

Classical Cryptography

Cryptography was once “the art of writing or solving codes.”

It has now become “the study of techniques for securing digital information, transactions, and distributed computation.”

We discuss first “private key ciphers,” aka “symmetric” or “one-key.”

encrypt, decrypt, key, plaintext, ciphertext.

Gen = key-generation algorithm. probabilistic, outputs k according to some distribution, usually uniform.

Enc = encryption algorithm: $c = E_k(m)$

Dec = decryption algorithm: $m = D_k(c)$

\mathcal{K} is set of all possible keys, usually a finite set.

\mathcal{M} is set of all possible messages, maybe all finite strings.

\mathcal{C} is set of all possible ciphertexts, completely determined by \mathcal{M} , \mathcal{K} and Enc.

Kerckhoffs' principle.

The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.

Security depends solely on the secrecy of the key, and not on Enc.

Easier to keep a short key secret than a long program.

If key is exposed, it is easier to replace it than Enc.

With many users, it is easier for all pairs to use just one Enc, and have different keys.

Today Kerckhoffs' principle is modified to require that the crypto algorithms be made public.

Public scrutiny is more likely find weaknesses.

It is better for flaws to be found by “ethical hackers” than the enemy.

If the algorithm is supposed to be secret, then reverse engineering can compromise it. Usually, the key is not subject to reverse engineering.

Public design allows for standards.

Attack scenarios:

Ciphertext-only.

Known-plaintext.

Chosen-plaintext.

Chosen-ciphertext.

Historic ciphers.

Caesar cipher = shift cipher — rotate the alphabet

Principle: Any secure encryption scheme must have a key space large enough so that an attacker cannot try all possible keys.

(Need at least 10^{18} or 2^{60} different keys.)

Mono-alphabetic cipher = simple substitution cipher: Key is an arbitrary permutation of the alphabet. (Have $26! \approx 2^{88}$ keys.)

No good. See Cryptoquip in *Exponent*.

To break, compute frequency of letters.

Use letter frequency to attack Caesar cipher.

Let p_i be the probability of the i -th letter in normal English. (So $p_4 = Pr[e] \approx 0.127$ is largest.)

Using tables of letter frequencies, one finds

$$\sum_{i=0}^{25} p_i^2 \approx 0.065.$$

If we let q_i be the probability of the i -th letter in the ciphertext from a Caesar cipher and compute

$$I_j = \sum_{i=0}^{25} p_i \cdot q_{(i+j \bmod 26)},$$

for each $0 \leq j \leq 25$, we will find $I_k \approx 0.065$ when k is the key and I_j for $j \neq k$ to be significantly different. Thus we can automate the attack.

Vigenère cipher = poly-alphabetic shift cipher:
Like Caesar, but use a key WORD to specify
a periodic sequence of alphabet rotations.

Example with key word BAND

M: RENA ISSA NCE

K: BAND BAND BAN

C: SEAD JSFD OCR

Easy to break if the length t of the key word is
known. It is just t Caesar ciphers intertwined.
Break each one with the automated computa-
tion of the previous slide.

How to find period length of key in Vigenère cipher.

Kasiski [1863] method: Look for repetitions in the ciphertext. They often occur at multiples of the period t . Look at the gcd of many of them.

Index of Coincidence [Friedman, 1920s] method (variation): For $s = 1, 2, 3, \dots$, define $J_s = \sum_{i=0}^{25} q_i^2$, where q_i is the frequency of the i -th letter in the subsequence $c_1, c_{1+s}, c_{1+2s}, \dots$. One finds $J_s \approx 0.065$ when $s = t$ and $J_s \approx 0.038 \approx 1/26$ when $s \neq t$.

The Vigenère cipher was invented by Giovan Battista Bellaso in 1553 and was known as “le chiffre indéchiffrable.” Even in 1917, *Scientific American* said it was “impossible of translation.” It was used by military forces from the 16th century to World War I. Experts could break it by the mid-nineteenth century.

We have used ciphertext-only attacks on the ciphers discussed so far. All these ciphers are trivial to break with a known-plaintext attack.

We have learned:

$|\mathcal{K}|$ must be large enough so that an attacker cannot search all of it.

But large $|\mathcal{K}|$ does not by itself imply security.

