George Theodorakopoulos

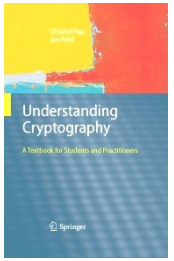
Foundations of Modern Security

Outline

- Symmetric and Asymmetric Encryption
- Using Encryption to ensure Secure Transmission (e.g. HTTPS)

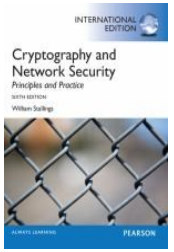
Textbooks

- Christof Paar and Jan Pelzl, "Understanding Cryptography," Springer, 2010.



Concise, some maths

- William Stallings, "Cryptography and Network Security: Principles and Practice," 6th ed., Prentice Hall, 2014.



More text, includes TLS

- David Kahn, "The Codebreakers: The story of Secret Writing," Scribner, 1996.

Crypto History

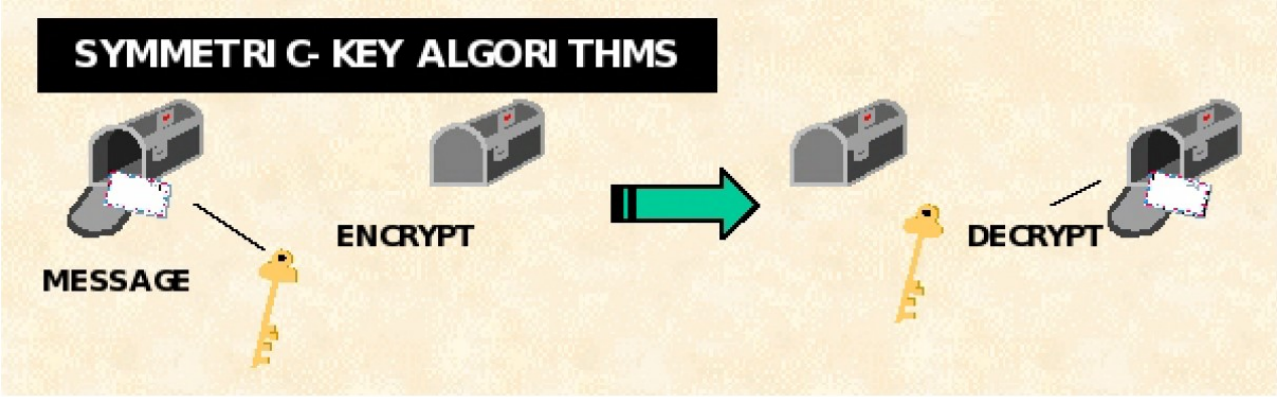
- Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone, "Handbook of applied cryptography," CRC press, 2010.

Mathematics (free online)

Encryption

Symmetric and Asymmetric Encryption, Key size, Brute force attack

Symmetric Encryption



Symmetric Encryption

- In symmetric encryption, the one-and-only key is used for decryption, so **it must be kept secret**. Otherwise confidentiality is lost.
- The problem then arises, how do we send the key to the receiver of our information without it being discovered?
- One option is to send it down another secure channel, separate from the data - sending "out of band." It could be done, e.g., over the phone.
- Not very dynamic! Another option is to encrypt the key itself. However, we now have another encryption key to send!

Symmetric Encryption

- Another problem is repudiation: If we are both using the same key, how can one of us prove in a court of law that the other one sent the message?
- DES (data encryption standard) and triple DES are two popular symmetric key encryption schemes that have been (or are still) used.
- AES is the new standard (used more and more)

Asymmetric Encryption

- In this scenario - two different keys are used, one for encryption and a different one for decryption.
- How can you decrypt information with one key and decrypt with a different key? - complex mathematics!

