Realtime
publishers
"Leading the Conversation"

The Shortcut Guide™ To

Extended Validation SSL Certificates

Dan Sullivan

Copyright Statement

Chapter 2: Overview of SSL and EV-SSL Certificates

SSL certificates are widely used in secure Web communications but because they function "behind the scenes," most users give little attention to how they work. This chapter will take a closer look at SSL certificates to answer several key questions:

- How are SSL certificates used?
- How are they implemented?
- What are their limitations?
- How are Extended Validation (EV) SSL certificates different?

In answering these questions, we will delve into both the technical and organizational issues that are involved with the use of SSL certificates.

Overview of SSL Certificates

SSL certificates are like identification cards for a device. They associate an identity, such as a Web site or a person, with a resource, such as a particular server or a client device. One of their purposes is to answer the question: how do I know that a server, which claims to host a particular Web site, is actually the authentic company server? Another purpose is to ensure confidential communications by encrypting traffic between a client and a server.

Proving Identity

Servers with certificates can essentially present those certificates in much the same way we use driver's licenses to prove our identity. SSL certificates are like driver's licenses in a couple of key ways. First, they are standardized. We expect a driver's license to have photo of the driver, a name, an address, a birth date, and an indication of the state that issued the license. Similarly, an SSL certificate will have a name of a server, the name of a trusted third party that issued the certificate as well as some additional cryptographic information that makes any tampering easy to detect.

The second similarity is that trusted third parties issue both driver's licenses and certificates. States issues driver's licenses; companies, known as Certification Authorities (CAs), issue certificates. Both types of third parties have procedures they use to verify the identity of the person, business, or organization requesting a licenses or SSL certificate. We trust these third parties to sufficiently verify identities, so we can trust that people, or servers, are who they claim they are when they posses one of these forms of identification.

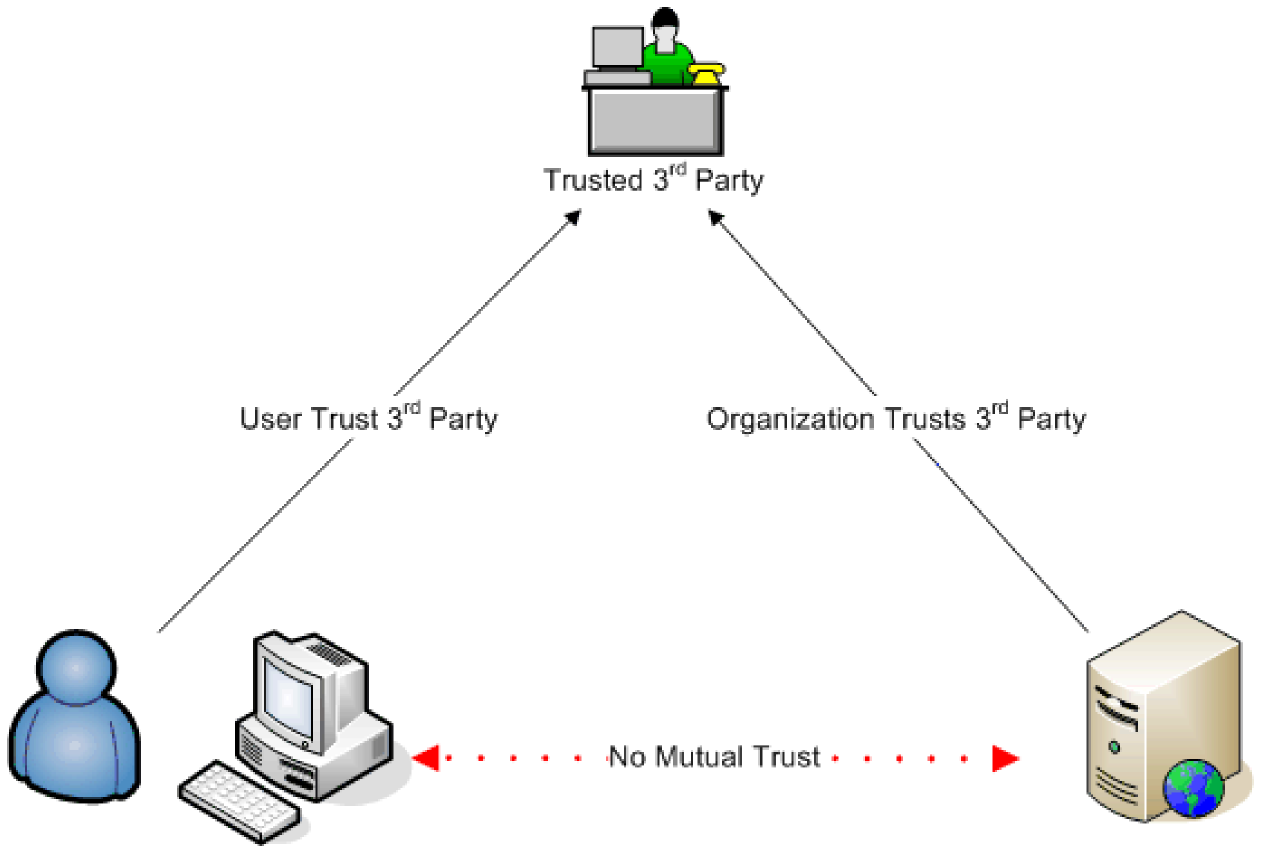


Figure 2.1: Parties that have not established mutual trust directly can establish trust indirectly through a mutually trusted third party.

