

# Web Security: Encryption & Authentication

---

Arnon Rungsawang

*fenganr@ku.ac.th*

**Massive Information & Knowledge Engineering**

Department of Computer Engineering

Faculty of Engineering

Kasetsart University, Bangkok, Thailand.



# Outline

---

- What are SSL/TLS?
- Symmetric vs. Asymmetric ciphers
- Shortcomings of both ciphers
- Cipher strength
- Digital certificates
- Certificate content
- Secure transaction

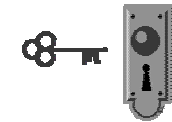


# SSL & TLS

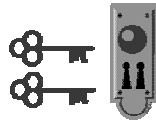
- **SSL** (Secure Sockets Layer), developed by the Netscape Communications, and **TLS** (Transport Layer Security), the open-standard replacement for SSL, are two protocols that add **encryption** and **authentication** onto the **TCP/IP** stack.
- Two main SSL/TLS features:
  - **Ciphers:** enable the encryption of data between two parties, i.e. a client and a server.
  - **Digital certificates:** provide the authentication of the two parties.

Why buy an  
**SSL**  
toolkit as a  
black-box when  
you can get an  
**open**  
one for  
**free** ?

# Two types of ciphers



Symmetric  
(Conventional)  
One Key  
Used for both  
Encryption  
and  
Decryption



Asymmetric  
(Public Key)  
Two Keys  
One Key Used  
for Encryption  
and  
One Key Used  
for Decryption

- **Symmetric** (secret-key ciphers)
  - Use a single key for both encrypting and decrypting data.
  - The encrypted data is secure only if the key can be securely distributed to both parties.
- **Asymmetric** (public-key ciphers)
  - **Public key** is used to encrypt data.
  - **Private key** is used to decrypt data.
  - Only the private key can decrypt the data that has been encrypted by its public key pair.
  - Data encoded with the public key is secure as long as the private key stays secure.
    - You can freely distribute your public key without security risk as long as you keep your private key secure.

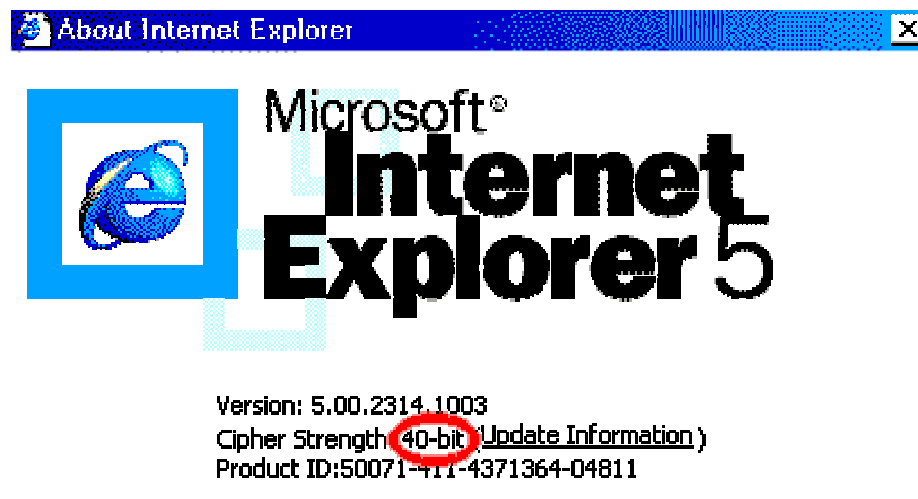
# Shortcomings of both ciphers

---

- **Symmetric** encoded data can be secure only so long as the key used is secured.
- **Asymmetric** encoded data require a longer processing time.
- **SSL/TLS** use both types of ciphers:
  - Use asymmetric cipher to securely exchange the symmetric key.
  - Use symmetric cipher during data transfer.

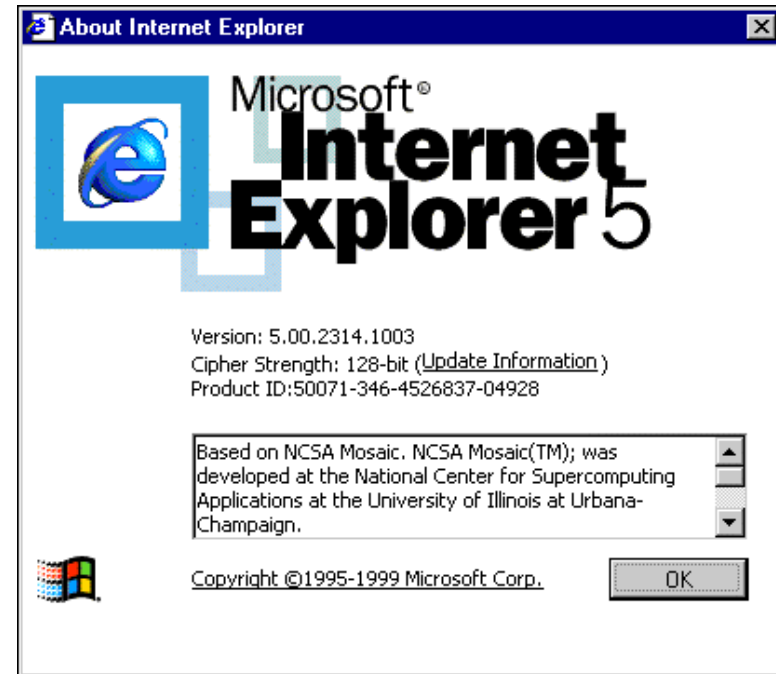
# Cipher strength

- Cipher size or strength plays an important role is **secure transaction**.
  - 40-bit or 56-bit cipher size in common web browser is considered **weak** since it can be cracked in approximately one week using current computer processing power.