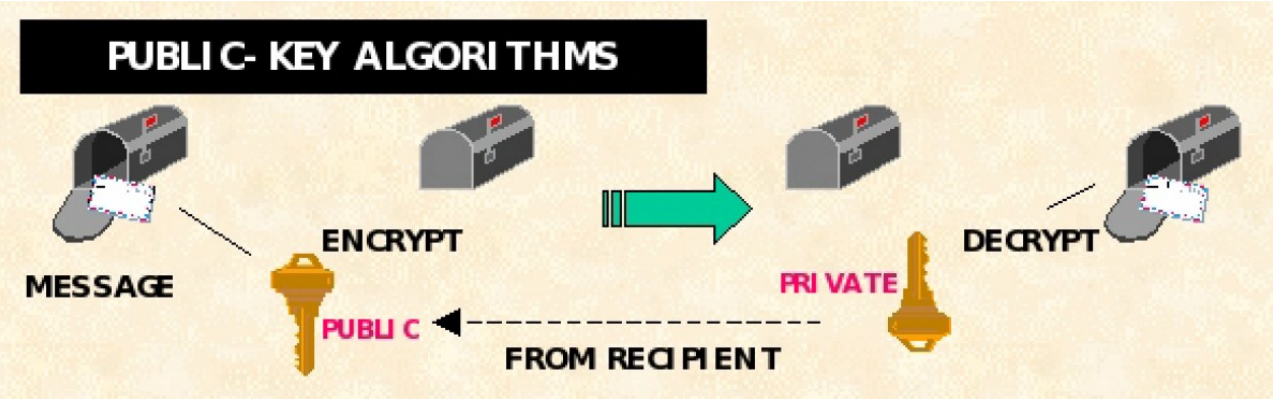
Asymmetric Encryption

- Each user who wants to receive encrypted messages must generate an encrypting key and a corresponding decrypting key on their computer or secure device (smartcard). The decrypting key never leaves their computer or device. The encrypting key is sent to everyone.

- The encrypting key is called the "public key" as it can be used by anyone wishing to encrypt information that is to be sent to the holder of the private key. Only the corresponding decrypting key can be used to decrypt information encrypted with the public key.

- The decryption key must be kept private, because anyone who has it can decrypt messages encrypted with the corresponding public key - hence the term "private key".

