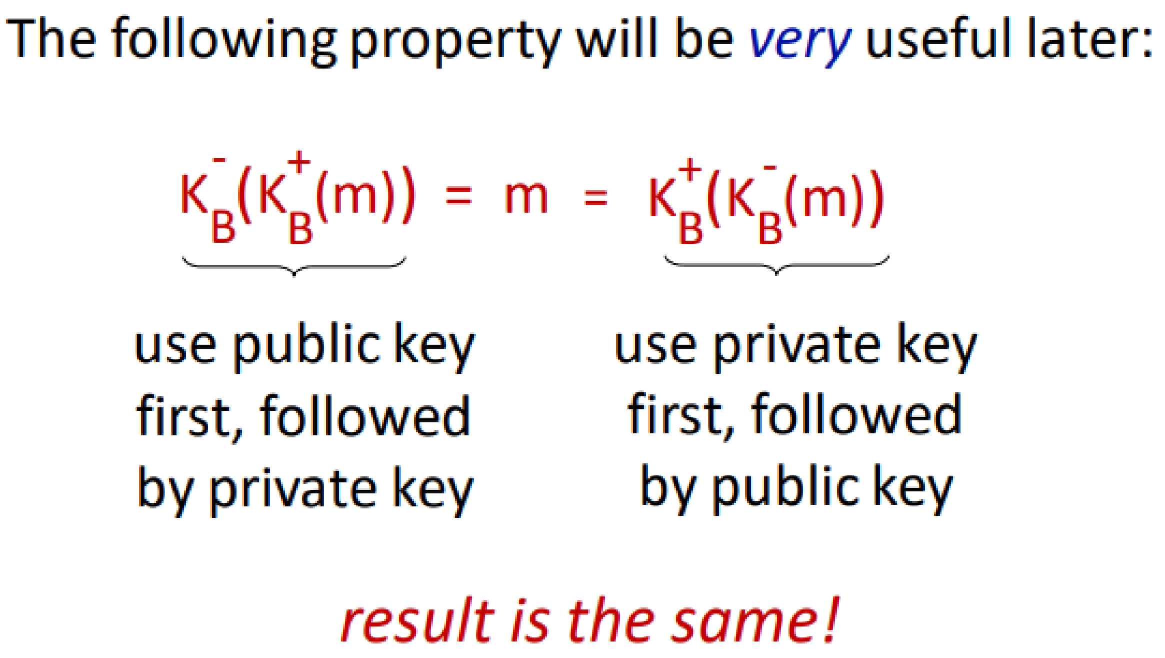
$c(I) = K_s(m(i) \oplus c(i-1))$

RSA computes a public key private pair by choosing two large primes p, q. These primes usually have upwards of 1024 bits each. (or more for rsa2048, etc.)

1. Compute $n = pq, z = (p-1)(q-1)$
2. Choose $e < n$ where n has no common factors with z. (e, z are relatively prime)
3. Choose d such that $ed - 1 mod z = 1$
4. Public key is (n, e) and private key is (n, d)

Encryption using RSA is performed as follows:

1. Encrypt message m = $c = m^e mod n$
2. Decrypt pattern $m = c^d mod n$



The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{} = m = \underbrace{K_B^+(K_B^-(m))}_{}$$

use public key first, followed by private key

use private key first, followed by public key

result is the same!

The reason RSA is secure is because you need to find factors of a prime number without knowing factors p and q. When the prime has upwards of 1024 bits or more, it requires immense amounts of computing power.

RSA is computionally exponentially expensive. A typical exchange can instead use RSA to exchange a symmetric session key which can be used with symmetric-key cryptography later.

8.3 Message Integrity and Digital Signatures

A receiver must be able to verify that the message was actually sent by the intended sender and that the message hasn't been modified in transit.

Cryptographic hash functions are mathematic functions that take an input string m and compute a hash of a known size (depending on the algorithm used). For any hash function, it must be computationally infeasible to find two messages x and y such that $H(x) = H(y)$. This is a hash collision and that's bad.

A message authentication code (MAC) is a technique used to verify that the sender actually sent the message. This is done by exchanging a secret HMAC code s which is used together with the message to compute the hash. The sender then sends $(m, H(m+s))$ to the receiver, which confirms by computing $H(m+s)$ themselves and verifying the hash. The most MAC today is HMAC.

A digital signature must be verifiable and impossible to impersonate. One way to add a digital signature is to encrypt the message or hash with one's private key. The receiver can then use the sender's public key to decrypt the message or hash to verify. For this to work, we need to ensure that the public key sent actually belongs to the sender. This is done by a Certification Authority (CA) which binds a public key to a person or unit and adds their own CA-signed certificate to verify that the public key belongs to the entity.

8.4 End-Point Authentication

Secure end-point authentication can typically be done with a password, but there are extra measures we need to take:

1. If there's just a plaintext password, someone may eavesdrop and steal the password in transmission
2. If we encrypt it, someone may perform a playback attack

3. If we encrypt it with a nonce and encrypt it using symmetric-key cryptography, then the receiver can decrypt it using the key and verify the nonce.

8.5 Securing E-Mail

Security measures are taken at each layer in the network model to preserve confidentiality, message integrity, and authentication. One way to do this is as follows:

- Sender: generates symmetric private key K_s
- Sender: encrypts the message with K_s for efficiency (instead of encrypting with private/public key)
- Sender: encrypts K_s with receiver's public key
- Sender: sends both $K_s(m)$ and $K_r^+(K_s)$ to receiver r
- Sender: signs and sends hash of her message with her private key K_s^-

The receiver may then:

- Use sender's public key to recover hash
- Use his private key to recover K_s
- Use K_s to decrypt m
- Validate the message and hash

8.6 Securing TCP Connections: TLS

Securing the transport layer is important to prevent people from stealing information from the application layer. If there was no security in place, attackers could steal your credit card information when purchasing something from Amazon or other important pieces of data.

