Standards and Protocols

Computer programmes store data in a particular format. Data is stored in binary and a programme will need to be able to recognize the code in order to operate properly. If there were no standards for coding computer data, one programme would not be able to interpret the data created by another. Portability of data, without standards, would be impossible. In the early days of computing there were a variety of standards. In effect computer languages were different, ASCII and EBCDIC. The International Standards Organisation (OSI) lays down ICT standards for file formats. E.g. JPEG, CSV, BMP etc.

The problem with standards is that as technology develops, these become out of date. To get international agreement on new standards is tricky and time consuming. Further problems include:

- Open systems based on old standards may be unacceptably slow.
- The full power of the machine may not be available and therefore increase reduced functionality or performance.
- Bespoke software designed for one hardware platform and ignoring standards, might make better use of particular hardware.

There are some very strict rules for web addresses—how they are put together, what each part does, etc. We need to look at two of these parts: the protocol, and the domain name.

The protocol

Protocols are the formal rules and procedures which need to be followed to allow data to be transmitted, received and correctly interpreted. Protocols allow the use of open systems, whereby machines from different manufacturers can communicate.

The protocol is the very first part of the web address. It's the http:// or https:// part. You generally don't need to type it in when you go to a web site, because most browsers will add it for you. These are two different protocols used for web pages and several other types of data. **A protocol**, in this context, is a description of how to introduce yourself properly, the browser equivalent of knowing what words to say to introduce yourself, and ask for a book from a librarian or a fried banana from a Thai street vendor. In the web world, the dominant protocol is **Hyper-Text Transfer Protocol, or HTTP**. The Hyper-Text is your web page; the transfer protocol tells your browser how to get it. **HTTPS is Hyper-Text Transfer Protocol over SSL.**

SSL stands for Secure Sockets Layer, and lately it's been renamed to Transport Layer Security, or TLS. SSL/TLS is an entire framework that provides two things: encryption and authentication. Encryption hides your data between the web site and your browser, preventing anyone from intercepting it along the way. Authentication lets you verify that the server you're visiting is really the server it says it is, and not some other server taking its place. So the first thing to check in the web address is, are you visiting a page that is protected by SSL/TLS? And you can tell this by looking at the protocol in the address bar of your browser—is it https, or http?

FTP or File Transfer Protocol allows for fast transfer of files to and from the Internet and checks for and corrects any errors during the transfer. Sometimes a programme is used for this transfer (such as cute ftp.) but often it is done automatically without the user knowing that it has happened.

Post Office Protocol 3 (POP3) defines the standards for the transfer of email between computers.

Point to Point Protocol (PPP) defines the communication between two directly connected computers. (Usually between the user, his telephone line, and the ISP)

Transmission Control Protocol/Internet Protocol (TCP/IP) A set of agreed standards (part of the OSI model) which allows providers and users to communicate with each other whatever hardware is being used. See the model at the end of these notes.

Protocols also include specifications for:

- Speed of transmission - depends on the speed of the modem or the line;
- Method of transmission - serial or parallel;
- Computer code used - ASCII or EBCDIC;
- Parity used - odd or even.

The speed of transmission is related to the bandwidth of the channel, for example an optical fibre, a wire, or a radio link.

Baseband carries one signal at a time, as 0 or 1 (off or on). The signals can travel very fast, but can only travel short distances. Booster equipment is needed every 300 metres.

Broadband carries multiple signals on carrier waves with the signals 0 and 1 carried as variations on the waves.

The speed at which data is sent is called the **baud rate**, measured in bits per second (bps). The character consists of 7 or 8 bits with a start and stop bit. Each character requires 10 bits, so a 56 k modem transmits about 5600 characters per second. Telephone lines operate on analogue signals (waves) rather than digital (on or off). Therefore a **modem** is needed which operates between 9600 and 56 000 bps.

A broadband modem transmits data at very much higher speeds.

Data can be transmitted one bit after another down a single channel. This is called **serial transmission**. Serial transmission can be **asynchronous**, where each character has a start and a stop code. The computers do not need to be kept in step, but the speed of transmission is low. In **synchronous** transmission the computers are kept in time, and the data are sent in timed sequences. This allows much faster transmission.

