How Encryption Works

by Jeff Tyson

The incredible growth of the Internet has excited businesses and consumers alike with its promise of changing the way we live and work. But a major concern has been just how secure the Internet is, especially when you're sending sensitive information through it.

Let's face it, there's a whole lot of information that we don't want other people to see, such as:

- Credit-card information
- Social Security numbers
- Private correspondence
- Personal details
- Sensitive company information
- Bank-account information

Friday, March 30, 2001

Item	Price*	Quantity	Gift Wrap*	Sub-Total
Motion Biker Sculpture Lamp Motion Biker Sculpture Lamp	$23.99	1	☐	$23.99

* Charges in parentheses denote additional shipping charges.
* Gift Wrap is $1.99 per item.

- Remove Item by setting its Quantity to 0.
- Remove Gift Wrap by unchecking the Gift Wrap checkbox.
- [illegible]

If the items above are as you wish them, please fill out a shipping address below.

Shipping Information

Mr./Mrs./Ms.

E-commerce relies on the ability to send information securely.

Information security is provided on computers and over the Internet by a variety of methods. A simple but straightforward security method is to only keep sensitive information on removable storage media like floppy disks. But the most popular forms of security all rely on **encryption**, the process of encoding information in such a way that only the person (or computer) with the **key** can decode it.

In this edition of **HowStuffWorks**, you will learn about encryption and authentication. You will also learn about public-key and symmetric-key systems, as well as hash algorithms.

In the Key of...

Computer encryption is based on the science of **cryptography**, which has been used throughout history. Before the digital age, the biggest users of cryptography were governments, particularly for military purposes. The existence of coded messages has been verified as far back as the Roman Empire. But most forms of cryptography in use these days rely on computers, simply because a human-based code is too easy for a computer to crack.

Most computer encryption systems belong in one of two categories:

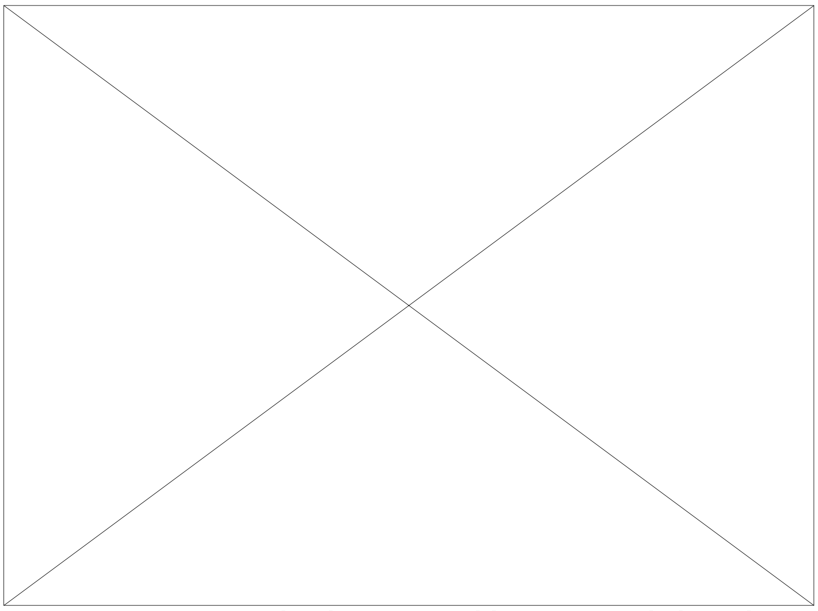
- Symmetric-key encryption
- Public-key encryption

Symmetric Key

In **symmetric-key encryption**, each computer has a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to another computer. Symmetric-key requires that you know which computers will be talking to each other so you can install the key on each one. Symmetric-key encryption is essentially the same as a secret code that each of the two computers must know in order to decode the information. The code provides the key to decoding the message. Think of it like this: You create a coded message to send to a friend in which each letter is substituted with the letter that is two down from it in the alphabet. So "A" becomes "C," and "B" becomes "D". You have already told a trusted friend that the code is "Shift by 2". Your friend gets the message and decodes it. Anyone else who sees the message will see only nonsense.

Public Key

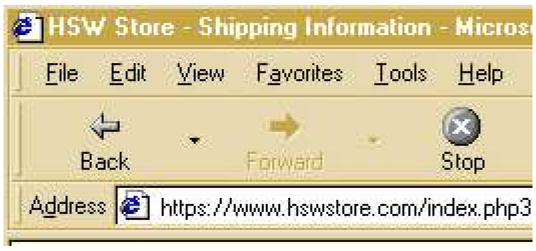
Public-key encryption uses a combination of a private key and a public key. The private key is known only to your computer, while the public key is given by your computer to any computer that wants to communicate securely with it. To decode an encrypted message, a computer must use the public key, provided by the originating computer, and its own private key. A very popular public-key encryption utility is called **Pretty Good Privacy** (PGP), which allows you to encrypt almost anything. You can find out more about PGP at the PGP site.



The sending computer encrypts the document with a symmetric key, then encrypts the symmetric key with the public key of the receiving computer. The receiving computer uses its private key to decode the symmetric key. It then uses the symmetric key to decode the document.

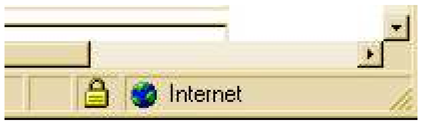
To implement public-key encryption on a large scale, such as a secure Web server might need, requires a different approach. This is where **digital certificates** come in. A digital certificate is basically a bit of information that says that the Web server is trusted by an independent source known as a **certificate authority**. The certificate authority acts as a middleman that both computers trust. It confirms that each computer is in fact who it says it is, and then provides the public keys of each computer to the other.

