**IX Security Considerations** **5 hours**

**Principle of Cryptography, Authentication, Encryption/Decryption, Digital Certificates, Digital signature, Secure Socket Layer, VPN**

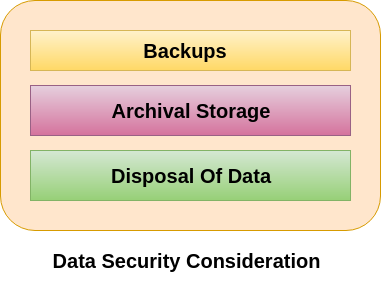
**Security Basic**

Internet security is a branch of computer security specifically related to not only Internet, often involving browser security and the World Wide Web[citation needed], but also network security as it applies to other applications or operating systems as a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information, which leads to a high risk of intrusion or fraud, such as phishing, online viruses, trojans, worms and more.

Many methods are used to protect the transfer of data, including encryption and from-the-ground-up engineering. The current focus is on prevention as much as on real time protection against well known and new threats. Many methods used for to protect the data of transfer.

Data security is the protection of programs and data in computers and communication systems against unauthorized access, modification, destruction, disclosure or transfer whether accidental or intentional by building physical arrangements and software checks. It refers to the right of individuals or organizations to deny or restrict the collection and use of information about unauthorized access. Data security requires system managers to reduce unauthorized access to the systems by building physical arrangements and software checks.

Data security uses various methods to make sure that the data is correct, original, kept confidentially and is safe. It includes-



* Ensuring the integrity of data.
* Ensuring the privacy of the data.
* Prevent the loss or destruction of data.

Data security consideration involves the protection of data against unauthorized access, modification, destruction, loss, disclosure or transfer whether accidental or intentional. Some of the important data security consideration are described below:

<https://www.javatpoint.com/data-security-consideration>

**Principle of Cryptography**

**What does Cryptography mean?**

Cryptography involves creating written or generated codes that allow information to be kept secret. Cryptography converts data into a format that is unreadable for an unauthorized user, allowing it to be transmitted without unauthorized entities decoding it back into a readable format, thus compromising the data..

Information security uses cryptography on several levels. The information cannot be read without a key to decrypt it. The information maintains its integrity during transit and while being stored. Cryptography also aids in nonrepudiation. This means that the sender and the delivery of a message can be verified.

Cryptography is also known as cryptology.

*Cryptography is used to secure and protect data during communication. It is helpful to prevent unauthorized person or group of users from accessing any confidential data. Encryption and decryption are the two essential functionalities of cryptography.*

In cryptography, as in any scientific subject, precision is important, and this applies in particular to the terminology, which is often used incorrectly by the layman. An example of this is confusion between the terms cryptography and cryptology, which are often used interchangeably in popular writing.

Cryptology is a broader subject, consisting of two branches: cryptography, the science of creating secure cryptosystems for converting data into a form that is unintelligible to unauthorized persons, and cryptanalysis, the science of 'attacking' cryptosystems in order to 'crack' them or at least discover their weaknesses.

When cryptanalysis reveals weaknesses in cryptosystems, cryptographers create more secure cryptosystems. Conversely, as cryptosystems become stronger, cryptanalysts try to discover more powerful methods of attacking them. Thus, cryptography and cryptanalysis are complementary.

The aim of cryptography is to convert any data in its original form, called the plaintext, into an incomprehensible form, known as the ciphertext. This process is called encryption. The reverse process of recovering the plaintext from the ciphertext is called decryption. In popular writing, one often finds the (incorrect) terms 'encoding' and 'decoding', but technically these have quite different meanings and should not be used in the present context.

It is important to understand that the plaintext need not necessarily be a textual message. It can be a computer file representing any type of date - an image, a database, etc.

Any particular cryptosystem is based on specific encryption and decryption algorithms. An algorithm is simply a computational procedure that follows some specific set of rules. An important general principle of modern cryptography, known as Kerckhoffs' Principle, is that the the algorithms defining a cryptosystem should be publicly known. Only then is it possible for the cryptosystem to be critically analysed by experts, so that users can have confidence in it.

The precise way in which the plaintext is encrypted by means of a specific algorithm depends on a secret key, which in practice is simply some large number. The implication of Kerckhoffs' Principle is that the security of the encryption relies on the secret key, and not on some secret encryption algorithm. It is easy to understand this by means of the following analogy: There are no secrets about how a combination padlock works - its mechanism is designed to open the lock when a particular sequence of numbers is dialed - but a locked padlock is secure because there are typically over a million possible combinations ('keys').