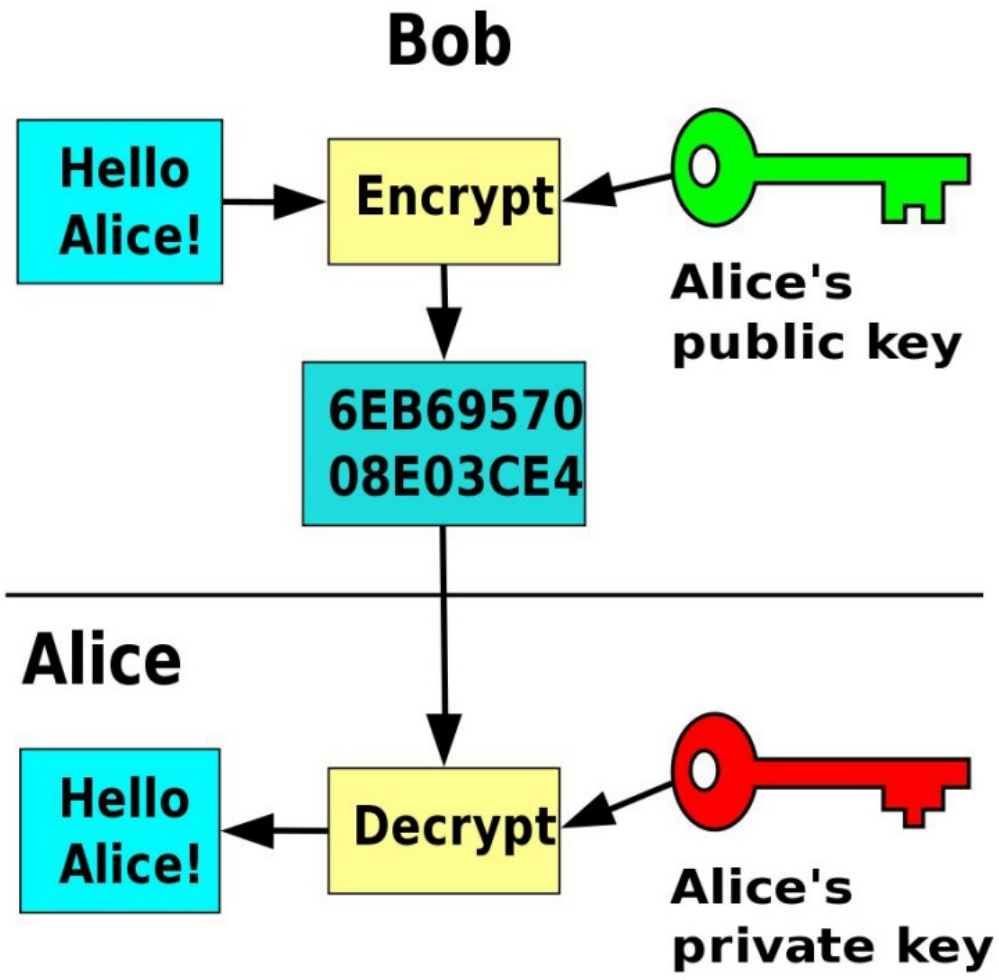
Asymmetric Encryption



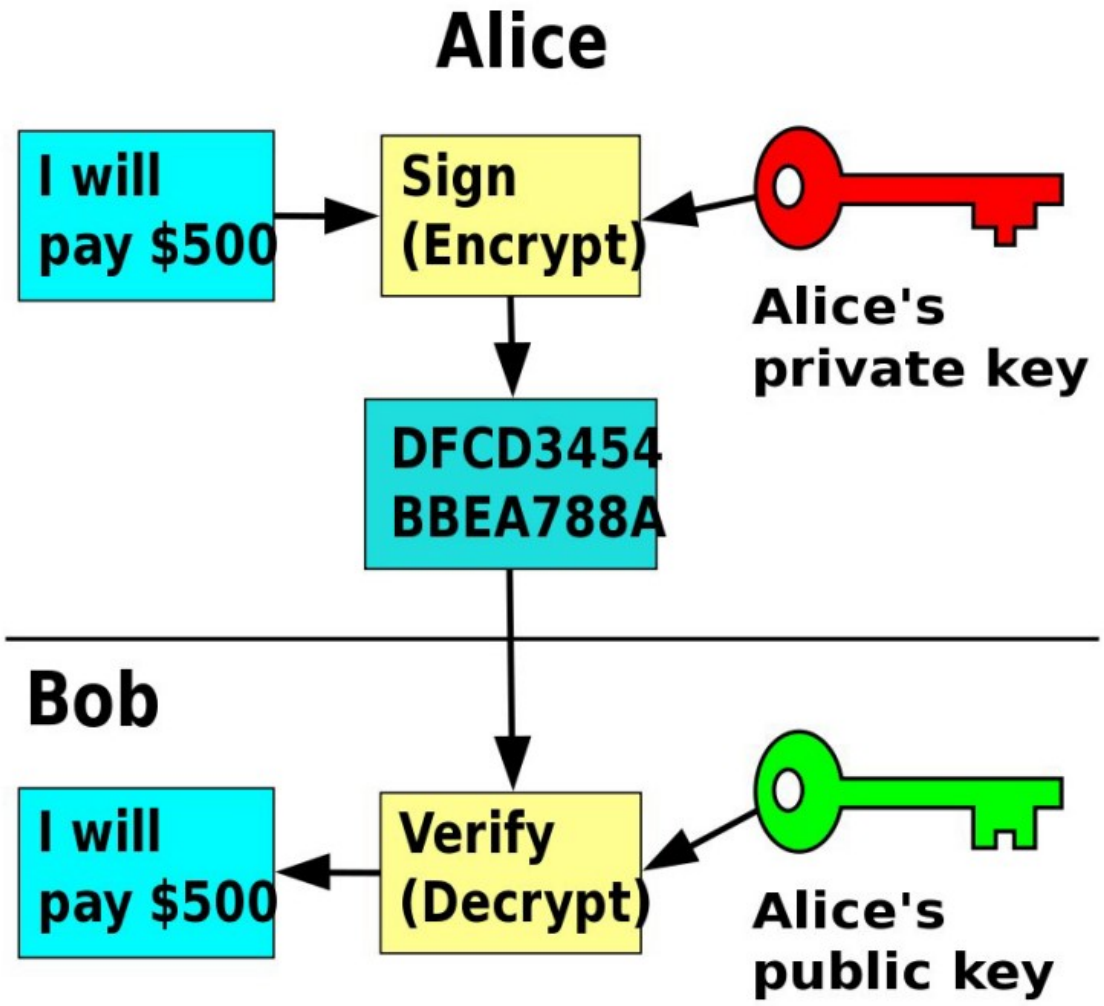
No longer necessary to send the encryption key "out of band" for confidentiality.

