

Unlocking the Secrets of Crypto

Sarah Granger 2002-08-13

Unlocking the Secrets of Crypto: Cryptography, Encryption, and Cryptology Explained

by [Sarah Granger](#)

last updated August 13, 2002

Encryption, decryption and code breaking came into the public consciousness in the 1980s with popularity of the movie [War Games](#). It became newsworthy in the 1990s with the legal battles surrounding PGP and the political discussion of the Clipper Chip. Now, with information security becoming more and more of a common concern, the terms encryption, cryptography and cryptology - commonly grouped together under the term "crypto" - are seeping into our daily language. Still, many people are unsure of what these terms refer to. The purpose of this article is to demystify crypto and break it down to simple tools that aid us in achieving satisfactory privacy and security.

[SearchSecurity.com](#) defines cryptography as the "science of information security", which is achieved "by processing data (generally referred to as plaintext) into unintelligible form (ciphertext), reversibly, without data loss." Cryptology is the mathematical science and theory that underlies crypto, while encryption is the actual process by which one applies cryptographic science, a form of encoding. The important concept to understand is that crypto is the application of mathematical algorithms to convert text into a form that is unintelligible to unauthorized viewers.

The History of Cryptography

According to Garfinkel and Spafford's [Practical UNIX and Internet Security](#), cryptography was used as a tactical measure as early as ancient Greece. Spartan generals exchanged secret messages on "ribbons of parchment that were bound spirally around a cylindrical staff." The receiver placed the received parchment on an identical staff in order to decipher its message.

More recently, World War I initiated a re-emergence of the tactical importance of cryptography, a fact that was signaled clearly by the founding of the U.S. Army Cipher Bureau. In the Second World War, the ability of the Allies to crack the Enigma machine, an encryption mechanism developed by Germany, proved to be one of the most pivotal developments of the war. The concept of basic digital computers, particularly the famed [Turing Machine](#), arose as part of the

Allied effort to breaking the German codes.

The National Security Agency (NSA)

After the war, various arms of the U.S. government fought over control of cryptography through the late 1940s. As a result of jurisdictional problems, the government decided to consolidate efforts under one agency, [The National Security Agency \(NSA\)](#), which officially began operations in 1952.

As the U.S. government's cryptographic arm, the NSA's job was, and continues to be, to quietly make and break codes. In the early 1970s, scientists and mathematicians working for the U.S. military began work on the [Data Encryption Standard \(DES\)](#). At the same time, with more and more data being transmitted over open networks, cryptography started becoming a more popular area of research and development outside the military. This would eventually lead to considerable conflict between private sector developers and the government.

Encryption as Munition

In 1993, the U.S. government made it illegal to export any cryptographic software with over 40-bit keys outside the U.S. without special munitions export licenses. [PGP \(Pretty Good Privacy\)](#), released for free by Phil Zimmermann in 1991, grew quickly in popularity as a tool for strong personal encryption. PGP was soon targeted as the NSA and U.S. Customs, who were concerned that having strong encryption in the hands of potential enemies of the state would endanger national security and had it designated as a munition.

As a result, it became illegal to export PGP, an edict that was particularly problematic given the borderless nature of the Internet. A legal battle ensued that raged on for a few years, and dissipated in 1996 as it became clear that PGP was not a real threat to U.S. national security. Instead, a compromise was reached so that crypto developed outside the military is no longer considered a munition.

Today, encryption technology is very robust. There are still some international export issues between countries, but most have been settled at least to the point of reasonable usability. Crypto algorithms can be patented and/or copyrighted. [RSA](#), for example, was patented by the parent company, [RSA Data Security](#), and released the algorithm to the public domain in 2000, a few days before the patent expired. Crypto, however, is one area where algorithms are often

left open for ease of developers further improving upon the algorithm and verifying its strength.

Practical Applications

Encryption is generally employed for four functions:

Authentication

Authentication is the process of verifying that the user is in fact the legitimate, authorized user. This is most commonly seen in encrypted passwords. Another example of authentication is digital signatures. The California Secretary of State says that: "a 'digitally-signed communication' is a message that has been processed by a computer in such a manner that ties the message to the individual that signed the message." In other words, they ensure that information that is being transmitted comes from the person it is supposed to come from (i.e., not a third-party spoofing the legitimate user's identity). There are many options for digital signatures. This generally requires a trusted third party, like [VeriSign](#).

Integrity

This refers to system integrity and data integrity. For example, encryption allows users to confirm that the system has not been breached and that the system retains its expected completeness, consistency and integrity.