**The Basic Principles**

**Authentication, Encryption/Decryption, Digital Certificates, Digital signature, Secure Socket Layer, VPN**

**1. Authentication**

This is another important principle of cryptography. In a layman’s term, authentication ensures that the message was originated from the originator claimed in the message. Now, one may think how to make it possible? Suppose, Alice sends a message to Bob and now Bob wants proof that the message has been indeed sent by Alice. This can be made possible if Alice performs some action on message that Bob knows only Alice can do. Well, this forms the basic fundamental of Authentication.

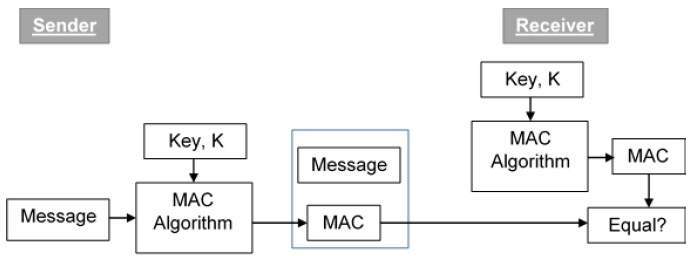
In this threat, the user is not sure about the originator of the message. Message authentication can be provided using the cryptographic techniques that use secret keys as done in case of encryption.

**Message Authentication Code (MAC)**

MAC algorithm is a symmetric key cryptographic technique to provide message authentication. For establishing MAC process, the sender and receiver share a symmetric key K.

Essentially, a MAC is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication.

The process of using MAC for authentication is depicted in the following illustration −



**Let us now try to understand the entire process in detail −**

1. The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.
2. Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and MAC is that MAC uses secret key during the compression.
3. The sender forwards the message along with the MAC. Here, we assume that the message is sent in the clear, as we are concerned of providing message origin authentication, not confidentiality. If confidentiality is required then the message needs encryption.
4. On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
5. The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.
6. If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified. As a bottom-line, a receiver safely assumes that the message is not the genuine.

**Limitations of MAC**

There are two major limitations of MAC, both due to its symmetric nature of operation −

**Establishment of Shared Secret.**

It can provide message authentication among pre-decided legitimate users who have shared key.

This requires establishment of shared secret prior to use of MAC.

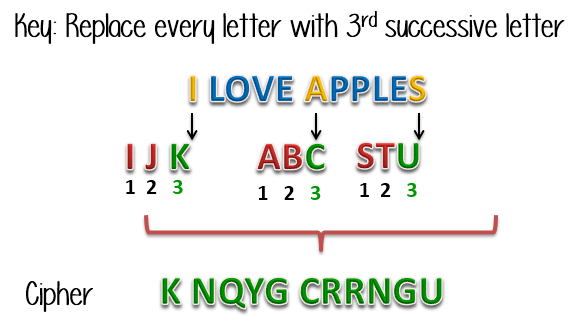
**Inability to Provide Non-Repudiation**

Non-repudiation is the assurance that a message originator cannot deny any previously sent messages and commitments or actions.

MAC technique does not provide a non-repudiation service. If the sender and receiver get involved in a dispute over message origination, MACs cannot provide a proof that a message was indeed sent by the sender.

Though no third party can compute the MAC, still sender could deny having sent the message and claim that the receiver forged it, as it is impossible to determine which of the two parties computed the MAC.

**2.1 Encryption**



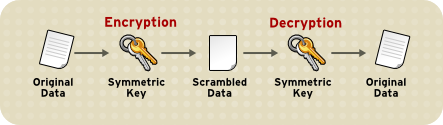
In a simplest form, encryption is to convert the data in some unreadable form. This helps in protecting the privacy while sending the data from sender to receiver. On the receiver side, the data can be decrypted and can be brought back to its original form. The reverse of encryption is called as decryption. The concept of encryption and decryption requires some extra information for encrypting and decrypting the data. This information is known as key. There may be cases when same key can be used for both encryption and decryption while in certain cases, encryption and decryption may require different keys.

Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. A cryptographic algorithm, also called a cipher, is a mathematical function used for encryption or decryption. In most cases, two related functions are employed, one for encryption and the other for decryption.

