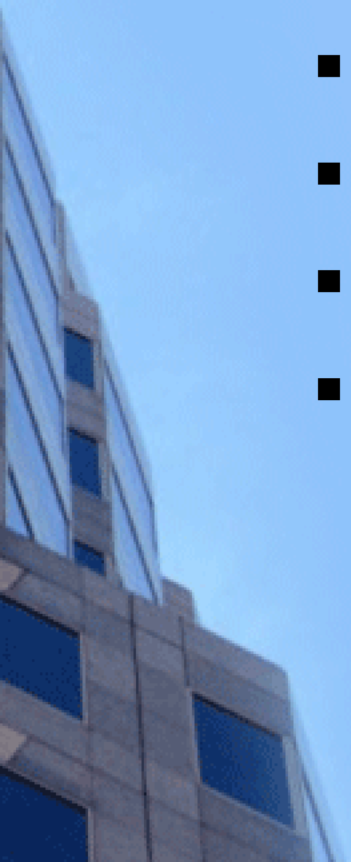
SSL

By: Martin Jarnes Olsen

Content



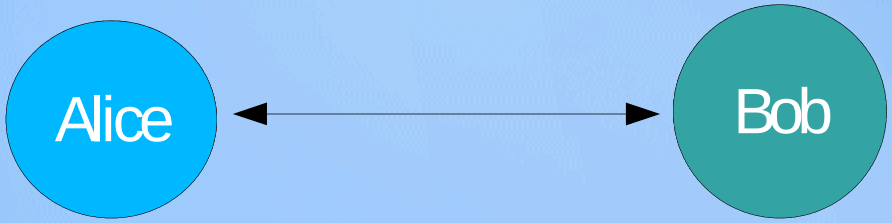
- Basics:
    - Cryptography, asymmetric and symmetric.
- Digital signatures.
- Certificates.
- Client/server interaction.
- OpenSSL.
- SSL in action.

Introduction

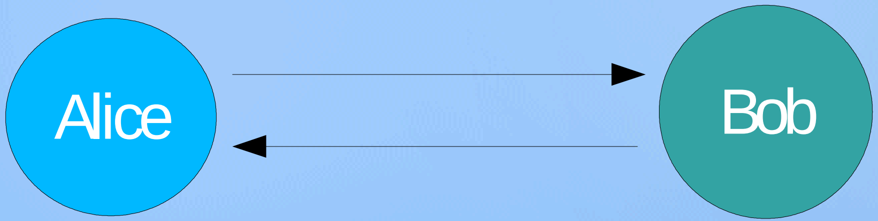
- SSL – **S**ecure **S**ocket **L**ayer is a protocol developed by Netscape for securely transfer of documents over the Internet.
- Development of the protocol started early in the 1990's and culminated in 1995 with the version we know today, SSLv3.
- The main role is to secure Internet traffic. This includes authentication, confidentiality and message integrity.
- SSL is not application specific and can be implemented by any application above the TCP layer.

Cryptography

- Two types:
  - Symmetric – uses same key for encryption and decryption:



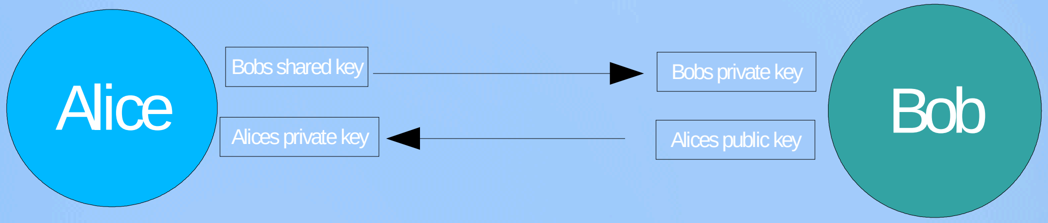
  - Asymmetric – uses different key for encryption:



- Problem: “In-the-middle” attack – user in the middle can fetch keys and encrypt/decrypt messages.
- Solution: PKI – **P**ublic **K**ey **I**nfrastructure

Cryptography (cont.) - PKI

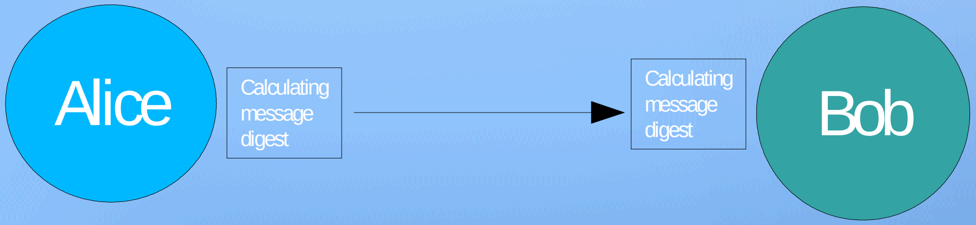
- Private key – not shared, used to decrypt messages encrypted with the public key.
- Public key – shared, others can encrypt messages with this key and only the private key can decrypt it.



- PKI is used by the asymmetric cryptography. Prevents “In-the-middle” attack, since no one in the middle has the private keys needed to decrypt the messages.

Digital signatures

- Used to ensure message integrity. Attached to each message sent through SSL.
- Digital signature consists of:
  - Hashed message digest – checksum of the message, hard to reverse.
  - Public key information.
- If Alice and Bob's message digest are not equal. Message integrity is not kept.



