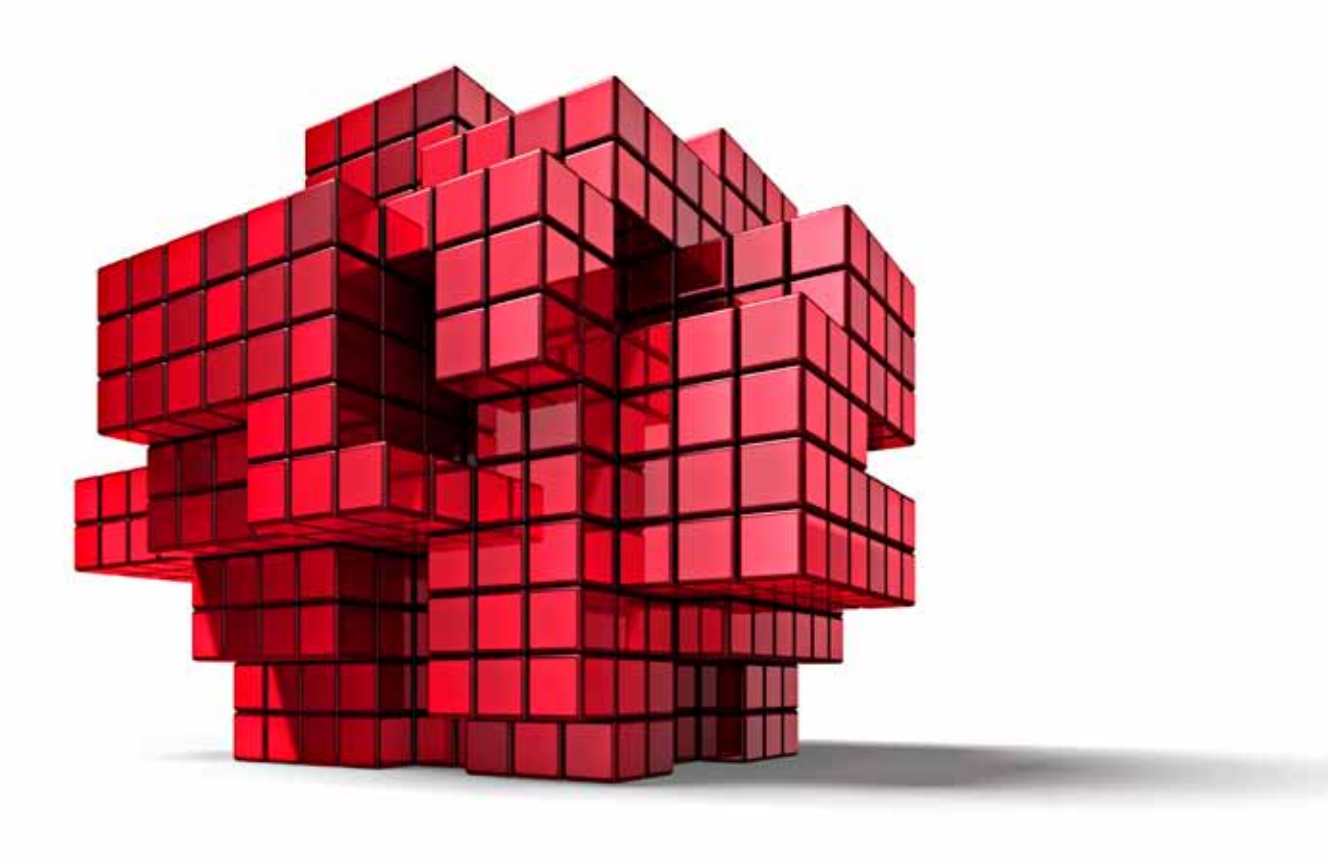
One Time Pad Encryption

The unbreakable encryption method



mils
ELECTRONIC

One Time Pad encryption is the only proven unbreakable encryption method

Dear Reader

One Time Pad encryption is a very simple, yet completely unbreakable cipher method. It has been used for decades in mils electronic cipher systems for encrypting our customers' sensitive data.

Over the years, we have perfected the implementation of One Time Pad encryption into our products. Today, our high level of automation, high capacity storage media, continuous key protection and huge One Time Pads offer our customers outstanding message security without sacrificing convenience.

This document will help you understand how One Time Pad can ensure complete privacy for your sensitive information.

With best regards,

Otto Kugler
CEO

mils ELECTRONIC

Characteristics of the One Time Pad encryption method

The One Time Pad encryption method is a binary additive stream cipher, where a stream of truly random keys is generated and then combined with the plain text for encryption or with the ciphertext for decryption by an ‘exclusive OR’ (XOR) addition.

