Introducción a la Criptografía

Gimenez, Christian

Universidad Nacional del Comahue
Facultad de Informática

GDG-DevFest 2018

Contenidos

Cuento de Alicia y Bob

Erase una vez...

Nuestro Personajes

- Alice, Bob, Carol, Dan/Dave, Erin
- Eve Trudy/Mallory
- Oscar, Faythe

Más: https://es.wikipedia.org/wiki/Alice_y_Bob

Cuento de Alicia y Bob

Erase una vez...

Nuestro Personajes

- Alice, Bob, Carol, Dan/Dave, Erin
- Eve Trudy/Mallory
- Oscar, Faythe

Más: https://es.wikipedia.org/wiki/Alice_y_Bob

¿Qué Sucede?

- Alice le pide a Faythe que le lleve una carta a Bob

Cuento de Alicia y Bob

Erase una vez...

Nuestro Personajes

- Alice, Bob, Carol, Dan/Dave, Erin
- Eve Trudy/Mallory
- Oscar, Faythe

Más: https://es.wikipedia.org/wiki/Alice_y_Bob

¿Qué Sucede?

- Alice le pide a Faythe que le lleve una carta a Bob
- Eve invita a Faythe a tomar té, le saca la carta y la lée.

Cuento de Alicia y Bob

Erase una vez...

Nuestro Personajes

- Alice, Bob, Carol, Dan/Dave, Erin
- Eve Trudy/Mallory
- Oscar, Faythe

Más: https://es.wikipedia.org/wiki/Alice_y_Bob

¿Qué Sucede?

- Alice le pide a Faythe que le lleve una carta a Bob
- Eve invita a Faythe a tomar té, le saca la carta y la lée.
- Trudy invita a Faythe a tomar mates, le saca la carta y la cambia.

Cuento de Alicia y Bob

Erase una vez...

Nuestro Personajes

- Alice, Bob, Carol, Dan/Dave, Erin
- Eve Trudy/Mallory
- Oscar, Faythe

Más: https://es.wikipedia.org/wiki/Alice_y_Bob

¿Qué Sucede?

- Alice le pide a Faythe que le lleve una carta a Bob
- Eve invita a Faythe a tomar té, le saca la carta y la lée.
- Trudy invita a Faythe a tomar mates, le saca la carta y la cambia.
- Bob se ofende al recibir la carta.

Criptografía Clásica

- Listo... Preparados...

Criptografía Clásica

- Listo... Preparados...
- ¡YA! 2 segundos.

¿Qué dice?

Elix jrkal

Criptografía Clásica

- Listo... Preparados...
- ¡YA! 2 segundos.
- Shhhh, no lo digas.

Criptografía Clásica

- Listo... Preparados...
- ¡YA! 2 segundos.
- Shhhh, no lo digas.
- ¡A copiar y a romper!

¿Qué dice?

Elix jrkal

¿Sirvió?

- ¿Sirvió a su propósito?

¿Sirvió?

- ¿Sirvió a su propósito?
- ¿Sabían el algoritmo? ¿La clave?

¿Sirvió?

- ¿Sirvió a su propósito?
- ¿Sabían el algoritmo? ¿La clave?
  - Principio de Kerckhoff.

Principio de Kerckhoff

- El algoritmo no requiere ser secreto.
- La fuerza del cifrado recae sobre la clave.

¿Sirvió?

- ¿Sirvió a su propósito?
- ¿Sabían el algoritmo? ¿La clave?
  - Principio de Kerckhoff.
- ¿Cuánto tiempo llevó romperlo?
  - Para cuando se rompe el cifrado César ya es tarde.
  - Complejidad computacional.

Principio de Kerckhoff

- El algoritmo no requiere ser secreto.
- La fuerza del cifrado recae sobre la clave.

Fortaleza de la Clave

En Nuestro Ejemplo

- Hay 26 posibles claves.
- Una computadora puede computarlas sin problemas.
  ¡En un segundo!

Fortaleza de la Clave

En Nuestro Ejemplo

- Hay 26 posibles claves.
- Una computadora puede computarlas sin problemas.
  ¡En un segundo!

Usando Algoritmos de Hoy en Día

Supongamos que la contraseña tiene 10 caracteres con:

- Números (10 posibles caracteres)
- Letras mayúsculas (26) y minúsculas (26) en inglés.

¿Cuántas contraseñas posibles?

Fortaleza de la Clave

En Nuestro Ejemplo

- Hay 26 posibles claves.
- Una computadora puede computarlas sin problemas.
  ¡En un segundo!

