Everything You Want to Know about the Cryptography behind SSL Encryption

Background

SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser; or a mail server and a mail client (e.g., Outlook). It allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely. To establish this secure connection, the browser and the server need an SSL Certificate.

But how is this accomplished? How is data encrypted so that no one—including the world's biggest super computers—can crack it?

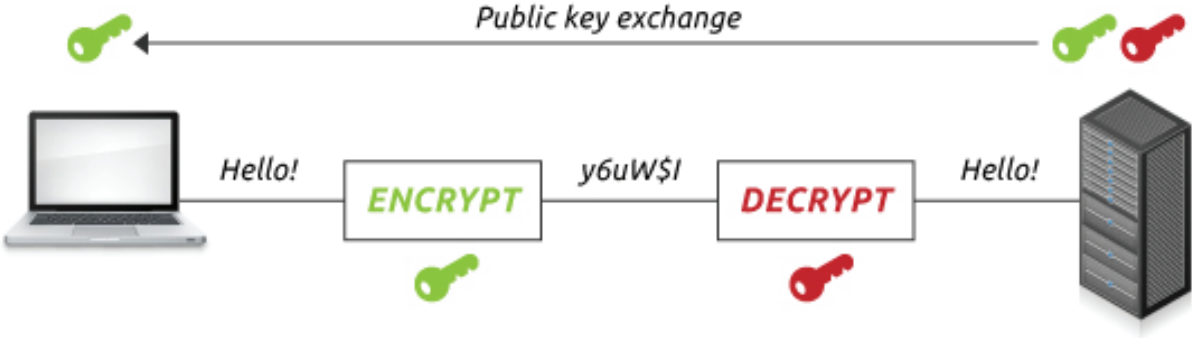
This article explains the technology at work behind the scenes of SSL encryption. It covers asymmetric and symmetric keys and how they work together to create an SSL-encrypted connection. It also covers different types of algorithms that are used to create these keys—including the mathematical equations that make them virtually impossible to crack.

Not sure you understand the basics of SSL Certificates and technology

Learn about SSL Certificates

Asymmetric Encryption

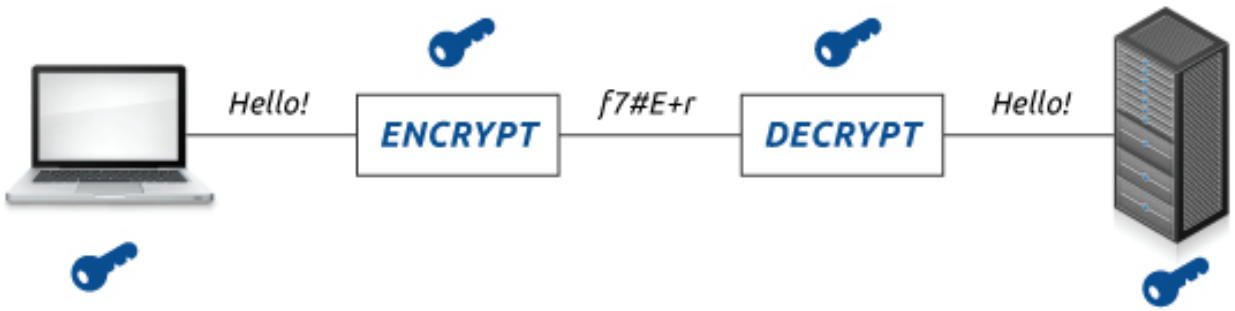
Asymmetric encryption (or public-key cryptography) uses a separate key for encryption and decryption. Anyone can use the encryption key (public key) to encrypt a message. However, decryption keys (private keys) are secret. This way only the intended receiver can decrypt the message. The most common asymmetric encryption algorithm is RSA; however, we will discuss algorithms later in this article.



Asymmetric keys are typically 1024 or 2048 bits. However, keys smaller than 2048 bits are no longer considered safe to use. 2048-bit keys have enough unique encryption codes that we won't write out the number here (it's 617 digits). Though larger keys can be created, the increased computational burden is so significant that keys larger than 2048 bits are rarely used. To put it into perspective, it would take an average computer more than 14 billion years to crack a 2048-bit certificate.

Symmetric Encryption

Symmetric encryption (or pre-shared key encryption) uses a single key to both encrypt and decrypt data. Both the sender and the receiver need the same key to communicate



crack. For example, a 128-bit key has 340,282,366,920,938,463,463,374,607,431,768,211,456 encryption code possibilities. As you can imagine, a 'brute force' attack (in which an attacker tries every possible key until they find the right one) would take quite a bit of time to break a 128-bit

Whether a 128-bit or 256-bit key is used depends on the encryption capabilities of both the server and the client software. SSL Certificates do not dictate what key size is used.

Which Is Stronger?

Since asymmetric keys are bigger than symmetric keys, data that is encrypted asymmetrically is tougher to crack than data that is symmetrically encrypted. However, this does not mean that asymmetric keys are better. Rather than being compared by their size, these keys should compared by the following properties: computational burden and ease of distribution.

