**Hypertext Transfer Protocol Secure** (**HTTPS**) is an extension of the [Hypertext Transfer Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (HTTP). It is used for [secure communication](https://en.wikipedia.org/wiki/Secure_communications) over a [computer network](https://en.wikipedia.org/wiki/Network_operating_system), and is widely used on the Internet.[[1]](https://en.wikipedia.org/wiki/HTTPS#cite_note-1)[[2]](https://en.wikipedia.org/wiki/HTTPS#cite_note-2) In HTTPS, the [communication protocol](https://en.wikipedia.org/wiki/Communication_protocol) is encrypted using [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) (TLS) or, formerly, Secure Sockets Layer (SSL). The protocol is therefore also referred to as **HTTP over TLS**,[[3]](https://en.wikipedia.org/wiki/HTTPS#cite_note-3) or **HTTP over SSL**.

The principal motivations for HTTPS are [authentication](https://en.wikipedia.org/wiki/Authentication) of the accessed [website](https://en.wikipedia.org/wiki/Website), and protection of the [privacy](https://en.wikipedia.org/wiki/Information_privacy) and [integrity](https://en.wikipedia.org/wiki/Data_integrity) of the exchanged data while in transit. It protects against [man-in-the-middle attacks](https://en.wikipedia.org/wiki/Man-in-the-middle_attack), and the bidirectional [encryption](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation) of communications between a client and server protects the communications against [eavesdropping](https://en.wikipedia.org/wiki/Eavesdropping) and [tampering](https://en.wikipedia.org/wiki/Tamper-evident#Tampering).[[4]](https://en.wikipedia.org/wiki/HTTPS#cite_note-httpse-4)[[5]](https://en.wikipedia.org/wiki/HTTPS#cite_note-5) In practice, this provides a reasonable assurance that one is communicating with the intended website without interference from attackers.

The authentication aspect of HTTPS requires a trusted third party to sign server-side [digital certificates](https://en.wikipedia.org/wiki/Public_key_certificate). This was historically an expensive operation, which meant fully authenticated HTTPS connections were usually found only on secured payment transaction services and other secured corporate information systems on the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). In 2016, a campaign by the [Electronic Frontier Foundation](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation) with the support of web browser developers led to the protocol becoming more prevalent.[[6]](https://en.wikipedia.org/wiki/HTTPS#cite_note-6) HTTPS is now used more often by web users than the original non-secure HTTP, primarily to protect page authenticity on all types of websites; secure accounts; and to keep user communications, identity, and web browsing private.[[7]](https://en.wikipedia.org/wiki/HTTPS#cite_note-7)

**Overview**

Further information: [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)

[](https://en.wikipedia.org/wiki/File:Internet2.jpg)

[URL](https://en.wikipedia.org/wiki/URL) beginning with the HTTPS scheme and the [WWW](https://en.wikipedia.org/wiki/World_Wide_Web) domain name label

The [Uniform Resource Identifier](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier) (URI) scheme *HTTPS* has identical usage syntax to the HTTP scheme. However, HTTPS signals the browser to use an added encryption layer of SSL/TLS to protect the traffic. SSL/TLS is especially suited for HTTP, since it can provide some protection even if only one side of the communication is [authenticated](https://en.wikipedia.org/wiki/Authentication). This is the case with HTTP transactions over the Internet, where typically only the [server](https://en.wikipedia.org/wiki/Web_server) is authenticated (by the client examining the server's [certificate](https://en.wikipedia.org/wiki/Public_key_certificate)).

HTTPS creates a secure channel over an insecure network. This ensures reasonable protection from [eavesdroppers](https://en.wikipedia.org/wiki/Eavesdropping) and [man-in-the-middle attacks](https://en.wikipedia.org/wiki/Man-in-the-middle_attack), provided that adequate [cipher suites](https://en.wikipedia.org/wiki/Cipher_suite) are used and that the server certificate is verified and trusted.

Because HTTPS piggybacks HTTP entirely on top of TLS, the entirety of the underlying HTTP protocol can be encrypted. This includes the request URL (which particular web page was requested), query parameters, headers, and cookies (which often contain identifying information about the user). However, because website addresses and [port](https://en.wikipedia.org/wiki/Port_(computer_networking)) numbers are necessarily part of the underlying [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP) protocols, HTTPS cannot protect their disclosure. In practice this means that even on a correctly configured web server, eavesdroppers can infer the IP address and port number of the web server, and sometimes even the domain name (e.g. www.example.org, but not the rest of the URL) that a user is communicating with, along with the amount of data transferred and the duration of the communication, though not the content of the communication.[[4]](https://en.wikipedia.org/wiki/HTTPS#cite_note-httpse-4)

Web browsers know how to trust HTTPS websites based on [certificate authorities](https://en.wikipedia.org/wiki/Certificate_authority) that come pre-installed in their software. Certificate authorities are in this way being trusted by web browser creators to provide valid certificates. Therefore, a user should trust an HTTPS connection to a website [if and only if](https://en.wikipedia.org/wiki/If_and_only_if) all of the following are true:

* The user trusts that the browser software correctly implements HTTPS with correctly pre-installed certificate authorities.
* The user trusts the certificate authority to vouch only for legitimate websites.
* The website provides a valid certificate, which means it was signed by a trusted authority.
* The certificate correctly identifies the website (e.g., when the browser visits "<https://example.com>", the received certificate is properly for "example.com" and not some other entity).
* The user trusts that the protocol's encryption layer (SSL/TLS) is sufficiently secure against eavesdroppers.