Usando Algoritmos de Hoy en Día

Supongamos que la contraseña tiene 10 caracteres con:

- Números (10 posibles caracteres)
- Letras mayúsculas (26) y minúsculas (26) en inglés.

¿Cuántas contraseñas posibles?

- $10 + 26 + 26 = 62$ caracteres que podemos usar.

Fortaleza de la Clave

En Nuestro Ejemplo

- Hay 26 posibles claves.
- Una computadora puede computarlas sin problemas.
  ¡En un segundo!

Usando Algoritmos de Hoy en Día

Supongamos que la contraseña tiene 10 caracteres con:

- Números (10 posibles caracteres)
- Letras mayúsculas (26) y minúsculas (26) en inglés.

¿Cuántas contraseñas posibles?

- $10 + 26 + 26 = 62$ caracteres que podemos usar.
- $VR_m^n = m^n$ es la Variación de m elementos en n posiciones.

Fortaleza de la Clave

En Nuestro Ejemplo

- Hay 26 posibles claves.
- Una computadora puede computarlas sin problemas.
  ¡En un segundo!

Usando Algoritmos de Hoy en Día

Supongamos que la contraseña tiene 10 caracteres con:

- Números (10 posibles caracteres)
- Letras mayúsculas (26) y minúsculas (26) en inglés.

¿Cuántas contraseñas posibles?

- $10 + 26 + 26 = 62$ caracteres que podemos usar.
- $VR_m^n = m^n$ es la Variación de m elementos en n posiciones.
- $62^{10} = 8{,}39299365868 \times 10^{17}$ posibles contraseñas.

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17} / 100000 = 8{,}393 \times 10^{12}$ segundos.

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17} / 100000 = 8{,}393 \times 10^{12}$ segundos.
- $8{,}393 \times 10^{12} / 60 = 1{,}399 \times 10^{11}$ minutos.

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17}/100000 = 8{,}393 \times 10^{12}$ segundos.
- $8{,}393 \times 10^{12}/60 = 1{,}399 \times 10^{11}$ minutos.
- $1{,}399 \times 10^{11}/60 = 2{,}331 \times 10^{9}$ horas.

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17}/100000 = 8{,}393 \times 10^{12}$ segundos.
- $8{,}393 \times 10^{12}/60 = 1{,}399 \times 10^{11}$ minutos.
- $1{,}399 \times 10^{11}/60 = 2{,}331 \times 10^{9}$ horas.
- $2{,}331 \times 10^{9}/24 = 97141130$ días.

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17}/100000 = 8{,}393 \times 10^{12}$ segundos.
- $8{,}393 \times 10^{12}/60 = 1{,}399 \times 10^{11}$ minutos.
- $1{,}399 \times 10^{11}/60 = 2{,}331 \times 10^{9}$ horas.
- $2{,}331 \times 10^{9}/24 = 97141130$ días.
- ¡$97141130/360 = 269836$ años!

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17}/100000 = 8{,}393 \times 10^{12}$ segundos.
- $8{,}393 \times 10^{12}/60 = 1{,}399 \times 10^{11}$ minutos.
- $1{,}399 \times 10^{11}/60 = 2{,}331 \times 10^{9}$ horas.
- $2{,}331 \times 10^{9}/24 = 97141130$ días.
- ¡$97141130/360 = 269836$ años!

Pero las contraseñas pueden ser más complejas realmente:

- ¡Pueden tener símbolos, letras de otros idiomas, etc!
- ¡Pueden tener más de 10 caracteres!

Fortaleza de la Clave

Si pensamos que la computadora puede probar 100000 contraseñas por segundo, va a tardar:

- $8{,}39299365868 \times 10^{17}/100000 = 8{,}393 \times 10^{12}$ segundos.
- $8{,}393 \times 10^{12}/60 = 1{,}399 \times 10^{11}$ minutos.
- $1{,}399 \times 10^{11}/60 = 2{,}331 \times 10^{9}$ horas.
- $2{,}331 \times 10^{9}/24 = 97141130$ días.
- ¡$97141130/360 = 269836$ años!

