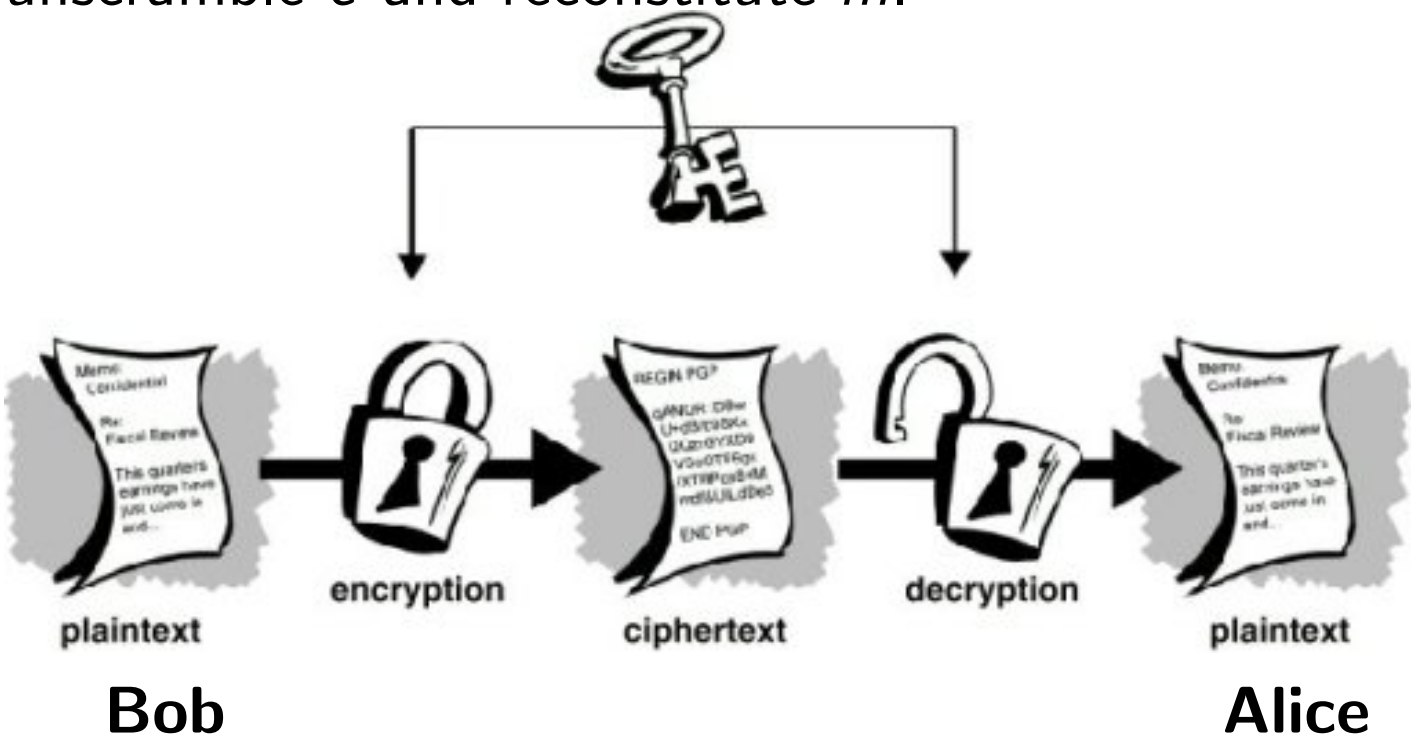
Criptografia — 2011/12

Symmetric and asymmetric ciphers

Different examples of ciphers have in common some features.

Bob wants to send a secret message to Alice. He uses a secret key k to scramble his plaintext message m and turn it into a ciphertext c. Alice, upon receiving c, uses the secret key k to unscramble c and reconstitute m.



If this procedure is to work properly, then both **Alice and Bob must possess copies of the secret key** k, and if the system is to provide security, then their adversary Eve must not know k, must not be able to guess k, and must not be able to recover m from c without knowing k.