It is possible to prove that a stream cipher encryption scheme is unbreakable if the following preconditions are met:

- ***The key must be as long as the plain text.***
- ***The key must be truly random.***
- ***The key must only be used once.***

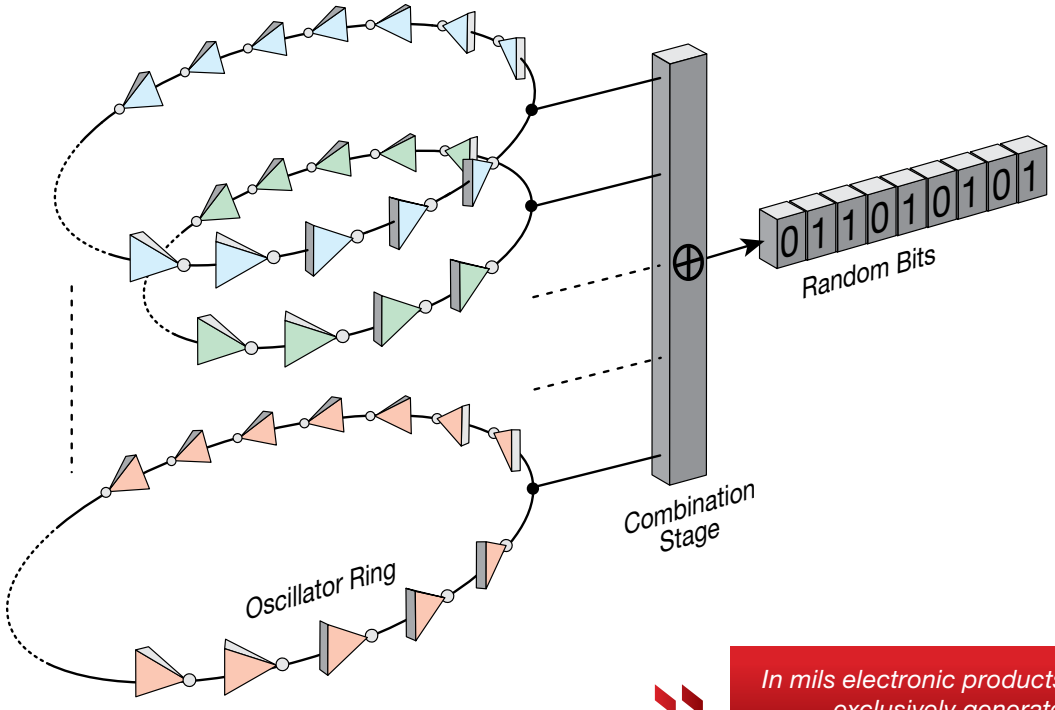
The One Time Pad implementation in mils electronic’s products fulfils all these requirements and therefore provides absolute protection for our customers’ sensitive information.

How does it work?

True random key generation

For One Time Pad encryption, a truly random key stream must be employed. The word ‘random’ is used in its most literal sense here.

In mils electronic products, all keys are exclusively generated by a ‘True Random Noise Source’. This Noise Source is incorporated into the hardware security token of each mils electronic application. Integration into the Security Token ensures tamper resistant protection and a very high key generation speed.



In mils electronic products, all keys are exclusively generated by a ‘True Random Noise Source’

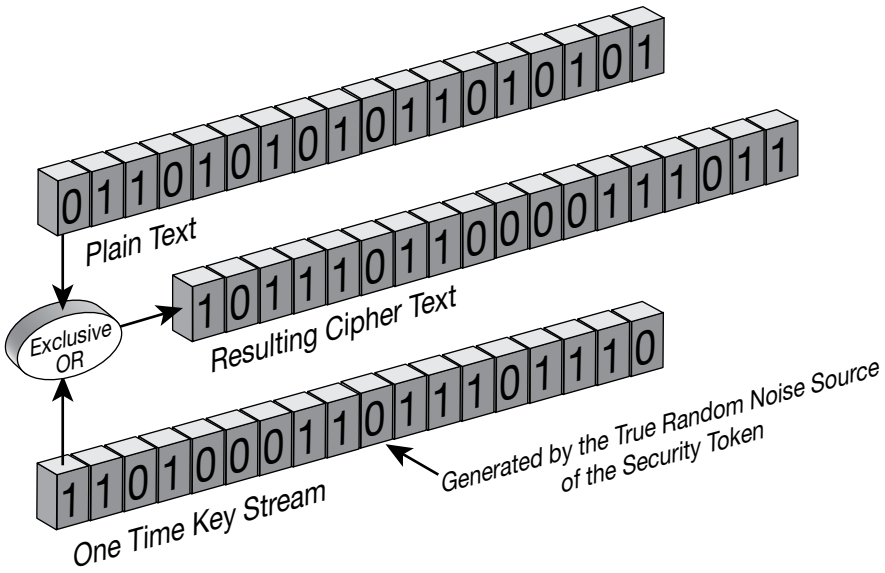
The random noise source derives its random bits by sampling a set of parallel ring oscillators, a reliable technology for obtaining truly random bits. This technique uses timing jitter and oscillator drift found in free-running CMOS ring oscillators as a source of randomness. Timing jitter is a random phenomenon caused by the thermal noise and local voltage variations present at each transistor of a ring oscillator.