With most modern cryptography, the ability to keep encrypted information secret is based not on the cryptographic algorithm, which is widely known, but on a number called a key that must be used with the algorithm to produce an encrypted result or to decrypt previously encrypted information. Decryption with the correct key is simple. Decryption without the correct key is very difficult, and in some cases impossible for all practical purposes.

The sections that follow introduce the use of keys for encryption and decryption.

1. [Symmetric-Key Encryption](https://developer.mozilla.org/en-US/docs/Archive/Security/Encryption_and_Decryption#Symmetric-Key_Encryption)
2. [Public-Key Encryption](https://developer.mozilla.org/en-US/docs/Archive/Security/Encryption_and_Decryption#Public-Key_Encryption)
3. [Key Length and Encryption Strength](https://developer.mozilla.org/en-US/docs/Archive/Security/Encryption_and_Decryption#Key_Length_and_Encryption_Strength)
4. **Symmetric-Key Encryption Section**

With symmetric-key encryption, the encryption key can be calculated from the decryption key and vice versa. With most symmetric algorithms, the same key is used for both encryption and decryption, as shown in Figure 1.

Implementations of symmetric-key encryption can be highly efficient, so that users do not experience any significant time delay as a result of the encryption and decryption. Symmetric-key encryption also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense.

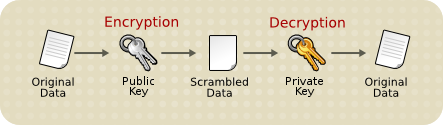
Symmetric-key encryption is effective only if the symmetric key is kept secret by the two parties involved. If anyone else discovers the key, it affects both confidentiality and authentication. A person with an unauthorized symmetric key not only can decrypt messages sent with that key, but can encrypt new messages and send them as if they came from one of the two parties who were originally using the key.

Symmetric-key encryption plays an important role in the SSL protocol, which is widely used for authentication, tamper detection, and encryption over TCP/IP networks. SSL also uses techniques of public-key encryption, which is described in the next section.

1. **Public-Key Encryption Section**

The most commonly used implementations of public-key encryption are based on algorithms patented by RSA Data Security. Therefore, this section describes the RSA approach to public-key encryption.

Public-key encryption (also called asymmetric encryption) involves a pair of keys-a public key and a private key-associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. Each public key is published, and the corresponding private key is kept secret. Data encrypted with your public key can be decrypted only with your private key. [Figure 2](https://developer.mozilla.org/en-US/docs/Archive/Security/Encryption_and_Decryption#Figure2) shows a simplified view of the way public-key encryption works.



The scheme shown in Figure 2 lets you freely distribute a public key, and only you will be able to read data encrypted using this key. In general, to send encrypted data to someone, you encrypt the data with that person's public key, and the person receiving the encrypted data decrypts it with the corresponding private key.

Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, it's possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL protocol.

As it happens, the reverse of the scheme shown in Figure 2 also works: data encrypted with your private key can be decrypted only with your public key. This would not be a desirable way to encrypt sensitive data, however, because it means that anyone with your public key, which is by definition published, could decrypt the data. Nevertheless, private-key encryption is useful, because it means you can use your private key to sign data with your digital signature-an important requirement for electronic commerce and other commercial applications of cryptography. Client software such as Firefox can then use your public key to confirm that the message was signed with your private key and that it hasn't been tampered with since being signed. "[Digital Signatures](https://developer.mozilla.org/en-US/docs/Digital_Signatures)" describes how this confirmation process works.

1. **Key Length and Encryption Strength Section**

Breaking an encryption algorithm is basically finding the key to the access the encrypted data in plain text. For symmetric algorithms, breaking the algorithm usually means trying to determine the key used to encrypt the text. For a public key algorithm, breaking the algorithm usually means acquiring the shared secret information between two recipients.

One method of breaking a symmetric algorithm is to simply try every key within the full algorithm until the right key is found. For public key algorithms, since half of the key pair is publicly known, the other half (private key) can be derived using published, though complex, mathematical calculations. Manually finding the key to break an algorithm is called a brute force attack.

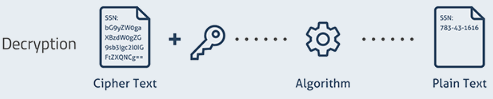
Breaking an algorithm introduces the risk of intercepting, or even impersonating and fraudulently verifying, private information.