Designing secure ciphers is hard.

Principles of Modern Cryptography

1. Formulate a rigorous and precise definition of security.

Important for design, usage and study.

An encryption scheme is secure if no adversary can ...

find the secret key when given a ciphertext.

find the plaintext when given a ciphertext.

find any character of the plaintext from the ciphertext.

get any meaningful info about the plaintext from the ciphertext.

compute any function of the plaintext from the ciphertext.

A crypto scheme is secure for a given use if no adversary of specified power can perform a specified break.

2. State precisely all unproved assumptions on which security depends. Make the assumptions minimal.

Assumptions may be tested (validated), compared, and used in proofs, as in 3.

Of course, one could assume that a crypto construction itself is secure. This is not done because older (tested) and simpler assumptions are better.

3. Prove security, as in 1, perhaps using assumptions, as in 2.

Proofs are often by reduction:

Assuming X is true, Construction Y is secure according to the given definition.

Informal definition: An encryption scheme is *perfectly secret* if it is provably secure even against an adversary with unbounded computing power, with no assumptions.

Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme with $|\mathcal{M}| > 1$.

We write $k \leftarrow \text{Gen}$ to mean that k is a random key produced by Gen.

When Enc is deterministic, we write $c := \text{Enc}_k(m)$ to mean that c is the ciphertext obtained when the plaintext m is enciphered with the key k .

It is possible that Enc is probabilistic. In this case we write $c \leftarrow \text{Enc}_k(m)$ to mean that c is a ciphertext obtained when the plaintext m is enciphered with the key k .

Dec is always deterministic and we write $m := \text{Dec}_k(c)$ to mean that m is the plaintext obtained when the ciphertext c is deciphered with the key k .

Write $Pr[K = k]$ for the probability that the key output by Gen is k . (That is, K is a random variable denoting the key value.) Often, $Pr[K = k] = 1/|\mathcal{K}|$ (uniform distribution).

Likewise, let M and C be random variables denoting the plaintext and the ciphertext. Thus, $Pr[M = m]$ is the probability that the plaintext is m and $Pr[C = c]$ is the probability that the ciphertext is c .

We will assume that both $Pr[M = m]$ and $Pr[C = c]$ are positive.

Formal definition: An encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ over a message space \mathcal{M} is *perfectly secret* if for every probability distribution over \mathcal{M} , every plaintext $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ with $\Pr[C = c] > 0$ we have

$$\Pr[M = m | C = c] = \Pr[M = m].$$

This definition says that Π is perfectly secret if the probability distributions over plaintexts and ciphertexts are independent.

Lemma 2.2: An encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ over a message space \mathcal{M} is perfectly secret if and only if for every probability distribution over \mathcal{M} , every plaintext $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ with $\Pr[C = c] > 0$ we have

$$\Pr[C = c | M = m] = \Pr[C = c].$$

Proof. Use Bayes' Theorem with $A = (C = c)$ and $B = (M = m)$, and the definition

$$\Pr[M = m | C = c] = \Pr[M = m].$$

Definition: If A and B are events with $Pr[B] > 0$, then the *conditional probability* of A given B is

$$Pr[A|B] = \frac{Pr[A \wedge B]}{Pr[B]}.$$

Thus

$$Pr[A \wedge B] = Pr[B] \cdot Pr[A|B].$$

Bayes' Theorem: If $Pr[B] > 0$, then

$$Pr[A|B] = \frac{Pr[A] \cdot Pr[B|A]}{Pr[B]}.$$

Proof: Eliminate $Pr[A \wedge B]$ from the equations

$$Pr[A \wedge B] = Pr[B] \cdot Pr[A|B],$$

$$Pr[A \wedge B] = Pr[A] \cdot Pr[B|A].$$

Perfect Indistinguishability

Perfect secrecy is equivalent to having the probability distribution of the ciphertext for a particular plaintext m be independent of m .

Lemma 2.3: An encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ over a message space \mathcal{M} is perfectly secret if and only if for every probability distribution over \mathcal{M} , every $m_0, m_1 \in \mathcal{M}$, and every $c \in \mathcal{C}$ we have

$$\Pr[C = c | M = m_0] = \Pr[C = c | M = m_1].$$

Proof: Use Lemma 2.2.

$$\begin{aligned} \Pr[C = c | M = m_0] &= \Pr[C = c] = \\ &= \Pr[C = c | M = m_1]. \end{aligned}$$

Adversarial Indistinguishability

Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$. Let \mathcal{A} be an adversary.

