

# Cryptology: History, Reasons, and Methods

Dan Evrard Nate Snyder  
Tyler Herzog Nicholas Tietz

Kent State University

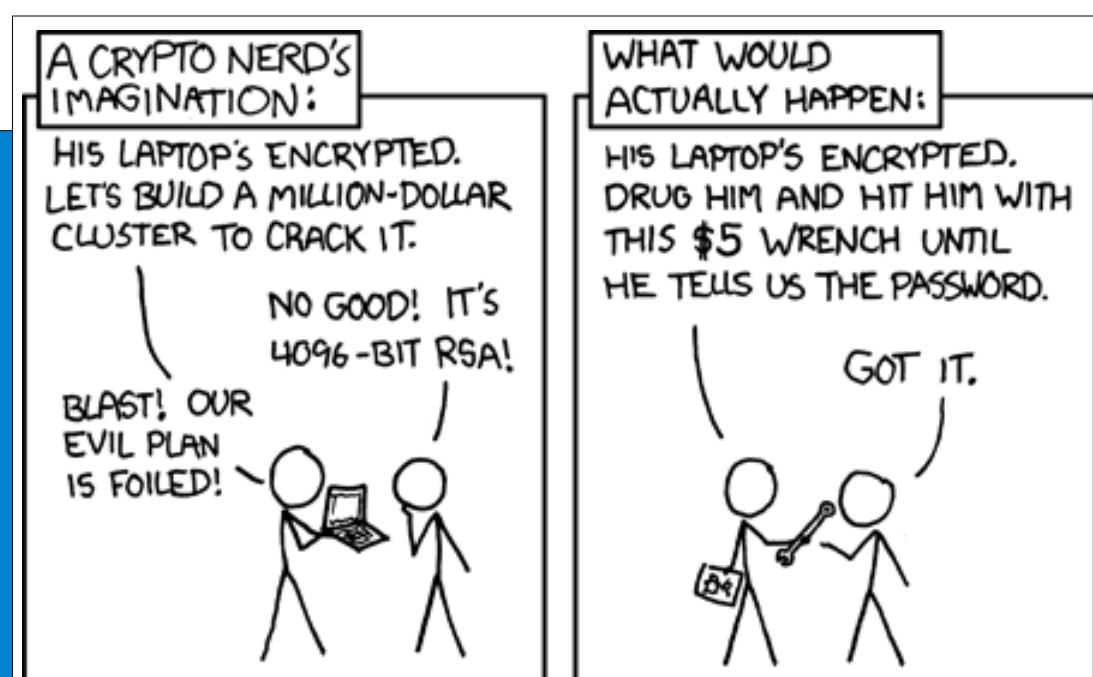


## Why do we need cryptology?

Cryptography is defined as “the science or study of the techniques of secret writing, esp. code and cipher systems, methods, and the like.” Cryptography is needed so that text can be kept secret. It is easy to imagine situations in ancient times where a writer who sent a message via courier would want to make sure that if the runner were intercepted, the interceptors could not read the message. Cryptography and encryption have been particularly important throughout history in times of war when a general would not want the enemy to figure out the plans he was distributing among his troops. Recently, the uses of cryptography have grown drastically. Cryptography is still important in times of war, but with the advent of computers and with it the vast amount of information being shared on the internet, there has been a need to create better, more efficient encryption strategies to protect private information, such as credit card numbers, private communications, and so on.

## History: Man to Machine

Cipher systems can be dated to as early 600 BC, where the Atbash or Caesar Ciphers which were simple, but effective. They did not need to be complex since most people of that time period did not know how to read, let alone decipher it. Simple human-based ciphers were used all the way up to World War 1, when mechanical encryption machines allowed for far more complex ciphers, but also lead to more complex deciphering machines. In the 1960s, computers created faster and harder to crack ciphers. Eventually, the technology age created a need to protect information that was being transmitted electronically. This was eventually accomplished with the use of public key cryptography. Encryption will continue to advance, because with each advance of encryption, an advance in cracking is quick to follow. No encryption method has existed for long periods of time without being cracked – modern encryption is just the latest step in the game.