Acquiring Certificates

CAs might have different procedures for issuing authenticated SSL certificates but some steps are common to all. First, the person or organization applying for the certificate will have to provide the CA basic information, which is validated by the CA as true and accurate, including:

- The name of the server or domain that will use the certificate; for example www.mybusinesssite.com or mybusinesssite.com
- The organization or business name requesting the certificate
- The type of server platform
- Contact information
- A certificate signing request (CSR)

These are all self-explanatory except for the last item, the CSR.

Different standards for issuing SSL certificates can lead to uncertainty on the part of users who depend on these certificates to verify the identity of businesses and organizations. EV SSL certificates address this issue.

The CSR is a cryptographic message that is created using the name of the domain or server that will use the SSL certificate along with a key, known only to the applicant. To create a CSR, the applicant first generates a private key/public key pair using software generally provided with Web servers. This key pair is used to encrypt and decrypt messages; one key essentially locks a message while the other unlocks it.

The details of this kind of public key encryption are described later in this chapter.

A program provided with Web and application servers is then used to generate the CSR. The server or domain name and the private key are used as inputs. CSRs are bound to both the domain name and the private key used to generate them so the certificate that is issued can only be installed on a server hosting that domain and by an administrator with the private key used to generate the CSR.

It is possible to use a single certificate to secure the same domain on multiple servers as long as the private key has been installed on all servers.

Installing an SSL Certificate

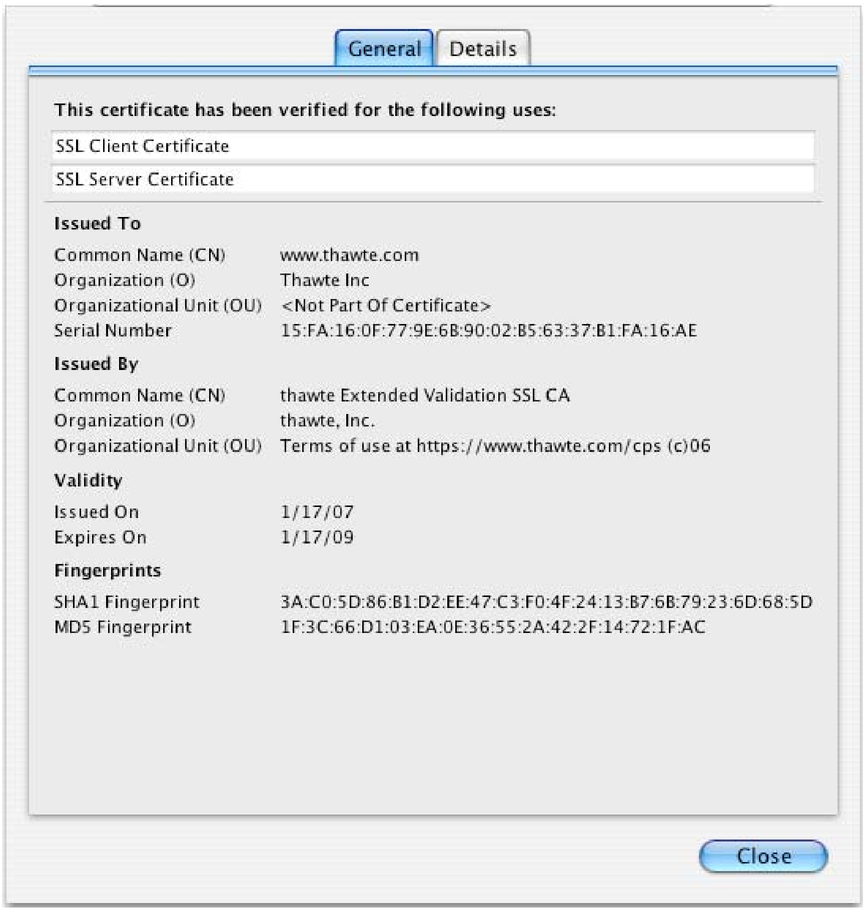
After your CA has created a certificate for a site, a file with a digitally encoded certificate is sent to the requestor. A certificate is delimited with a ‘Begin Certificate’ and ‘End Certificate’ tag instead of the ‘BEGIN CERTIFICATE REQUEST’ and ‘END CERTIFICATE REQUEST’ used in CSRs.

The installation process will vary by server type but the essential steps are:

- Copy the CA’s certificate to the server. This certificate contains information, such as the CA’s public key, needed to verify certificates from that CA.
- Copy the server’s certificate to the server.
- Update the server configuration to register the new certificate. For example, a systems administrator would edit the http daemon configuration file to include the location of the server’s certificate, the private key used to generate the CSR, and the location of the CA’s certificate.

Once these steps are done, the server can then authenticate itself to clients and establish encrypted communication between the server and a client’s browser.

Before delving into an example of how SSL certificates are used to secure a session between the server and a client, we will first examine some of the details that underlie the SSL protocol.



This certificate has been verified for the following uses:

SSL Client Certificate

SSL Server Certificate

Issued To

Common Name (CN)	www.thawte.com
Organization (O)	Thawte Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	15:FA:16:0F:77:9E:6B:90:02:B5:63:37:B1:FA:16:AE

Issued By

Common Name (CN)	thawte Extended Validation SSL CA
Organization (O)	thawte, Inc.
Organizational Unit (OU)	Terms of use at https://www.thawte.com/cps (c)06

Validity

Issued On	1/17/07
Expires On	1/17/09

Fingerprints

SHA1 Fingerprint	3A:C0:5D:86:B1:D2:EE:47:C3:F0:4F:24:13:B7:6B:79:23:6D:68:5D
MD5 Fingerprint	1F:3C:66:D1:03:EA:0E:36:55:2A:42:2F:14:72:1F:AC

Figure 2.2: A typical server certificate contains details about the domain as well as the encryption algorithms and parameters.