Local variations in voltage and temperature will cause each ring to oscillate faster (or slower) over time - resulting in a random drift relative to the other rings. As the frequency of each oscillator randomly drifts with each cycle, the output stream becomes random relative to the lower frequency sampling rate.

The encryption process

One Time Pad keys are used in pairs. One copy of the key is kept by each user and the keys are distributed securely prior to encryption.

The confidentiality and authenticity of the One Time Pad keys are assured by continuous protection during their distribution and storage. This guarantees that outsiders will not be able to misuse the key (e.g. by copying or altering the key during distribution).

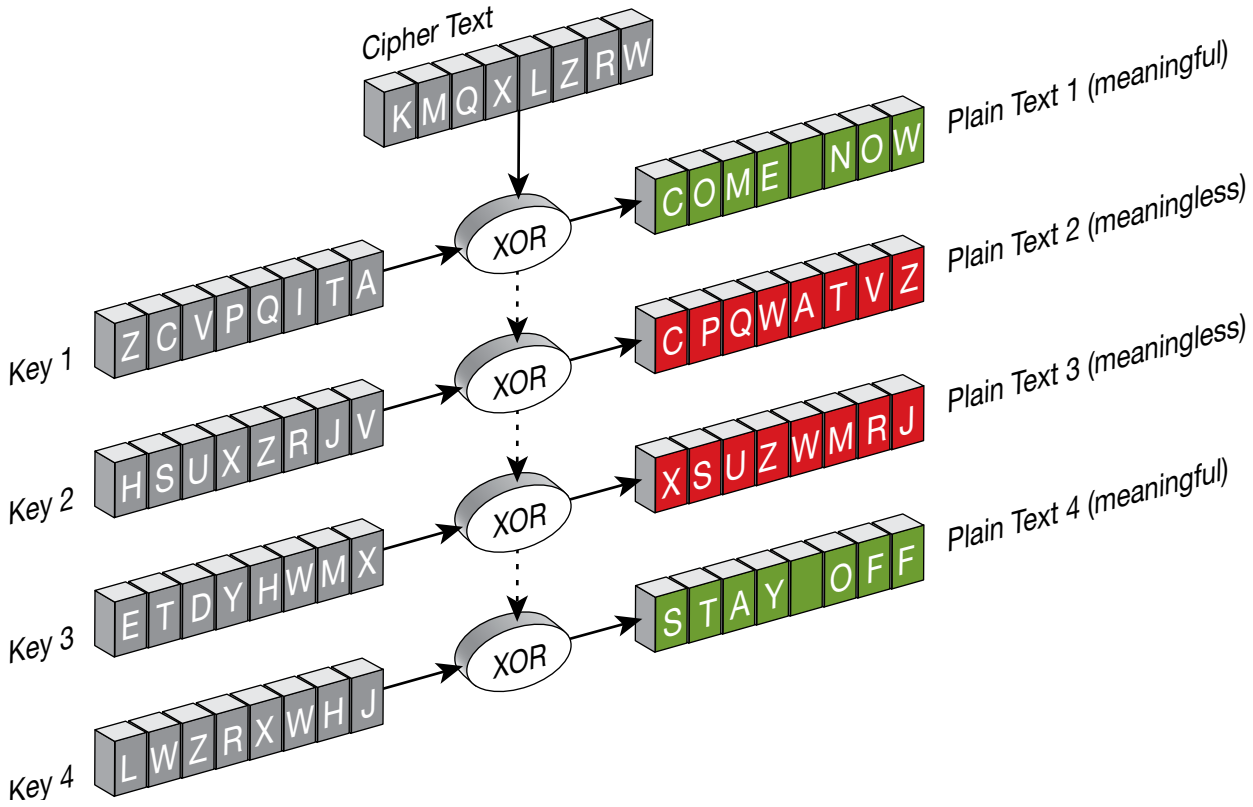


- To encrypt plain text data, the sender uses a key string equal in length to the plain text. The key is used by mixing (XOR-ing) bit by bit, always a bit of the key with a bit of the plain text to create a bit of cipher text.
- This cipher text is then sent to the recipient.
- At the recipient's end, the encoded message is mixed (XOR-ed) with the duplicate copy of the One Time Key and the plain text is restored.
- Both sender's and recipient's keys are automatically destroyed after use, to ensure re-application of the same key is not possible.