Confidentiality

Confidentiality, or the protection of data from unauthorized viewing, is the most basic form of data security. By obscuring the message as ciphertext, encryption allows senders of messages to obscure the original message so that the confidentiality of the message is assured. For example: encrypting raw data like one's entire hard drive can be done simply to ensure that if the network has been hacked, the data on the disk is still secure from unauthorized viewing by someone on the outside. Encryption can also be used to ensure the confidentiality of e-mail messages being sent over the Internet.

Non-repudiation

Non-repudiation refers to the ability of both parties in a transaction to prove that the other party has in fact participated in a communication. In other words, neither party can effectively

deny that they have exchanged messages with the other party. This may be pertinent in situations in which intellectual property is discussed. Non-repudiation is generally considered to constitute undeniable proof of delivery through multiple types of assurances by binding the users and the transaction including origin of data, the data sender, and time of receipt. The use of encryption keys to encrypt and decrypt messages can provide such proof.

How Encryption Works

Any form of data can be encrypted, including single bits, small strings such as passwords, individual files or huge databases of multiple files. Encryption algorithms work in different ways, so the use often dictates the proper algorithm. And sometimes a combination of strong algorithms is used to ensure even tighter security.

Methodology

In essence, encryption is said to occur when data is passed through an algorithm, or a series of mathematical operations that generate an alternate form of that data. Crypto expert, [Bruce Schneier](#) defines cryptographic algorithms as "the mathematical function used for encryption and decryption." Because the algorithm is a set procedure, it can be reversed so that the encrypted data can be unencrypted and restored to its original, readable form.

An example of a very basic cipher is the Captain Midnight Secret Decoder Ring. In this case, a ring exists with the 26 letters of the alphabet on it. The sender & receiver of the message agree how far to turn the ring to encode/decode messages in order to encrypt message in question. For instance, they may decide to set a coding pattern of minus one letter, so that the letters in plaintext (non-encrypted) message are replaced by the letter preceding it in the alphabet. This can best be exemplified by the moview [2001: A Space Odyssey](#), in which the computer HAL 9000 got its name by running "IBM" through this very basic algorithm. This is most definitely not strong crypto and can be broken in a half second by the most basic computer, but it's fun.

The mathematical concept behind cryptography is that certain equations are incredibly difficult to solve unless very specific criteria are met. "An algorithm is considered computationally secure if it cannot be broken with available (current or future) resources," explains Schneier. Strong cryptographic algorithms can be made stronger by publishing the algorithm and allowing anyone and everyone to try and prove its insecurity. This is why open source suits crypto development well.

Key-Based Ciphers

A key is the means by which messages are encrypted and decrypted. Key-based encryption is generally considered to be very secure. With key-based ciphers, even if an attacker knows the ciphertext (the encrypted form of the data) and the algorithm, neither the plaintext (data to be secured) nor the key can be determined from the attacker's information.

One way in which security of the encryption can be strengthened is by increasing the length of the key. In terms of crypto, key length equates to the number of possible options of keys available to decrypt a certain code. Key length has an exponential effect on the strength of encryption. An 8-bit key has 2^8 (256) possible outcomes. 40-bit encryption provides 2^{40} or 10^{12} (1 trillion) possible keys for decrypting ciphertext. To understand this more clearly, think of this analogy: assume we have 10 keys in front of us to open a door. That's annoying enough. Try 10^3 (1000). That will take all day. 10^{10} (10,000,000,000) would take forever. In the same manner, each time the key length is increased, we increase the number of keys that potential crackers have to try in order to crack the encrypted message.

Key-Based Algorithms

The two types of key-based algorithms are symmetric and asymmetric.

Symmetric (Single Key) Cipher

In symmetric encryption, the same key is used for encryption and decryption and must be kept secret, thereby requiring the sender and receiver to agree on the same key before making any data transmissions. According to [Mark Russinovich](#): "symmetric algorithms are typically very fast, which makes them suitable for encrypting large amounts of data, such as file data." DES is the most common symmetric cipher.

Digital Encryption Standard (DES), which was adopted in 1977, has a 56-bit key. It was proven insecure in 1997 and cracked in 56 hours by a machine built by the [Electronic Frontier Foundation](#) in 1998. This gave rise to Triple DES, a way of using DES on itself for greater security. Typically this would consist of a three-step encryption operation, i.e. encrypt the file with 1 key, decrypt it with another, and encrypt it again with a third.