SSL is actually an older protocol that has been supplanted by the newer Transport Layer Protocol (TSL), however they are quite similar and TSL is often referred to by the older name of SSL. For convenience, this book uses the more commonly used designation ‘SSL’ and will use ‘TSL’ only when describing a feature found in TSL but not SSL.

SSL Encryption Technology

The ability to secure confidential information and verify the identity of parties on the Internet, or any public network for that matter, depends on cryptography. The SSL protocol supports a number of different encryption algorithms; users are able to select algorithms that best balance requirements for security and speed. Although the details of the various encryption algorithms is beyond the scope of this guide, we will examine two categories of algorithms, known as symmetric or private key cryptography and asymmetric or public key cryptography. Both are used in the SSL protocol.

Symmetric Key Cryptography

To encrypt a message, one needs an encryption algorithm and a key. The key is a sequence of bits that is input to the algorithm along a message (known as a clear text) to produce the encrypted message (known as the cipher text).

The cipher text cannot be decrypted unless one has the key or can discover the key by cryptanalytic technique ("cracking the code"). When using symmetric key cryptography, the same key is used for both encoding and decoding a message, so we have two problems: keeping the key from being cracked and, most problematic of all, keeping it secret.

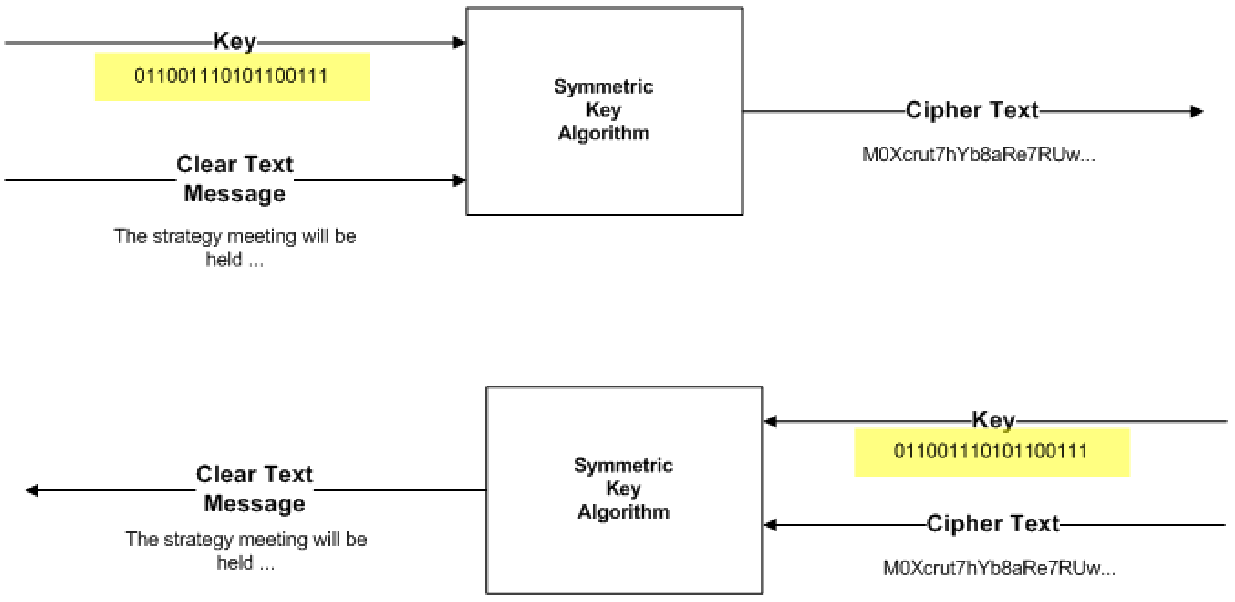


Figure 2.3: In symmetric key cryptography, the same key is used to encrypt and decrypt a message.

Cracking Codes

In theory, all keys can be cracked if only by brute-force trial and error; however, the longer the key, the more time that is required to crack the key. There was a time when keys tens of bits in length were sufficient to protect messages. In fact, the U.S. government established the Data Encryption Standard (DES) standard in 1977 and specified the use of 56-bit keys. By the late 1990s, DES-encrypted messages were being cracked on specialized hardware designed to search all possible keys. DES is no longer considered secure, primarily because of its short key length. DES has been replaced by the Advanced Encryption Standard (AES) that uses 128- to 256-bit length keys.

📖 For more about DES, see the standard specification at http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf; for more information about the AES standard, see http://csrc.nist.gov/CryptoToolkit/aes/rijndael/.

In addition to brute-force techniques to find keys, encryption can be compromised using cryptanalytic techniques such as:

- A cipher-text only attack in which the cryptanalyst only has a piece of encoded text. This generally requires a large quantity of cipher text.
- A known plain-text attack in which both the plain text and cipher text of a message are available.
- A chosen plain-text attack in which the attacker is somehow able to get a piece of selected text encrypted
- An adaptive chosen plain-text attack is possible when the attacker can continue having pieces of selected text encrypted and decides what text to encrypt based on what was learned from previous encryptions of selected text

These kinds of attacks can provide information that can help narrow the search for the key. In practice, cryptanalysis is exceedingly difficult and, practically speaking, not a significant threat to SSL.