# Cipher strength (2)

- 128-bit strength ciphers are **not unbreakable**, but involve a larger time and resource commitment that reduces the usefulness of the data being sought.
  - We can crack a weak ciphered credit card data and go shopping within a week.
  - We must take time and employ much resource (albeit money) to crack a strong ciphered credit card data and perhaps, if success, can go shopping within 6 month!



# Digital certificates

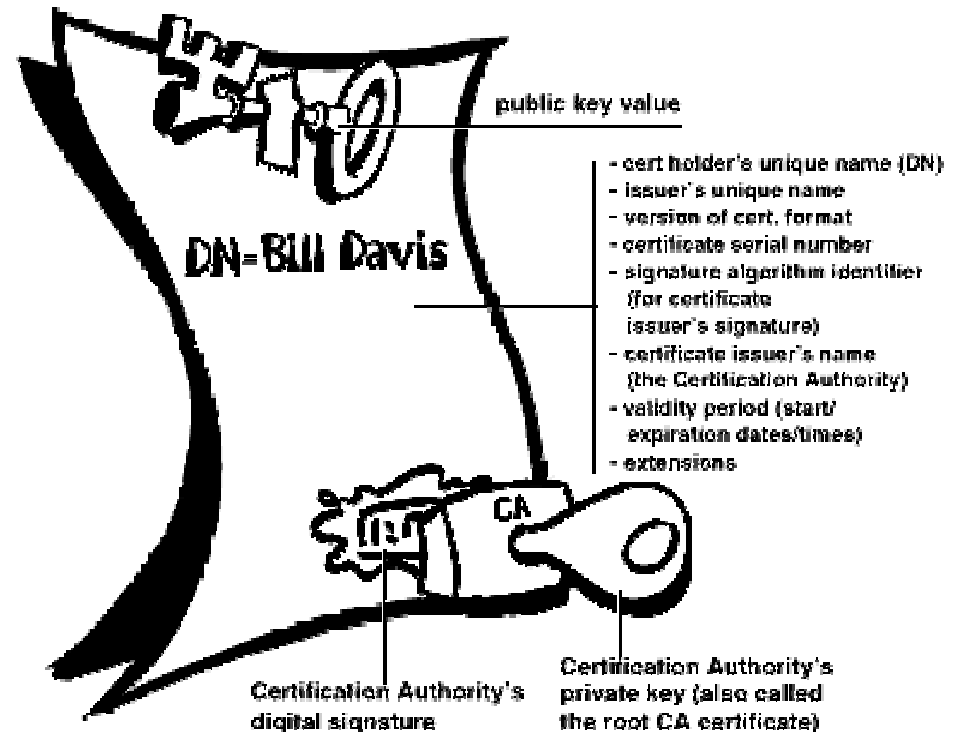


- Allow **authentication** of the parties engaged in a secure transaction.
- Two types of certificates:
  - Server certificates
  - Client certificates
- Both types of certificates are in the X.509 format, and issued by **Certificate Authorities** (albeit **CAs**, such as VeriSign, Thawte, Entrust) that act as a trusted third party.
- CAs verifies the identity of the two parties.
- Server certificates are required in and SSL/TLS transaction, client certificates are in general not needed.



# Certificate content

- A **serial** number
- **Name** of the CA
- **Period** the certificate is valid for
- **Information** of the party in question, such as name, street address and/or email address
- **Subject's** public key
- A **signature** from the issuing CA



## Secure transaction using SSL/TLS

- The client requests a **secure transaction** (by accessing a URL with https) and lets the server know what **ciphers** and **key sizes** it can handle.
- The server sends the requested **server certificate**, which contains the server's **public key** in a "**package**" that has been encrypted by a CA. It also sends a list of ciphers and key sizes in order of priority.
- The client:
  - **Generate** a new symmetric "**session**" key based on the priority list sent by the server.
  - **Compare** the CA that issued the certificate to its list of trusted CAs, **verify** that the certificate has not expired, and **check** that the certificate is used by the server that is listed in the certificate.



## Secure transaction using SSL/TLS (2)

- The client **encrypts** a copy of the new session key with the **public key** sent by the server.
- The client then sends the new encrypted key to the server.
- The server **decrypts** the new session key with its own **private key**.
- At this point, both the client and server have the **same secured session key** which can now be used to encrypt and decrypt the rest of the transferred data.  
If the server wishes to verify the client it will now ask for the client certificate.

\*100% secure ordering 

**Kiddicare**

will undercut any website or shop  
If you find the same Product sold in the uk for less.  
Email us and we will beat their price

Terms and Condition Apply  
[Click here for more details](#)

# Web Security: Encryption & Authentication



Any  
questions  
?