Pero las contraseñas pueden ser más complejas realmente:

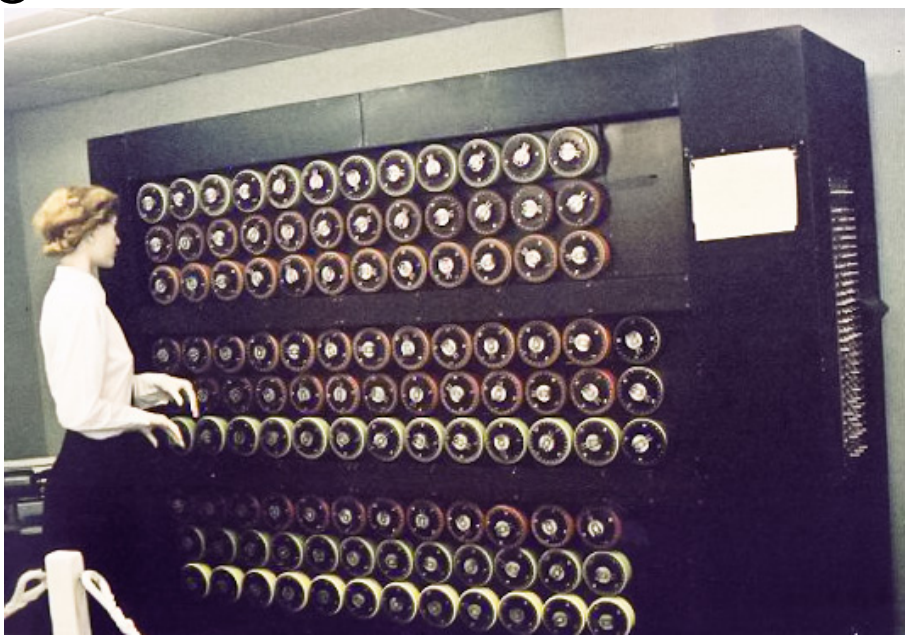
- ¡Pueden tener símbolos, letras de otros idiomas, etc!
- ¡Pueden tener más de 10 caracteres!

Por fuerza bruta se tarda muchísimo aún si usaramos un diccionario.

Usos en la Historia Moderna

Enigma vs. Bombe

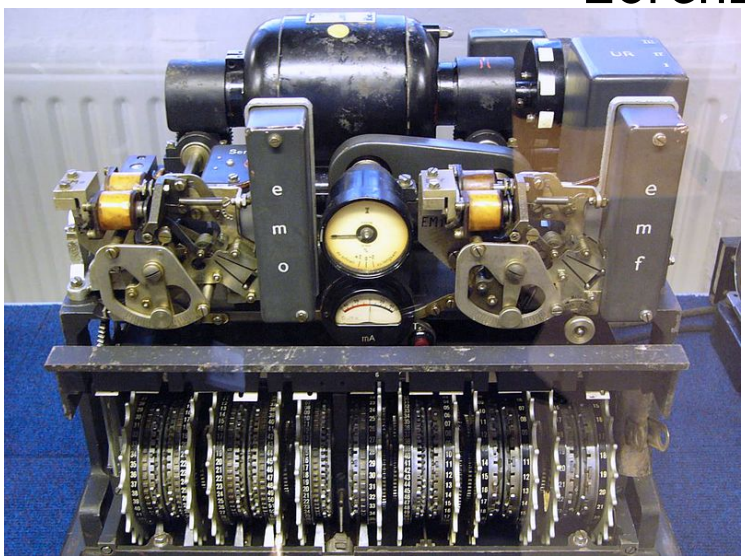
vs.



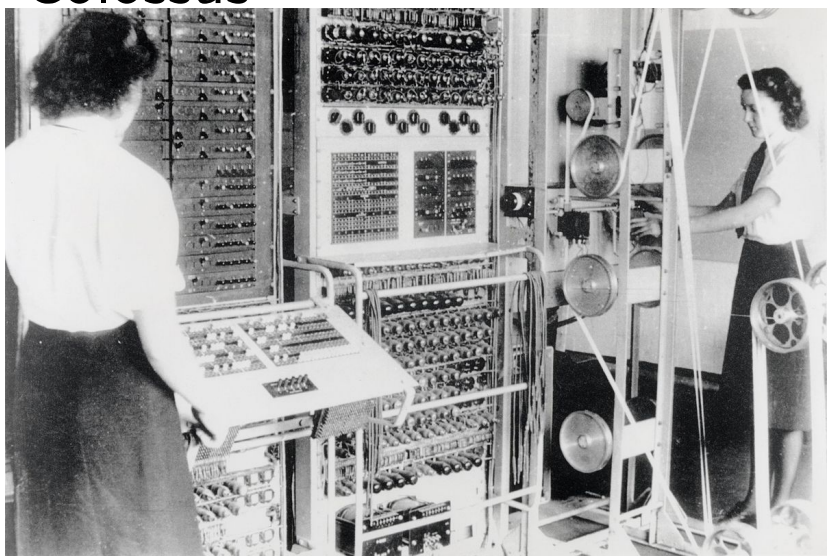
Imágenes obtenidas desde Wikipedia.org:
Enigma Machine: https://commons.wikimedia.org/wiki/File:EnigmaMachine.jpg bajo la licencia de Domino Público.
Turing Bombe: https://commons.wikimedia.org/wiki/File:TuringBombeBletchleyPark.jpg bajo las licencias GNU FDL, CC-By-SA 3.0, CC-By 2.5.