A popular implementation of public-key encryption is the **Secure Sockets Layer** (SSL). Originally developed by Netscape, SSL is an Internet security protocol used by Internet browsers and Web servers to transmit sensitive information. SSL recently became part of an overall security protocol known as **Transport Layer Security** (TLS).


HSW Store - Shipping Information - Micros
File
Edit
View
Favorites
Tools
Help
Back
Forward
Stop
Address
https://www.hswstore.com/index.php3

Look for the "s" after "http" in the address whenever you are about to enter sensitive information, such as a credit-card number, into a form on a Web site.

In your browser, you can tell when you are using a secure protocol, such as TLS, in a couple of different ways. You will notice that the "http" in the address line is replaced with "https," and you should see a small padlock in the status bar at the bottom of the browser window.



The padlock symbol lets you know that you are using encryption.

Public-key encryption takes a lot of computing, so most systems use a combination of public-key and symmetry. When two computers initiate a secure session, one computer creates a symmetric key and sends it to the other computer using public-key encryption. The two computers can then communicate using symmetric-key encryption. Once the session is finished, each computer discards the symmetric key used for that session. Any additional sessions require that a new symmetric key be created, and the process is repeated.

Hash This!

The key in public-key encryption is based on a **hash value**. This is a value that is computed from a base input number using a **hashing algorithm**. Essentially, the hash value is a summary of the original value. The important thing about a hash value is that it is nearly impossible to derive the original input number without knowing the data used to create the hash value. Here's a simple example:

Input number	Hashing algorithm	Hash value
10,667	**Input # x 143**	**1,525,381**

You can see how hard it would be to determine that the value 1,525,381 came from the multiplication of 10,667 and 143. But if you knew that the multiplier was 143, then it would be very easy to calculate the value 10,667. Public-key encryption is actually much more complex than this example, but that is the basic idea.

Public keys generally use complex algorithms and very large hash values for encrypting, including 40-bit or even 128-bit numbers. A 128-bit number has a possible 2^{128} or 3,402,823,669,209,384,634,633,746,074,300,000,000,000,000,000,000,000,000,000,000,000,000 different combinations! This would be like trying to find one particular grain of sand in the Sahara Desert.

Are You Authentic?

As stated earlier, encryption is the process of taking all of the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Another process, **authentication**, is used to verify that the information comes from a trusted source. Basically, if information is "authentic," you know who created it and you know that it has not been altered in any way since that person created it. These two processes, encryption and authentication, work hand-in-hand to create a secure environment.

There are several ways to authenticate a person or information on a computer:

- **Password** - The use of a user name and password provides the most common form of authentication. You enter your name and password when prompted by the computer. It checks the pair against a secure file to confirm. If either the name or the password does not match, then you are not allowed further access.
- **Pass cards** - These cards can range from a simple card with a magnetic strip, similar to a credit card, to sophisticated smart cards that have an embedded computer chip.
- **Digital signatures** - A digital signature is basically a way to ensure that an electronic document (e-mail, spreadsheet, text file) is authentic. The **Digital Signature Standard** (DSS) is based on a type of public-key encryption method that uses the **Digital Signature Algorithm** (DSA). DSS is the format for digital signatures that has been endorsed by the U.S. government. The DSA algorithm consists of a private key, known only by the originator of the document (the signer), and a public key. The public key has four parts, which you can learn more about at this page. If anything at all is changed in the document after the digital signature is attached to it, it changes the value that the digital signature compares to, rendering the signature invalid.

Recently, more sophisticated forms of authentication have begun to show up on home and office computer systems. Most of these new systems use some form of **biometrics** for authentication. Biometrics uses biological information to verify identity. Biometric authentication methods include:

- Fingerprint scan
- Retina scan
- Face scan
- Voice identification

Another secure-computing need is to ensure that the data has not been corrupted during transmission or encryption. There are a couple of popular ways to do this:

- **Checksum** - Probably one of the oldest methods of ensuring that data is correct, checksums also provide a form of authentication because an invalid checksum suggests that the data has been compromised in some fashion. A checksum is determined in one of two ways. Let's say the checksum of a packet is 1 byte long. A byte is made up of 8 bits, and each bit can be in one of two states, leading to a total of 256 (2^8) possible combinations. Since the first combination equals zero, a byte can have a maximum value of 255.
  - If the sum of the other bytes in the packet is 255 or less, then the checksum contains that exact value.
  - If the sum of the other bytes is more than 255, then the checksum is the remainder of the total value after it has been divided by 256.

Let's look at a checksum example:

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**	**Total**	**Checksum**

212	**232**	**54**	**135**	**244**	**15**	**179**	**80**	**1,151**	**127**

- **1,151 / 256 = 4.496 (round to 4)**
- **4 x 256 = 1,024**
- **1,151 - 1,024 = 127**

- **Cyclic Redundancy Check** (CRC) - CRCs are similar in concept to checksums, but they use polynomial division to determine the value of the CRC, which is usually 16 or 32 bits in length. The good thing about CRC is that it is very accurate. If a single bit is incorrect, the CRC value will not match up. Both checksum and CRC are good for preventing random errors in transmission but provide little protection from an intentional attack on your data. Symmetric- and public-key encryption techniques are much more secure.

All of these various processes combine to provide you with the tools you need to ensure that the information you send or receive over the Internet is secure. In fact, sending information over a computer network is often much more secure than sending it any other way. Phones, especially cordless phones, are susceptible to eavesdropping, particularly by unscrupulous people with radio scanners. Traditional mail and other physical mediums often pass through numerous hands on the way to their destination, increasing the possibility of corruption. Understanding encryption, and simply making sure that any sensitive information you send over the Internet is secure (remember the "https" and padlock symbol), can provide you with greater peace of mind.