Symmetric keys are smaller than asymmetric, so they require less computational burden. However, symmetric keys also have a major disadvantage—especially if you use them for securing data transfers. Because the same key is used for symmetric encryption and decryption, both you and the recipient need the key. If you can walk over and tell your recipient the key, this isn't a huge deal. However, if you have to send the key to a user halfway around the world (a more likely scenario) you need to worry about data security.

Asymmetric encryption doesn't have this problem. As long as you keep your private key secret, no one can decrypt your messages. You can distribute the corresponding public key without worrying who gets it. Anyone who has the public key can encrypt data, but only the person with the private key can decrypt it.

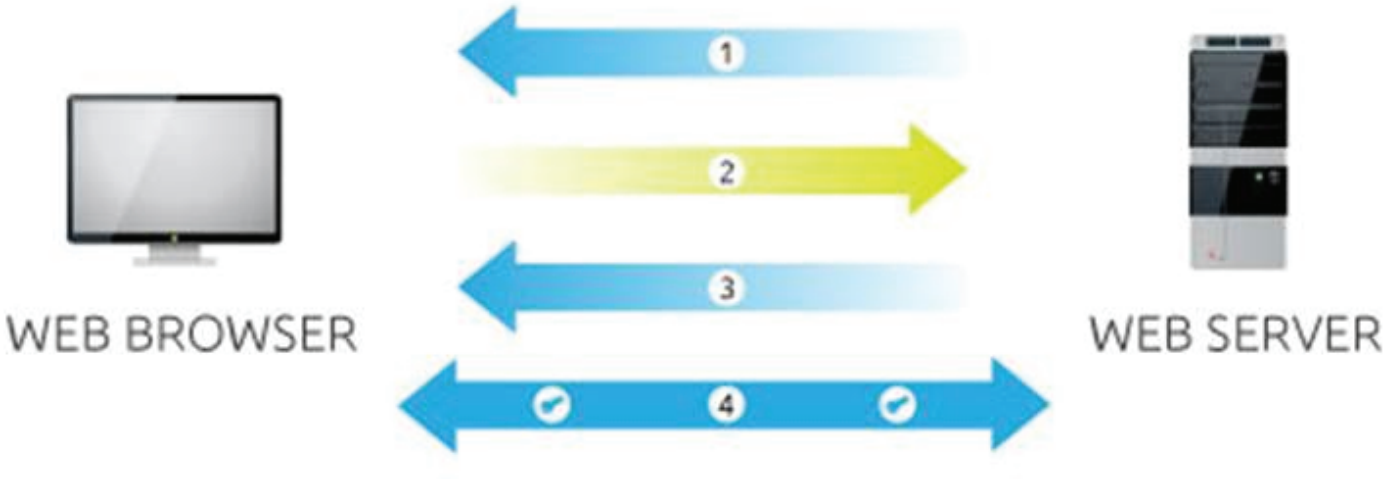
Public-Key Encryption Algorithms

Public-key cryptography (asymmetric) uses encryption algorithms like RSA and Elliptic Curve Cryptography (ECC) to create the public and private keys. These algorithms are based on the intractability* of certain mathematical problems.

With asymmetric encryption it is computationally easy to generate public and private keys, encrypt messages with the public key, and decrypt messages with the private key. However, it is extremely difficult (or impossible) for anyone to derive the private key based only on the public key.

How SSL Uses both Asymmetric and Symmetric Encryption

Public Key Infrastructure (PKI) is the set of hardware, software, people, policies, and procedures that are needed to create, manage, distribute, use, store, and revoke digital certificates. PKI is also what binds keys with user identities by means of a Certificate Authority (CA). PKI uses a hybrid cryptosystem and benefits from using both types of encryption. For example, in SSL communications, the server's SSL Certificate contains an asymmetric public and private key pair. The session key that the server and the browser create during the SSL Handshake is symmetric. This is explained further in the diagram below.



1. Server sends a copy of its asymmetric public key.

2. Browser creates a symmetric session key and encrypts it with the server's asymmetric public key. Then sends it to the server.

3. Server decrypts the encrypted session key using its asymmetric private key to get the symmetric session key.

4. Server and Browser now encrypt and decrypt all transmitted data with the symmetric session key. This allows for a secure channel because only the browser and the server know the symmetric session key, and the session key is only used for that session. If the browser was to connect to the same server the next day, a new session key would be created.

ECC

Elliptic curve cryptography (ECC) relies on the algebraic structure of elliptic curves over finite fields. It is assumed that discovering the discrete logarithm of a random elliptic curve element in connection to a publicly known base point is impractical.

The use of elliptic curves in cryptography was suggested by both Neal Koblitz and Victor S. Miller independently in 1985; ECC algorithms entered common use in 2004.