However, it is still necessary to preserve the integrity of the encryption key, otherwise an attacker might substitute his own.

An Example


Bob
Hello Alice!
Encrypt
Alice's public key
6EB69570 08E03CE4
Alice
Hello Alice!
Decrypt
Alice's private key

What if we invert the keys?



Alice cannot repudiate having written the message.

Asymmetric Encryption

- If asymmetric encryption is so powerful, why do you need symmetric encryption? The reason is speed.

- RSA is a commonly used asymmetric algorithm and is used in many implementations that utilize this type of encryption.

Public and Private Keys

- The private key must stay private.
- If you encrypt with the private key then what is the problem?
- What is the point of the following two scenarios?
- Encrypt with private. Decrypt with public.
- Encrypt with public. Decrypt with private.

Encryption Summary

- The encryption/decryption of information takes (processor) time, and can slow down the data transfer process.
- In addition to that, Asymmetric Cryptography is generally more processor intensive than Symmetric Cryptography.
- Asymmetric encryption offers more secure key sharing so is more suited to dynamic applications running in untrusted networks such as Web Services, while Symmetric encryption is faster and may be used where speed is the ultimate requirement.

Encryption Summary

- Encryption is clearly a very powerful tool to achieve security goals, but there are many different options and combinations that could be used in an implementation.
- The selection of options/combinations should be based on a consideration of the potential risks to information and the impact of identified threats, as well as the purpose of the application/task it is supporting.

Key Size

- In cryptography, **key size** or **key length** is the size (usually measured in bits) of the key used in a cryptographic algorithm (such as a cipher).
- A key should be large enough that a **brute force attack** (possible against any encryption algorithm) is infeasible - i.e., would take too long to execute.
- To achieve perfect secrecy, it is necessary for the key length to be at least as large as the message to be transmitted and only used once (The One-time pad is a cipher with such a key length, and it achieves perfect secrecy).

Key Size

- In light of this, and the practical difficulty of managing such long keys, modern cryptographic practice has discarded the notion of perfect secrecy as a requirement for encryption, and instead focuses on computational security.

- Under this definition, the computational requirements of breaking an encrypted text must be infeasible for an attacker.

Brute Force Attack

- Even if a cipher is unbreakable by exploiting structural weaknesses in the algorithm, it is possible to run through the entire space of keys in what is known as a brute force attack.

- Since longer keys require more work to brute force search, a long enough key will require more work than is feasible. Thus, the length of the key is important in resisting this type of attack.

Brute Force Attack

- With a key of length n bits, there are 2^n possible keys. This number grows extremely rapidly as *n* increases.

- Computing power is increasing but even so this still leaves the key lengths currently considered acceptable (128, 256, 512 bits) well out of reach.

Symmetric Key Lengths

- When the Data Encryption Standard (DES) cipher was released in 1977, a key length of 56 bits was thought to be sufficient so as to limit the 'strength' of encryption available to non-US users. DES can now be decrypted in minutes.
- DES has been replaced by Triple DES, which has 168-bit keys.
- The Advanced Encryption Standard (AES) published in 2001 uses a key size of (at minimum) 128 bits. It also can use keys up to 256 bits. 128 bits is currently thought, by many observers, to be sufficient for the foreseeable future for symmetric algorithms of AES's quality.

Asymmetric Key Lengths

- Public key cryptosystems are cracked in a different way to symmetric algorithms. Usually, cracking them is done by mathematical techniques (e.g., factoring the product of two primes) that are faster than trying all possible keys by brute force. Thus, asymmetric algorithm keys must be longer for equivalent resistance to attack to symmetric algorithm keys.

- As of 2003 RSA Security claims that 1024-bit RSA keys are equivalent in strength to 80-bit symmetric keys, 2048-bit RSA keys to 112-bit symmetric keys and 3072-bit RSA keys to 128-bit symmetric keys.