> The same cannot be said for all encryption algorithms. For example, the once popular Wireless Encryption Protocol (WEP) is easily cracked and should not be used. Only rigorously analyzed encryption algorithms, such as those commonly used in the SSL protocol, should be considered secure.

Cracking codes is and always will be a potential problem for symmetric key cryptography. However, the work effort required to crack algorithms used in SSL is so high, attackers are likely to search for an easier route to decrypting messages.

Keeping Keys Secret

When using symmetric key cryptography, we need a secure method for exchanging keys. This is something of a bootstrapping problem. We need to exchange keys in order to have secure communication over the Internet, but we can’t exchange keys over the Internet because the lower level protocols, such as TCP and UDP, are not secure.

One option is to use some other means of exchanging keys. This “out of channel” communication, as it is known has its drawbacks. It is not practical to have out of channel communication for high volume, short-lived transactions. For example, consider buying a book from an online retailer. The customer chooses a book and proceeds to checkout. Before entering her shipping and credit card information she needs to make or receive a phone call with a special code. Once she has the code, she enters it into her browser and establishes a secure session with the retailer. She then goes on to provide her name, address, and credit card number. Other methods of exchanging keys—for example, using email rather than the browser session—are still subject to man-in-the-middle attacks and other eavesdropping attacks.

Out of channel communications do have their place when additional levels of verification are required with relatively infrequent but important transactions. For example, when a bank customer signs up for online banking, the bank calls the customer at the home phone number on record to provide a verification code that must be entered into the online form.

Fortunately, practical methods have been developed for exchanging keys using the other broad group of encryption algorithms, known as public key cryptography. Before moving onto public key cryptography, it is worth noting that in spite of the difficulties in exchanging keys, symmetric key cryptography holds an important advantage over public key algorithms: implementations of symmetric key algorithms are much faster than public key methods. By combining symmetric key and public key cryptography, we can have the best of both worlds.

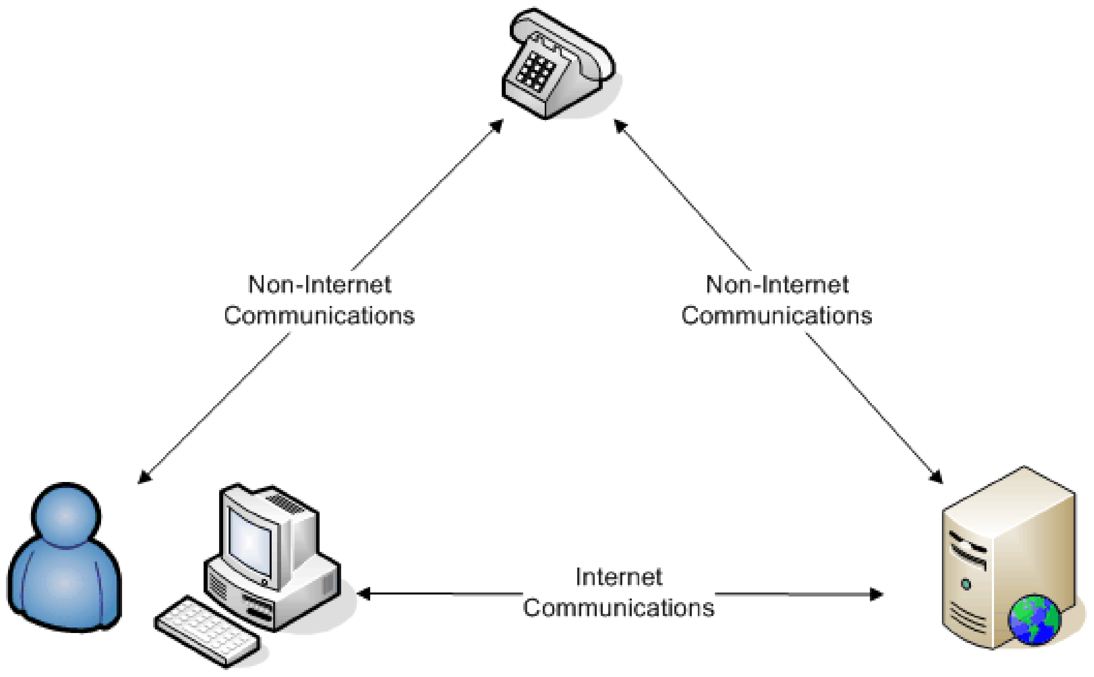


Figure 2.4: Out of channel communications increase the complexity of and time required to complete a transaction and is practical only for low volume, high security transactions.

Public Key Cryptography

Public key cryptography, also known as asymmetric cryptography, uses two keys instead of one. One key is used to encrypt a message and the other is used to decrypt it. It’s like having one key to lock a door and another to unlock it. We wouldn’t worry if someone had a key to lock the front door of our house as long as no one else had the key to unlock it. The same goes for public key cryptography.