In **parallel transmission** all the bits that make up a character with a parity bit are transmitted simultaneously. This allows for rapid transmission of data, but requires each bit to have its own wire to travel in. Hence parallel cables are thick. Parallel transmission can only be done up to about 5 metres.

An example of a parallel cable is that between a computer and a printer. Many computers are nowadays connected to peripherals by a Universal Serial Bus (USB).

The domain name

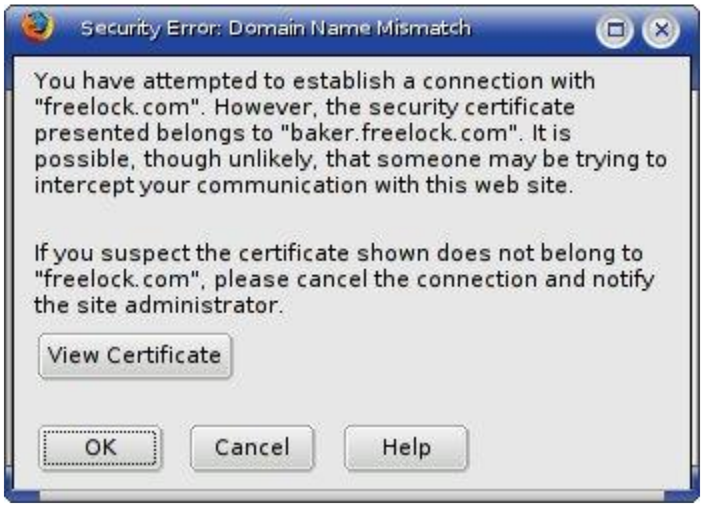
The other part of the web address to look at is the domain name. The domain name is everything after the https:// part up to the next forward slash ("/"). A domain name is broken up into domain "parts." The further right you go in the domain name, the more significant the part is.

.com is called the Top-Level Domain, or TLD, of this domain name. Each country has its own two-letter TLD, such as .us, .uk, .au, .tv, and then there are the three-letter TLDs we're all familiar with: .com, .net, .org, .edu, .gov, .biz and a few other less common ones. These top-level domains are controlled by a designated registry, and copied into what are called the Root Domain Name Servers. There are 14 of them, scattered around the world.

Moving to the left in the domain name, if the actual domain we're looking at is freelock.com. When your browser asks for www.freelock.com, it first goes to the Root Domain Name Servers to ask for where to find the directory for the domain freelock.com. The Root Domain Name Servers tell your browser to go to the IP address 69.55.225.251. Your browser then asks the name server where to find www, which could be an actual computer, or it could be another domain part.

SSL/TLS authentication

SSL/TLS does two things: encrypts traffic, and authenticates the server at the other end. The encryption part is simple, from a user point of view—if you see the lock icon in the bottom right corner of the window, the connection is encrypted. But without authentication, you could just be talking very privately with the garage-based scam artist posing as your bank.


Security Error: Domain Name Mismatch

You have attempted to establish a connection with "freelock.com". However, the security certificate presented belongs to "baker.freelock.com". It is possible, though unlikely, that someone may be trying to intercept your communication with this web site.

If you suspect the certificate shown does not belong to "freelock.com", please cancel the connection and notify the site administrator.

Figure 2. Firefox warns you if there's a problem authenticating the SSL certificate

That's why authentication is important. Authentication provides some assurance that you're at the real Paypal.com, and not a scam site. It works by checking something called a certificate that the web server presents to your browser, before sending any data. Your browser does a number of checks on the certificate, and if it appears to be valid, and matches the domain name in the address bar, it shows the lock icon in the status bar and gets the page. If it detects anything wrong, it gives you an authentication warning.

You've probably seen authentication warnings in your browser. Figure 2 shows one. There are several things your browser checks to determine if the certificate is legitimate: Does the domain name in the certificate match the domain name in the address bar? Has the certificate expired? Is the certificate signed by a trusted authority? Is the signing authority certificate valid, and current?