TLS has three stages; handshake, key derivation, and data transfer

- The handshake stage: TCP connection is created. Accepts the server's public key and verifies it against the CA. Create Secret Master Key and encrypt it with the server's public key. The server decrypts the encrypted Secret Master key.
- The key derivation stage: Generates four keys: A-B encryption key and A-B HMAC, B-A encryption key, and B-A HMAC. These keys are derived from the Secret Master

Key which is why it's called the key derivation stage. Both parties know of all four keys + Secret Master Key now.

- Data transfer is split into records, add a MAC, and encrypt with the corresponding key. Reverse the process at the receiving end. Individual sequence numbers are also used to prevent attacks from removing, replaying, or shuffling records.

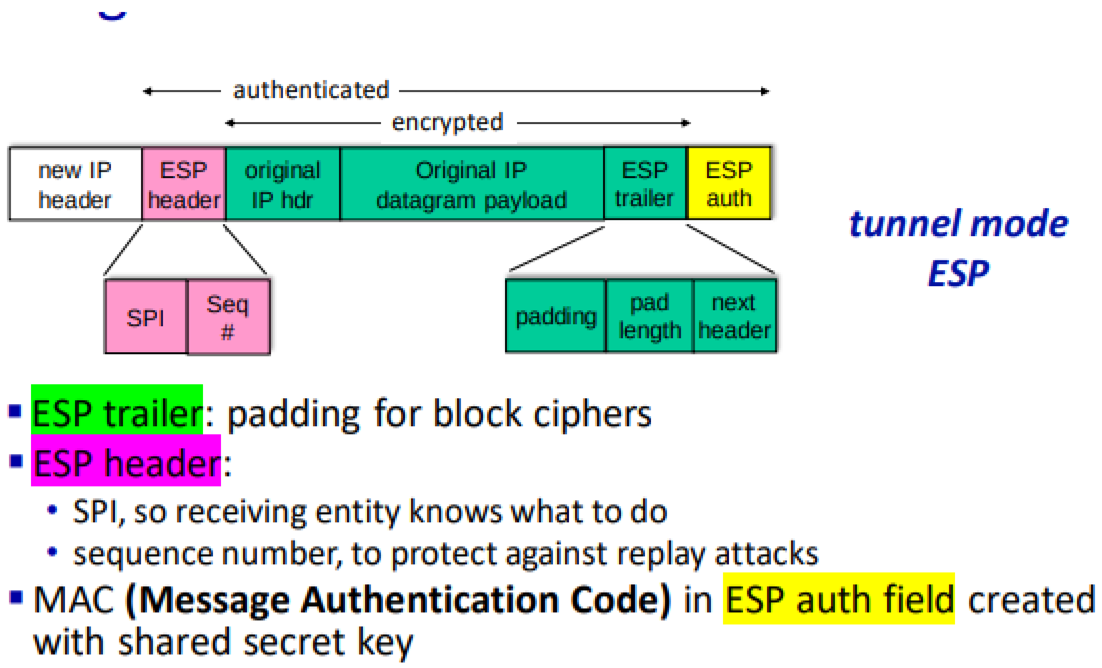
8.7 Network Layer Security

IPsec provides datagram-level encryption, authentication, and integrity. It has two operation modes; transport and tunnel.

1. Transport mode: only the datagram payload is encrypted and authenticated
2. Tunnel mode: entire datagram is encrypted, and it's encapsulated in a new datagram with new IP header

There are two primary IPsec protocols: AH and ESP. Both provide source authentication and data integrity, but AH does not provide confidentiality.

Before transmission, a security association is established. It's unidirectional so you need two to communicate. IP is connectionless, but IPsec is connection-oriented.



- ESP trailer: padding for block ciphers
- ESP header:
  - SPI, so receiving entity knows what to do
  - sequence number, to protect against replay attacks
- MAC **(Message Authentication Code)** in ESP auth field created with shared secret key

At the source router:

1. Append ESP trailer to original datagram
2. Encrypt result using algorithm and key specified by SA.
3. Append the ESP header in front of the result
4. Create authentication MAC using algorithm and key specified by SA.
5. Append MAC, forming the final payload
6. Create a new IP header with the payload

IPsec uses sequence numbers, initialized to zero which is increased each time a datagram is sent on the SA. This is to prevent sniffing and replay attacks.

Setting up the SA for each router is impractical for large networks with thousands of endpoints. Instead, we use IPsec Internet Key Exchange (IKE).

8.8

8.9 Operational Security: Firewalls and Intrusion Detection Systems

Firewalls have three goals; allow authorized people through, deny intruders, and be impenetrable themselves. There exist three types of firewalls:

1. Traditional Packet Filter: inspects each diagram, and decides based on datagram information to let through. Typically based on protocol, host, TCP flags, etc.
2. Stateful Packet Filter: tracks TCP connections to block non-sensical packets.
3. Application gateway: Filter packets on application data to a specific application. Blocks all application connections not originating from the gateway.

Firewalls have a few limitations, namely:

- IP spoofing cannot be detected.
- If multiple apps need special treatment, each will need its own gateway.
- Client software must know how to contact the proxy gateway.
- Filters often use an “all or nothing” policy for UDP.
- Highly protected sites still suffer attacks.

Intrusion detection systems operate on TCP/IP headers only and have no correlation check on sessions. They examine strings in the packet against know viruses or attack strings.

They also examine the correlation among multiple packets to prevent port scanning, networking mapping, and denial-of-service attacks.