Usos en la Historia Moderna

Lorenz vs. Colossus



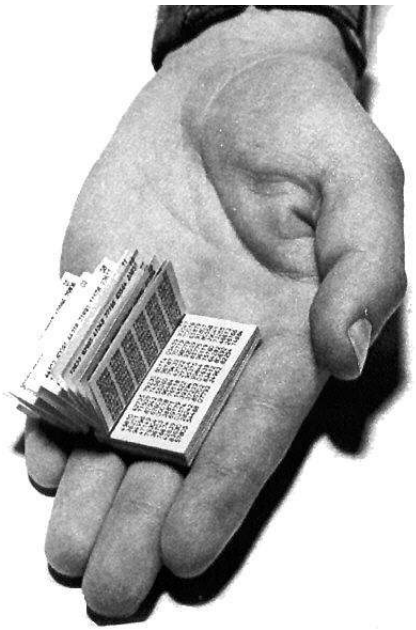
vs.



Imágenes obtenidas desde Wikipedia.org bajo licencia de Dominio Público:
https://en.wikipedia.org/wiki/Lorenz_cipher y
https://en.wikipedia.org/wiki/Colossus_computer

Usos en la Historia Moderna

OTP vs. Proyecto VENONA



vs.

OTP obtenida desde ranum.com (Copyright(C) 1995 Marcus J. Ranum. All rights reserved): http://www.ranum.com/security/computer_security/papers/otp-faq/
VENONA obtenida desde Wikipedia.org (Imágen de Dominio Público): https://en.wikipedia.org/wiki/VENONA

Sip, en Mr. Robot también

Tapa de Mr. Robot 2da temporada, obtenida desde IMDB.com (© 1990-2018 IMDb.com, Inc.):
https://www.imdb.com/title/tt4158110/

Sip, en Mr. Robot También

¿Qué dirá?

