DEFINITION

Hypertext Transfer Protocol Secure (HTTPS)

Rahul Awati

What is Hypertext Transfer Protocol Secure (HTTPS)?

Hypertext Transfer Protocol Secure (HTTPS) is a protocol that secures communication and data transfer between a user's web browser and a website. HTTPS is the secure version of HTTP.

The protocol protects users against eavesdroppers and man-in-the-middle (MitM) attacks. It also protects legitimate domains from domain name system (DNS) spoofing attacks.

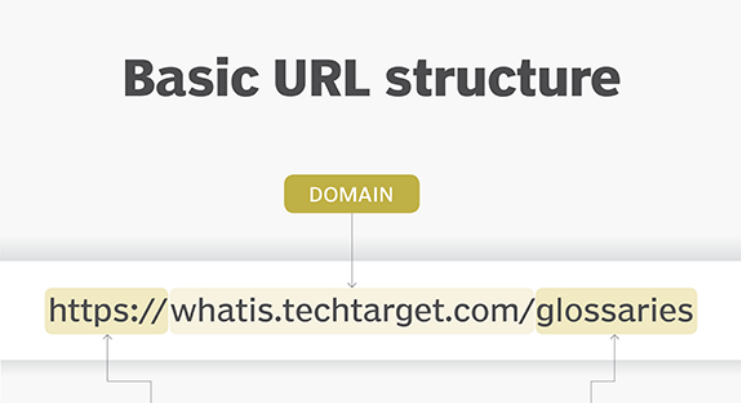
HTTPS plays a significant role in securing websites that handle or transfer sensitive data, including data handled by online banking services, email providers, online retailers, healthcare providers and more. Simply put, any website that requires login credentials or involves financial

HTTP vs. HTTPS

A malicious actor can easily impersonate, modify or monitor an HTTP connection. HTTPS provides protection against these vulnerabilities by encrypting all exchanges between a web browser and web server. As a result, HTTPS ensures that no one can tamper with these transactions, thus securing users' privacy and preventing sensitive information from falling into the wrong hands.

HTTPS is not a separate protocol from HTTP. Rather, it is a variant that uses Transport Layer Security (TLS)/Secure Sockets Layer (SSL) encryption over HTTP to secure communications. When a web server and web browser talk to each other over HTTPS, they engage in what's known as a *handshake* -- an exchange of TLS/SSL certificates -- to verify the provider's identity and protect the user and their data.

An HTTPS URL begins with *https://* instead of *http://*. Most web browsers show that a website is secure by displaying a closed padlock symbol to the left of the URL in the browser's address bar. In some browsers, users can click on the padlock icon to check if an HTTPS-enabled website's digital certificate includes identifying information about the website owner, such as their name or company name.



HTTPS is a variant of HTTP that uses TLS/SSL encryption over HTTP to secure communications.

How is HTTPS superior to HTTP?

In HTTP, the information shared over a website may be intercepted, or sniffed, by any bad actor snooping on the network. This is especially risky if a user is accessing the website over an unsecured network, such as public Wi-Fi. Since all HTTP communications happen in plaintext, they are highly vulnerable to on-path MitM attacks.

HTTPS ensures that all communications between the user's web browser and a website are completely encrypted. Even if cybercriminals intercept the traffic, what they receive looks like garbled data. This data can be converted to a readable form only with the corresponding decryption tool -- that is, the private key.

Encryption in HTTPS

HTTPS is based on the TLS encryption protocol, which secures communications between two parties. TLS uses asymmetric public key infrastructure for encryption. This means it uses two different keys:

1. **The private key.** This is controlled and maintained by the website owner and resides on the web server. It decrypts information that is encrypted by the public key.
2. **The public key.** This is available to users who want to securely interact with the server via their web browser. The information encrypted by the public key can only be decrypted by the private key.