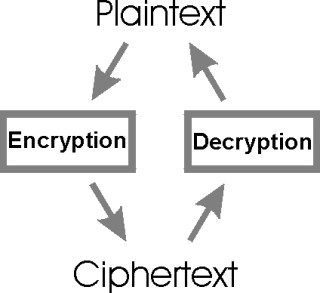
The key strength of an algorithm is determined by finding the fastest method to break the algorithm and comparing it to a brute force attack.

For symmetric keys, encryption strength is often described in terms of the size or length of the keys used to perform the encryption: in general, longer keys provide stronger encryption. Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher. Roughly speaking, 128-bit RC4 encryption is 3 x 1026 times stronger than 40-bit RC4 encryption. (For more information about RC4 and other ciphers used with SSL, see "[Introduction to SSL](https://developer.mozilla.org/en/Introduction_to_SSL).") An encryption key is considered full strength if the best known attack to break the key is no faster than a brute force attempt to test every key possibility.

Different ciphers may require different key lengths to achieve the same level of encryption strength. The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based. Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values.

Because it is relatively trivial to break an RSA key, an RSA public-key encryption cipher must have a very long key, at least 1024 bits, to be considered cryptographically strong. On the other hand, symmetric-key ciphers can achieve approximately the same level of strength with an 80-bit key for most algorithms.

**2.2** **Decryption**[](https://www.guru99.com/images/1/040419_0451_Encryptionv2.png)**:** Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer. This method is performed by un-encrypting the text manually or by using keys used to encrypt the original data.

**Definition**: The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the encrypted information so that an authorized user can only decrypt the data because decryption requires a secret key or password.

**Description:** One of the reasons for implementing an encryption-decryption system is privacy. As information travels over the Internet, it is necessary to scrutinise the access from unauthorized organisations or individuals. Due to this, the data is encrypted to reduce data loss and theft. Few common items that are encrypted include text files, images, e-mail messages, user data and directories. The recipient of decryption receives a prompt or window in which a password can be entered to access the encrypted data. For decryption, the system extracts and converts the garbled data and transforms it into words and images that are easily understandable not only by a reader but also by a system. Decryption can be done manually or automatically. It may also be performed with a set of keys or passwords.

There are many methods of conventional cryptography, one of the most important and popular method is Hill cipher Encryption and Decryption, which generates the random Matrix and is essentially the power of security. Decryption requires inverse of the matrix in Hill cipher. Hence while decryption one problem arises that the Inverse of the matrix does not always exist. If the matrix is not invertible then the encrypted content cannot be decrypted. This drawback is completely eliminated in the modified Hill cipher algorithm. Also this method requires the cracker to find the inverse of many square matrices which is not computationally easy. So the modified Hill-Cipher method is both easy to implement and difficult to crack.

**Why use Encryption and Decryption?**

Here, are important reasons for using encryption:

1. Helps you to protect your confidential data such as passwords and login id
2. Provides confidentiality of private information
3. Helps you to ensure that that the document or file has not been altered
4. Encryption process also prevents plagiarism and protects IP
5. Helpful for network communication (like the internet) and where a hacker can easily access unencrypted data.
6. It is an essential method as it helps you to securely protect data that you don't want anyone else to have access.

| **Parameter** | **Encryption** | **Decryption** |
| --- | --- | --- |
| What is | It is a process of converting normal data into an unreadable form. It helps you to avoid any unauthorized access to data | It is a method of converting the unreadable/coded data into its original form. |
| Process | Whenever the data is sent between two separate machines, it is encrypted automatically using a secret key. | The receiver of the data automatically allows you to convert the data from the codes into its original form. |
| Location of Conversion | The person who is sending the data to the destination. | The receiver receives the data and converts it. |
| Example | An employee is sending essential documents to his/her manager. | The manager is receiving the essential documents from his/her employee. |
| Use of Algorithm | The same algorithm with the same key is used for the encryption-decryption process. | The only single algorithm is used for encryption and decryption with a pair of keys where each use for encryption and decryption. |
| Major function | Transforming humanly understandable messages into an incomprehensible and obscure form that can not be interpreted. | It is a conversion of an obscure message into an understandable form which is easy to understand by a human. |

**Summary**

1. Cryptography is used to secure and protect data during communication.
2. Encryption is a process which transforms the original information into an unrecognizable form.
3. Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer.
4. Encryption method helps you to protect your confidential data such as passwords and login id.
5. Public, Private, Pre-Shared and Symmetric are important keys used in cryptography.
6. An employee is sending essential documents to his/her manager is an example of an encryption method.
7. The manager is receiving the essential encrypted documents from his/her employee and decrypting it is an example of a decryption method.