HTTPS is especially important over insecure networks and networks that may be subject to tampering. Insecure networks, such as public [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) access points, allow anyone on the same local network to [packet-sniff](https://en.wikipedia.org/wiki/Packet_analyzer) and discover sensitive information not protected by HTTPS. Additionally, some free-to-use and paid [WLAN](https://en.wikipedia.org/wiki/Wireless_LAN) networks have been observed tampering with webpages by engaging in [packet injection](https://en.wikipedia.org/wiki/Packet_injection) in order to serve their own ads on other websites. This practice can be exploited maliciously in many ways, such as by injecting [malware](https://en.wikipedia.org/wiki/Malware) onto webpages and stealing users' private information.[[8]](https://en.wikipedia.org/wiki/HTTPS#cite_note-8)

HTTPS is also important for connections over the [Tor anonymity network](https://en.wikipedia.org/wiki/Tor_(anonymity_network)), as malicious Tor nodes could otherwise damage or alter the contents passing through them in an insecure fashion and inject malware into the connection. This is one reason why the [Electronic Frontier Foundation](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation) and the Tor project started the development of [HTTPS Everywhere](https://en.wikipedia.org/wiki/HTTPS_Everywhere),[[4]](https://en.wikipedia.org/wiki/HTTPS#cite_note-httpse-4) which is included in the Tor Browser Bundle.[[9]](https://en.wikipedia.org/wiki/HTTPS#cite_note-9)

As more information is revealed about global [mass surveillance](https://en.wikipedia.org/wiki/Mass_surveillance) and criminals stealing personal information, the use of HTTPS security on all websites is becoming increasingly important regardless of the type of Internet connection being used.[[10]](https://en.wikipedia.org/wiki/HTTPS#cite_note-10)[[11]](https://en.wikipedia.org/wiki/HTTPS#cite_note-11) Even though [metadata](https://en.wikipedia.org/wiki/Metadata) about individual pages that a user visits might not be considered sensitive, when aggregated it can reveal a lot about the user and compromise the user's privacy.[[12]](https://en.wikipedia.org/wiki/HTTPS#cite_note-12)[[13]](https://en.wikipedia.org/wiki/HTTPS#cite_note-13)[[14]](https://en.wikipedia.org/wiki/HTTPS#cite_note-deployhttpscorrectly-14)

Deploying HTTPS also allows the use of [HTTP/2](https://en.wikipedia.org/wiki/HTTP/2) (or its predecessor, the now-deprecated protocol [SPDY](https://en.wikipedia.org/wiki/SPDY)), which is a new generation of HTTP designed to reduce page load times, size, and latency.

It is recommended to use [HTTP Strict Transport Security](https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security) (HSTS) with HTTPS to protect users from man-in-the-middle attacks, especially [SSL stripping](https://en.wikipedia.org/wiki/Moxie_Marlinspike#SSL_stripping).[[14]](https://en.wikipedia.org/wiki/HTTPS#cite_note-deployhttpscorrectly-14)[[15]](https://en.wikipedia.org/wiki/HTTPS#cite_note-15)

HTTPS should not be confused with the seldom-used [Secure HTTP](https://en.wikipedia.org/wiki/Secure_Hypertext_Transfer_Protocol) (S-HTTP) specified in [RFC 2660](https://tools.ietf.org/html/rfc2660).

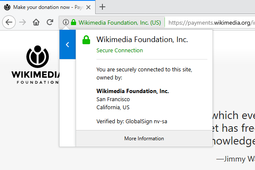
**Usage in websites**

As of April 2018, 33.2% of Alexa top 1,000,000 websites use HTTPS as default,[[16]](https://en.wikipedia.org/wiki/HTTPS#cite_note-16) 57.1% of the Internet's 137,971 most popular websites have a secure implementation of HTTPS,[[17]](https://en.wikipedia.org/wiki/HTTPS#cite_note-17) and 70% of page loads (measured by Firefox Telemetry) use HTTPS.[[18]](https://en.wikipedia.org/wiki/HTTPS#cite_note-18)

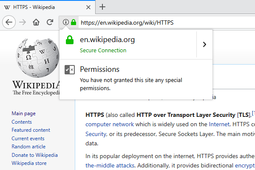
**Browser integration**

Most [browsers](https://en.wikipedia.org/wiki/Web_browser) display a warning if they receive an invalid certificate. Older browsers, when connecting to a site with an invalid certificate, would present the user with a [dialog box](https://en.wikipedia.org/wiki/Dialog_box) asking whether they wanted to continue. Newer browsers display a warning across the entire window. Newer browsers also prominently display the site's security information in the [address bar](https://en.wikipedia.org/wiki/Address_bar). [Extended validation certificates](https://en.wikipedia.org/wiki/Extended_validation_certificate) turn the address bar green in newer browsers. Most browsers also display a warning to the user when visiting a site that contains a mixture of encrypted and unencrypted content. Additionally, many [web filters](https://en.wikipedia.org/wiki/Content-control_software) return a security warning when visiting prohibited websites.

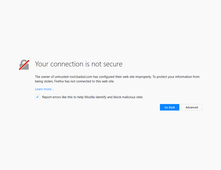
Comparison between different kinds of [SSL/TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security) certificates  
(Using [Firefox](https://en.wikipedia.org/wiki/Firefox) as an example)

* [](https://en.wikipedia.org/wiki/File:Wikimedia_donation_page_with_extended_validation_certificate_in_firefox.png)

Many web browsers, including Firefox (shown here), use the [address bar](https://en.wikipedia.org/wiki/Address_bar) to tell the user that their connection is secure, often by coloring the background.

* [](https://en.wikipedia.org/wiki/File:SSL_Certificate_Info_Box_In_Firefox.png)

When accessing a site only with a common certificate, the address bar of [Firefox](https://en.wikipedia.org/wiki/Firefox) turns green. For some other [browsers](https://en.wikipedia.org/wiki/Web_browser), a "lock" sign may appear.

* [](https://en.wikipedia.org/wiki/File:Firefox_blocking_a_site_with_bad_TLS_configuration.png)

Most web browsers alert the user when visiting sites that have invalid security certificates.