The advantage of the ECC algorithm over RSA is that the key can be smaller, resulting in improved speed and security. The disadvantage lies in the fact that not all services and applications are interoperable with ECC-based SSL Certificates.

RSA

RSA is based on the presumed difficulty of factoring large integers (integer factorization). Full decryption of an RSA ciphertext is thought to be infeasible on the assumption that no efficient algorithm exists for integer factorization.

A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but only someone with knowledge of the prime factors can feasibly decode the message.

RSA stands for Ron Rivest, Adi Shamir, and Leonard Adleman— the men who first publicly described the algorithm in 1977.

Pre-Shared Key Encryption Algorithms

Pre-shared key encryption (symmetric) uses algorithms like Twofish, AES, or Blowfish, to create keys—AES currently being the most popular. All of these encryption algorithms fall into two types: stream ciphers and block ciphers. Stream ciphers apply a cryptographic key and algorithm to each binary digit in a data stream, one bit at a time. Block ciphers apply a cryptographic key and algorithm to a block of data (for example, 64 sequential bits) as a group. Block ciphers are currently the most common symmetric encryption algorithm.

*Note: Problems that can be solved in theory (e.g., given infinite time), but which in practice take too long for their solutions to be useful are known as intractable problems.

What is an SSL Certificate and How Does it Work?

Secure Sockets Layer (SSL) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser, or a mail server and a mail client (e.g., Outlook).

SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely. Normally, data sent between browsers and web servers is sent in plain text—leaving you vulnerable to eavesdropping. If an attacker is able to intercept all data being sent between a browser and a web server, they can see and use that information.

More specifically, SSL is a security protocol. Protocols describe how algorithms should be used. In this case, the SSL protocol determines variables of the encryption for both the link and the data being transmitted.

All browsers have the capability to interact with secured web servers using the SSL protocol. Howev-

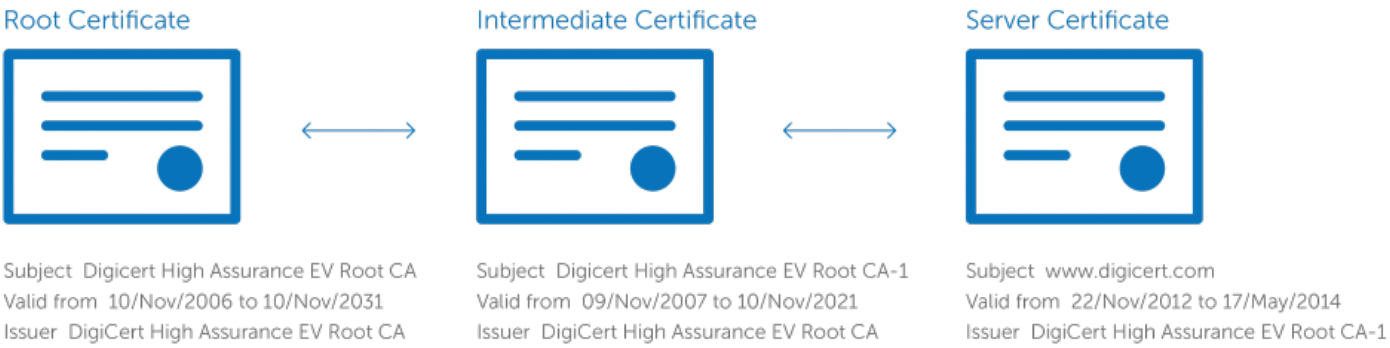
What is an SSL Certificate and How Does it Work?

SSL certificates create an encrypted connection and establish trust.

One of the most important components of online business is creating a trusted environment where potential customers feel confident in making purchases. SSL certificates create a foundation of trust by establishing a secure connection. To assure visitors their connection is secure, browsers provide special visual cues that we call EV indicators -- anything from a green padlock to branded URL bar.

SSL certificates have a key pair: a public and a private key. These keys work together to establish an encrypted connection. The certificate also contains what is called the "subject," which is the identity of the certificate/website owner.

To get a certificate, you must create a Certificate Signing Request (CSR) on your server. This process creates a private key and public key on your server. The CSR data file that you send to the SSL Certificate issuer (called a Certificate Authority or CA) contains the public key. The CA uses the CSR data file to create a data structure to match your private key without compromising the key itself. The CA never sees the private key.



The most important part of an SSL certificate is that it is digitally signed by a trusted CA, like DigiCert. Anyone can create a certificate, but browsers only trust certificates that come from an organization on their list of trusted CAs. Browsers come with a pre-installed list of trusted CAs, known as the Trusted Root CA store. In order to be added to the Trusted Root CA store and thus become a Certificate Authority, a company must comply with and be audited against security and authentication standards established by the browsers.

An SSL Certificate issued by a CA to an organization and its domain/website verifies that a trusted third party has authenticated that organization's identity. Since the browser trusts the CA, the browser now trusts that organization's identity too. The browser lets the user know that the website is secure, and the user can feel safe browsing the site and even entering their confidential information.