**3. Digital Certificates**

*A Digital Certificate is an electronic "password" that allows a person, organizaion to exchange data securely over the Internet using the public key infrastructure (PKI). Digital Certificate is also known as a public key certificate or identity certificate.*

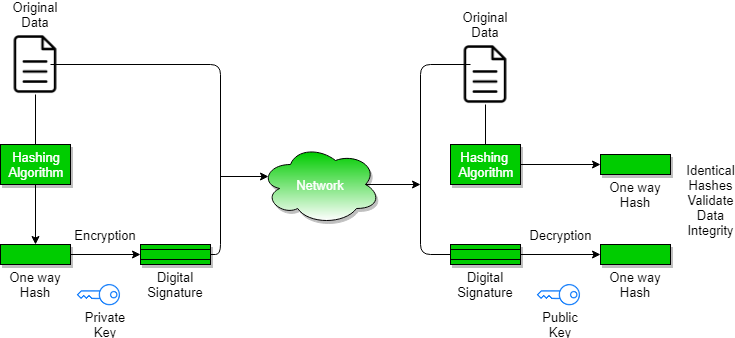
Digital Certificates are a means by which consumers and businesses can utilise the security applications of Public Key Infrastructure (PKI). PKI comprises of the technology to enables secure e-commerce and Internet based communication.

A digital signature is a *“mathematical technique used to validate the authenticity and integrity of a message, software or digital document.”*

1. **Key Generation Algorithms**: Digital signature are electronic signatures, which assures that the message was sent by a particular sender. While performing digital transactions authenticity and integrity should be assured, otherwise, the data can be altered or someone can also act as if he was the sender and expect a reply.
2. **Signing Algorithms**: To create a digital signature, signing algorithms like email programs create a one-way hash of the electronic data, which is to be signed. The signing algorithm then encrypts the hash value using the private key (signature key). This encrypted hash along with other information like the hashing algorithm is the digital signature. This digital signature is appended with the data and sent to the verifier. The reason for encrypting the hash instead of the entire message or document is that a hash function converts any arbitrary input into a much shorter fixed length value. This saves time as now instead of signing a long message a shorter hash value has to be signed and moreover hashing is much faster than signing.
3. **Signature Verification Algorithms**: Verifier receives Digital Signature along with the data. It then uses Verification algorithm to process on the digital signature and the public key (verification key) and generates some value. It also applies the same hash function on the received data and generates a hash value. Then the hash value and the output of the verification algorithm are compared. If they both are equal, then the digital signature is valid else it is invalid.

**The steps followed in creating digital signature are:**

1. Message digest is computed by applying hash function on the message and then message digest is encrypted using private key of sender to form the digital signature. (digital signature = encryption (private key of sender, message digest) and message digest = message digest algorithm(message)).
2. Digital signature is then transmitted with the message.(message + digital signature is transmitted)
3. Receiver decrypts the digital signature using the public key of sender.(This assures authenticity, as only sender has his private key so only sender can encrypt using his private key which can thus be decrypted by sender’s public key).
4. The receiver now has the message digest.
5. The receiver can compute the message digest from the message (actual message is sent with the digital signature).
6. The message digest computed by receiver and the message digest (got by decryption on digital signature) need to be same for ensuring integrity.
7. Message digest is computed using one-way hash function, i.e. a hash function in which computation of hash value of a is easy but computation of a from hash value of a is very difficult.



**Digital Certificate**

Digital certificate is issued by a trusted third party which proves sender's identity to the receiver and receiver’s identity to the sender.

A digital certificate is a certificate issued by a Certificate Authority (CA) to verify the identity of the certificate holder. The CA issues an encrypted digital certificate containing the applicant’s public key and a variety of other identification information. Digital signature is used to attach public key with a particular individual or an entity.

**Digital certificate contains-**

1. Name of certificate holder.
2. Serial number which is used to uniquely identify a certificate, the individual or the entity identified by the certificate
3. Expiration dates.
4. Copy of certificate holder's public key.(used for encrypting messages and digital signatures)
5. Digital Signature of the certificate issuing authority.
6. Digital certificate is also sent with the digital signature and the message.

