“TEST - Multiply-by-two”

Example of source-code package
to explain the delivery procedure

X 2

Package’s LOGO

Source-code package overview

What the code do, which language is used

Type of processing, references to public data on Internet, language (C, C++, Matlab, Perl, ..), processor’s assembly..

What it does not

Remove any type of confusion before next slides which give the detailed APIs

Which compilation tool is used with which revision and operating system

List of the Matlab libraries used, C language syntax version, where can the compilers be downloaded, any specific information related to the OS used.

CPU and RAM consumption with or without floating point accelerator

Performance data for MHz, RAM, wait-states (map-file is part of the package when applicable), which CPU revision is assumed with which instruction-set.

Source-code package details

List of the APIs

APIs with detailed parameters. A “main” function is provided as example.

Data example

Explanations for the provided data given as activation examples (file formats, data scaling ..)

Exceptions

For example : what happens when the input files have not the right format.

Computation results

File formats of the results, graphical view.

Code extracts and other details

APIs code extracts

Screen-shots of the program usage.

List of the files in the package

Screen-shots of the file listings.

Deciphering process

How to decipher the source-code and file listings

“TEST - Multiply-by-two”

Deciphering tools

X 2

Package’s LOGO

Source-code delivery procedure

Each software package contains :

1. An overview in PDF format, with APIs, performances, …
2. A “main” calling the subroutine
3. Test-patterns and corresponding results
4. The source codes

{1,2,3} are free of access, {4} is encrypted with a picture used as a key.

Once your payment is accepted we give you the key to decipher it.

Encryption process (done offline, the key is never on the site)

The source-code folder is packed and compressed :

tar cvfz src.tar.gz source_code_folder

The compressed file is encrypted using a picture used as key :

perl xor_img.pl src.tar.gz key.jpg src.otp

The result is translated to a text file format :

base64 src.otp > file_to_download.txt

Deciphering process

The file found on our site is translated to binary format :

base64 -d file_downloaded.txt > src.otp

Deciphering made from the picture-key you received by mail from us :

perl xor_img.pl src.tar.gz key.jpg src.otp

The result source-code folder is unpacked and uncompressed :

tar xvfz src.tar.gz

ls source_code_folder*

Example

```
tar    cvfz       src.tar.gz P6_TEST_PROT
P6_TEST_PROT/
P6_TEST_PROT/TEST.m
P6_TEST_PROT/TEST_TOP.m

perl   xor_img.pl src.tar.gz KEY0.jpg src.otp
base64 src.otp >  src.txt

cat src.txt
4FP34H/kvBBJRe2VTG7beAzOfTZvJMAIgHRZNn36UfbUnxlu22sqD8BVX2K1Z0420v/+VnDVI8y8
WdWszPOVbJF2/c0zM8aA85xj/2768mfw1nYRjBojBq7f0FI/j5AmCDT4wL0zyfQ4RfQsgbEsgO8o
+CxxKTN3+Mo+AezFyyLHzih3IuJauUBE9GK0Oqv8eS5qt/l7p0Ph3cN0pfKG2m76F+Zx/ioNMy2C
oSr4CO54AmM3+GHiTmT93N1rglrf9m27AqOW9hbRHz5T+GM9joAQro9eNEXAHbrFFsE68yQAp39d
8qovVB67LPG7tlBoUfPEL410taQ/v6DWLEcOcMQdSCqjRk0ItLptl7qmRqqlnoHkP3skEmhqjw7Q
Viq7QlU5xrKjom2kB2X0ukJXN7w8tFNMMMVKaQdSSOBt5wA1S3jKeodzNQpMWRuqlMZxZZJtN8hZ
oKFf1TiiYEoRV2REhTytI1jHQXFn57YM+m0QMdZaujo2nNRgk3+yK3KWbmi1LVYdgtFadQKT+sSe
9cbx83tOqueTamY5MJBK6Zc3VJE1UIxNZvuRsYa1zWxru5cR+xsw7se1H0K1Zc+xmrq86gYetqTV
6Mgc8uvPsZBIGNEw2yi8Z3Rv0zPebEaFf/wRLVkfHslYGXTHykoogKoM6hTfW4SAqwtUQQdXTAha
VxVFUhZIXRtzYCR2Y5lHk36DbBCHeFF6gg==

base64 -d src.txt > src.otp
perl   xor_img.pl src.otp KEY0.jpg src.tar.gz
tar    xvfz       src.tar.gz
P6_TEST_PROT/
P6_TEST_PROT/TEST.m
P6_TEST_PROT/TEST_TOP.m
```

Source-code folder “./P6_TEST_PROT” packed and compressed to file “src.tar.gz”.

