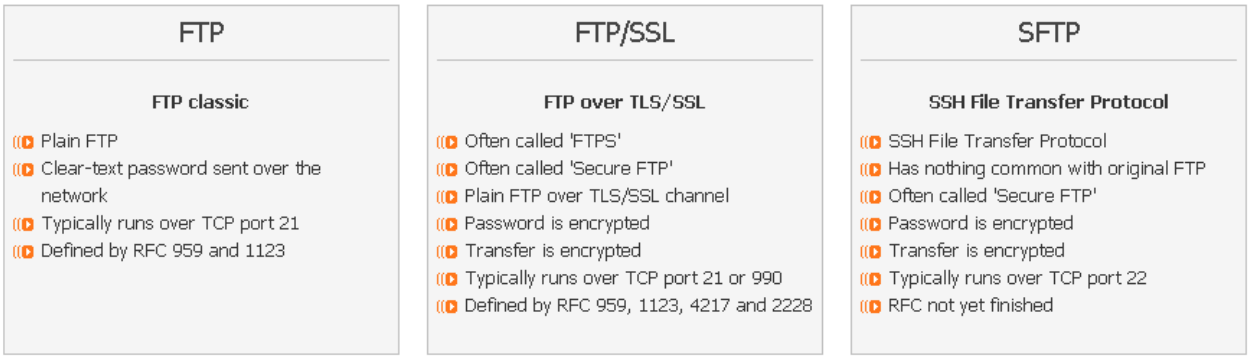
WHAT'S THE DIFFERENCE BETWEEN SECURE FTP, FTP/SSL, SFTP, FTPS, FTP, SCP…?

There are several different secure file transfer protocols that are, unfortunately, named in a very confusing way that often makes it difficult to distinguish one from another. The aim of this page is to provide some guidelines to make it easier to determine which is which.



COMMUNICATION PROTOCOLS

Basically, there are the following file transfer protocols around:

FTP – the plain old FTP protocol that has been around since **1970s**. The acronym stands for "File Transfer Protocol". It usually runs over **TCP port 21**.

SFTP – another, completely different file transfer protocol that has nothing to do with FTP. SFTP runs over an SSH session, usually on **TCP port 22**. It has been around since late 1990s. The acronym actually stands for "SSH File Transfer Protocol".

SCP – a variant of **BSD** rcp utility that transfers files over **SSH** session. The SCP protocol has been mostly superseded by the more comprehensive **SFTP** protocol and some implementations of the "**scp**" utility actually use SFTP instead.

SECURE COMMUNICATION LAYERS

Additionally, there are the following two secure communication layers:

SSH – a protocol that allows establishing a secure channel between the local and the remote computer. Serves as an underlying channel for associated protocols such as secure shell, port forwarding, SFTP or SCP. While it is possible to run the (***slightly modified***) plain old FTP protocol over SSH, this is not very common, fortunately. File transfer over SSH is nearly always done using **SFTP** or SCP.

TLS – this is almost generally known primarily by its old name – **SSL** - and provides a way of securing otherwise unsecure protocols such as **HTTP**, **SMTP**, **POP3** or **FTP**. Please note that SSL 3.1 is called TLS 1.0, and therefore TLS 1.0 is a newer version of the protocol than SSL 3.0, despite the lower version number.

HTTP over SSL is often called **HTTPS**, and **FTP over SSL** is often called **FTPS** and has two variants, explicit (starts as an unencrypted FTP session and is secured on client request) and ***implicit*** (is secured right from the beginning and therefore needs a separate TCP port, usually **990**).

The ***implicit*** mode is deprecated, but still widely used.

SECURE FILE TRANSFER PROTOCOLS, OR FITTING IT ALL TOGETHER

In an ideal world, the information above should be just enough. Unfortunately, this is not the case. The file transfer protocols are also referred to by other names, and even the names that only refer to a one single protocol are often mistakenly used for the wrong protocol by (***understandably***) confused authors.

FTP – should be only used for the plain old FTP protocol.

SFTP – should be only used for SFTP, the SSH file transfer protocol. However, people often shorten Secure FTP into SFTP – this is not correct, because the **S** in **SFTP** does not stand for Secure, but for **SSH**.

SFTP2 – this confusing name is used by some vendors to highlight the obvious fact that their **SFTP** protocol runs over **SSH2**. For all practical purposes, consider this to be a synonym of SFTP, because SSH1 has been deprecated for many years.

Secure FTP – this name is the most confusing, because it is used to refer to either of the two different protocols. Whenever this name is used, it is necessary to specify whether the SSH-based or SSL-based file transfer protocol is meant.

SSH FTP, FTP over SSH – fortunately, these names are not used very often. They usually refer to SFTP, the SSH file transfer protocol. Even though it is possible to run the (***slightly modified***) plain old **FTP** protocol over SSH, this is not very common.

FTP/SSL, FTP/TLS, **FTPover SSL**, **FTP over TLS, FTPS** – should be only used for FTP over TLS/SSL.
SFTP over SSL – although the **SFTP** protocol can utilize any underlying data stream, in practice **SFTP** over anything other that SSH is very rare. It is much more likely the term was used by mistake in place of either "**SFTP over SSH**" or "**FTP over SSL**".

SCP – should be only used for scp protocol/utility, a variant of BSD **rcp**. Some applications with **SCP** in its name now use **SFTP** by default instead – examples of this practice are **WinSCP** application and **scp2** utility.

TFTP is yet another file transfer protocol different from any of above.