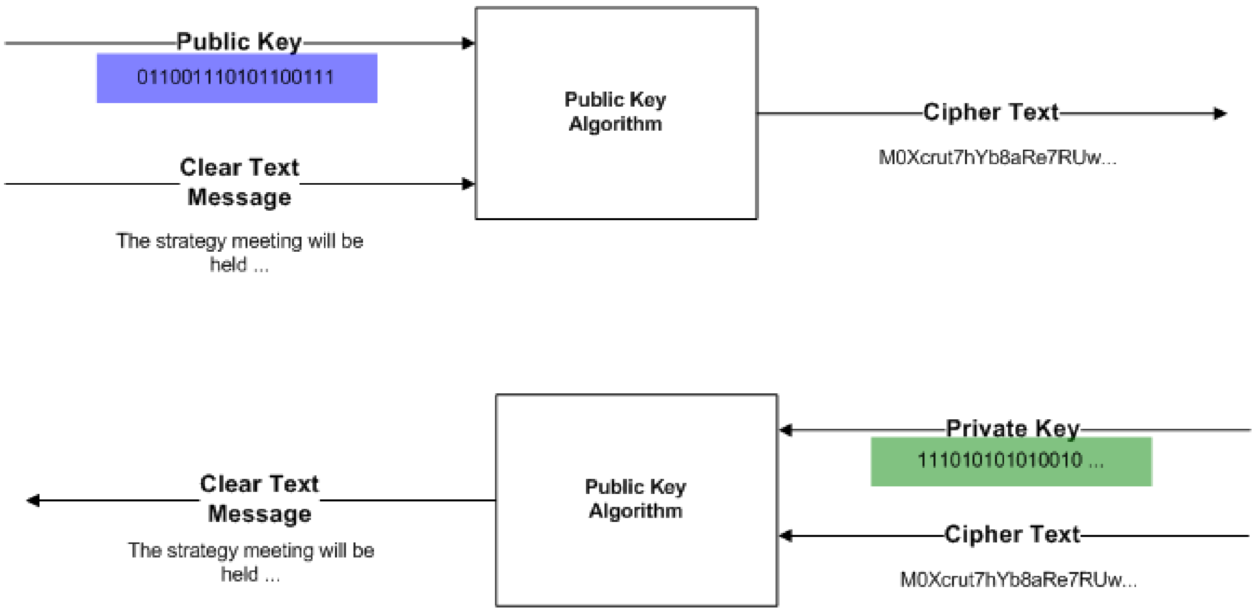


Figure 2.5: Public key algorithms use two keys; one is kept secret and the other is publicly available, eliminating the requirement for exchanging a shared key.

An important characteristic of the public and private keys is that there is no practical way to determine one of them given the other. Someone can safely assume that their public key can be shared widely and not have to worry that somehow an attacker will figure out their private key. This feature stems from the way key pairs are created.

Key generation for public key cryptography uses mathematical problems that are relatively easy to perform but the reverse, or inverse, operation is much harder. Take a simple example. Adding 5 + 3 is just as easy as subtracting (addition's inverse operation) 8 – 3. Now consider multiplying two large prime numbers (these can be divided without a remainder only themselves and 1). That is easy, but given the product of those two large primes it is quite difficult, practically impossible, factor that number back into the two prime numbers you started with. There are other mathematical problems that have similar characteristics. They are collectively known as "trapdoor functions" because it is easy to go through one way and much harder to go through the other way.

Trapdoor functions are useful when solving the problem of secretly exchanging keys. Suppose two people want to exchange confidential messages. They can each share their public key with the other. In the case of a large organization, they may store their key in a directory so that anyone can use it to send encrypted messages that only the person holding the corresponding private key can decrypt. Each person generates a random number, encrypts it with the recipient's public key, and sends it. Only the recipient will be able to recover the random number because the private key is needed to decrypt the message. Now each party has a pair of random numbers known to each. This pair is used to generate a session key that both can use to communicate with less computationally intensive private key cryptography.

We have now established how keys can be securely exchanged allowing for efficiently encrypted communications. This meets the confidentiality requirement of secure communication. There still remains the need to ensure messages are not tampered with in transit. This is done with message authentication codes.

Message Authentication Codes and Message Integrity

Even though an attacker may not be able to read an encrypted message, using a man in the middle attack, someone could intercept the message, scramble the order of bytes or substitute their own data. The attacker could at the very least disrupt the communications. To avoid this problem, SSL uses another cryptographic method known as hash functions.

Hash Functions

Hash functions are calculations that take a variable-length string of data as input and produce a fixed-length string as output. This would be quite unremarkable if it were not for two facts. First, even a slight change in the original input will result in a change in the output; and second, it is highly unlikely that two different input strings will produce the same output string. These features of hash make them quite useful for detecting tampering.

Consider an example. A commonly used hash function is called MD5. When MD5 is applied to the string

1. `The strategy meeting will be held Wednesday at 10:00 am`

The following string is output

2. `0a7a94d707e95d0fe69b551dc0b9cdb8`

If we change the input string slightly by adding a period to

3. `The strategy meeting will be held Wednesday at 10:00 am.`

The new output is:

4. `b88b587881b945a5eadc8ca96bc5a0d3`

By sending the value of a hash function along with a message, the recipient can tell if the message has changed in transit. That is, of course, assuming the hash value has not changed either. This brings us to the next step in calculating a message authentication code.

Protecting Message Integrity

To ensure that the hash value is protected, it is encrypted with the sender’s private key. It can then be decrypted using the sender’s public key. To make sure no one other than the intended recipient can decrypt the hash value, the encrypted hash value is combined with the message text and encrypted using the recipient’s public key. Now, only someone with the recipient’s private key will be able to get the hash value.