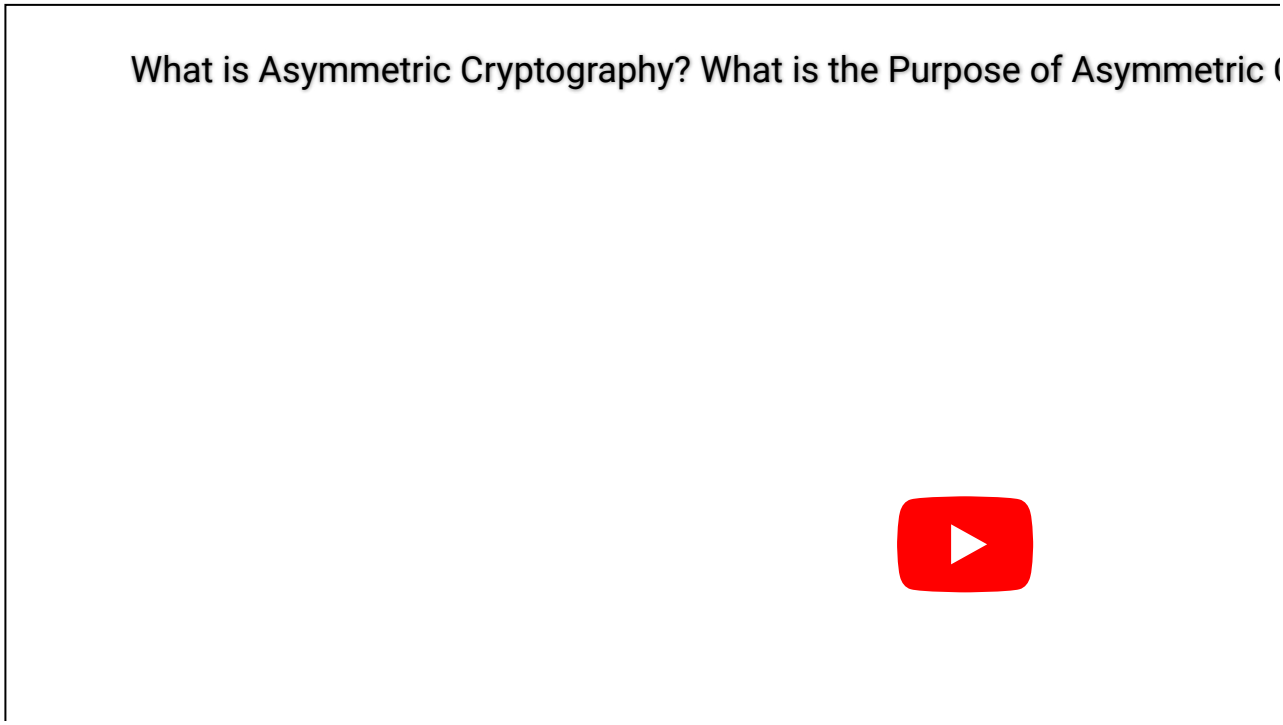
How HTTPS works

As noted in the previous section, HTTPS works over SSL/TLS with public key encryption to distribute a shared symmetric key for data encryption and authentication. It uses port 443 by default, whereas HTTP uses port 80. All secure transfers require port 443, although the same port supports HTTP connections as well.

Before a data transfer starts in HTTPS, the browser and the server decide on the connection parameters by performing an SSL/TLS handshake. The handshake is also important to establish a secure connection.

Here's how the entire process works:

1. The client browser and the web server exchange "hello" messages.
2. Both parties communicate their encryption standards with each other.
3. The server shares its certificate with the browser.
4. The client verifies the certificate's validity.
5. The client uses the public key to generate a pre-master secret key.
6. This secret key is encrypted using the public key and shared with the server.
7. The client and server compute the symmetric key based on the value of the secret key.
8. Both sides confirm that they have computed the secret key.
9. Data transmission uses symmetric encryption.



Example of how HTTPS works

Suppose a customer visits a retailer's e-commerce website to purchase an item. When the customer is ready to place an order, they are directed to the product's order page. The URL of this page starts with *https://*, not *http://*.

To place the order, the customer is prompted to enter some personal details (e.g., their name and shipping address), as well as financial data (e.g., their credit card number). HTTPS encrypts this data to ensure that it cannot be compromised or stolen by an unauthorized party, such as a hacker or cybercriminal.