The [Electronic Frontier Foundation](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation), opining that "In an ideal world, every web request could be defaulted to HTTPS", has provided an add-on called HTTPS Everywhere for [Mozilla Firefox](https://en.wikipedia.org/wiki/Mozilla_Firefox), [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome), [Chromium](https://en.wikipedia.org/wiki/Chromium_(web_browser)), and [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), that enables HTTPS by default for hundreds of frequently used websites.[[19]](https://en.wikipedia.org/wiki/HTTPS#cite_note-19)[[20]](https://en.wikipedia.org/wiki/HTTPS#cite_note-20)

**Security**

Main article: [Transport Layer Security § Security](https://en.wikipedia.org/wiki/Transport_Layer_Security#Security)

The security of HTTPS is that of the underlying TLS, which typically uses long-term [public](https://en.wikipedia.org/wiki/Public-key_cryptography) and private keys to generate a short-term [session key](https://en.wikipedia.org/wiki/Session_key), which is then used to encrypt the data flow between the client and the server. [X.509](https://en.wikipedia.org/wiki/X.509) certificates are used to authenticate the server (and sometimes the client as well). As a consequence, [certificate authorities](https://en.wikipedia.org/wiki/Certificate_authority) and [public key certificates](https://en.wikipedia.org/wiki/Public_key_certificate) are necessary to verify the relation between the certificate and its owner, as well as to generate, sign, and administer the validity of certificates. While this can be more beneficial than verifying the identities via a [web of trust](https://en.wikipedia.org/wiki/Web_of_trust), the [2013 mass surveillance disclosures](https://en.wikipedia.org/wiki/2013_mass_surveillance_disclosures) drew attention to certificate authorities as a potential weak point allowing [man-in-the-middle attacks](https://en.wikipedia.org/wiki/Man-in-the-middle_attack).[[21]](https://en.wikipedia.org/wiki/HTTPS#cite_note-21)[[22]](https://en.wikipedia.org/wiki/HTTPS#cite_note-22) An important property in this context is [forward secrecy](https://en.wikipedia.org/wiki/Forward_secrecy), which ensures that encrypted communications recorded in the past cannot be retrieved and decrypted should long-term secret keys or passwords be compromised in the future. Not all web servers provide forward secrecy.[[23]](https://en.wikipedia.org/wiki/HTTPS#cite_note-ecdhe-23)[[*needs update*](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Dates_and_numbers#Chronological_items)]

For HTTPS to be effective, a site must be completely hosted over HTTPS. If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable to attacks and surveillance. Additionally, [cookies](https://en.wikipedia.org/wiki/HTTP_cookie) on a site served through HTTPS must have the [secure attribute](https://en.wikipedia.org/wiki/Secure_cookie) enabled. On a site that has sensitive information on it, the user and the session will get exposed every time that site is accessed with HTTP instead of HTTPS.[[14]](https://en.wikipedia.org/wiki/HTTPS#cite_note-deployhttpscorrectly-14)

**Technical**

**Difference from HTTP**

HTTPS [URLs](https://en.wikipedia.org/wiki/URL) begin with "https://" and use [port](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers) 443 by default, whereas, [HTTP](https://en.wikipedia.org/wiki/HTTP) URLs begin with "http://" and use port 80 by default.

HTTP is not encrypted and thus is vulnerable to [man-in-the-middle](https://en.wikipedia.org/wiki/Man-in-the-middle) and [eavesdropping attacks](https://en.wikipedia.org/wiki/Eavesdropping_attack), which can let attackers gain access to website accounts and sensitive information, and modify webpages to inject [malware](https://en.wikipedia.org/wiki/Malware) or advertisements. HTTPS is designed to withstand such attacks and is considered secure against them (with the exception of HTTPS implementations that use deprecated versions of SSL).

**Network layers**

HTTP operates at the highest layer of the [TCP/IP model](https://en.wikipedia.org/wiki/TCP/IP_model), the [Application layer](https://en.wikipedia.org/wiki/Application_layer); as does the [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security) security protocol (operating as a lower sublayer of the same layer), which encrypts an HTTP message prior to transmission and decrypts a message upon arrival. Strictly speaking, HTTPS is not a separate protocol, but refers to the use of ordinary [HTTP](https://en.wikipedia.org/wiki/HTTP) over an [encrypted](https://en.wikipedia.org/wiki/Encryption) SSL/TLS connection.

HTTPS encrypts all message contents, including the HTTP headers and the request/response data. With the exception of the possible [CCA](https://en.wikipedia.org/wiki/Chosen-ciphertext_attack) cryptographic attack described in the [limitations](https://en.wikipedia.org/wiki/HTTPS#Limitations) section below, an attacker should at most be able to discover that a connection is taking place between two parties, along with their domain names and IP addresses.

**Server setup**

To prepare a web server to accept HTTPS connections, the administrator must create a [public key certificate](https://en.wikipedia.org/wiki/Public_key_certificate) for the web server. This certificate must be signed by a trusted [certificate authority](https://en.wikipedia.org/wiki/Certificate_authority) for the web browser to accept it without warning. The authority certifies that the certificate holder is the operator of the web server that presents it. Web browsers are generally distributed with a list of [signing certificates of major certificate authorities](https://en.wikipedia.org/wiki/Root_certificate) so that they can verify certificates signed by them.

**Acquiring certificates**

A number of commercial [certificate authorities](https://en.wikipedia.org/wiki/Certificate_authority) exist, offering paid-for SSL/TLS certificates of a number of types, including [Extended Validation Certificates](https://en.wikipedia.org/wiki/Extended_Validation_Certificate).