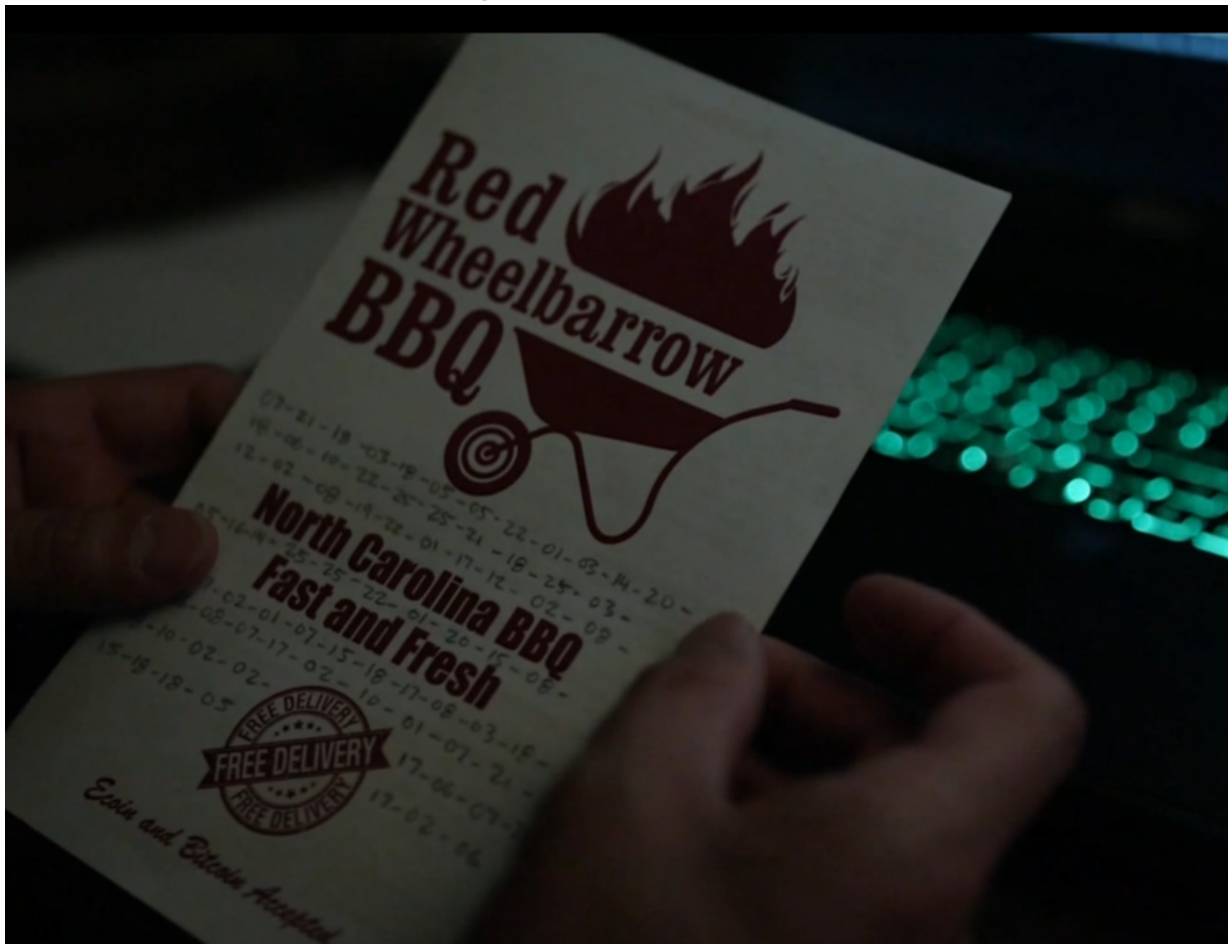


Imagen obtenida desde GeekWire.com (© 2011-2018 GeekWire, LLC):
https://www.geekwire.com/2016/mr-robot-rewind-code-cracking-mysterious-mind-bending-episode-11/

¿Qué dirá? ¿Lo desciframos?

07-21-18-03-18-05-05-22-01-03-14-20-
18-06-10-22-25-25-21-18-25-03-
12-02-08-19-22-01-17-12-02-08-
05-16-14-25-25-22-01-20-15-08-
07-17-02-01-07-15-18-17-08-03-18-
17-16-08-07-17-02-10-01-07-21-
18-10-02-02-17-06-07-21-18-12-
15-18-18-05-17-02-06

Criptografía Moderna

- Deja de ser un arte.
  - ¿El algoritmo se puede romper?
  - ¿Qué tan seguro es?
  - ¿Qué propiedades existen para los algoritmos de cifrados?

Criptografía Moderna

- Deja de ser un arte.
  - ¿El algoritmo se puede romper?
  - ¿Qué tan seguro es?
  - ¿Qué propiedades existen para los algoritmos de cifrados?
- Formalizado con argumentos matemáticos.
  Se puede saber con más certeza:
  - Las vulnerabilidades.
  - Cuánto se tarda en romperlo dependiendo de la fortaleza del atacante.
  - ¿Qué hace falta para crear nuevos protocolos?

Criptografía Moderna

- Deja de ser un arte.
  - ¿El algoritmo se puede romper?
  - ¿Qué tan seguro es?
  - ¿Qué propiedades existen para los algoritmos de cifrados?
- Formalizado con argumentos matemáticos.
  Se puede saber con más certeza:
  - Las vulnerabilidades.
  - Cuánto se tarda en romperlo dependiendo de la fortaleza del atacante.
  - ¿Qué hace falta para crear nuevos protocolos?
- Criptografía Simétrica
- Aparece la Criptografía Asimétrica

Aplicaciones

Ufff... ¡Muchas!

HTTP vs. HTTPS

- ¡Vamos a ver la diferencia! ¡¡¡A sniffear!!!

Aplicaciones

Ufff... ¡Muchas!

HTTP vs. HTTPS

- ¡Vamos a ver la diferencia! ¡¡¡A sniffear!!!

Computación

- En redes HTTPS, POP3S, SMTPS, etc.
- Sistemas bancarios. ¿Se imaginan Home Banking sin HTTPS?
- Consolas remotas SSH ¿Se imaginan las contraseñas de login y sudo?

Aplicaciones

Ufff... ¡Muchas!

HTTP vs. HTTPS

- ¡Vamos a ver la diferencia! ¡¡¡A sniffear!!!

Computación

- En redes HTTPS, POP3S, SMTPS, etc.
- Sistemas bancarios. ¿Se imaginan Home Banking sin HTTPS?
- Consolas remotas SSH ¿Se imaginan las contraseñas de login y sudo?

DNS y Autenticación

- Con técnicas de cifrado se puede autenticar un mensaje.
- Ej.: Paquetes de GNU/Linux se autentican.
- UNComa tuvo que autenticar sus DNS hace un par de días.