Criptografia — 2011/12

Symmetric and asymmetric ciphers

We formulate the notion of a cryptosystem in abstract mathematical terms. There are many reasons why this is desirable. In particular, it allows us to highlight similarities and differences between different systems, while also providing a framework within which we can **rigorously analyze the security of a cryptosystem against various types of attacks**.

Returning to Bob and Alice, we observe that they must share knowledge of the secret key k. using the secret key, they can both encrypt and decrypt messages, so Bob and Alice have equal (symmetric) knowledge and abilities. For this reason, ciphers of this sort are known as symmetric ciphers.

Mathematically, a symmetric cipher uses a key k chosen from a space (i.e. a set) of possible keys $\mathcal{K}$ to encrypt a plaintext message m chosen from a space of possible messages $\mathcal{M}$, and the result of the encryption process is a ciphertext c belonging to a space of possible ciphertexts $\mathcal{C}$.

Thus encryption may be viewed as a function

$$e : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$$

whose domain $\mathcal{K} \times \mathcal{M}$ is the set of pairs (k, m) consisting of a key k and a plaintext m and whose range is the space of ciphertexts $\mathcal{C}$. Similarly, decryption is a function

$$d : \mathcal{K} \times \mathcal{C} \to \mathcal{M}.$$

Of course, **we want the decryption function to "undo" the results of the encryption function**. Mathematically, this is expressed by the formula

$$\mathbf{d}(\mathbf{k}, \mathbf{e}(\mathbf{k}, \mathbf{m})) = \mathbf{m} \qquad \text{for all } \mathbf{k} \in \mathcal{K} \text{ and all } \mathbf{m} \in \mathcal{M}.$$

It is sometimes convenient to write the dependence on k as a subscript. then for each key k, we get a pair of functions

$$e_k : \mathcal{M} \to \mathcal{C} \quad \text{and } d_k : \mathcal{C} \to \mathcal{M}$$

satisfying the decryption property

$$\mathbf{d_k(e_k(m)) = m} \qquad \text{for all } \mathbf{m} \in \mathcal{M}.$$

In particular, e_k must be a one-to-one function, since if $e_k(m) = e_k(m')$, then $m = d_k(e_k(m)) = d_k(e_k(m')) = m'$.

It is safest for Alice and Bob to assume that Eve knows the encryption method that is being employed. In mathematical terms, this means that Eve knows the functions e and d. What Eve does not know is the particular key k that Alice and Bob are using.

For example, if Alice and Bob use a simple substitution cipher, they should assume that Eve is aware of this fact.

This illustrates a basic premise of modern cryptography called Kerckhoff's principle, which says that **the security of a cryptosystem should depend only on the secrecy of the key, and not on the secrecy of the encryption algorithm itself**.

If $(\mathcal{K}, \mathcal{M}, \mathcal{C}, e, d)$ is to be a successful cipher, it must have the following properties:

1. For any key $k \in \mathcal{K}$ and plaintext $m \in \mathcal{M}$, it must be **easy to compute** the ciphertext $\mathbf{e_k(m)}$.
2. For any key $k \in \mathcal{K}$ and ciphertext $c \in \mathcal{C}$, it must be **easy to compute** the ciphertext $\mathbf{d_k(c)}$.
3. Given one or more ciphertexts $c_1, \cdots, c_n \in \mathcal{C}$ encrypted using the key $k \in \mathcal{K}$, it must be **very difficult to compute** any of the corresponding plaintexts $\mathbf{d_k(c_1)}, \cdots, \mathbf{d_k(c_n)}$ **without knowledge of k**.

   There is a fourth property that is desirable, although it is more difficult to achieve.

4. Given one or more pairs of plaintexts and their corresponding ciphertexts, $(m_1, c_1), \cdots, (m_m, c_n)$, it must be difficult to decrypt any ciphertext c that is not in the given list without knowing k. This is known as security against a chosen plaintext attack.

Notice that the simple substitution cipher does not have Property 4, since even a single plaintext/ciphertext pair (m, c) reveals most of the encryption table. Thus simple substitution ciphers are vulnerable to chosen plaintext attacks.