Certificates

- How can Alice trust Bob's server?
- Certificates are used to authenticate servers. It is a digital document that will attest to the binding of a public key. Help prevent someone to impersonate the server with a false key.
- SSL uses X.509 certificate standard. Contains information about entity, name and public key. This information is then validated by a CA.
- CA – **C**ertificate **A**uthority, trusted third party. For example VeriSign, 995$/year.
- Sometimes the CA is not very well known. This CA can be validated by a more well known CA. Certificate chaining.

Client/server

- The client initiates SSL traffic.
- The server responds, negotiates cipher suites.
- SSL uses three protocols:
  - Handshake – the client automatically authenticates the server. The server has the option of not authenticating the client. Ciphers are negotiated. Uses symmetric cryptography, but symmetric key is sent with PKI.
  - Record – All SSL messages are encapsulated into the Record protocol. This includes the handshake and the alert.
  - Alert – if server or client detects an error, an alert is sent. Three types; warning, critical and fatal. If fatal, SSL transaction is terminated.

Working with OpenSSL

- Generating a private key, example:

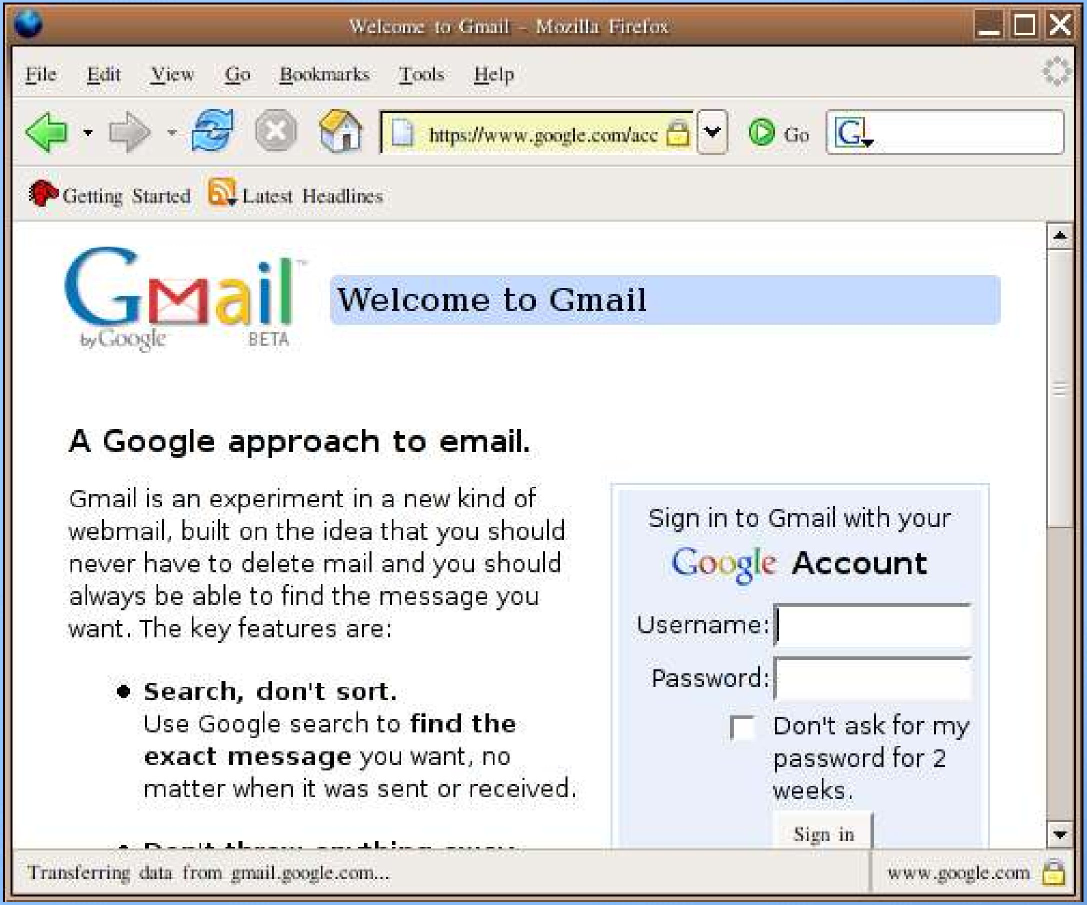
  -
    ```
    > openssl genrsa -des3 -out filename.key 1024
    ```
- CSR – **C**ertificate **S**igning **R**equest, send CA enough information to create certificate without sending the entire private key, example:

  -
    ```
    > openssl req -new -key filename.key -out filename.csr
    ```
- Creating a certificate, example:

  -
    ```
    > openssl req -new -key filename.key -x509 -out filename.crt
    ```

SSL in action

- SSL enabled web server. HTTPS://
  - Gmail: https://www.google.com/accounts/ServiceLogin..


Welcome to Gmail - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://www.google.com/acc
Go
Getting Started
Latest Headlines
Gmail by Google BETA

Welcome to Gmail

A Google approach to email.

Gmail is an experiment in a new kind of webmail, built on the idea that you should never have to delete mail and you should always be able to find the message you want. The key features are:

- **Search, don't sort.** Use Google search to **find the exact message** you want, no matter when it was sent or received.

Sign in to Gmail with your Google Account

Username:

Password:

Don't ask for my password for 2 weeks.

Sign in

Transferring data from gmail.google.com...

www.google.com

SSL in action (cont.)

- SSL enabled FTP. SFTP
- SSL enable remote login. SSH – **S**ecure **S**hell.
- Developing applications with SSL:
  - javax.net.ssl
- Tunneling, wrapping sockets inside SSL sockets. This technique allows you to secure all traffic at Application level.
  - Stunnel