[Let's Encrypt](https://en.wikipedia.org/wiki/Let%27s_Encrypt), launched in April 2016,[[24]](https://en.wikipedia.org/wiki/HTTPS#cite_note-softpedia-launch-24) provides free and automated service that delivers basic SSL/TLS certificates to websites.[[25]](https://en.wikipedia.org/wiki/HTTPS#cite_note-25) According to the [Electronic Frontier Foundation](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation), Let's Encrypt will make switching from HTTP to HTTPS "as easy as issuing one command, or clicking one button."[[26]](https://en.wikipedia.org/wiki/HTTPS#cite_note-26) The majority of web hosts and cloud providers now leverage Let's Encrypt, providing free certificates to their customers.

**Use as access control**

The system can also be used for client [authentication](https://en.wikipedia.org/wiki/Authentication) in order to limit access to a web server to authorized users. To do this, the site administrator typically creates a certificate for each user, which the user loads into their browser. Normally, the certificate contains the name and e-mail address of the authorized user and is automatically checked by the server on each connection to verify the user's identity, potentially without even requiring a password.

**In case of compromised secret (private) key**

|  |  |
| --- | --- |
|  | This section needs to be **updated**. The reason given is: Out of date. Please update this article to reflect recent events or newly available information. *(April 2019)* |

An important property in this context is [perfect forward secrecy](https://en.wikipedia.org/wiki/Forward_secrecy) (PFS). Possessing one of the long-term asymmetric secret keys used to establish an HTTPS session should not make it easier to derive the short-term session key to then decrypt the conversation, even at a later time. [Diffie–Hellman key exchange](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange) (DHE) and [Elliptic curve Diffie–Hellman](https://en.wikipedia.org/wiki/Elliptic_curve_Diffie%E2%80%93Hellman) key exchange (ECDHE) are in 2013 the only schemes known to have that property. In 2013, only 30% of Firefox, Opera, and Chromium Browser sessions used it, and nearly 0% of Apple's [Safari](https://en.wikipedia.org/wiki/Safari_(web_browser)) and [Microsoft Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) sessions.[[23]](https://en.wikipedia.org/wiki/HTTPS#cite_note-ecdhe-23) TLS 1.3, published in August 2018, dropped support for ciphers without forward secrecy. As of February 2020, 96.6% of web servers surveyed support some form of forward secrecy, and 52.1% will use forward secrecy with most browsers.[[27]](https://en.wikipedia.org/wiki/HTTPS#cite_note-27)

A certificate may be revoked before it expires, for example because the secrecy of the private key has been compromised. Newer versions of popular browsers such as [Firefox](https://en.wikipedia.org/wiki/Firefox),[[28]](https://en.wikipedia.org/wiki/HTTPS#cite_note-28) [Opera](https://en.wikipedia.org/wiki/Opera_(web_browser)),[[29]](https://en.wikipedia.org/wiki/HTTPS#cite_note-29) and [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) on [Windows Vista](https://en.wikipedia.org/wiki/Windows_Vista)[[30]](https://en.wikipedia.org/wiki/HTTPS#cite_note-30) implement the [Online Certificate Status Protocol](https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol) (OCSP) to verify that this is not the case. The browser sends the certificate's serial number to the certificate authority or its delegate via OCSP (Online Certificate Status Protocol) and the authority responds, telling the browser whether the certificate is still valid or not.[[31]](https://en.wikipedia.org/wiki/HTTPS#cite_note-31)The CA may also issue a [CRL](https://en.wikipedia.org/w/index.php?title=Certificate_Revocation_list&action=edit&redlink=1) to tell people that these certificates are revoked.

**Limitations**

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) encryption can be configured in two modes: *simple* and *mutual*. In simple mode, authentication is only performed by the server. The mutual version requires the user to install a personal [client certificate](https://en.wikipedia.org/wiki/Client_certificate) in the web browser for user authentication.[[32]](https://en.wikipedia.org/wiki/HTTPS#cite_note-32) In either case, the level of protection depends on the correctness of the [implementation](https://en.wikipedia.org/wiki/Implementation) of the software and the [cryptographic algorithms](https://en.wikipedia.org/wiki/Cipher) in use.

SSL/TLS does not prevent the indexing of the site by a [web crawler](https://en.wikipedia.org/wiki/Web_crawler), and in some cases the [URI](https://en.wikipedia.org/wiki/Uniform_resource_identifier) of the encrypted resource can be inferred by knowing only the intercepted request/response size.[[33]](https://en.wikipedia.org/wiki/HTTPS#cite_note-33) This allows an attacker to have access to the [plaintext](https://en.wikipedia.org/wiki/Plaintext) (the publicly available static content), and the [encrypted text](https://en.wikipedia.org/wiki/Ciphertext) (the encrypted version of the static content), permitting a [cryptographic attack](https://en.wikipedia.org/wiki/Chosen-ciphertext_attack).

Because [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security) operates at a protocol level below that of HTTP and has no knowledge of the higher-level protocols, TLS servers can only strictly present one certificate for a particular address and port combination.[[34]](https://en.wikipedia.org/wiki/HTTPS#cite_note-34) In the past, this meant that it was not feasible to use [name-based virtual hosting](https://en.wikipedia.org/wiki/Virtual_hosting#Name-based) with HTTPS. A solution called [Server Name Indication](https://en.wikipedia.org/wiki/Server_Name_Indication) (SNI) exists, which sends the hostname to the server before encrypting the connection, although many old browsers do not support this extension. Support for SNI is available since [Firefox](https://en.wikipedia.org/wiki/Firefox) 2, [Opera](https://en.wikipedia.org/wiki/Opera_(web_browser)) 8, [Apple Safari](https://en.wikipedia.org/wiki/Safari_(web_browser)) 2.1, [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) 6, and [Internet Explorer 7](https://en.wikipedia.org/wiki/Internet_Explorer_7) on [Windows Vista](https://en.wikipedia.org/wiki/Windows_Vista).[[35]](https://en.wikipedia.org/wiki/HTTPS#cite_note-35)[[36]](https://en.wikipedia.org/wiki/HTTPS#cite_note-36)[[37]](https://en.wikipedia.org/wiki/HTTPS#cite_note-37)

From an architectural point of view:

* An SSL/TLS connection is managed by the first front machine that initiates the TLS connection. If, for any reasons (routing, traffic optimization, etc.), this front machine is not the application server and it has to decipher data, solutions have to be found to propagate user authentication information or certificate to the application server, which needs to know who is going to be connected.
* For SSL/TLS with mutual authentication, the SSL/TLS session is managed by the first server that initiates the connection. In situations where encryption has to be propagated along chained servers, session timeOut management becomes extremely tricky to implement.
* Security is maximal with mutual SSL/TLS, but on the client-side there is no way to properly end the SSL/TLS connection and disconnect the user except by waiting for the server session to expire or by closing all related client applications.

