Seguridad en Sistemas de Información

Curso en Zacatenco y Tamaulipas [Q2 2013]

Luis J. Dominguez Perez

Cinvestav, Junio 17 a Julio 10 de 2013

Contenido, sección 1

Bibliotecas de funciones

NaCl: Networking and Cryptography library

- NaCl (”salt”) es una biblioteca de alta velocidad de software para comunicación de red, cifrado, descifrado, firmas, etc.
- Provee las operaciones básicas para la construcción de herramientas criptográficas de alto nivel
- Ayuda a mejorar la seguridad al proveer usabilidad, y al mejorar la velocidad de las aplicaciones

- Selecciona las primitivas criptográficas adecuadas, por lo que el usuario no hace decisiones erróneas.

Instalación

- Sitio web: http://nacl.cr.yp.to/install.html

```
wget http://hyperelliptic.org/nacl/nacl-20110221.tar.bz2
bunzip2 < nacl-20110221.tar.bz2 | tar -xf -
cd nacl-20110221
./do
```

La compilación tarda un poco

Ejemplo - Firma de documentos

```
#include "crypto_sign.h"

std::string pk;
std::string sk;
std::string m = "This is a test message";
std::string m2;
std::string sm;

pk = crypto_sign_keypair(&sk);

std::cout << m << std::endl;
sm = crypto_sign(m, sk);
std::cout << sm << std::endl;
m2 = crypto_sign_open(sm, pk);
std::cout << m2 << std::endl;
```

MIRACL - Multiprecision Integer and Rational Arithmetic C/C++ Library.

- MIRACL es una biblioteca altamente eficiente y portable de aritmética racional y de interos multiprecisión en C/C++.
- Su área de aplicación es la implementación de sistemas de Criptografía de Llave Pública y protocolos.
- Provee código fuente, y recientemente se volvió GPL

- Es ampliamente utilizada en proyectos de seguridad

Instalación

- Sitio web: http://www.certivox.com

```
unzip -j -aa -L miracl.zip
bash linux64

(bash linux64_cpp)
```

- Utiliza gcc o g++ según sea el caso

Contenido de la configuración

- Definiciones de la biblioteca

```
/*
 *   MIRACL compiler/hardware definitions - mirdef.h
 *   Copyright (c) 1988-2011 Shamus Software Ltd.
 *   For C++ build of library
 */

#define MR_LITTLE_ENDIAN
#define MIRACL 64
#define mr_utype long
#define mr_unsign64 unsigned long
#define MR_IBITS 32
#define MR_LBITS 64
#define mr_unsign32 unsigned int
#define MR_FLASH 52
#define MAXBASE ((mr_small)1<<(MIRACL-1))
#define MR_BITSINCHAR 8
#define MR_CPP
```

- Dentro de linux64_hpp se pueden escoger qué programas incluir en la biblioteca para hacerla más ligera

Ejemplo - RSA

```
/* simplest possible secure rsa implementation */
/* Uses IEEE-1363 standard methods  */
/* See IEEE-1363 section 11.2 IFES, page 57 */
/* cl /O2 rsa.c p1363.c miracl.lib    */

#include <stdio.h>
#include "p1363.h"
int main()
{
int i,bytes,res;
octet m,m1,c,e,raw;
if_public_key pub;
if_private_key priv;
csprng RNG;
/* some true randomness needed here */
OCTET_INIT(&raw,100); /* should be filled with 100 random bytes */

CREATE_CSPRNG(&RNG,&raw); /* initialise strong RNG */

bytes=IF_KEY_PAIR(NULL,&RNG,1024,65537,&priv,&pub);
/* generate random public & private key */
OCTET_INIT(&m,bytes); /* allocate space for plaintext and ciphertext */
OCTET_INIT(&m1,bytes);
OCTET_INIT(&c,bytes); OCTET_INIT(&e,bytes);
```

Ejemplo - RSA

```
OCTET_JOIN_STRING((char *)"Hello World\n",&m);

EME1_ENCODE(&m,&RNG,1023,NULL,SHA256,&e); /* OAEP encode message m to e  */
/* must be less than 1024 bits */

IFEP_RSA(NULL,&pub,&e,&c); /* encrypt encoded message */
IFDP_RSA(NULL,&priv,&c,&m1); /* ... and then decrypt it */

EME1_DECODE(&m1,1023,NULL,SHA256,&m1); /* decode it */

OCTET_PRINT_STRING(&m1); /* print out decrypted/decoded message */

OCTET_KILL(&m); OCTET_KILL(&m1); /* clean up afterwards */
OCTET_KILL(&c); OCTET_KILL(&raw); OCTET_KILL(&e);

IF_PUBLIC_KEY_KILL(&pub); /* kill the keys */
IF_PRIVATE_KEY_KILL(&priv);
}
```

PyCrypto: Python Cryptography Toolkit

- Es una colección de funciones hash seguras, y varios algoritmos de cifrado
- Viene incluída en SAGE
- Se utiliza para escribir herramientas de administración seguras, escribir protocolos y servicios de red
- Provee un espacio de trabajo para experimentar con algoritmos criptográficos, y protocolos en general

- Funciona para al menos Python 2.1 a 3.3

Instalación

- El sitio web es: https://www.dlitz.net/software/pycrypto/

- Está lista para Distutils, por lo que se instala así:
  - python setup.py build
  - python setup.py

Sin embargo, puede ya estar instalada en su computadora

Ejemplo - Cifrado AES

```
from Crypto.Cipher import AES
obj = AES.new('This is a key123', AES.MODE_CBC, 'This is an IV456')
message = "The answer is no"
ciphertext = obj.encrypt(message)
ciphertext
obj2 = AES.new('This is a key123', AES.MODE_CBC, 'This is an IV456')
obj2.decrypt(ciphertext)
```