In our list of four desirable properties for a cryptosystem, we have left open the question of what exactly is meant by the words "easy" and "hard". For now, we informally take **"easy"** to mean computable in less than a second on a typical desktop computer and **"hard"** to mean that all of the computing power in the world would require several years (at least) to perform the computation.

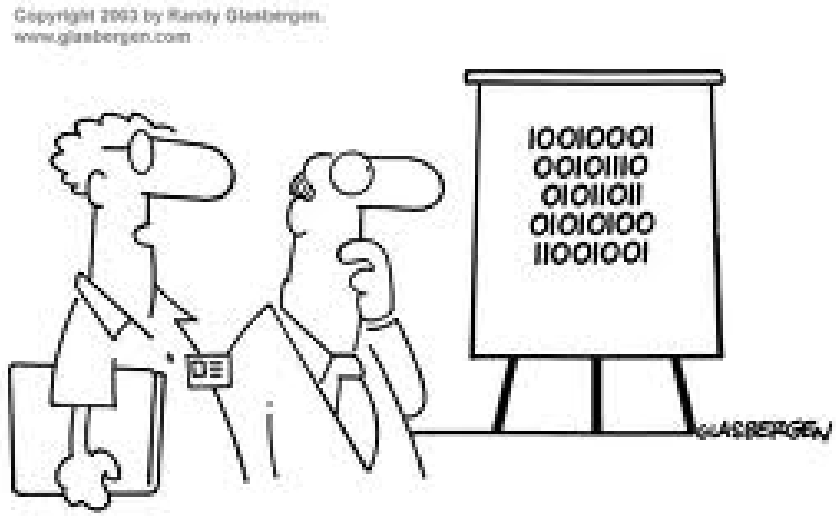
Criptografia — 2011/12

Symmetric and asymmetric ciphers: encoding schemes

It is convenient to view keys, plaintexts, and ciphertexts as numbers and to write those numbers in binary form.

Your computer may use **ASCII** code to store data, where each character is coded by eight bits.

The word "bit" is an abbreviation for binary digit, and ASCII is an acronym for American Standard Code for Information Interchange.



"We've devised a new security encryption code. Each digit is printed upside down."

Criptografia — 2011/12

Symmetric and asymmetric ciphers: encoding schemes

An encoding scheme is a method of converting one sort of data into another sort of data, for example, converting text into numbers.

The **distinction between an encoding scheme and an encryption scheme** is one of intent. An encoding scheme is assumed to be entirely public knowledge and used by everyone for the same purposes. An encryption scheme is designated to hide information from anyone who does not possess the secret key. Thus an encoding scheme, like an encryption scheme, consists of an encoding function and its inverse decoding function, but for an encoding scheme, both functions are public knowledge and should be fast and easy to compute.

Criptografia — 2011/12

Symmetric and asymmetric ciphers: encoding schemes

With the use of an encoding scheme, a plaintext or ciphertext may be viewed as a sequence of binary blocks, where each block consists of eight bits, i.e., of a sequence of eight ones and zeros. A block of eight bits is called a byte.

For human comprehension, a byte is often written as a decimal number between 0 and 255.

Computers often operate on more than one byte at a time. For example, a 64-bit processor operates on eight bytes at a time.

In using an enconding as described, it is convenient to view the elements of the plaintext space $\mathcal{M}$ as consisting of bit strings of a fixed length B, i.e., strings of exactly B ones and zeros. We call B the blocksize of the cipher.

A general plaintext message then consists of a list of message blocks chosen from $\mathcal{M}$, and the encryption function transforms the message blocks into a list of ciphertext blocks in $\mathcal{C}$, where each block is a sequence of B bits. If the plaintext ends with a block of fewer than B bits, we pad the end of the block with zeros.

Keep in mind that this encoding process, which converts the original plaintext message into a sequence of blocks of bits in $\mathcal{M}$, is public knowledge.