A valid certificate does not guarantee that a site is legitimate

Just like a domain name, anybody can get a certificate. It costs a little more—between $35 and $800 a year, depending on the certificate authority—but it's easy to do. Anybody can run their own certificate authority, but if the signing certificate isn't built into browsers visiting sites signed with it, you'll get warned when you visit the site A valid certificate provides a guarantee that somebody you trust (the certificate authority) has verified that the web site you're visiting really is the one in your browser's address bar.

SSL warnings would alert you if your DNS server has been hijacked, and you're visiting a fake site. It will also tell you if a web master is too cheap to buy a certificate from a trusted authority, or if they've been lazy renewing their certificates.

What's a certificate?

Public Key Encryption made secure electronic communications possible. Public Key Encryption involves two keys, or codes used to encrypt or decrypt data. One key is public, shared with the world at large. The other is secret, only stored on your computer. In public key encryption, whatever you encrypt with the public key can only be decrypted by the secret key—you cannot even decrypt it with the public key you used to encrypt it.

This may sound counter-intuitive, but think about the difference between multiplication and division—it's much easier to multiply two large numbers than to divide one from the other. This is the underlying principle that makes public key encryption possible.

In the other direction, you can use a secret key to digitally sign a chunk of data, and verify the signature with the public key. This turns out to be a very useful thing to use to guarantee the authenticity of a message.

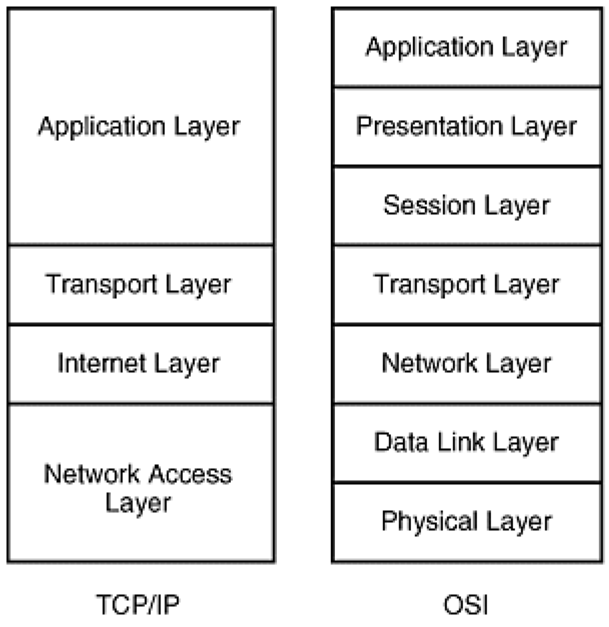
A certificate is a public key, combined with information about the owner of the corresponding secret key, digitally signed by somebody else—the Certificate Authority. So when somebody wants to run an encrypted web server, here's what they do:

- Create a brand new pair of keys, public and secret.
- Send the public key, along with name, address, site name, and other details to a certificate authority, such as VeriSign, GeoTrust, Thawte, or their local webmaster as a certificate signing request.
- The certificate authority uses their secret key to sign the certificate signing request, and the result is a signed certificate.
- The site owner installs the secret key and the signed certificate on the server.

Now, when you visit the site, here's what happens:

- Your browser connects to the server, and asks for its certificate.
- Your browser verifies the signature of the certificate authority. If it doesn't recognize the certificate authority, or the details in the certificate do not match the web address or have expired, or anything else weird, it pops up a warning.
- If the certificate is properly verified, or if you tell your browser to go ahead even though it couldn't verify the signature, your browser creates a new, random symmetrical key, encrypts it with the public key in the certificate, and sends it back to the server.
- From then on, both the server and your web browser have a big, shared key they use to encrypt all data going back and forth. Your browser will show the lock icon, so that you can easily see that your surfing is protected.

The OSI Model (TCP/IP)


Application Layer
Transport Layer
Internet Layer
Network Access Layer
TCP/IP
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer
OSI