Periodismo y Hacktivismo

- Edward Snowden y Julian Asange los usaban continuamente.
- Cifrado de discos y respaldos seguro en caso de que caigan en manos ajenas.
- Chat con OTR (obsoleto ahora se aconseja OMEMO o OpenPGP) para evitar que se filtre información antes de tiempo.
- Mail cifrados.
- Anonimato: TOR, FreeNet, etc. Usan cifrado por computación distribuida.

Periodismo y Hacktivismo

- Edward Snowden y Julian Asange los usaban continuamente.
- Cifrado de discos y respaldos seguro en caso de que caigan en manos ajenas.
- Chat con OTR (obsoleto ahora se aconseja OMEMO o OpenPGP) para evitar que se filtre información antes de tiempo.
- Mail cifrados.
- Anonimato: TOR, FreeNet, etc. Usan cifrado por computación distribuida.

Criptomonedas

- Se utiliza la criptografía para autenticar transacciones.
- Se firma digitalmente nuevas monedas.
- ¿Para guardar las monedas en tu Wallet?

Criptografía Simétrica

Criptografía Simétrica

- Alice y Bob comparten la misma clave.
- Ambos conocen el algoritmo a utilizar.

Criptografía Simétrica

Criptografía Simétrica

- Alice y Bob comparten la misma clave.
- Ambos conocen el algoritmo a utilizar.

Pros y Contras

- Rápida.
- Hay algoritmos para flujos de datos.
- Ciphertext no ocupa tanto espacio.
- Clave es corta y memorizable.

Criptografía Simétrica

Criptografía Simétrica

- Alice y Bob comparten la misma clave.
- Ambos conocen el algoritmo a utilizar.

Pros y Contras

- Rápida.
- Hay algoritmos para flujos de datos.
- Ciphertext no ocupa tanto espacio.
- Clave es corta y memorizable.
- Necesitan de un medio seguro para compartirla.
- ¿Qué pasa si Alice vive en Argentina y Bob en Japón?
- ¿Y si son muchas personas?

GPG Simétrica

GNU Privacy Guard

- Implementación del estándar OpenPGP ← ¿Recuerdan a Kerckhoff?
- Están en todos los GNU/Linuxs por defecto ← ¿Recuerdan las aplicaciones?

Se puede descargar de: www.gnupg.org

GPG Simétrica

GNU Privacy Guard

- Implementación del estándar OpenPGP ← ¿Recuerdan a Kerckhoff?
- Están en todos los GNU/Linuxs por defecto ← ¿Recuerdan las aplicaciones?

Se puede descargar de: www.gnupg.org

¿Qué protocolos puedo usar?

```
gpg2 --version --verbose
```

GPG Simétrica

GNU Privacy Guard

- Implementación del estándar OpenPGP ← ¿Recuerdan a Kerckhoff?
- Están en todos los GNU/Linuxs por defecto ← ¿Recuerdan las aplicaciones?

Se puede descargar de: www.gnupg.org

¿Qué protocolos puedo usar?

```
gpg2 --version --verbose
```

Cifrar y Descifrar

Cifrar: `gpg2 --symmetric --armor`

Descifrar: `gpg2`

Criptografía Simétrica

Ejemplo de Cifrado

Texto: `hola mundo`

Clave: `hola`

```
-----BEGIN PGP MESSAGE-----

jA0EBwMC4IuUSdLhQgnk0kABk0A0FkaWeBTc9LvkQM78V5nFmu6lVorXaak8h5yJ
p/DK/UZBWplQ/ycrU5/B4eB9TryS5FnOEoBBWpBBfo4l
=owdv
-----END PGP MESSAGE-----
```

Descifrando

- Ejecutar gpg2
- Pegar el texto desde `----BEGIN PGP MESSAGE-----` hasta `-----END PGP MESSAGE-----` inclusive.
- Pedirá la contraseña que se usó para cifrar.
- Presionar Control+d (GNU/Linux) o Control+c para salir.

Criptografía Asimétrica

- Alice y Bob se crean un par de claves. Conocen el algoritmo.
- Clave pública: Se puede mostrar y se comparte.
- Clave privada: ¡Nunca se muestra!

Criptografía Asimétrica

- Alice y Bob se crean un par de claves. Conocen el algoritmo.
- Clave pública: Se puede mostrar y se comparte.
- Clave privada: ¡Nunca se muestra!

Cifrar y Descifrar

- Alice usa la clave pública de Bob para cifrar su texto.
- Bob descifra usando su clave privada.

Criptografía Asimétrica

- Alice y Bob se crean un par de claves. Conocen el algoritmo.
- Clave pública: Se puede mostrar y se comparte.
- Clave privada: ¡Nunca se muestra!