The order then reaches the server where it is processed. Once the order is successfully placed, the user receives an acknowledgement from the server, which also travels in encrypted form and displays in their web browser. This acknowledgement is decrypted by the browser's HTTPS sublayer.

HTTPS and the CIA triad

HTTPS guarantees the CIA triad, which is a foundational element in information security:

- HTTPS encrypts the website visitor's connection and hides cookies, URLs and other types of sensitive metadata.
- HTTPS ensures that any data transferred between the visitor and the website cannot be tampered with or modified by a hacker.
- HTTPS ensures that the user accesses the actual website and not a fake version.

Advantages of HTTPS

HTTPS offers numerous advantages over HTTP connections:

- **Data and user protection.** HTTPS prevents eavesdropping between web browsers and web servers and establishes secure communications. It thus protects the user's privacy and protects sensitive information from hackers. This is critical for transactions involving personal or financial data.
- **Improved user experience.** When customers know that a website is authentic and protects their data, it instills confidence and trust. In addition, HTTPS increases data transfer speeds by reducing the size of the data.
- **Search engine optimization (SEO).** HTTPS websites usually rank higher in search engine results pages, which is a significant advantage for organizations looking to boost their digital presence through SEO.

Common mistakes to avoid when adapting HTTPS connection

While HTTPS can enhance website security, implementing it improperly can negatively affect a site's security and usability. Common mistakes include the following issues.

Problem	Solution
Expired certificates	Always ensure that the site certificate is up to date.
Missing certificate for all host names	Get a certificate for all host names that the site serves to avoid certificate name mismatch errors.
Server Name Indication (SNI) support	Ensure that the web server supports SNI and that the audience uses SNI-supported browsers.
Crawling and indexing issues	Ensure that the HTTPS site is not blocked from crawling using robots.txt. Also, enable proper indexing of all pages by search engines.
Content	Ensure that content matches on both HTTP and HTTPS pages.

Are HTTPS connections vulnerable to attacks?

While HTTPS is more secure than HTTP, neither is immune to cyber attacks. HTTPS connections may be vulnerable to the following malicious activities:

- **Cryptanalysis or protocol weakness.** Threat actors may use cryptanalysis or exploit potential weaknesses to compromise the HTTPS connection.
- **Attacks on the client computer.** Attackers may install a malicious root certificate into the client computer or browser trust store, thereby compromising the HTTPS connection.
- **Manipulating a certificate authority.** Attackers can manipulate or compromise a certificate authority to obtain a rogue certificate that is mistakenly trusted by major browsers.

See what the most important email security protocols are.

This was last updated in March 2022

Continue Reading About Hypertext Transfer Protocol Secure (HTTPS)

- How to encrypt and secure a website using HTTPS
- Infoblox's Cricket Liu explains DNS over HTTPS security issues
- 6 questions to ask before evaluating secure web gateways
- Prevent man-in-the-middle attacks on apps, CI/CD toolchains
- 5-step checklist for web application security testing

Related Terms

automated testing

Automated testing is a software testing technique that automates the process of validating the functionality of software and ... See complete definition

continuous integration (CI)

Continuous integration (CI) is a software development practice in which frequent, isolated changes are immediately tested and ... See complete definition

garbage in, garbage out (GIGO)

Garbage in, garbage out, or GIGO, refers to the idea that in any system, the quality of output is determined by the quality of ... See complete definition

Dig Deeper on Software testing tools and techniques

man-in-the-middle attack (MitM)

By: Kinza Yasar

certificate authority (CA)

By: Rahul Awati

cookie

By: Sean Kerner

SSL (secure sockets layer)

By: TechTarget Contributor

CLOUD COMPUTING APPLICATION ARCHITECTURE IT OPERATIONS JAVA AWS

Cloud Computing

HPE bets big on public cloud offering for AI

HPE is entering the AI public cloud provider market -- but is it ready? Read more about its AI offerings for HPE GreenLake and ...

Refining HPE GreenLake as it sets its sights on everything

HPE's Bryan Thompson talks about how HPE GreenLake has become synonymous with the brand, and looks to its future and how the ...