$\text{PrivK}_{\mathcal{A}, \Pi}^{\text{eav}}$ is this experiment:

1. \mathcal{A} outputs $m_0, m_1 \in \mathcal{M}$.
2. An entity running the experiment uses Gen to generate a random key k and choose a random bit $b \leftarrow \{0, 1\}$. Then the entity computes $c \leftarrow \text{Enc}_k(m_b)$ and gives c to \mathcal{A} .
3. \mathcal{A} outputs a bit b' .
4. The output is 1 if $b' = b$, else 0. Write $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{eav}} = 1$ and say \mathcal{A} *succeeded* if the output is 1.

Proposition 2.5: The encryption scheme $(\text{Gen}, \text{Enc}, \text{Dec})$ over a message space \mathcal{M} is perfectly secret if and only if

$$\Pr \left[\text{PrivK}_{\mathcal{A}, \Pi}^{\text{eav}} = 1 \right] = \frac{1}{2}$$

for every adversary \mathcal{A} .

The One-Time Pad

Also called Vernam's cipher (1917).

Let $a \oplus b$ denote the bitwise XOR of the bit strings a and b . Recall $a \oplus a = 0$ and $a \oplus 0 = a$ for any a .

The one-time pad encryption scheme is defined as follows.

1. Fix an integer $\ell > 0$. Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1\}^\ell$, bit strings of length ℓ bits.
2. Gen chooses a bit string k of length ℓ with uniform distribution; that is, each string has probability $2^{-\ell}$.
3. $c := \text{Enc}_k(m) = m \oplus k$.
4. $m := \text{Dec}_k(c) = c \oplus k$.

Theorem 2.6: The one-time pad encryption scheme is perfectly secret.

Proof: Fix a probability distribution on \mathcal{M} . Fix $m \in \mathcal{M}$ and $c \in \mathcal{C}$. Note that

$$\begin{aligned} Pr[C = c|M = m] &= Pr[M \oplus K = c|M = m] = \\ &= Pr[m \oplus K = c] = \\ &= Pr[K = m \oplus c] = \frac{1}{2^\ell}. \end{aligned}$$

This holds for all distributions and all m , so for all m_0 and m_1

$$Pr[C = c|M = m_0] = \frac{1}{2^\ell} = Pr[C = c|M = m_1].$$

By Lemma 2.3, the encryption scheme is perfectly secret.

The one-time pad must not be used more than once because if $c = m \oplus k$ and $c' = m' \oplus k$, then

$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = m \oplus m'.$$

Theorem 2.7. Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be a perfectly secret encryption scheme with message space \mathcal{M} and key space \mathcal{K} . Then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: Assume $|\mathcal{K}| < |\mathcal{M}|$. Assume the uniform distribution on \mathcal{M} . Let $c \in \mathcal{C}$ occur. Let $\mathcal{M}(c)$ be the set of all possible decryptions m of c . Clearly $|\mathcal{M}(c)| \leq |\mathcal{K}| < |\mathcal{M}|$, so there is some $m' \notin \mathcal{M}(c)$. But then

$$\Pr[M = m' | C = c] = 0 \neq \Pr[M = m'],$$

so the scheme is not perfectly secret.

This theorem shows that the only way to get perfect secrecy is with keys as long as the plaintext. For most applications this is unacceptable.

Shannon's Theorem

Theorem 2.8. Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme over \mathcal{M} for which $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$. The scheme is perfectly secret if and only if

1. Gen chooses $k \in \mathcal{K}$ with uniform distribution, and
2. $\forall m \in \mathcal{M} \forall c \in \mathcal{C} \exists! k \in \mathcal{K} c := \text{Enc}_k(m)$.

This theorem shows that if an encryption scheme is perfectly secret for one probability distribution on \mathcal{M} , then it is perfectly secret for every probability distribution on \mathcal{M} ,

Shannon's theorem is easy to apply. It is easy to prove using it that the one-time pad is perfectly secret.