Why is One Time Pad encryption unbreakable?

The popular scientific explanation

With One Time Pad encryption, the key used for encoding the message is completely random and is as long as the message itself. That is why the only possible attack to such a cipher is a brute force attack.



The ‘Brute Force’ attack

Brute force attacks use exhaustive trial and error methods in order to find the key that has been used for encrypting the plain text. This means that every possible combination of key bits must be used to decrypt the cipher text. The correct key would be the one that produces a meaningful plain text.

Unlimited computing power is useless

Let's assume an eavesdropper has intercepted a One Time Pad encrypted message and that he has unlimited computing power and time. A brute force attack would be very expensive for a plain text of reasonable size. For example, typical e-mail messages are at least 200 bytes long, requiring the testing of 1.600 bits. Even if the eavesdropper is both willing and able to do this, the following paragraph will describe why unlimited computational power will not compromise the system.

Attackers must try every possible key

Since all One Time Keys are equally likely and come from a completely unpredictable noise source that is provably random, the attacker has to test all possible key strings.

Impossible to guess the right plain text

If he used every possible key string to decrypt the cipher text, all potential plain text strings with the same length as the original plain text data would appear. As illustrated on the left-hand side, most of these potential plain text strings would not make sense; however, every meaningful string the same length as the original plain text data would also appear as a potential plain text string.

Without knowing the applied OTP, the eavesdropper has no way of finding out which meaningful string is the original plain text. Thus, trying all possible keys doesn't help the attacker at all, since all possible plain texts are equally likely decryptions of the cipher text.

Why is One Time Pad encryption unbreakable?

The mathematical proof

According to *Alfred Menezes* et al. in their book, *Handbook of Applied Cryptography,* a system can be called perfectly secret, or unconditionally secure, when observing cipher text gives an eavesdropper no additional information about the original plain text string. If we let L be the number of bits in the plain text string, then i ranges from 1 to L in the following definitions:

pi = the i^{th} bit in the plain text string

ci = the i^{th} bit in the cipher text string

ki = the i^{th} bit in the key string

P(pi) = the probability that pi was sent

P(pi | ci) = the probability that pi was sent given that ci was observed

A system can be called perfectly secret when **P(pi) = P(pi | ci)**. This section will prove that an One Time Pad system is perfectly secret.

In traditional stream cipher systems, the most common method of mixing plain text data bits with key bits is by performing the XOR operation on the corresponding bits. XOR is short for exclusive OR. The following is a table that defines XOR (you can think of column a as a bit of plain text and column b as its corresponding key bit):

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

The sender makes cipher text by XOR-ing plain text and key one bit at a time:

ci = pi XOR ki (1)

where ci, pi, and ki are as defined above.

Because the One Time Pad key is completely random and unpredictable, two conclusions can be drawn:

- First, the probability of observing any particular One Time Pad key bit is equal to the probability of observing any other One Time Key bit.
- Second, knowing all the previous values of key in a sequence tells us nothing about the next key bit.

By stating another definition,

P(ki) = the probability that ki was used to create ci

the first conclusion drawn above can be written as

P(ki=1) = P(ki=0) = 1/2 for all i. (2)

In other words, a bit of One Time Key is just as likely to be a 1 as a 0 at any time. The second conclusion drawn above allows us to consider triples of key, cipher text, and plain text for a particular value of i without regard for other triples.

Equation (1) leads to an important observation: knowing any two of {pi , ci , ki } determines the third. Likewise, given one of {pi , ci , ki }, a second one can be written in terms of the third. For example,

P(ci=1 | ki=0) = P(pi=1);

in other words, if we know for a fact that the key bit is 0, then plain text and cipher text must be equal.

In order to show that P(pi | ci) = P(pi), we first need to show P(ci) = P(ci | pi). Using equation (1), we will do this explicitly by first deriving the distribution of P(ci). Next, we will derive the distribution of P(ci | pi) given that the plain text bit is a 0 and then given that it is a 1.