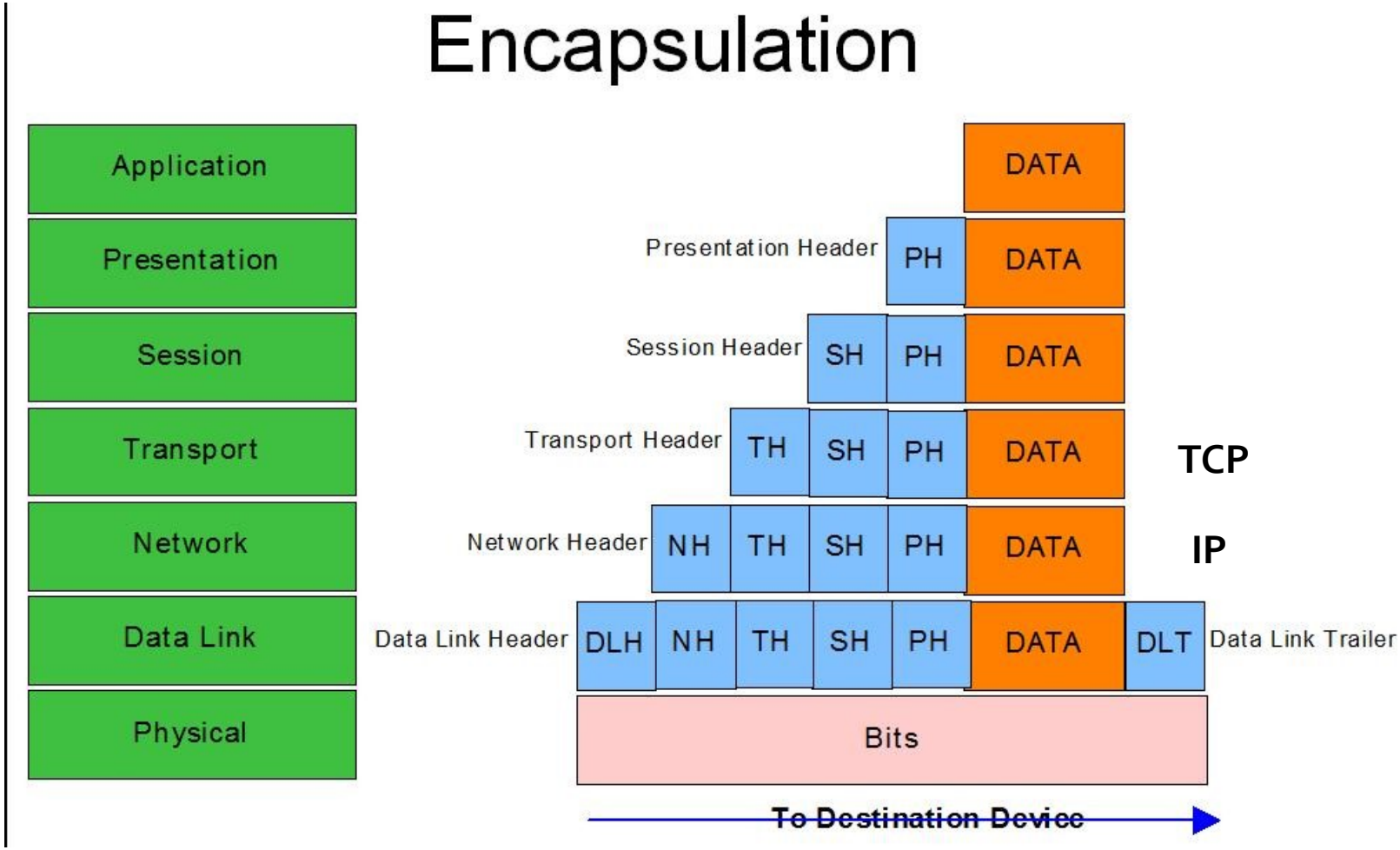
Asymmetric Key Lengths

- RSA claimed that 1024-bit keys were likely to become crackable some time between 2006 and 2010. They were cracked in March 2010 at the University of Michigan. RSA claim that 2048-bit keys are sufficient until 2030.