Diffie, Whitfield. "INFORMATION SECURITY: 50 YEARS BEHIND, 50 YEARS AHEAD." *Communications of the ACM* 51.1 (2008): 55-57. Academic Search Complete. EBSCO. Web. 14 Mar. 2010.  
Stalling, William. *Cryptography and Network Security: Principles and Practice*. 5th ed. Prentice Hall, 2006. Print.

## Caesar's Cipher

Caesar's cipher, or Caesar's shift, is one of the most common encryption methods. It is a substitution cipher involving the shifting of letters. The shift can be any amount in either direction, resulting in an encrypted plaintext.

For example, using a shift of 5 left, the alphabet can be viewed as follows:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
F G H I J K L M N O P Q R S T U V W X Y Z A B C D E

This cipher is named after its creator, Julius Caesar, who used it to protect messages he sent in the military. This cipher is not very complex, making it simple to crack. The simplest way to do this, especially with the advent of computers which can manipulate strings quickly, is to simply try all possible keys and check which yields the correct text (Stallings 40).

## Vigènere Cipher

The Vigènere Cipher is a form of alphabet encryption involving multiple uses of a Caesar's shift determined by the letters in a keyword. The cipher was originally developed by Giovan Battista Bellaso in 1553. It was later credited to Blaise de Vigènere sometime in the 1800s. The Vigènere Cipher is stronger than the Caesar's cipher because of the multiple shifts. However, the way it works is rather simple. The tool used for this is a table of alphabets, or a “Vigènere square.” When encrypting a word, different alphabet rows are used depending on a repeated keyword. The keyword can be anything. For example, let's say the plaintext to be encrypted is “THEMUSICNOTE.” Whoever is encrypting the message chooses a keyword to use and repeats it until the number of characters matches. For example, we will use the keyword “WATER.” It is repeated until the characters match to look like this: “WATERWATERWA.” The first letter of the plaintext is T, so using the column T and row W, you find the letter that is replaced. In this case, it is P. The rest of the code encrypted is as follows:

Plaintext: THEMUSICNOTE  
Keyword: WATERWATERWA  
Ciphertext: PHXQLOIVRFPE

This is much more difficult to decipher, as the letter frequency is disrupted. There are many tests to help decipher the ciphertext, such as the Friedman test and Kasiski examination, but knowing the length of the keyword makes it much easier.

## Data Encryption Standard

Data Encryption Standard (DES) is a block cipher that was approved as a standard in 1976. It has a block size of 64 bits, but only 56 of these bits are actually used by the algorithm, so its key length is actually 56 bits. The encryption process is relatively simple. First, the 64 bits of plaintext undergo an initial permutation. The text is then divided into two 32-bit sections which are processed alternately. The encryption is then broken up into sixteen rounds in which one of the 32-bit sections undergoes the Feistel function and is then combined with the other section using an XOR operation. The two sections are then swapped before the next round. After the final round though, the two sections are NOT swapped. The final permutation takes place at the end to undo the initial permutation. This process does have limitations, as it can be brute forced easily, due to a small key size. Triple DES is considerably safer, but can still be cracked given enough time and resources, making RSA or Diffie-Hellman generally better choices than DES or TDES.

## Diffie-Hellman Algorithm

The Diffie-Hellman symmetric key-exchange protocol has a few key advantages: it does not need known keys before communication begins, and is extremely difficult to crack because of the discrete log problem. It has a major weakness, however, as it is subject to man-in-the-middle attacks. The process is relatively simple:

- Let the two computers involved be named A and B.
- These two computers agree on a prime modulus  $p$  and a base  $b$  coprime to  $p$ , both of which are public.
- A chooses a random prime number  $s$  such that  $1 < s < p - 1$ . A then computes  $S = b^s \% p$ .
- B chooses a random prime number  $t$  such that  $1 < t < p - 1$ . B then computes  $T = b^t \% p$ .
- A sends  $S$  to B, and B sends  $T$  to A.  $S$  and  $T$  are both public information now, but  $s$  and  $t$  are both still private.
- A computes  $K = T^s \% p$ , and B computes  $K = S^t \% p$ .
- These are the same since  $T^s = (b^t)^s = b^{(t*s)} = (b^s)^t = S^t \pmod{p}$ .

Through this process, A and B are able to obtain a private key.