Distribution of $P(c_i)$

$P(ci=1) = P(ci=1 \mid ki=1) P(ki=1) + P(ci=1 \mid ki=0) P(ki=0)$

by the definition of conditional probability

$= P(pi=0) P(ki=1) + P(pi=1) P(ki=0)$ by equation (1)

$= P(pi=0) (1/2) + P(pi=1) (1/2)$ by equation (2)

$= (1/2) [P(pi=0) + P(pi=1)]$ regrouping

$= 1/2$ since pi can only be 1 or 0

$P(ci=0) = P(ci=0 \mid ki=1) P(ki=1) + P(ci=0 \mid ki=0) P(ki=0)$

by the definition of conditional probability

$= P(pi=1) P(ki=1) + P(pi=0) P(ki=0)$ by equation (1)

$= P(pi=1) (1/2) + P(pi=0) (1/2)$ by equation (2)

$= (1/2) [P(pi=1) + P(pi=0)]$ regrouping

$= 1/2$ since pi can only be 1 or 0

Distribution of $P(c_i \mid p_i)$

If pi=0:

P(ci=0 | pi=0) = P(ki=0) by equation (1)

= 1/2 by equation (2)

P(ci=1 | pi=0) = P(ki=1) by equation (1)

= 1/2 by equation (2)

If pi=1:

P(ci=0 | pi=1) = P(ki=1) by equation (1)

= 1/2 by equation (2)

P(ci=1 | pi=1) = P(ki=0) by equation (1)

= 1/2 by equation (2)

It is clear from the distributions derived above that **P(ci | pi) = P(ci)**. Recall that a system can be called perfectly secret when **P(pi) = P(pi | ci)**. Using the definition of conditional probability, the joint probability, P(pi and ci), the probability that pi and ci are observed, can be written in the following two (equivalent) forms:

P(pi and ci) = P(ci | pi) P(pi)

and

P(pi and ci) = P(pi | ci) P(ci).

Combining the two equations gives

P(pi | ci) P(ci) = P(ci | pi) P(pi).

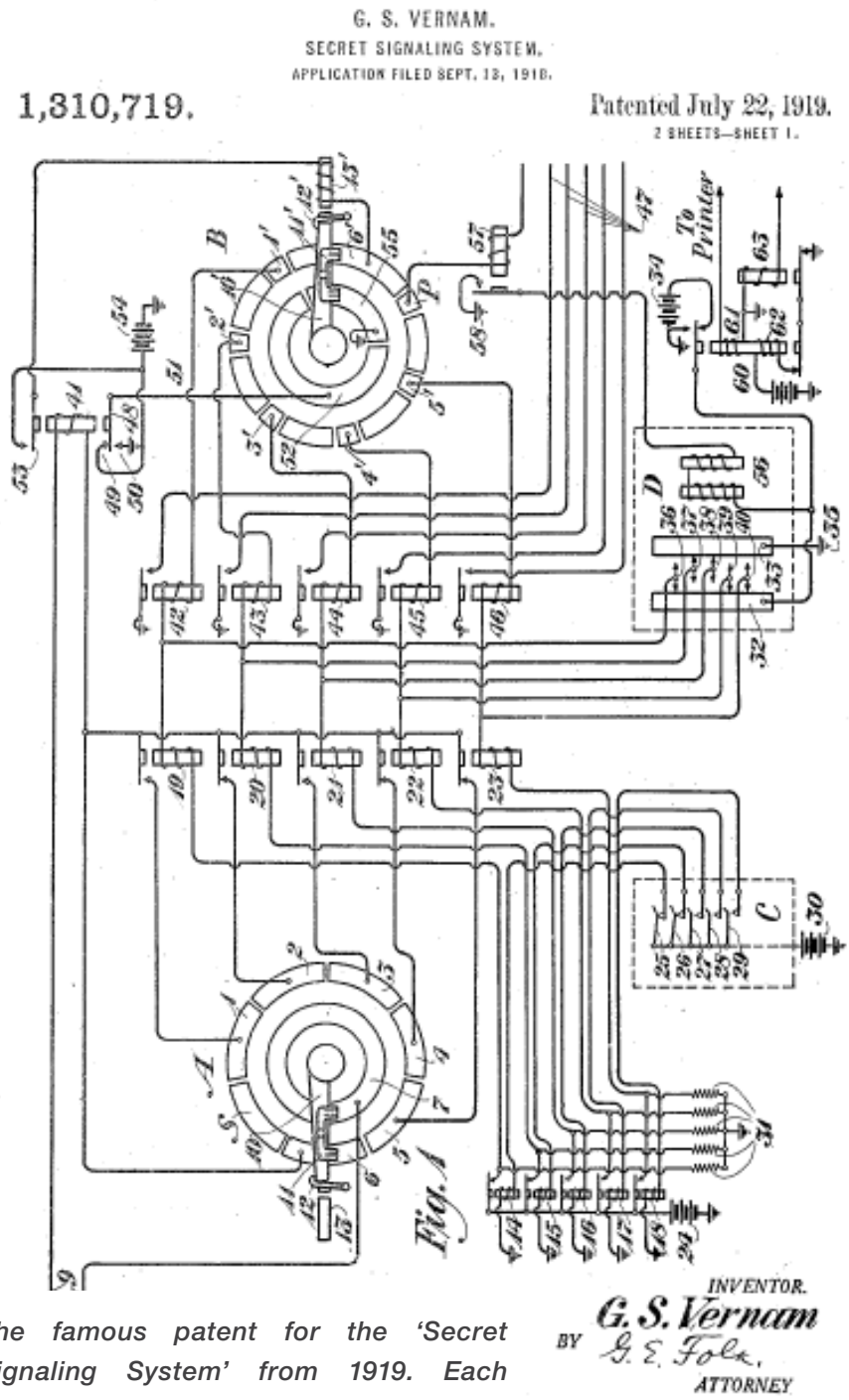
Since **P(ci | pi) = P(ci)** as shown above, these two terms cancel, leaving **P(pi | ci) = P(pi)**, which is the condition for perfect secrecy.