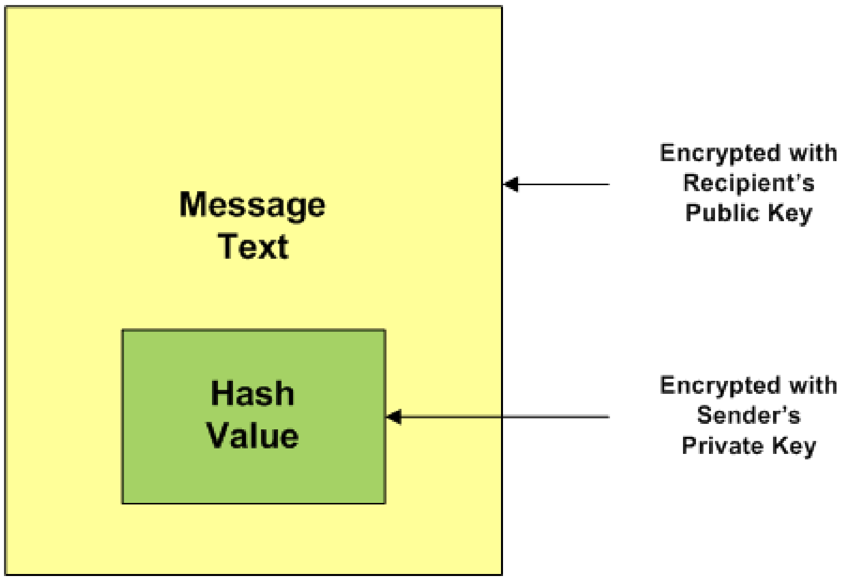

Message Text
Hash Value
Encrypted with Recipient's Public Key
Encrypted with Sender's Private Key

Figure 2.6: Doubly encrypting the hash value provides confidentiality as well as non-repudiation.

A further benefit is that if we can decrypt the hash value using the sender's public key, we can be sure the message was sent by someone in possession of the senders' private key. This provides some level of assurance that the sender is actually who we believe it to be and thus provides another important aspect of secure communications—non-repudiation.

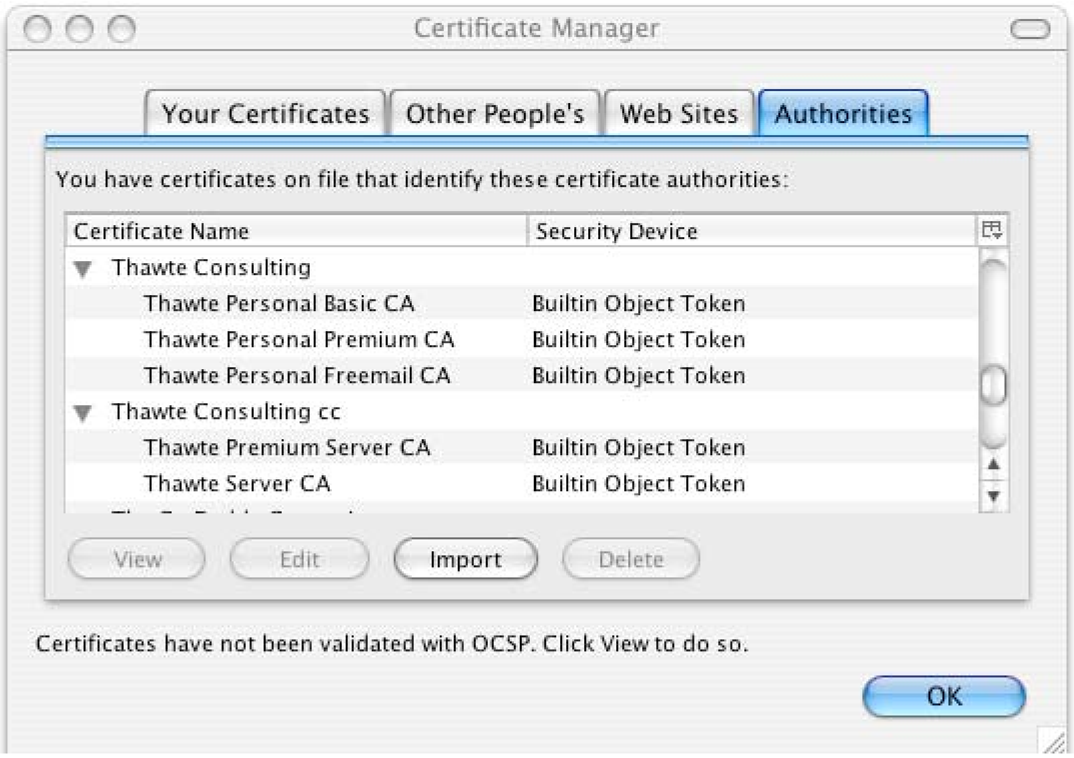
To summarize, SSL-based communications use public key cryptography techniques to exchange a shared secret key, known as a session key, which is then used to encrypt communications using a symmetric key algorithm.

Example SSL Session

Several steps are required to set up a secure communication channel between a client and server:

1. The client sends a message to the server over an unsecured channel indicating its intent to start an SSL session. In the message, the client includes details about the client's capabilities, such as:
    a. The version of SSL the client uses
    b. Cipher algorithms supported
    c. Session-specific data
2. The server responds by sending similar information about SSL version, cipher algorithms, and session-specific data. In addition, the server sends its SSL certificate.

