# Behind the Scenes of SSL Cryptography

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.
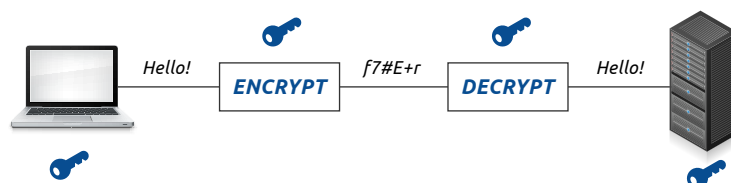
Asymmetric Encryption

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 paper.
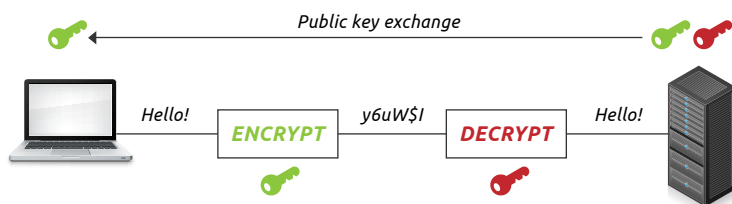
Asymmetric keys are typically 1024 or 2048 bits. However, keys smaller than 2048 bits are no longer considered safe to use. 2048-bit keys have enough unique encryption codes that we won't write out the number here (it's 617 digits). Though larger keys can be created, the increased computational burden 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.

Symmetric key sizes are typically 128 or 256 bits—the larger the key size, the harder the key is to 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 key.

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.

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.

WEB BROWSER   WEB SERVER
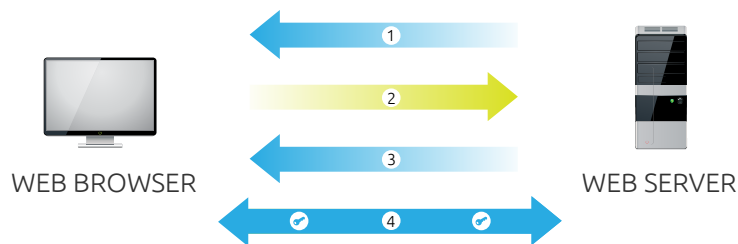
1. **Server** sends a copy of its asymmetric public key.

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

3. **Server** decrypts the asymmetric public key with 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.

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.

* *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.*

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.

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.

To learn more about DigiCert's SSL Certificates call 1-855-800-3444 or visit www.digicert.com

digicert®
Your Success is Built on Trust™