The history of One Time Pad encryption

The One Time Pad encryption method is nothing new. In 1917, Gilbert Vernam invented a cipher solution for a teletype machine. U.S. Army Captain Joseph Mauborgne realized that the character on the key tape could be completely random. Together, they introduced the first One Time Pad encryption system.

Gilbert Vernam

Joseph Mauborgne

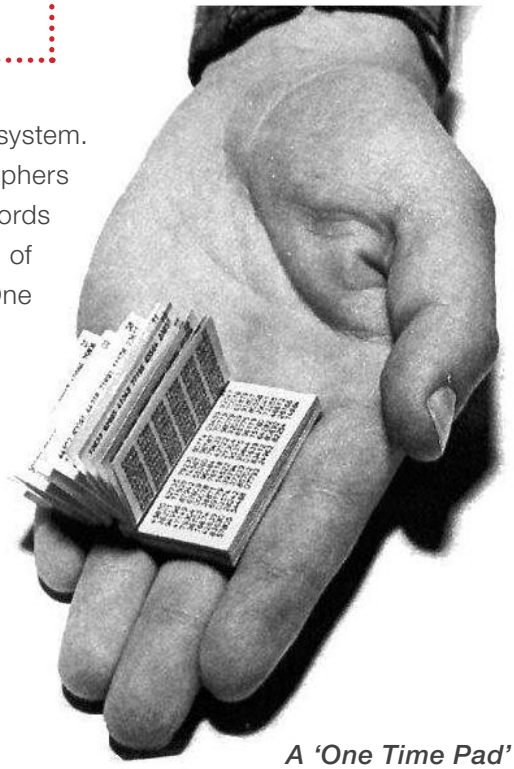


The famous patent for the 'Secret Signaling System' from 1919. Each character in a message was combined with a character on a paper tape key.

The 'Sigsaly' system

Since then, One Time Pad systems have been widely used by governments around the world. Outstanding examples of a One Time Pad system include the 'hot line' between the White House and the Kremlin and the famous Sigsaly speech encryption system.