3. The client uses the server's certificate to authenticate the server. This involves several checks, including:
    a. Verifying that the current date is within the valid date range specified on the certificate.
    b. Checking the client's list of trusted certification authorities stored in the client's browser (see Figure 2.6). If the certificate is not issued by a trusted CA, then the server is not authenticated.
    c. The public key of the CA, provided along with other information about the trusted CA in the browser's list of trusted CAs, is used to verify the digital signature of the CA on the certificate. If the digital signature is not validated, the server is not authenticated.
    d. The client compares the domain name on the certificate and the domain name of the server. If they are different, then the server is not authenticated.
4. Once the server's certificate is validated, the client generates a piece of secret data. (The details of how this is done will depend on which cipher was chosen by the client and the server). The client then encrypts the secret data with the public key provided in the server's certificate and sends it to the server.
5. The server decrypts the message with the secret data using its private key. It then calculates a master secret. (The exact steps depend on the protocol negotiated by the client and server).



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
▼ Thawte Consulting	
Thawte Personal Basic CA	Builtin Object Token
Thawte Personal Premium CA	Builtin Object Token
Thawte Personal Freemail CA	Builtin Object Token
▼ Thawte Consulting cc	
Thawte Premium Server CA	Builtin Object Token
Thawte Server CA	Builtin Object Token

Certificates have not been validated with OCSP. Click View to do so.

Figure 2.7: Browsers are preconfigured with lists of trusted CAs.

6. The client uses the secret data it had sent to the server to calculate the master secret as well.
7. The client and the server use the master secret to calculate a session key that is used for further encrypted communication.
8. The client and server exchange messages indicating they will use the shared session key for future communication and that the setup process is complete.

At this point, communications between the client and the server are encrypted until the session is terminated.

It should be noted, the example session described is the more common kind of SSL authentication. A second kind, known as mutual authentication, includes additional steps in which the client device sends its SSL certificate to the server, which in turn verifies the certificate in the same way the client verifies the server's certificate. The basic principles are the same in both kinds of authentication.

Issues Successfully Addressed by SSL

The SSL protocol has demonstrated by its continued widespread use that it is an effective means of establishing and maintaining secure communications channels over untrusted networks, such as the Internet. The important issues successfully addressed by SSL include:

- The use of public key cryptography within SSL has solved the problem of securely sharing a private key (which must be kept secret) that can be used to both encrypt and decrypt messages. SSL uses this more computationally intensive type of cryptography to exchange information needed to create a shared secret key; it is not used for encrypting the contents of communication between the client and server.
- Symmetric key cryptography is much more efficient than public key cryptography and is used for the bulk of the encryption work in an SSL session.
- SSL must work with a wide range of clients and servers that may support different versions of SSL and different kinds of ciphers, so SSL incorporates an initial exchange of information and a negotiation to find a mutually supported set of parameters.
- SSL's support of a variety of encryption algorithms provides clients and servers a range of options for encryption, allowing for longer key lengths when requirements warrant it. The protocol also uses older as well as newer encryption algorithms, providing some degree of backward compatibility to clients and servers that may not have upgraded to improved ciphers.

Of course, a chain is only as strong as its weakest link. SSL strong links are built on the sound mathematical foundation of cryptography; the weaker links in the security chain stem from the concerns with the initial authentication of an organization requesting a certificate from a CA.

Limitations of SSL: Lack of Standards

The SSL protocol is well designed with respect to preventing eavesdropping and avoiding successful man in the middle attacks. It is less concerned with the processes and procedures that a person or organization must go through to acquire a certificate.

Lack of Authentication Standards

The SSL protocol depends on the existence of a trusted third party. It is assumed that parties that want to communicate over a secure channel can agree on an organization that will vouch for the identity of holders of SSL certificates. The protocol does not address some key issues related to authenticating a person or organization before issuing a certificate:

- What constitutes sufficient proof of identity?
- Are there varying levels of proof?
- If so, how will certificates represent the varying levels of proof?
- How can one be sure different CAs follow the same standards for identifying a party?

These issues all move us from the realm of cryptography and network protocols into the often more complex organizational and procedural issues that surround CAs.

Varying Levels of Certification

Rarely in business or government operations is there a situation in which one size fits all. Security requirements are especially variable. Consider a simple analogy with locks on doors.

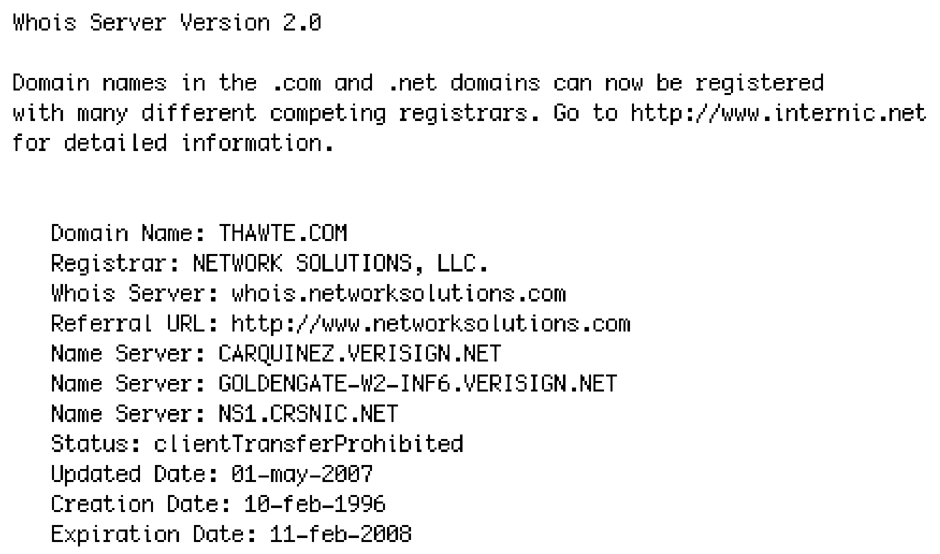
Sometimes a relatively inexpensive and weak lock is sufficient to meet one’s needs, for example, to keep a toddler from getting into a cabinet filled with chemical cleaners. One could invest in a stronger lock, but it would not add any advantages to the existing solution. An entire house, however, is likely to have stronger locks that will better protect its inhabitants and their possessions. The additional cost and effort required to use the better locks is well justified. Finally, a bank, an obvious target for thieves, will use specialized locks and additional security measures to protect its assets.