Cifrar y Descifrar

- Alice usa la clave pública de Bob para cifrar su texto.
- Bob descifra usando su clave privada.

Firmas Digitales

- Integridad y autenticación
- Alice usa su clave privada para firmar un texto.
- Bob usa la clave pública de Alice para verificar el texto.

Criptografía Asimétrica

Pros y Contras

- Claves más grandes y seguras.

Criptografía Asimétrica

Pros y Contras

- Claves más grandes y seguras.
- ¡No hace falta compartir una contraseña!

Criptografía Asimétrica

Pros y Contras

- Claves más grandes y seguras.
- ¡No hace falta compartir una contraseña!
- Claves no memorizables: ¡son archivos!

Criptografía Asimétrica

Pros y Contras

- Claves más grandes y seguras.
- ¡No hace falta compartir una contraseña!
- Claves no memorizables: ¡son archivos!
- Muy Lento. No soporta streaming.

Criptografía Asimétrica

Pros y Contras

- Claves más grandes y seguras.
- ¡No hace falta compartir una contraseña!
- Claves no memorizables: ¡son archivos!
- Muy Lento. No soporta streaming.
- Ciphertexts "grandes".

Criptografía Asimétrica

Crear Par de Claves

Sólo se hace una vez, a menos que se quieran varios pares.

- `gpg2 --gen-key`

Criptografía Asimétrica

Crear Par de Claves

Sólo se hace una vez, a menos que se quieran varios pares.

- `gpg2 --gen-key`

Cifrar y Descifrar

- `gpg2` (Por defecto asume asimétrica)
- `gpg2 --decrypt --armor`
- `gpg2 --encrypt --armor`

Criptografía Asimétrica

Crear Par de Claves

Sólo se hace una vez, a menos que se quieran varios pares.

- `gpg2 --gen-key`

Cifrar y Descifrar

- `gpg2` (Por defecto asume asimétrica)
- `gpg2 --decrypt --armor`
- `gpg2 --encrypt --armor`

Firmar y Verificar

- `gpg2` (Por defecto verifica)
- `gpg2 --verify --armor`
- `gpg2 --clear-sign --armor --default-key ID`

Asimétrica: Gestión de Claves

Listar claves públicas: `gpg2 --list-keys`

Listar claves con cierto texto: `gpg2 --list-keys TEXTO`

Exportar una clave pública: `gpg2 --export-keys TEXTO`

`TEXTO` puede ser un nombre, correo o ID.

Asimétrica: Gestión de Claves

Listar claves públicas: `gpg2 --list-keys`

Listar claves con cierto texto: `gpg2 --list-keys TEXTO`

Exportar una clave pública: `gpg2 --export-keys TEXTO`

`TEXTO` puede ser un nombre, correo o ID.

Así se muestran las claves

```
pub   rsa2048 2018-09-16 [SC] [expires: 2020-09-15]
      C3DED999449A50B18E564C83463FC5D945853BF0
uid           [ultimate] devfest2018 <devfest2018@gmail.com>
sub   rsa2048 2018-09-16 [E] [expires: 2020-09-15]
```

Asimétrica: Gestión de Claves

Listar claves públicas: `gpg2 --list-keys`
Listar claves con cierto texto: `gpg2 --list-keys TEXTO`
Exportar una clave pública: `gpg2 --export-keys TEXTO`

`TEXTO` puede ser un nombre, correo o ID.

Así se muestran las claves

```
pub   rsa2048 2018-09-16 [SC] [expires: 2020-09-15]
      C3DED999449A50B18E564C83463FC5D945853BF0
uid           [ultimate] devfest2018 <devfest2018@gmail.com>
sub   rsa2048 2018-09-16 [E] [expires: 2020-09-15]
```

El ID ("huella dactilar" ó "fingerprint") identifica una única clave pública/privada. Se puede usar completa o los últimos 8 dígitos. Útil si tenemos varias claves públicas.
En nuestro ejemplo, estos dos IDs referencian a la misma clave:

- `C3DED999449A50B18E564C83463FC5D945853BF0`
- `45853BF0`

Asimétrica: Cifrar y Descifrar

Cifrar

- `gpg2 --encrypt --armor`
- Dar un ID, correo o nombre. ENTER.
- Se pueden agregar varios destinatarios.
- Para terminar, dejar en blanco y presionar ENTER.
- Escribir el mensaje.
- Control+d (GNU/Linux) o Control+c para terminar.

Asimétrica: Cifrar y Descifrar

Cifrar

- `gpg2 --encrypt --armor`
- Dar un ID, correo o nombre. ENTER.
- Se pueden agregar varios destinatarios.
- Para terminar, dejar en blanco y presionar ENTER.
- Escribir el mensaje.
- Control+d (GNU/Linux) o Control+c para terminar.