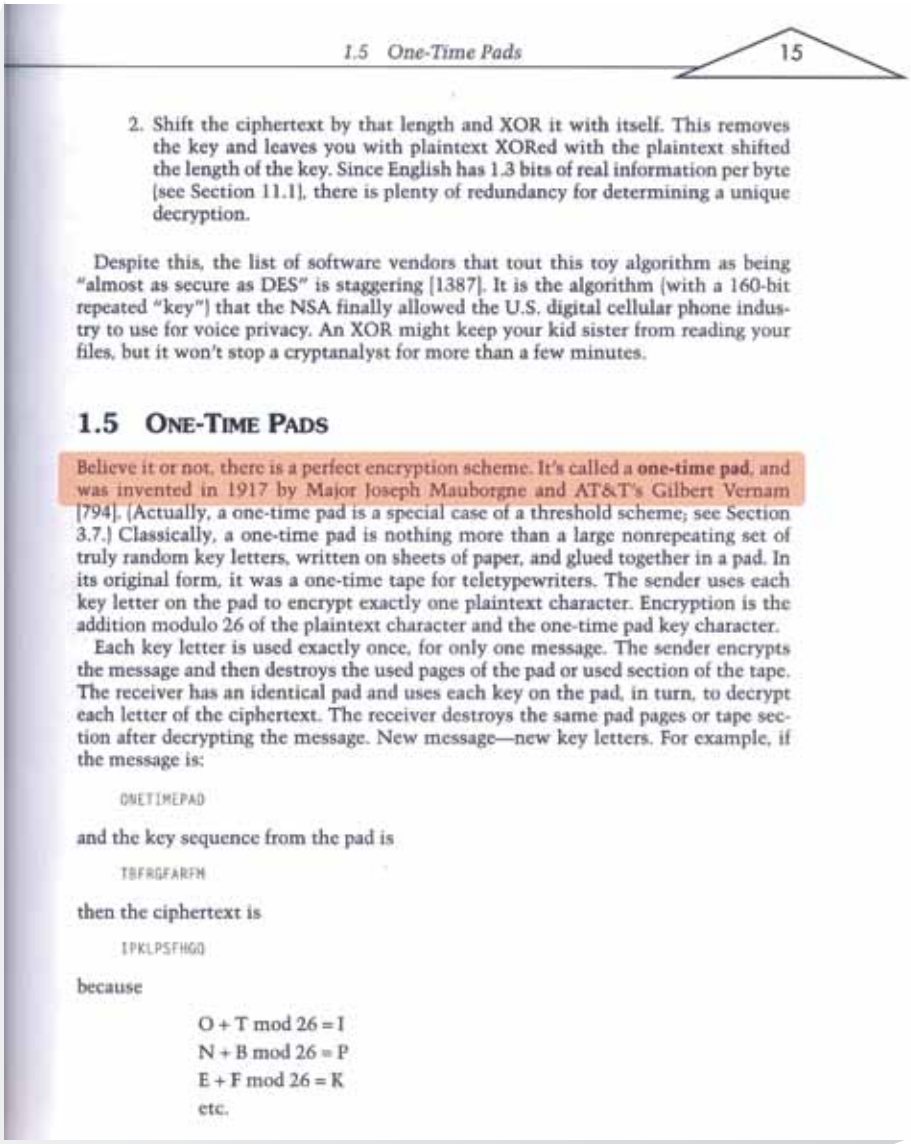
Another development was the paper pad system. Diplomats had long used codes and ciphers for confidentiality. For encryption, words and phrases were converted to groups of numbers and then encrypted by using a One Time Pad.



A 'One Time Pad'

To date, customers in more than 60 countries have benefited from the unconditional security that the One Time Pad encryption of our products provide.

Further readings



2. Shift the ciphertext by that length and XOR it with itself. This removes the key and leaves you with plaintext XORed with the plaintext shifted the length of the key. Since English has 1.3 bits of real information per byte (see Section 11.1), there is plenty of redundancy for determining a unique decryption.

Despite this, the list of software vendors that tout this toy algorithm as being "almost as secure as DES" is staggering [1387]. It is the algorithm (with a 160-bit repeated "key") that the NSA finally allowed the U.S. digital cellular phone industry to use for voice privacy. An XOR might keep your kid sister from reading your files, but it won't stop a cryptanalyst for more than a few minutes.

1.5 One-Time Pads

Believe it or not, there is a perfect encryption scheme. It's called a **one-time pad**, and was invented in 1917 by Major Joseph Mauborgne and AT&T's Gilbert Vernam [794]. (Actually, a one-time pad is a special case of a threshold scheme; see Section 3.7.) Classically, a one-time pad is nothing more than a large nonrepeating set of truly random key letters, written on sheets of paper, and glued together in a pad. In its original form, it was a one-time tape for teletypewriters. The sender uses each key letter on the pad to encrypt exactly one plaintext character. Encryption is the addition modulo 26 of the plaintext character and the one-time pad key character.

Each key letter is used exactly once, for only one message. The sender encrypts the message and then destroys the used pages of the pad or used section of the tape. The receiver has an identical pad and uses each key on the pad, in turn, to decrypt each letter of the ciphertext. The receiver destroys the same pad pages or tape section after decrypting the message. New message—new key letters. For example, if the message is:

ONETIMEPAD

and the key sequence from the pad is

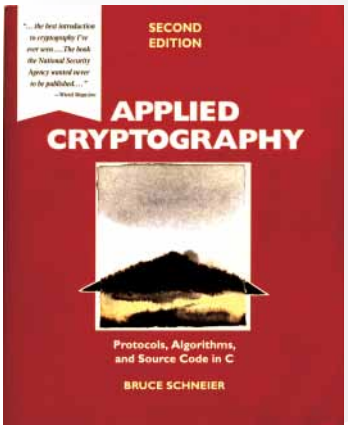
TBFRGFARFM

then the ciphertext is

IPKLPSFHGQ

because

O + T mod 26 = I
N + B mod 26 = P
E + F mod 26 = K
etc.