Domain-Only Certificates

In the case of online transactions, different needs dictate different levels of security and authentication. A Web master running a site for a local basketball league wants to allow coaches to use the site to post practice schedules and other team-related information. The Web master does not want anyone else changing those schedules, so she implements a user login. Being security conscious, she also does not want clear text passwords sent over the Internet, so she uses the SSL protocol to establish a secure channel between clients and her Web server.

This is a relatively low-security environment. There are no financial transactions, no exchange of confidential personal information, and no potential for significant loss of intellectual property. Coaches, if they are concerned at all about submitting their usernames and passwords, would likely want nothing more than to be assured that the transaction is encrypted. In this case, simply having a certificate that verifies the identity of the domain is sufficient.

Domain-only certificates typically validate that the requestor of a certificate is authorized to use that domain. These certificates are inexpensive, largely because the validation process can be automated. Information about the owners of domain names is readily available from utility programs, such as whois. (See Figure 2.8 for example output of whois).



```
Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.


   Domain Name: THAWTE.COM
   Registrar: NETWORK SOLUTIONS, LLC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: CARQUINEZ.VERISIGN.NET
   Name Server: GOLDENGATE-W2-INF6.VERISIGN.NET
   Name Server: NS1.CRSNIC.NET
   Status: clientTransferProhibited
   Updated Date: 01-may-2007
   Creation Date: 10-feb-1996
   Expiration Date: 11-feb-2008
```

Figure 2.8: Information about domain owners is publicly and programmatically available on the Internet. This easy-to-access information is used for domain-only validations.

Domain-only certificates have lowered the cost of using SSL, which has been a benefit to many. Unfortunately, they have also lowered the cost of starting phishing sites that look legitimate. They have also led some companies to use lower-grade certificates rather than authenticated certificates to protect sensitive data. More extensive authentication procedures should be used for most business-oriented domains.

Full-Company Validation

When CAs use full validation procedures, they look for more rigorous proof of the identity of the person, business, or organization requesting a certificate. They will still go through the same steps as the domain-only validation, but in addition they will do things such as:

- Verify the existence of a physical address of the person, company, or organization
- Check government records to verify a business is legally established
- Require copies of documentation, such as a driver’s license for a person or incorporation papers for a company

With full-company validation, one cannot simply register a domain name and acquire a certificate; the requestor must be able to demonstrate the company has some established legal identity. Again, there are varying levels of certification involved depending on the issuing CA.

Problems with Varying Levels of Certification

The biggest problem with varying levels of certification is that these variances are not apparent to users who are expected to trust these certificates. When a browser establishes an SSL session with a server, the same lock icon will appear on the browser whether the server certificate is domain-only or full-company validation. A phishing site can look as legitimate as a real bank's site.

The root of this problem was that there were no well-defined standards for authenticating businesses. Two different CAs may have different procedures for full company certification. One company may check government records to see if a business by a certain name has been established while another will make more rigorous checks to see that the company is actually still actively in business. These variations in current practices, along with the rise of phishing scams, have undermined trust in online commerce and prompted the industry to respond with a new type of SSL certificate that does not suffer from these deficiencies.

EV SSL Certificates

EV SSL certificates use the same cryptography and network protocols as SSL certificates but they improve the certification process to address the weakness outlined earlier. The standards for EV SSL certificates have been established by a governing body known as the CA/Browser Forum (http://www.cabforum.org/). Before an EV SSL certificate is issued, the CA conducts a thorough and standardized process to verify the identity of the requestor. The steps include:

- Verifying the entity physically exists
- The entity is legally recognized
- The entity is actively conducting business or other operations
- The identity of the entity matches the identity on legal records
- The entity has legitimate use of the domain
- The individual requesting the certificate is an authorized representative of the company in question

CAs that issue EV SSL certificates are also subject to audits, performed by WebTrust, a professional assurances organization, to demonstrate that proper policies, procedures, and training measures are in place to ensure quality control.

Details of the authentication and auditing process are described in Chapter 3.

In addition, most high-security browsers such as Microsoft IE7 now provide additional visual cues to users when a site uses EV SSL certificates. This eliminates the problem of how a user is to know the level of verification and authenticate behind a certificate.

Details about the how browsers make use of EV SSL certificates are described in Chapter 4.

Summary

SSL certificates have long been used to support trust between parties using the Internet for commerce and other transactions. Using sound cryptographic techniques, SSL has been able to provide an efficient means for the confidential exchange of information and guarantees of message integrity in those exchanges. In spite of these advantages, weaknesses in how certificates are issued and the level of information about certificates provided to users have undermined the trust that had been built. In response to these issues, CAs and browser developers have established EV SSL certificates and a set of standards governing their processing and use that seek to re-establish levels of trust needed for an environment conducive to online commerce and other transactions.