Encryption using the key-file “KEY0.jpg” and conversion to text format.

Dump of the text.

Conversion to binary, deciphering, unpacking and recovery of the original source-code folder.

Screen-shot

```
$ ls -l
total 1088
-rwxrwx---+ 1 FirmwareDevelopments None  69777 26 mars  20:36 KEY0.jpg
-rwxrwx---+ 1 FirmwareDevelopments None   1472 26 mars  20:01 list_test.png
-rwxrwx---+ 1 FirmwareDevelopments None    538 26 mars  21:19 src.otp
-rwxrwx---+ 1 FirmwareDevelopments None    545 16 mai   12:20 src.tar.gz
-rwxrwx---+ 1 FirmwareDevelopments None 826413 26 mars  21:17 src.txt
drwxrwx---+ 1 FirmwareDevelopments None      0 16 mai   11:57 src_folder
-rwxrwx---+ 1 FirmwareDevelopments None    545 16 mai   12:21 src_test.otp
-rwxrwx---+ 1 FirmwareDevelopments None      0 16 mai   12:17 src_test.txt
-rwxrwx---+ 1 FirmwareDevelopments None 121814 26 mars  21:13 TEST.pdf
-rwxrwx---+ 1 FirmwareDevelopments None  46911 26 mars  21:13 TEST.pptx
-rwxrwx---+ 1 FirmwareDevelopments None  28672 26 mars  20:30 XOR.exe
-rwxrwx---+ 1 FirmwareDevelopments None    477 26 mars  14:05 XOR_IMG.pl

FirmwareDevelopments@syndicap-PC /cygdrive/c/tmp/P6_TEST
$ tar cvfz src.tar.gz src_folder
src_folder/
src_folder/TEST.m
src_folder/TEST_TOP.m

FirmwareDevelopments@syndicap-PC /cygdrive/c/tmp/P6_TEST
$ perl XOR_IMG.pl src.tar.gz key0.jpg src_test.otp

FirmwareDevelopments@syndicap-PC /cygdrive/c/tmp/P6_TEST
$ base64 src_test.otp > src_test.txt

FirmwareDevelopments@syndicap-PC /cygdrive/c/tmp/P6_TEST
$ perl XOR_IMG.pl src_test.otp key0.jpg src.tar.gz

FirmwareDevelopments@syndicap-PC /cygdrive/c/tmp/P6_TEST
$ tar xvfz src.tar.gz
src_folder/
src_folder/TEST.m
src_folder/TEST_TOP.m

FirmwareDevelopments@syndicap-PC /cygdrive/c/tmp/P6_TEST
```

Cipher programs in Perl and C

```
#!/usr/bin/perl
open INPUT, "<$ARGV[0]" ;   binmode INPUT;
open INPUTKEY, "<$ARGV[1]"; binmode INPUTKEY;
open OUTPUT, ">$ARGV[2]";   binmode OUTPUT;
$i = 0;
while (($n = read INPUT, $byte, 1) != 0)
{
    read INPUTKEY, $key, 1;
    $xored = $byte ^ $key;
            print OUTPUT $xored;
    $i = $i+1;
    if ($i > 8191)
    {
        close INPUTKEY; open INPUTKEY, "<$ARGV[1]"; binmode INPUTKEY;
        $i = 0;
    }
}
close INPUT; close OUTPUT; close INPUTKEY;
```

```
#include <stdio.h>
void main (int argc, char *argv[])
{
  FILE *out_file, *inp_file, *key_file;
  int i, n;
  char ci, ck, co;

  if ((0 == (inp_file = fopen (argv[1], "rb"))) ||
      (0 == (key_file = fopen (argv[2], "rb"))) ||
      (0 == (out_file = fopen (argv[3], "wb")))
     )  { printf("fopen error \n");}
  else
  {   i = 0;
      n = fread (&ci, 1, 1, inp_file);
      while (n > 0)
      {
        fread  (&ck, 1, 1, key_file);
        co = ci ^ ck;
        fwrite (&co, 1, 1, out_file);
        i++;
        if (i > 8191)
        {
          fclose (key_file);
          key_file = fopen (argv[2], "rb");
          i = 0;
        }
        n = fread (&ci, 1, 1, inp_file);
      }
      fclose (out_file);
      fclose (key_file);
      fclose (inp_file);
  }
}
```