**Digital certificate vs digital signature:**

Digital signature is used to verify authenticity, integrity, non-repudiation, i.e. it is assuring that the message is sent by the known user and not modified, while digital certificate is used to verify the identity of the user, maybe sender or receiver. Thus, digital signature and certificate are different kind of things but both used for security. Most websites use digital certificate to enhance trust of their users.

**Why security is needed on the Internet?**

The number of people and businesses online is continuing to increase. As access becomes faster and cheaper such people will spend even more time connected to the Internet for personal communication and business transactions.

The Internet is an open communications network that was not originally designed with security in mind. Criminals have found they can exploit its vulnerabilities for fraudulent gain. If the Internet is to succeed as a business and communications tool users must be able to communicate securely.

**What does security provide?**

**Identification / Authentication:**

The persons / entities with whom we are communicating are really who they say they are.

**Confidentiality:**

The information within the message or transaction is kept confidential. It may only be read and understood by the intended sender and receiver.

**Integrity:**

The information within the message or transaction is not tampered accidentally or deliberately with in route without all parties involved being aware of the tampering.

**Non-Repudiation:**

The sender cannot deny sending the message or transaction, and the receiver cannot deny receiving it.

**Access Control:**

Access to the protected information is only realized by the intended person or entity.

All the above security properties can be achieved and implemented through the use of Public Key Infrastructure (in particular Digital Certificates).

1. **Secure Socket Layer**

SSL (Secure Sockets Layers) is a process that manages the security of transactions made on the Internet. The SSL standard was developed by Netscape, together with Mastercard, Bank of America, MCI and Silicon Graphics. It is based on a public-key encryption process to guarantee that data sent over the Internet remain secure. Its principle involves establishing a secure (encrypted) communication channel between two machines (a client and a server) after an authentication phase.

The SSL system is independent of the protocol used, which means it can secure transactions made on the Web via the HTTP protocol as well as connections via the FTP, POP and IMAP protocols. SSL acts as an additional layer, making it possible to guarantee secure data, that is located between the application layer and the transport layer (TCP protocol for example).

As such, SSL is transparent for the user (this means the user may not know he is using SSL). For example, a user using an Internet browser to connect to an e-commerce website protected by SSL will send encrypted data without having to perform any special operation.

Almost all browsers now support the SSL protocol. Netscape Navigator, for example, displays a locked padlock to indicate a connection to an SSL secure website and an open padlock in the opposite case, whereas Microsoft Internet Explorer displays a padlock only for a connecton to an SSL secure site.

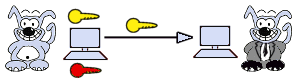
|  |  |
| --- | --- |
| **in Internet Explorer** | **in Mozilla** |
| viewing an SSL secure connection in Internet Explorer | viewing an SSL secure connection in Mozilla |

An SSL secure web server has a [URL](https://ccm.net/contents/295-url) that starts with *https://*, where the "s" of course means *secure*.

In mid-2001, the SSL patent that had until then belonged to Netscape was bought by the *IETF*(*Internet Engineering Task Force*) and was renamed **TLS** (*Transport Layer Security*).

**How SSL 2.0 works?**

Transaction security with SSL 2.0 is based on an exchange of keys between a client and a server. An SSL secure transaction is made according to the following model:

1. Firstly, the client connects to the commercial site protected by SSL and asks it for authentication. The client also sends the list of cryptosystems it supports, sorted in descending order by key length.
2. The server receiving the request sends a certificate to the client, containing the server's public key signed by a certification authority (CA), as well as the name of the cryptosystem that is highest on the list it is compatible with (the length of the encryption key - 40 bits or 128 bits - will be that of the shared cryptosystem having the largest key size).
3. The client verifies the certificate's validity (and therefore the merchant's authenticity), then creates a random secret key (more precisely a supposedly random block), encrypts this key with the server's public key, and then sends the server result (the session key).
4. The server is capable of decrypting the session key with its private key. As such, the two entitites have a shared key that only they know. The remaining transactions can be made using the session key, guaranteeing the integrity and confidentiality of exchanged data.

**SSL 3.0**

SSL 3.0 aims to authenticate the server vis-à-vis the client and possibly the client vis-à-vis the server.