Encryption and decryption are done one block at a time, so it suffices to study the process for a single plaintext block, i.e., for a single $m \in \mathcal{M}$. This, of course, is why it is convenient to break a message up into blocks.

A message can be of arbitrary length, so it is nice to be able to focus the cryptographic process on a single piece of fixed length.

The plaintext block m is a string of B bits, which for concreteness we identify with the corresponding number in binary form. In other words, we identify $\mathcal{M}$ with the set of integers m satisfying $0 \leq m < 2^B$ via

$$\underbrace{m_{B-1} \cdots m_2\, m_1\, m_0}_{\text{list of } B \text{ bits of } m} \leftrightarrow \underbrace{m_{B-1} \cdot 2^{B-1} + \cdots + m_2 \cdot 2^2 + m_1 \cdot 2 + m_0}_{\text{integer between 0 and } 2^B - 1}.$$

Here $m_0, \cdots, m_{B-1}$ are each 0 or 1.

Similarly, we identify the key space $\mathcal{K}$ and the ciphertext space $\mathcal{C}$ with sets of integers corresponding to bit strings of a certain blocksize. For notational convenience, we denote the blocksizes for keys, plaintexts, and ciphertexts by B_k, B_m and B_c. They need not be the same. Thus we have identified $\mathcal{K}$, $\mathcal{M}$ and $\mathcal{C}$ with sets of positive integers

$$\mathcal{K} = \{k \in \mathbb{Z} : 0 \leq k < 2^{B_k}\},$$

$$\mathcal{M} = \{m \in \mathbb{Z} : 0 \leq m < 2^{B_m}\},$$

$$\mathcal{C} = \{c \in \mathbb{Z} : 0 \leq c < 2^{B_c}\}.$$

An important question immediately arises: how large should Alice and Bob make the set $\mathcal{K}$, or equivalently, **how large should they choose the key blocksize B_k?**

If B_k is too small, then Eve can check every number from 0 to $2^{B_k} - 1$ until she finds Alice and Bob's key. More precisely, since Eve is assumed to know the descryption algorithm d (Kerckhoff's principle), she takes each $k \in \mathcal{K}$ and uses it to compute $d_k(c)$.

Assuming that Eve is able to distinguish between valid and invalid plaintexts, eventually she will recover the message.

This attack is known as an exhaustive search attack (also sometimes referred as a brute-force attack), since Eve exhaustively searches through the key space.

With current technology, an exhaustive search is considered to be infeasible if the space has at least 2^{80} elements. Thus Bob and Alice should definitely choose $B_k \geq 80$ [state of the art in 2008].

For many cryptosystems there are refinements on the exhaustive search attack that effectively replace the size of the space with its square root. These methods are based on the principle that it is easier to find matching objects (collisions) in a set than it is to find a particular object in the set. We describe some of these **meet-in-the-middle** or **collision attacks** latter. If meet-in-the-middle attacks are available, then Alice and Bob should choose $B_k \geq 160$.

Criptografia — 2011/12

Symmetric and asymmetric ciphers: examples of symmetric ciphers

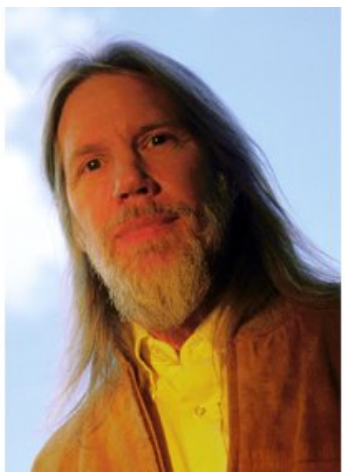
If Alice and Bob want to exchange messages using a symmetric cipher, they must first mutually agree on a secret key k. This is fine if they have the opportunity to meet in secret or if they are able to communicate once over a secure channel. But what if they do not have this opportunity and if every communication between them is monitored by their adversary Eve?

