Theoretical Cryptography

February 24, 2011

Basic notions

A plaintext message is the original message in a ‘readable’(*) form.

(*) We always consider a plaintext message as a sequence of numbers (or a single number) instead of a sequence of characters.

An encoding scheme converts letters into numbers (usually in binary form). We assume that this scheme is well-known (public domain) and does not involve a real encryption.

EXAMPLE: In the previous simple encryption algorithms the encoding scheme is A–Z $\longrightarrow$ 0–25.

The standard encoding scheme in practice is ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange).

Symbols, control characters, letters	$\longrightarrow$	numbers 0–255 (or 0–127)
A–Z	$\mapsto$	65–90
a–z	$\mapsto$	97–122

UTF-8, a fully compatible extension of ASCII broadly employed in Linux machines, is more popular in web pages than the original ASCII since 2007.

The Python command to calculate the ASCII code of a character is `ord`. Its inverse is `chr`.

```
for c in 'How are you?':
        print c, '=', ord(c)
```

```
H = 72          H = 01001000
o = 111         o = 01101111
w = 119         w = 01110111
  = 32            = 00100000
a = 97          a = 01100001
r = 114         r = 01110010
e = 101         e = 01100101
  = 32            = 00100000
y = 121         y = 01111001
o = 111         o = 01101111
u = 117         u = 01110101
? = 63          ? = 00111111
```

Use `print c, '=', bin(ord(c))[2:].rjust(8,'0')` to get the 8-digit binary representation of the second column.

Notation

To encrypt (to hide) a plaintext message we use a key (a secret number) to produce a ciphertext (the encrypted message).

We denote

$$\begin{aligned}\mathcal{K} &= \text{Space of all possible keys}\\ \mathcal{M} &= \text{Space of all possible plaintext messages}\\ \mathcal{C} &= \text{Space of all possible ciphertext messages}\end{aligned}$$

In many instances $\mathcal{M} = \mathcal{C}$.

With this notation a cryptosystem is a pair of maps

$$\boxed{\begin{array}{l} e : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C} \text{ the encryption map} \\ d : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M} \text{ the decryption map} \end{array}}$$

[For the sake of simplicity use $e_k(m)$ and $d_k(c)$ instead of $e(k, m)$ and $d(k, c)$]
Such that for every $k_1 \in \mathcal{K}$, the encryption key, there is a $k_2 \in \mathcal{K}$, the decryption key, satisfying

$$\boxed{d_{k_2}\big(e_{k_1}(m)\big) = m \qquad \forall m \in \mathcal{M}}$$

EXAMPLE: In the affine ciphers we can consider that the key is $k = (a, b)$ and $e_k(x) = ax + b$ and $d_k(x) = a^{-1}x - a^{-1}b$.

Private and Public key cryptosystems

EXAMPLE: In the previous simple encryption algorithms the decryption key k_2 is easily derived from the encryption key k_1. We can in fact assume that $k_1 = k_2$ and implement the easy function on d.

We say that $k_1 = k_2$ (or k_2 easily derived from k_1) corresponds to symmetric cipher or private key cryptosystems.

The notation is self-explanatory:

- symmetric: encrypter and decrypter have equal knowledge.
- private: the encryption key cannot be published

The opposite is called asymmetric cipher or public key cryptosystems.

In this case k_2 cannot be easily computed from k_1

You know how to encrypt **but** You know how to decrypt without extra information

Slipping a note under the closed door of an office is in some sense an example of this phenomenon. Anyone can hide the information in this way but only the owner of the office has the key to recover the information.

At first sight it seems impossible to dessign a computer based public key cryptosystem (with a channel, www, monitored by attackers). This course will show that this prejudge is wrong.

Requirements (Our wish list for a cryptosystem)

1. $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$ the function $e_k(m)$ must be easy to compute.
2. $\forall k \in \mathcal{K}$, $\forall c \in \mathcal{C}$ the function $d_k(c)$ must be easy to compute.
3. Given $c \in \mathcal{C}$ it must be very difficult to compute $d_k(c)$ without knowledge of k.

---

4. The chosen plaintext attack must be unfeasible
   i.e. given a (small) set of decrypted messages $(c_1, d_k(c_1))$, $(c_2, d_k(c_2)), \ldots$ $(c_n, d_k(c_n))$ it must be difficult to decrypt any other message.

In practice we also assume one of the *Kerckhoff's principles*: The security lies on the key. The algorithm itself "must be able to fall into the hands of the enemy'. This is a polite way of saying that attackers are not stupid and they know every book and result on cryptography.

With respect tp the last requirement note that using the encoding scheme A-Z $\longleftrightarrow$ 0-25 and an affine chiper, if we know that TRY is encrypted as EQN or equivalently that EQN is decrypted as TRY, then we know

$$f: \begin{array}{rcr} \mathtt{T} = 19 & \longrightarrow & 4 = \mathtt{E} \\ \mathtt{R} = 17 & \longrightarrow & 16 = \mathtt{Q} \\ \mathtt{Y} = 24 & \longrightarrow & 13 = \mathtt{N} \end{array} \quad \Rightarrow \quad \begin{cases} a \cdot 19 + b = 4 \\ a \cdot 17 + b = 16 \end{cases}$$

Solving the system in $\mathbb{Z}/26\mathbb{Z}$

$$\left.\begin{array}{r} 19a + b = 4 \\ 17a + b = 16 \end{array}\right\} \Rightarrow 2a = -12 \equiv 14\,(26) \Rightarrow a \equiv 7\,(26) \Rightarrow b \equiv 1\,(26)$$

Then the secret key is $a = 7$, $b = 1$ and we can decrypt any other message.