**Behind the Scenes of SSL Cryptography**

**Everything You Want to Know about the Cryptography behind SSL Encryption**

**Background**

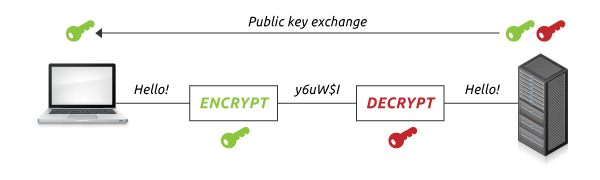
SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser; or a mail server and a mail client (e.g., Outlook). It allows sensitive information such as credit card numbers, social security 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.

**Asymmetric Encryption**

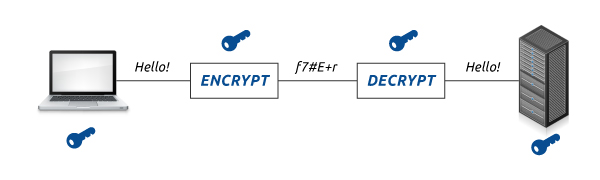
Asymmetric encryption (or public-key cryptography) uses a separate key for encryption and decryption. Anyone can use the encryption key (public key) to encrypt a message. However, decryption keys (private keys) are secret. This way only the intended receiver can decrypt the message. The most common asymmetric 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.



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?**

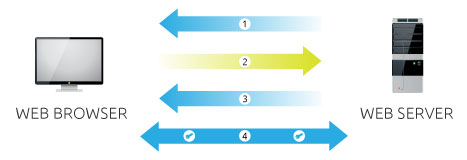
Since asymmetric keys are bigger than symmetric keys, data that is encrypted asymmetrically is tougher to crack than data that is symmetrically encrypted. However, this does not mean that asymmetric keys are better. Rather than being compared by their size, these keys should compared by the following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



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.

**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\*](https://www.digicert.com/ssl-cryptography.htm#note) of certain mathematical problems.

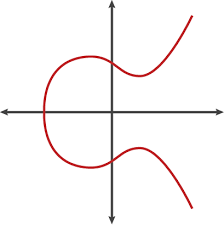
With asymmetric encryption it is computationally easy to generate public and private keys, encrypt messages with the public key, and decrypt messages with the private key. However, it is extremely difficult (or impossible) for anyone to derive the private key based only on the public key.

#### **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.

**ECC**

[Elliptic curve cryptography](https://www.digicert.com/ecc.htm) (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.

**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.

1. **VPN**

A virtual private network (VPN) is programming that creates a safe and encrypted connection over a less secure network, such as the public internet. A VPN works by using the shared public infrastructure while maintaining privacy through security procedures and tunneling protocols. In effect, the protocols, by encrypting data at the sending end and decrypting it at the receiving end, send the data through a "tunnel" that cannot be "entered" by data that is not properly encrypted. An additional level of security involves encrypting not only the data, but also the originating and receiving network addresses.

In the early days of the internet, VPNs were developed to provide branch office employees with an inexpensive, safe way to access corporate applications and data. Today, VPNs are often used by remote corporate employees, gig economy freelance workers and business travelers who require access to sites that are geographically restricted. The two most common types of VPNs are remote access VPNs and site-to-site VPNs.

A Virtual Private Network is a network that is constructed to allow connecting to a private network from a machine not physically present on the network. This allows connecting to say, a private company network, over the internet instead of being physically connected to their network. A connection is formed over an encrypted tunnel to the private network, often emulating creating a separate network interface on the users computer to allow the connection to work as if it were physically connected.

A common use case for a VPN is using a VPN provider to encrypt all network traffic through their connection, to protect online privacy and identity. During the connection process, a VPN sets up an encrypted tunnel for communication to take place through - meaning that all data both to and from your connection is encrypted, before it is sent and after it is received. This means that your data is protected from eavesdropping not just from say, people on the network you're connecting through (such as a public WiFi network), but even from your internet service provider inspecting what packets they are transferring for you.

When it comes to online privacy, a VPN provider shifts where you must place your trust - instead of trusting anyone else on your network (such as WiFi eavesdroppers or router-level inspection), as well as your internet service provider and any intermediaries between them and the destination of your connection (a connection being anything from a web page request to a streaming video connection), you instead must only trust your VPN provider. For this reason, many VPN providers have policies such as not keeping connection/data logs, and try to be transparent about how their network operates.