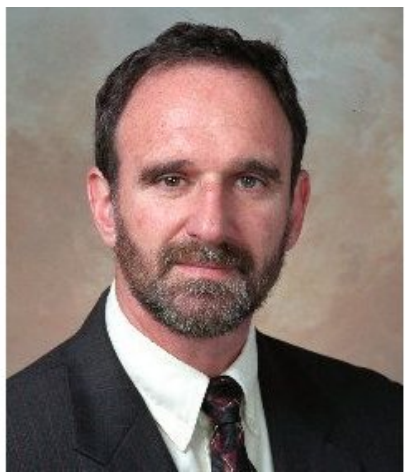
Is it possible for Alice and Bob to exchange a secret key under these conditions?

Criptografia — 2011/12

Symmetric and asymmetric ciphers: asymmetric ciphers make a first appearance

Most people's first reaction is that it is not possible, since Eve sees every piece of information that Alice and Bob exchange.





It was the brilliant insight of **Diffie and Hellman** (the history is actually somewhat more complicated than this) that under certain hypotheses, it is possible.

The search for efficient (and provable) solutions to this problem, which is called public key (or asymmetric) cryptography, forms one of the most interesting parts of mathematical cryptography.

Criptografia — 2011/12

Symmetric and asymmetric ciphers: asymmetric ciphers make a first appearance

We start by describing a nonmathematical way to visualize public key cryptography.

Alice buys a safe with narrow slot in the top and puts her safe in a public location. Everyone in the world is allowed to examine the safe and see that it is securely made. Bob writes his message to Alice on a piece of paper and slips it through the slot in the top of the safe. Now only a person with the key to the safe, which presumably means only Alice, can retrieve and read Bob's message.

In this scenario, **Alice's public key is the safe**, the encryption algorithm is the process of putting the message in the slot, and **the decryption algorithm is the process of opening the safe with the key**.

Note that this setup is not far-fetched; it is used in the real world. For example, the night deposit slot at a bank has this form, although in practice the "slot" must be well protected to prevent someone from inserting a long thin pair of tongs and extracting other people's deposits.

A useful feature of our "safe-with-a-slot" cryptosystem, which it shareswith actual public key cryptosystems, is that Alice needs to put only one safe in a public location, and then everyone in the world can use it repeatedly to send encrypted messages to Alice. **There is no need for Alice to provide a separate safe for each of her correspondents.**

And there is also **no need for Alice to open the safe and remove Bob's message before someone else uses it to send Alice a message**.

We are now ready to give a mathematical formulation of an asymmetric cipher.

As usual, there are spaces of keys $\mathcal{K}$, plaintexts $\mathcal{M}$, and ciphertexts $\mathcal{C}$. However $k \in \mathcal{K}$ is a pair of keys,

$$\mathbf{k} = (\mathbf{k_{priv}}, \mathbf{k_{pub}})$$

called the private key and the public key, respectively.

For each public key k_{pub} there is a corresponding encryption function

$$\mathbf{e_{k_{pub}}} : \mathcal{M} \to \mathcal{C},$$

and for each private key k_{priv} there is a corresponding dencryption function

$$\mathbf{d_{k_{priv}}} : \mathcal{C} \to \mathcal{M}.$$

These have the property that if the pair (k_{priv}, k_{pub}) is in the key space $\mathcal{K}$, then

$$d_{k_{priv}}(e_{k_{pub}}(m)) = m \qquad \text{for all } m \in \mathcal{M}.$$

If an asymmetric cipher is to be secure, it must be **difficult** for Eve **to compute the decryption function $d_{k_{priv}}$, even if she knows the public key k_{pub}.**

Notice that under this assumption, Alice can send k_{pub} to Bob using an insecure communication channel, and Bob can send back the ciphertext $e_{k_{pub}}(m)$, without worrying that Eve will be able to decrypt the message.

To easily decrypt, it is necessary to know the private key k_{priv}, and presumably Alice is the only person with that information.

The private key is sometimes called Alice's trapdoor information, because it provides a trapdoor (i.e. a shortcut) for computing the inverse function of $e_{k_{pub}}$.

The fact that the encryption and the decription keys k_{pub} and k_{priv} are different makes the cipher asymmetric