Using Encryption to ensure Secure Transmission

SSL/TLS

OSI layers



http://abigailabanilla.com/network/osi-encapsulation-terminology/

TCP/IP

- The network-layer protocol used on the Internet is known as the Internet Protocol (IP).
- The two transport-layer protocols used are Transport Control Protocol (TCP) and User Datagram Protocol (UDP).
- **These protocols provide no security guarantees.**

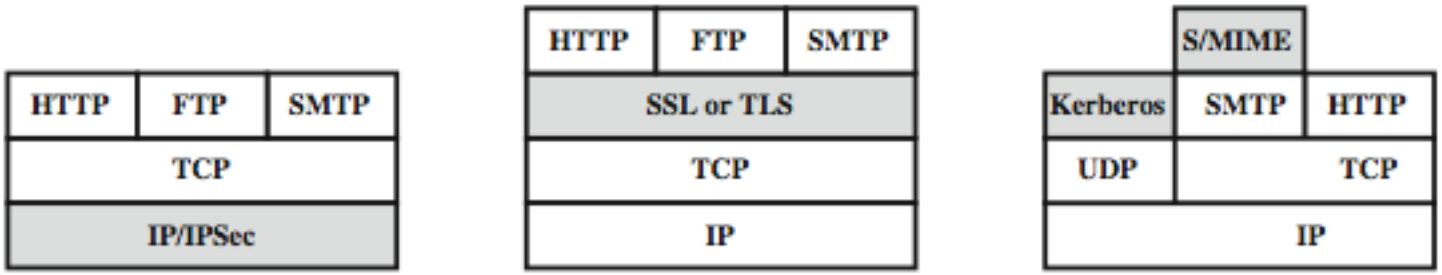
Vulnerability of TCP/IP

- Packets are transmitted in clear text

- Difficult to verify:
  1. The claimed client/server is the true client/server (Authentication).
  2. Data has not been modified in transit (Integrity).
  3. Data has not been viewed by a third party while in transit (Confidentiality).

- How can we provide a solution to these vulnerabilities?

~~SSL~~ TLS



(a) Network Level (b) Transport Level (c) Application Level

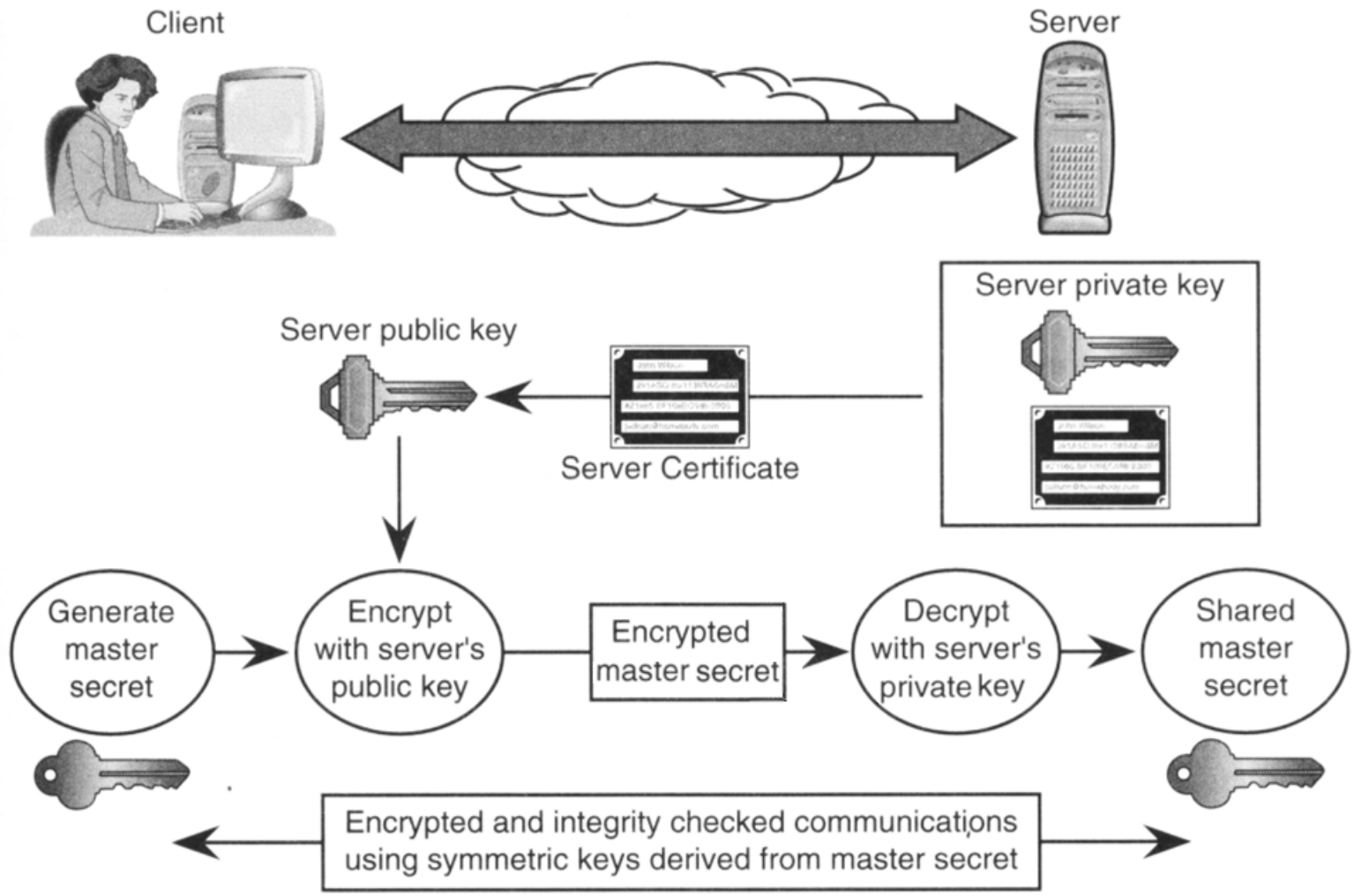
Security can be provided in various layers:

- IPSec is at the network layer
- ~~SSL~~ TLS is at the transport layer
- S/MIME is at the application layer (for email)

~~SSL~~ TLS

- ~~SSL~~ TLS allows applications to communicate across a network in a way designed to prevent eavesdropping, tampering, and message forgery.
- Provides endpoint authentication and message confidentiality over an unsecured network using cryptography.
- Typically, only the server is authenticated, meaning that the end user can be sure with whom they are communicating.
  - Example: In HTTPS, a web browser (end user) communicates with a web server

~~SSL~~ TLS


Client
Server
Server private key
Server public key
Server Certificate
Generate master secret
Encrypt with server's public key
Encrypted master secret
Decrypt with server's private key
Shared master secret
Encrypted and integrity checked communications using symmetric keys derived from master secret

~~SSL~~ TLS

Involves three basic phases. The first and second are setting up the connection, the third transfers the data.

1. Peer negotiation for algorithm support
   - "Which ciphers do you support?"
     "Here is the list of the ones that I support."
2. Authentication and key exchange
3. Symmetric cipher encryption and message authentication

~~SSL~~ TLS

First phase: Establish security capabilities

Client and server negotiate cipher suites, which determine

- Key exchange algorithms (RSA, Diffie-Hellman)
- Symmetric Ciphers to be used (AES, DES).
- Hash functions (MD5, SHA-1)

~~SSL~~ TLS

- TLS client/server negotiate a stateful connection by using a handshaking procedure, agreeing on various parameters used to establish the connection's security.
- Handshake begins when a client connects to a TLS-enabled server requesting a secure connection, and presents a list of supported ciphers and hash functions.
- From this list, the server picks the strongest cipher and hash function that it also supports and notifies the client of the decision.

~~SSL~~ TLS

Second Phase: Server Authentication and Key Exchange

- The server sends back its identification in the form of a digital certificate. The certificate usually contains the server name (SN), the trusted certificate authority (CA), and the server's public key (PK_S).

  Certificate ≈
  "CA vouches that key PK_S belongs to server S"

- The client may contact the trusted CA that issued the certificate and confirm that the certificate is authentic before proceeding.

~~SSL~~ TLS

- In order to generate the session keys used later for the secure transmission of data, the client encrypts a random number with the server's public key, and sends the result to the server. Only the server can decrypt it (with its private key).
- From the random number, both parties generate key material for encryption and for integrity protection.
  - **Different keys for different objectives**

~~SSL~~ TLS

Phase three: Data exchange

- The previous phase concludes the handshake and begins the secured connection, which is encrypted and integrity-protected with the key material until the connection closes.
- If any one of the above steps fails, the TLS handshake fails, and the connection is not created.