A sophisticated type of [man-in-the-middle attack](https://en.wikipedia.org/wiki/Man-in-the-middle_attack) called SSL stripping was presented at the 2009 [Blackhat Conference](https://en.wikipedia.org/wiki/Black_Hat_Briefings). This type of attack defeats the security provided by HTTPS by changing the https: link into an http: link, taking advantage of the fact that few Internet users actually type "https" into their browser interface: they get to a secure site by clicking on a link, and thus are fooled into thinking that they are using HTTPS when in fact they are using HTTP. The attacker then communicates in clear with the client.[[38]](https://en.wikipedia.org/wiki/HTTPS#cite_note-38) This prompted the development of a countermeasure in HTTP called [HTTP Strict Transport Security](https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security).

HTTPS has been shown vulnerable to a range of [traffic analysis](https://en.wikipedia.org/wiki/Traffic_analysis) attacks. Traffic analysis attacks are a type of [side-channel attack](https://en.wikipedia.org/wiki/Side-channel_attack) that relies on variations in the timing and size of traffic in order to infer properties about the encrypted traffic itself. Traffic analysis is possible because SSL/TLS encryption changes the contents of traffic, but has minimal impact on the size and timing of traffic. In May 2010, a research paper by researchers from [Microsoft Research](https://en.wikipedia.org/wiki/Microsoft_Research) and [Indiana University](https://en.wikipedia.org/wiki/Indiana_University_Bloomington) discovered that detailed sensitive user data can be inferred from side channels such as packet sizes. The researchers found that, despite HTTPS protection in several high-profile, top-of-the-line web applications in healthcare, taxation, investment, and web search, an eavesdropper could infer the illnesses/medications/surgeries of the user, his/her family income, and investment secrets.[[39]](https://en.wikipedia.org/wiki/HTTPS#cite_note-39) Although this work demonstrated the vulnerability of HTTPS to traffic analysis, the approach presented by the authors required manual analysis and focused specifically on web applications protected by HTTPS.

The fact that most modern websites, including Google, Yahoo!, and Amazon, use HTTPS causes problems for many users trying to access public Wi-Fi hot spots, because a Wi-Fi hot spot login page fails to load if the user tries to open an HTTPS resource.[[40]](https://en.wikipedia.org/wiki/HTTPS#cite_note-40) Several websites, such as [neverssl.com](http://neverssl.com/) and [nonhttps.com](http://www.nonhttps.com), guarantee that they will always remain accessible by HTTP.[[41]](https://en.wikipedia.org/wiki/HTTPS#cite_note-41)[[42]](https://en.wikipedia.org/wiki/HTTPS#cite_note-42)

**History**

[Netscape Communications](https://en.wikipedia.org/wiki/Netscape_Communications) created HTTPS in 1994 for its [Netscape Navigator](https://en.wikipedia.org/wiki/Netscape_Navigator) web browser.[[43]](https://en.wikipedia.org/wiki/HTTPS#cite_note-43) Originally, HTTPS was used with the [SSL](https://en.wikipedia.org/wiki/Secure_Sockets_Layer) protocol. As SSL evolved into [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) (TLS), HTTPS was formally specified by [RFC 2818](https://tools.ietf.org/html/rfc2818) in May 2000. Google announced in February 2018 that its Chrome browser would mark HTTP sites as "Not Secure" after July 2018.[[44]](https://en.wikipedia.org/wiki/HTTPS#cite_note-44) This move was to encourage website owners to implement HTTPS, as an effort to secure the internet.

**See also**

* [Bullrun (decryption program)](https://en.wikipedia.org/wiki/Bullrun_(decryption_program)) – a secret anti-encryption program run by the US [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency)
* [Computer security](https://en.wikipedia.org/wiki/Computer_security)
* [curl-loader](https://en.wikipedia.org/wiki/Curl-loader)
* [Diameter protocol](https://en.wikipedia.org/wiki/Diameter_protocol)
* [HTTPsec](https://en.wikipedia.org/wiki/HTTPsec)
* [Moxie Marlinspike](https://en.wikipedia.org/wiki/Moxie_Marlinspike)
* [Opportunistic encryption](https://en.wikipedia.org/wiki/Opportunistic_encryption)
* [Stunnel](https://en.wikipedia.org/wiki/Stunnel)

**References**

 [*"Secure your site with HTTPS"*](https://support.google.com/webmasters/answer/6073543?hl=en)*. Google Support. Google, Inc.* [*Archived*](https://web.archive.org/web/20150301023624/https:/support.google.com/webmasters/answer/6073543?hl=en) *from the original on 2015-03-01. Retrieved 2018-10-20.*

  [*"What is HTTPS?"*](https://www.instantssl.com/ssl-certificate-products/https.html)*.* [*Comodo CA Limited*](https://en.wikipedia.org/wiki/Comodo_Group)*.* [*Archived*](https://web.archive.org/web/20150212105201/https:/www.instantssl.com/ssl-certificate-products/https.html) *from the original on 2015-02-12. Retrieved 2018-10-20. Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP [...]*