Originally entitled Rijndael, the [Advanced Encryption Standard \(AES\)](#) was adopted by NIST in

2000 out of a competition of numerous algorithms developed in the crypto community. The cipher has a variable block length and key length, usable with keys of 128, 192, or 256 bits to encrypt blocks with a length of 128, 192 or 256 bits. According to the [Rijndael home page](#), "both block length and key length can be extended very easily to multiples of 32 bits."

Asymmetric Encryption

In asymmetric encryption, which is also known as public key encryption, the key used to encrypt differs from that used to decrypt. The key used to encrypt is generally called the "public key" and the decryption key is the "private key" or "secret key". The cool thing about asymmetric ciphers is that they can authenticate transactions across a public channel, providing a receipt ("signing") when the private key is used on the receiving end.

If copies of any keys are stored anywhere outside the main user's traditional storage location, they are considered to be in escrow. Some tools, such as PGP, put these in [certificate authorities \(CA\)](#) for trust in the public keys. Its key certificate contains the public key, user ID related to the key, date created, and if desired, a list of digital signatures on the key ring, signed by people who "attest to that key's veracity." [RSA \(Rivest-Shamir-Adleman\)](#), released in 1977, is the most common asymmetric cipher. It is now in the public domain, open for anyone to use or improve.

Hash Cipher

In essence, a hash is a string of text that is generated by a message that serves as a unique identifier of that message. The text of the message, as well as date and time of submission, is put into the hash algorithm, which spits out a string of text. The hash is used to verify the document integrity by placing the document through the hash functionP: "The recipient decrypts both the message and the hash, produces another hash from the received message, and compares the two hashes. If they're the same, there is a very high probability that the message was transmitted intact" ([Webopedia, "Hashing"](#)). [MD5](#) is the most widely used hash algorithm, with a 128-bit fingerprint. MD5 is also in the public domain. SHA1 has a 160-bit output, and is often used for digital signatures.

One-Time Pads

Invented in 1917, only one-time pads are theoretically unbreakable. Visualize a pad with a code

on one side, keys on the other. Each key is used exactly once and for one message. Key letters are generated randomly, each random letter used once on one "pad" to decrypt one character. Each new message requires a new pad. Length of keys is equals message length.

Encryption Tools

An abundance of commercial and open source options are available now for various crypto applications. Here are some of the most widely used (and generally more secure) tools:

PGP

PGP, which stands for Pretty Good Privacy, is undoubtedly the most famous encryption tool ever. It is based on the RSA algorithm and is most commonly applied to e-mail transmissions. In fact, according to Phil Zimmermann, creator of the PGP tool, over 90% of e-mail encryption is done with PGP. PGP is available on a few sites now as a commercial product, further developed by Network Solutions (NAI) in the late 90s, but they no longer support the product. PGP Freeware is available all over the world. [The International PGP Home Page](#) is the best resource for that. All forms of PGP can be found through [Zimmermann's own site](#).

GPG

[GPG \(Gnu Privacy Guard\)](#) replaced PGP to some extent as a more widely available open solution internationally due to its development outside the United States GPG is gaining in popularity, particularly after NAI shut down the commercial support of PGP and with the rise of Linux.

S/KEY

The best way to secure a password, S/KEY is a true one-time pad based password scheme. Passwords are long, so that key length is long. To use S/KEY the user must carry around a list of one-time passwords for each password entry. The keys should never be stored on-line.

SSH and SCP

SSH (Secure Shell) is a secure version of Telnet, whereas SCP (Secure Copy) is a secure version of copy, which also replaces the insecure FTP. They work by authenticating known hosts using RSA.

SSL

Developed by Netscape, the Secure Socket Layer handshake protocol authenticates servers and clients by using the RSA algorithm. OpenSSL is a newer version available for open platforms.

Encryption tools are constantly evolving. With greater open source development and use, crypto is in a perfect position for the future. Open PGP and GPG lead the way, with OpenSSH and OpenSSL running on OpenBSD, etc. Bugs crop up occasionally, but developers in the open source community are usually able to resolve them quickly.

