# Introducción a la Criptografía

GDG-DevFest 2018

Gimenez, Christian

15 sep 2018

## Contents

# 1 Datos del Taller

**Nombre del Taller** Introducción a la criptografía

**Reseña** Mostrar los principios básicos de la criptografía desde un punto de vista práctico y teórico. Se pretende que el participante comprenda el uso, las aplicaciones, la importancia y las herramientas que actualmente se disponen para realizar las tareas básicas de cifrado. También, se describirán algunas interfaces de programación en algunos lenguajes para llevar a cabo estas tareas. Finalmente, se mostrarán ejemplos de aplicación para que se pueda observar la importancia en cuanto a la seguridad y privacidad de la información.

**Requisitos para el Taller** GNU/Linux instalado o el paquete GPG2 desde `https://www.gnupg.org` para el sistema operativo que posea.

**Short Bio** Docente en la Universidad Nacional del Comahue y recibido como Licenciado en Ciencias de la Computación de dicha institución. Integro el Grupo de Investigación de Lenguajes e Inteligencia Artificial (GILIA) el cual estoy involucrado en el desarrollo de herramientas Web para la Web Semántica. Además, soy parte del Grupo de Usuario de Lignux de Allen (ULA) con los que comparto afición por el Software Libre, la computación y la electrónica.

**Nombre del que da el taller** Christian Gimenez

## 1.1 Licencias

En esta sección se indica la licencia de este documento y de los archivos que puede generar.

### 1.1.1 Este Documento (DevFest2018.org)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see `http://www.gnu.org/licenses/`.

### 1.1.2 Archivos Python Generados

```
# -*- coding: utf-8 -*-
'''
GNU GPG Ejemplo de uso.



:copyright: 2018 Christian Gimenez
:author: Christian Gimenez
:license: GPL v3 (see COPYING.txt or LICENSE.txt file for more information)
'''
#
# gpgEj1.py
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

```
# -*- coding: utf-8 -*-
'''
GNU GPG Ejemplo de uso.



:copyright: 2018 Christian Gimenez
:author: Christian Gimenez
:license: GPL v3 (see COPYING.txt or LICENSE.txt file for more information)
'''
#
# gnupgEj2.py
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

### 1.1.3 Archivos Ruby Generados

```
# -*- coding: utf-8 -*-
#
# Copyright 2018 Christian Gimenez
#
# Author: Christian Gimenez
#
# gpgme.rb
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
```

```
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

### 1.1.4 Archivos PHP Generados

```
<?php
/*

   Copyright 2018 Christian Gimenez

   Author: Christian Gimenez

   gpg.php

   This program is free software: you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation, either version 3 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program.  If not, see <http://www.gnu.org/licenses/>.
 */
```

# 2 Temas

- Cuento presentación de Alice Bob

- Mostrar un ROT por 5 segs. que Pablo lo copie y lo decifre.
- Explicar que eso se usaba y que sirvió, pero si otros lo copian no sirve.
- Breve reseña histórica: criptografía clásica vs moderna.
- Mostrar Wireshark y acceso a una página sin SSL/TLS. Aplicaciones.

# 3 Python

## 3.1 gnupg o pretty_bad_protocol Package

El paquete `gnupg` provée la clase GPG que puede generar claves y recibirlas del servidor de gnupg.net. Es mantenido por isislovecruft y el código fuente está disponible en su repositorio en GitHub.

Se puede instalar usando `pip3 install --user pretty-bad-protocol` o `pip3 install --user gnupg` para versiones legacy.

Éste paquete no utiliza las claves del sistema. En cambio, guarda las claves en otro directorio al del comando de terminal GPG.

### 3.1.1 Importar una Clave Pública

Para importar e inicializar una instancia GPG se ejecuta lo siguiente.

```
from pretty_bad_protocol.gnupg import GPG
gpg = GPG()
```

### 3.1.2 Recibir Claves

Con este comando se recibe una clave pública con el ID `C936B214EB93830D`.

```
a = gpg.recv_keys('hkp://keys.gnupg.net', '8EA85F43420045020C30EC02C936B214EB93830D')
a.counts
```

### 3.1.3 Crear Claves

Los datos del par de claves se genera con el siguiente comando.

```
key_settings = gpg.gen_key_input(key_type='RSA',
 key_length=1024,
 passphrase='foo')
key_settings
```

Con el siguiente comando se genera el par de claves con los datos de `key_settings`.

```
key = gpg.gen_key(key_settings)
key
```

#### 3.1.4 Cifrar

```
message = "The crow flies at midnight."
encrypted = gpg.encrypt(message, key.fingerprint, passphrase='foo')
str(encrypted)
```

#### 3.1.5 Descifrar

```
decrypted = gpg.decrypt(str(encrypted), passphrase='foo', always_trust=True)
str(decrypted)
```

### 3.2 GPG Package

El paquete `gpg` es mantenido por gnupg.org. Utiliza las claves que ya están en el sistema, por lo que se puede utilizar el comando de terminal `gpg2` o Kleopatra para gestionar las claves.

Se puede instalar utilizando `pip3 install --user gpg`.

```
import gpg
c = gpg.Context(armor=True)
c
```

#### 3.2.1 Listar Claves

Para listar las claves públicas generadas o importadas.

```
g = c.keylist('devfest2018')
ks = []
for i in g:
    ks.append(i)
```

#### 3.2.2 Cifrar

```
enc = c.encrypt('hola mundo'.encode(), recipients=ks, sign=False)
str(enc[0])
```

### 3.2.3 Descifrar

```
dec = c.decrypt(enc[0])
dec[0]
```

# 4 Ruby

La gema `gpgme` provee una interfaz para gestionar las claves de GPG que utiliza el sistema.

Se puede instalar por medio de `gem install gpgme`.

```
require 'gpgme'
c = GPGME::Crypto.new :armor => true
```

## 4.1 Cifrar

```
d = c.encrypt "hola mundo", :recipients => ['devfest2018']
d.to_s
```

## 4.2 Descifrar

```
d2 = c.decrypt d.to_s
d2.to_s
```

# 5 PHP

La librería estándar posee un paquete llamado `gpg`. Se puede utilizar para gestionar las claves que GPG posee en el sistema.

Se instala utilizando el gestor de paquetes propia de la distribución GNU/Linux que se está utilizando. También se puede utilizad PECL: `pecl install gpg`.

Para empezar a usarlo, se requiere crear un contexto.

```
$res = gnupg_init();
```

## 5.1 Cifrar

Primero, se debe indicar cuáles claves usar para cifrar con la función `gnupg_addencryptkey()` y para firmar digitalmente con `gnupg_addsignkey()`.

```
gnupg_addencryptkey($res, "christian.gimenez@fi.uncoma.edu.ar");
gnupg_addsignkey($res, "christian.gimenez@fi.uncoma.edu.ar");
$cipher = gnupg_encryptsign($res, 'Hello World');
```

### 5.2 Descifrar

Para descifrar, se requiere de una variable donde se guarda el contenido. La función `gnupg_decryptverify()` guarda el resultado en `$plaintext` y devuelve en `$info` información del estado.

```
$plaintext = "";
$info = gnupg_decryptverify($res, $cipher, $plaintext);
var_dump($plaintext)
print_r($info);
```