  *Network Working Group (May 2000).* [*"HTTP Over TLS"*](https://tools.ietf.org/html/rfc2818)*. The Internet Engineering Task Force.* [*Archived*](https://web.archive.org/web/20181031095731/https:/tools.ietf.org/html/rfc2818) *from the original on 2018-10-31. Retrieved 2018-10-20.*

  [*"HTTPS Everywhere FAQ"*](https://www.eff.org/https-everywhere/faq)*. 2016-11-08.* [*Archived*](https://web.archive.org/web/20181114011956/https:/www.eff.org/https-everywhere/faq/) *from the original on 2018-11-14. Retrieved 2018-10-20.*

  [*"Usage Statistics of Default protocol https for Websites, July 2019"*](https://w3techs.com/technologies/details/ce-httpsdefault/all/all)*. w3techs.com.* [*Archived*](https://web.archive.org/web/20190801134536/https:/w3techs.com/technologies/details/ce-httpsdefault/all/all) *from the original on 2019-08-01. Retrieved 2019-07-20.*

  [*"Encrypting the Web"*](https://www.eff.org/encrypt-the-web)*. Electronic Frontiers Foundation.* [*Archived*](https://web.archive.org/web/20191118094200/https:/www.eff.org/encrypt-the-web) *from the original on 18 November 2019. Retrieved 19 November 2019.*

  [*"HTTP, HTTPS, SSL Over TLS"*](https://www.belugacdn.com/blog/everything-you-need-to-know-about-https-ssltls/)*. November 2019.* [*Archived*](https://web.archive.org/web/20191213004928/https:/www.belugacdn.com/blog/everything-you-need-to-know-about-https-ssltls/) *from the original on 2019-12-13. Retrieved 2019-11-19.*

  [*"Hotel Wifi JavaScript Injection"*](https://justinsomnia.org/2012/04/hotel-wifi-javascript-injection/)*. JustInsomnia. 2012-04-03.* [*Archived*](https://web.archive.org/web/20181118154608/https:/justinsomnia.org/2012/04/hotel-wifi-javascript-injection/) *from the original on 2018-11-18. Retrieved 2018-10-20.*

  *The Tor Project, Inc.* [*"What is Tor Browser?"*](https://www.torproject.org/projects/torbrowser.html.en)*. TorProject.org.* [*Archived*](https://archive.is/20130717222709/https:/www.torproject.org/projects/torbrowser.html.en) *from the original on 2013-07-17. Retrieved 2012-05-30.*

  *Konigsburg, Eitan; Pant, Rajiv; Kvochko, Elena (2014-11-13).* [*"Embracing HTTPS"*](https://open.blogs.nytimes.com/2014/11/13/embracing-https/)*. The New York Times.* [*Archived*](https://web.archive.org/web/20190108190000/https:/open.blogs.nytimes.com/2014/11/13/embracing-https/) *from the original on 2019-01-08. Retrieved 2018-10-20.*

  *Gallagher, Kevin (2014-09-12).* [*"Fifteen Months After the NSA Revelations, Why Aren't More News Organizations Using HTTPS?"*](https://freedom.press/news-advocacy/fifteen-months-after-the-nsa-revelations-why-arenat-more-news-organizations-using-https/)*. Freedom of the Press Foundation.* [*Archived*](https://web.archive.org/web/20180810204919/https:/freedom.press/news-advocacy/fifteen-months-after-the-nsa-revelations-why-arenat-more-news-organizations-using-https/) *from the original on 2018-08-10. Retrieved 2018-10-20.*

  [*"HTTPS as a ranking signal"*](https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html)*. Google Webmaster Central Blog. Google Inc. 2014-08-06.* [*Archived*](https://web.archive.org/web/20181017052432/https:/webmasters.googleblog.com/2014/08/https-as-ranking-signal.html) *from the original on 2018-10-17. Retrieved 2018-10-20. You can make your site secure with HTTPS (Hypertext Transfer Protocol Secure) [...]*

  *Grigorik, Ilya; Far, Pierre (2014-06-26).* [*"Google I/O 2014 - HTTPS Everywhere"*](https://www.youtube.com/watch?v=cBhZ6S0PFCY)*. Google Developers.* [*Archived*](https://web.archive.org/web/20181120144918/https:/www.youtube.com/watch?v=cBhZ6S0PFCY) *from the original on 2018-11-20. Retrieved 2018-10-20.*

  [*"How to Deploy HTTPS Correctly"*](https://www.eff.org/https-everywhere/deploying-https)*. 2010-11-15.* [*Archived*](https://web.archive.org/web/20181010233702/https:/www.eff.org/https-everywhere/deploying-https) *from the original on 2018-10-10. Retrieved 2018-10-20.*

  [*"HTTP Strict Transport Security"*](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security)*. Mozilla Developer Network.* [*Archived*](https://web.archive.org/web/20181019171534/https:/developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security) *from the original on 2018-10-19. Retrieved 2018-10-20.*

  [*"HTTPS usage statistics on top 1M websites"*](https://statoperator.com/research/https-usage-statistics-on-top-websites/)*. StatOperator.com.* [*Archived*](https://web.archive.org/web/20190209055130/https:/statoperator.com/research/https-usage-statistics-on-top-websites/) *from the original on 2019-02-09. Retrieved 2018-10-20.*

  [*"Qualys SSL Labs - SSL Pulse"*](https://www.ssllabs.com/ssl-pulse/)*. www.ssllabs.com. 2018-04-03.* [*Archived*](https://web.archive.org/web/20171202155646/https:/www.ssllabs.com/ssl-pulse/) *from the original on 2017-12-02. Retrieved 2018-10-20.*

  [*"Let's Encrypt Stats"*](https://letsencrypt.org/stats/)*. LetsEncrypt.org.* [*Archived*](https://web.archive.org/web/20181019221028/https:/letsencrypt.org/stats/) *from the original on 2018-10-19. Retrieved 2018-10-20.*