Warning: Crypto is Not Always Secure

Given enough time and computing power, all key-based algorithms are theoretically breakable depending on time and computing resources. While encryption sounds like it's the best thing for security, there are a few things to consider:

- **Key length** - Keys can be cracked by brute force and the right hacker. Most current data considers keys shorter than 128 bit keys to be at risk for a brute force attack with symmetric algorithms. So far, Triple DES and AES have yet to be broken. This is due to their extraordinary key lengths. As for asymmetric or public key lengths, Bruce Schneier recommends 1280 bits through 2005 for individuals, 1536 for corporations, and 2048 for governments.
- **Programmer error** - Even with strong encryption algorithms, some software programs contain other bugs that can make passwords and such easy to hack which, in turn, means that no matter how good the encryption is, the supposedly secured information can be read.
- **User error** - It is said in encryption circles that using encryption poorly is worse than not using encryption at all. While this might be an overstatement, the point is that encryption may provide a false sense of security in some circumstances: a user may disregard fundamental security practices for a file if it is encrypted. This is a security weakness because some information should never be sent – encrypted or not.
- **Obscurity** - The term "security by obscurity" can be found often in recent literature generally referring to hiding important data, rather than securing it in a known location. Bruce Schneier uses the example of a letter, locked in a safe hidden in an undisclosed location within one state. That is security by obscurity. Strong security in this case would be locking a letter in a safe, knowing that safe's location, and providing design

specifications of that safe along with 100 safes of the same type to anyone who wants to try and break in. Security by obscurity may work for keeping physical items safe from the random burglar, but it will not work for data on a network.

- **Cryptanalysis** - Cryptanalysis is the study of breaking ciphers (uncovering codes & algorithms). Assuming an algorithm or product is breakable, cryptanalysis can be conducted in a variety of ways, the most common being the brute force method. Brute force means trying all keys until the end result resembles estimated plaintext. This can only realistically be done with special hardware, and/or multiple machines running in parallel. Sometimes codes are deduced by using ciphertext of several messages, definitely with better results if some plaintext is available as well. The 56-bit key broken in 1995 took one week and 120,000 processors.

The Future of Crypto

Crypto today seamlessly works in the background more than in the past. Crypto is becoming easier to install and understand, and more types are available to the average user. As a result, NIST created the federal Public Key Infrastructure or PKI. According to the [University of Toronto PKI Assessment Project](#), "PKI is the term generally used to describe the laws, policies, standards, and software that regulate or manipulate certificates and public and private keys. All of this implies a set of standards for applications that use encryption." Currently, the NIST PKI is mainly a center of information on various encryption tools and technologies.

As GPG's popularity grows, PGP's future as a commercial product remains up in the air. Zimmermann is currently working to buy the software back from NAI in order to further develop and guide the direction of the company. After all of Zimmermann's battles, he's still fighting: "If I succeed in getting PGP out of NAI's hands, I will devise a strategy to prevent PGP from falling into another intellectual property black hole."

Cryptography is increasingly becoming the centerpiece in the attempt by corporate interests to protect their intellectual property. This is particularly true in the case of the TCPA (Trusted Computing Platform Alliance) or Microsoft's Palladium. Furthermore, the attempt to apply cryptography to protect copyrighted material against unauthorized reproduction is still creating tremendous debate, particularly regarding the CBDTPA (Consumer Broadband and Digital Television Promotion Act), not to mention the BPDG (Broadcast Protection Discussion Group). The general idea behind all of these groups and concepts is to mandate strong security through crypto hardware, which greatly benefits the entertainment industry, while potentially

threatening the Open Source community. The jury's still out on most of these, but crypto is a large piece of the puzzle.

Recommended Reading

For more specifics on crypto, read the bible: [Applied Cryptography](#) by Bruce Schneier.

Relevant Links

[International Association for Cryptologic Research](#)

[Crypto Resources](#)

Cryptonomicon.net

[What is MD5?](#)

Jim Ellis, CERT

[Secure Hash Standard](#)

FIPS (Federal Information Processing Standards Publication)

[Practical UNIX and Internet Security, O'Reilly](#)

Garfinkel, Simson & Spafford, Gene

[Non-Repudiation in the Digital Environment](#)

Adrien McCullagh and William Caelli, First Monday

[NIST PKI Program](#)

National Institute of Standards and Technology

[A Brief History of Cryptography & Cryptanalysis](#)

Oliver Pell

[NSA History](#)

Pike, John

[Inside Encrypting File System](#)

Mark Russinovich

[Applied Cryptography: Protocols, Algorithms, and Source Code in C](#)

Bruce Schneier

[Block Encryption for the 21st Century](#)

Bruce Schneier, Dr. Dobbs's Journal

Is 1024 Bits Enough?

Bruce Schneier, Crypto-Gram, Counterpane Internet Security, Inc.

Secrets and Lies: Digital Security in a Networked World

Bruce Schneier, Wiley & Sons

University of Toronto PKI Assessment Project

University of Toronto

Crypto Bibliography

Phil Zimmerman

Where to Get PGP

Phil Zimmerman

[Privacy Statement](#)

Copyright 2006, SecurityFocus