Schneier, Bruce:

Applied Cryptography: Protocols, Algorithms, and Source Code in C.
1996, John Wiley and Sons, Inc.
New York, Chichester, Brisbane, Toronto, Singapore

a nonlinear combining function on the outputs of several LFSRs (§6.3.1), using a nonlinear filtering function on the contents of a single LFSR (§6.3.2), and using the output of one (or more) LFSRs to control the clock of one (or more) other LFSRs (§6.3.3). Two concrete proposals for clock-controlled generators, the alternating step generator and the shrinking generator are presented in §6.3.3. §6.4 presents a stream cipher not based on LFSRs, namely SEAL. §6.5 concludes with references and further chapter notes.

6.1.1 Classification

Stream ciphers can be either symmetric-key or public-key. The focus of this chapter is symmetric-key stream ciphers; the Blum-Goldwasser probabilistic public-key encryption scheme (§8.7.2) is an example of a public-key stream cipher.

6.1 Note (*block vs. stream ciphers*) Block ciphers process plaintext in relatively large blocks (e.g., $n \geq 64$ bits). The same function is used to encrypt successive blocks; thus (pure) block ciphers are *memoryless*. In contrast, stream ciphers process plaintext in blocks as small as a single bit, and the encryption function may vary as plaintext is processed; thus stream ciphers are said to have memory. They are sometimes called *state ciphers* since encryption depends on not only the key and plaintext, but also on the current state. This distinction between block and stream ciphers is not definitive (see Remark 7.25); adding a small amount of memory to a block cipher (as in the CBC mode) results in a stream cipher with large blocks.

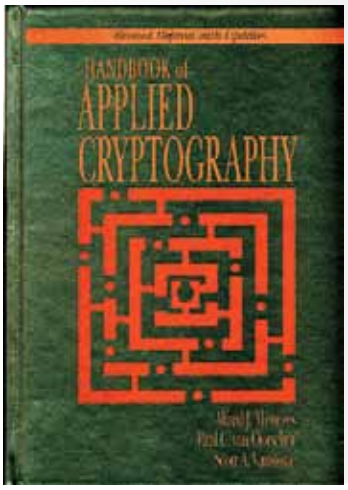
(i) The one-time pad

Recall (Definition 1.39) that a *Vernam cipher* over the binary alphabet is defined by

$$c_i = m_i \oplus k_i \quad \text{for } i = 1, 2, 3 \ldots ,$$

where $m_1, m_2, m_3, \ldots$ are the plaintext digits, $k_1, k_2, k_3, \ldots$ (the *keystream*) are the key digits, $c_1, c_2, c_3, \ldots$ are the ciphertext digits, and $\oplus$ is the XOR function (bitwise addition modulo 2). Decryption is defined by $m_i = c_i \oplus k_i$. If the keystream digits are generated independently and randomly, the Vernam cipher is called a *one-time pad*, and is unconditionally secure (§1.13.3(i)) against a ciphertext-only attack. More precisely, if M, C, and K are random variables respectively denoting the plaintext, ciphertext, and secret key, and if $H()$ denotes the entropy function (Definition 2.39), then $H(M|C) = H(M)$. Equivalently, $I(M;C) = 0$ (see Definition 2.45): the ciphertext contributes no information about the plaintext.

Shannon proved that a necessary condition for a symmetric-key encryption scheme to be unconditionally secure is that $H(K) \geq H(M)$. That is, the uncertainty of the secret key must be at least as great as the uncertainty of the plaintext. If the key has bitlength k, and the key bits are chosen randomly and independently, then $H(K) = k$, and Shannon's necessary condition for unconditional security becomes $k \geq H(M)$. The one-time pad is unconditionally secure regardless of the statistical distribution of the plaintext, and is optimal in the sense that its key is the smallest possible among all symmetric-key encryption schemes having this property.



Menezes, Alfred J., Paul C. van Oorschot, and Scott A. Vanstone:

Handbook of Applied Cryptography
1997, CRC Press
Boca Raton, New York, London, Tokyo

mils ELECTRONIC

mils electronic gesmbh & cokg | leopold-wedl-strasse 16 | 6068 mils | austria
t +43 5223 57710-0 | f +43 5223 57710-110 | info@mils.com | www.mils.com