  *Eckersley, Peter (2010-06-17).* [*"Encrypt the Web with the HTTPS Everywhere Firefox Extension"*](https://www.eff.org/deeplinks/2010/06/encrypt-web-https-everywhere-firefox-extension)*. EFF blog.* [*Archived*](https://web.archive.org/web/20181125102636/https:/www.eff.org/deeplinks/2010/06/encrypt-web-https-everywhere-firefox-extension) *from the original on 2018-11-25. Retrieved 2018-10-20.*

  [*"HTTPS Everywhere"*](https://www.eff.org/https-everywhere)*. EFF projects. 2011-10-07.* [*Archived*](https://web.archive.org/web/20110605022218/https:/www.eff.org/https-everywhere) *from the original on 2011-06-05. Retrieved 2018-10-20.*

  *Singel, Ryan (2010-03-24).* [*"Law Enforcement Appliance Subverts SSL"*](https://www.wired.com/2010/03/packet-forensics/)*. Wired.* [*Archived*](https://web.archive.org/web/20190117142906/https:/www.wired.com/2010/03/packet-forensics/) *from the original on 2019-01-17. Retrieved 2018-10-20.*

  *Schoen, Seth (2010-03-24).* [*"New Research Suggests That Governments May Fake SSL Certificates"*](https://www.eff.org/deeplinks/2010/03/researchers-reveal-likelihood-governments-fake-ssl)*. EFF.* [*Archived*](https://web.archive.org/web/20160104234608/https:/www.eff.org/deeplinks/2010/03/researchers-reveal-likelihood-governments-fake-ssl) *from the original on 2016-01-04. Retrieved 2018-10-20.*

  *Duncan, Robert (2013-06-25).* [*"SSL: Intercepted today, decrypted tomorrow"*](https://news.netcraft.com/archives/2013/06/25/ssl-intercepted-today-decrypted-tomorrow.html)*. Netcraft.* [*Archived*](https://web.archive.org/web/20181006021916/https:/news.netcraft.com/archives/2013/06/25/ssl-intercepted-today-decrypted-tomorrow.html) *from the original on 2018-10-06. Retrieved 2018-10-20.*

  *Cimpanu, Catalin (2016-04-12).* [*"Let's Encrypt Launched Today, Currently Protects 3.8 Million Domains"*](https://news.softpedia.com/news/let-s-encrypt-launched-today-currently-protects-3-8-million-domains-502857.shtml)*. Softpedia News.* [*Archived*](https://web.archive.org/web/20190209055129/https:/news.softpedia.com/news/let-s-encrypt-launched-today-currently-protects-3-8-million-domains-502857.shtml) *from the original on 2019-02-09. Retrieved 2018-10-20.*

  *Kerner, Sean Michael (2014-11-18).* [*"Let's Encrypt Effort Aims to Improve Internet Security"*](http://www.eweek.com/security/let-s-encrypt-effort-aims-to-improve-internet-security)*. eWeek.com. Quinstreet Enterprise. Retrieved 2018-10-20.*

  *Eckersley, Peter (2014-11-18).* [*"Launching in 2015: A Certificate Authority to Encrypt the Entire Web"*](https://www.eff.org/deeplinks/2014/11/certificate-authority-encrypt-entire-web)*.* [*Electronic Frontier Foundation*](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation)*.* [*Archived*](https://web.archive.org/web/20181118160126/https:/www.eff.org/deeplinks/2014/11/certificate-authority-encrypt-entire-web) *from the original on 2018-11-18. Retrieved 2018-10-20.*

  [*Qualys SSL Labs*](https://en.wikipedia.org/wiki/Qualys)*.* [*"SSL Pulse"*](https://web.archive.org/web/20190215213454/https:/www.ssllabs.com/ssl-pulse/)*. Archived from* [*the original*](https://www.ssllabs.com/ssl-pulse/) *(3 February 2019) on 15 February 2019. Retrieved 25 February 2019.*

  [*"Mozilla Firefox Privacy Policy"*](https://www.mozilla.org/en-US/privacy/)*.* [*Mozilla Foundation*](https://en.wikipedia.org/wiki/Mozilla_Foundation)*. 2009-04-27.* [*Archived*](https://web.archive.org/web/20181018063732/https:/www.mozilla.org/en-US/privacy/) *from the original on 2018-10-18. Retrieved 2018-10-20.*

  [*"Opera 8 launched on FTP"*](https://news.softpedia.com/news/Opera-8-launched-on-FTP-1330.shtml)*.* [*Softpedia*](https://en.wikipedia.org/wiki/Softpedia)*. 2005-04-19.* [*Archived*](https://web.archive.org/web/20190209055128/https:/news.softpedia.com/news/Opera-8-launched-on-FTP-1330.shtml) *from the original on 2019-02-09. Retrieved 2018-10-20.*

  *Lawrence, Eric (2006-01-31).* [*"HTTPS Security Improvements in Internet Explorer 7"*](https://technet.microsoft.com/en-us/windows/bb250503(v=vs.60))*.* [*MSDN*](https://en.wikipedia.org/wiki/MSDN)*.* [*Archived*](https://web.archive.org/web/20181020223806/https:/technet.microsoft.com/en-us/windows/bb250503(v=vs.60)) *from the original on 2018-10-20. Retrieved 2018-10-20.*

  *Myers, Michael; Ankney, Rich; Malpani, Ambarish; Galperin, Slava; Adams, Carlisle (1999-06-20).* [*"Online Certificate Status Protocol – OCSP"*](https://tools.ietf.org/html/rfc2560)*.* [*Internet Engineering Task Force*](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force)*.* [*Archived*](https://web.archive.org/web/20110825095059/http:/tools.ietf.org/html/rfc2560) *from the original on 2011-08-25. Retrieved 2018-10-20.*

  [*"Manage client certificates on Chrome devices – Chrome for business and education Help"*](https://support.google.com/chrome/a/answer/6080885?hl=en)*. support.google.com.* [*Archived*](https://web.archive.org/web/20190209055127/https:/support.google.com/chrome/a/answer/6080885?hl=en) *from the original on 2019-02-09. Retrieved 2018-10-20.*

  *Pusep, Stanislaw (2008-07-31).* [*"The Pirate Bay un-SSL"*](https://www.exploit-db.com/docs/english/13026-the-pirate-bay-un-ssl.pdf) *(PDF).* [*Archived*](https://web.archive.org/web/20180620001518/https:/www.exploit-db.com/docs/english/13026-the-pirate-bay-un-ssl.pdf) *(PDF) from the original on 2018-06-20. Retrieved 2018-10-20.*

  [*"SSL/TLS Strong Encryption: FAQ"*](https://httpd.apache.org/docs/2.0/ssl/ssl_faq.html#vhosts)*. apache.org.* [*Archived*](https://web.archive.org/web/20181019105423/http:/httpd.apache.org/docs/2.0/ssl/ssl_faq.html#vhosts) *from the original on 2018-10-19. Retrieved 2018-10-20.*

  *Lawrence, Eric (2005-10-22).* [*"Upcoming HTTPS Improvements in Internet Explorer 7 Beta 2"*](https://blogs.msdn.microsoft.com/ie/2005/10/22/upcoming-https-improvements-in-internet-explorer-7-beta-2/)*.* [*Microsoft*](https://en.wikipedia.org/wiki/Microsoft)*.* [*Archived*](https://web.archive.org/web/20180920113838/https:/blogs.msdn.microsoft.com/ie/2005/10/22/upcoming-https-improvements-in-internet-explorer-7-beta-2/) *from the original on 2018-09-20. Retrieved 2018-10-20.*

  [*"Server Name Indication (SNI)"*](https://blog.ebrahim.org/2006/02/21/server-name-indication-sni/)*. inside aebrahim's head. 2006-02-21.* [*Archived*](https://web.archive.org/web/20180810173628/https:/blog.ebrahim.org/2006/02/21/server-name-indication-sni/) *from the original on 2018-08-10. Retrieved 2018-10-20.*

  *Pierre, Julien (2001-12-19).* [*"Browser support for TLS server name indication"*](https://bugzilla.mozilla.org/show_bug.cgi?id=116169)*. Bugzilla. Mozilla Foundation.* [*Archived*](https://web.archive.org/web/20181008070112/https:/bugzilla.mozilla.org/show_bug.cgi?id=116169) *from the original on 2018-10-08. Retrieved 2018-10-20.*

  [*"sslstrip 0.9"*](https://moxie.org/software/sslstrip/index.html)*.* [*Archived*](https://web.archive.org/web/20180620042059/https:/moxie.org/software/sslstrip/index.html) *from the original on 2018-06-20. Retrieved 2018-10-20.*

  *Shuo Chen; Rui Wang; XiaoFeng Wang; Kehuan Zhang (2010-05-20).* [*"Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow"*](https://www.microsoft.com/en-us/research/publication/side-channel-leaks-in-web-applications-a-reality-today-a-challenge-tomorrow/)*. Microsoft Research.* [*IEEE*](https://en.wikipedia.org/wiki/Institute_of_Electrical_and_Electronics_Engineers) *Symposium on Security & Privacy 2010.* [*Archived*](https://web.archive.org/web/20180722120329/https:/www.microsoft.com/en-us/research/publication/side-channel-leaks-in-web-applications-a-reality-today-a-challenge-tomorrow/) *from the original on 2018-07-22. Retrieved 2018-10-20.*

  *Guaay, Matthew (2017-09-21).* [*"How to Force a Public Wi-Fi Network Login Page to Open"*](https://zapier.com/blog/open-wifi-login-page/)*.* [*Archived*](https://web.archive.org/web/20180810143254/https:/zapier.com/blog/open-wifi-login-page/) *from the original on 2018-08-10. Retrieved 2018-10-20.*

  [*"NeverSSL"*](http://neverssl.com/)*.* [*Archived*](https://web.archive.org/web/20180901224536/http:/neverssl.com/) *from the original on 2018-09-01. Retrieved 2018-10-20.*

  [*"a non https website"*](http://www.nonhttps.com)*.* [*Archived*](https://web.archive.org/web/20181017052428/http:/www.nonhttps.com/) *from the original on 2018-10-17. Retrieved 2018-10-20.*

  *Walls, Colin (2005).* [*Embedded Software: The Works*](https://books.google.com/books?id=FLvsis4_QhEC&pg=PA344)*. Newnes. p. 344.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*0-7506-7954-9*](https://en.wikipedia.org/wiki/Special:BookSources/0-7506-7954-9)*.* [*Archived*](https://web.archive.org/web/20190209055124/https:/books.google.com/books?id=FLvsis4_QhEC&pg=PA344) *from the original on 2019-02-09. Retrieved 2018-10-20.*

* 1.  [*"A secure web is here to stay"*](https://blog.chromium.org/2018/02/a-secure-web-is-here-to-stay.html)*. Chromium Blog.* [*Archived*](https://web.archive.org/web/20190424215132/https:/blog.chromium.org/2018/02/a-secure-web-is-here-to-stay.html) *from the original on 2019-04-24. Retrieved 2019-04-22.*

**External links**

|  |  |
| --- | --- |
|  | Wikimedia Commons has media related to [***HTTPS***](https://commons.wikimedia.org/wiki/Category:HTTPS). |

* [RFC 2818: HTTP Over TLS](https://tools.ietf.org/html/rfc2818)
* [RFC 5246: The Transport Layer Security Protocol 1.2](https://tools.ietf.org/html/rfc5246)
* [RFC 6101: The Secure Sockets Layer (SSL) Protocol Version 3.0](https://tools.ietf.org/html/rfc6101)
* [How HTTPS works ...in a comic!](https://howhttps.works:)