Descifrar

- gpg2
- Pegar el texto como en el de asimétrica.
- Si nuestra clave privada tiene contraseña, la va a pedir.

Asimétrica: Firmar y Verificar

Firmar

- `gpg2 --clear-sign --armor --default-key TEXTO`
  `TEXTO` puede ser ID, nombre o correo.
- Escribir el mensaje.
- Control+d (GNU/Linux) o Control+c para terminar.
- Pedirá la contraseña de la clave privada.

Cifrar y Firmar

`gpg2 --encrypt --sign --armor --default-key TEXTO`

- ¡No se usa `--clear-sign`!
- Procedimiento es el mismo para cifrar.

Asimétrica: Firmar y Verificar

Ejemplo de un texto firmado (clear-sign)

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

hola mundo
-----BEGIN PGP SIGNATURE-----

iQFKBAEBCAA0FiEEw97ZmUSaULGOVkyDRj/F2UWFO/AFAlue1RUWHGRldmZlc3Qy
MDE4QGdtYWlsLmNvbQAKCRBGP8XZRYU78C2sB/9LRaC7ntqCtiX888zpoaa4MUZc
jz+2q0h7SkFlUnQHovnKxqA+is5WHSs5iV58rAe1qi0Lr73uA+AzjXTEECA861iz
pwmMHuskLQ44f3YijLdHc8Q62quwuLlZ4VEi1ldNf6A6TEmyZ5lAVvRsE/N8UCVl
QsJroidG4dVfbFsu0vsqnlSeGYjyX7nFW8uDs2JRcqsS/R2DHdGhWHHn4/JBBta8
S9J20V87JDmxPyQxJz25eQM46QEUGgnbLNyp6HgIOA/5xMF+q8qzNtvB97pKxjWS
6mgsNfHnOE0R5QIwxMxdrHXUD7sDJu5wkFQBm8NZDfhr9WqzAL8dy+j3MzKw
=ldDr
-----END PGP SIGNATURE-----
```

Asimétrica: Firmar y Verificar

Verificar

- gpg2
- Pegar el texto completo (como se muestra en el ejemplo).

Asimétrica: Firmar y Verificar

Verificar

- gpg2
- Pegar el texto completo (como se muestra en el ejemplo).

Válido

```
gpg: Signature made dom 16 sep 2018 19:11:33 -03
gpg:                using RSA key C3DED999449A50B18E564C83463FC5D945853BF0
gpg:                issuer "devfest2018@gmail.com"
gpg: Good signature from "devfest2018 <devfest2018@gmail.com>" [ultimate]
```

Asimétrica: Firmar y Verificar

Verificar

- gpg2
- Pegar el texto completo (como se muestra en el ejemplo).

Válido

```
gpg: Signature made dom 16 sep 2018 19:11:33 -03
gpg:                using RSA key C3DED999449A50B18E564C83463FC5D945853BF0
gpg:                issuer "devfest2018@gmail.com"
gpg: Good signature from "devfest2018 <devfest2018@gmail.com>" [ultimate]
```

Inválido

Cuando no dice “Good Signature” es por:

- ¡El texto fue modificado! y/o
- ¡La firma fue modificada! y/o
- ¡La firma no coincide con la clave pública!

GPG En PHP

PECL

gnupg: http://php.net/manual/en/book.gnupg.php

Cifrar

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

Descifrar

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

En PHP

Todo acerca de Crypto

Tooodo lo que PECL puede ofrecer:
http://php.net/manual/en/refs.crypto.php
OpenSSL
http://php.net/manual/en/function.openssl-encrypt.php

En Python

gnupg Package

gnupg: https://pypi.org/project/gnupg/
https://github.com/isislovecruft/python-gnupg

```
pip3 install --user gnupg
```

Usa un directorio diferente al de GPG para almacenar claves.

Generar Claves

```
from gnupg import GPG
gpg = GPG()
key_settings = gpg.gen_key_input(key_type='RSA', key_length=1024,
passphrase='foo')
key = gpg.gen_key(key_settings)
```

Cifrar y Descifrar

```
message = "The crow flies at midnight."
encrypted = str(gpg.encrypt(message, key.fingerprint))
decrypted = str(gpg.decrypt(encrypted))
```

En Python - A falta de uno, ¡hay muchos!

gpg

Mantenido por www.gnupg.org.

```
pip3 install --user gpg
```

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

Buscar una clave

```
k = c.keylist('micorreo@host.com')
lst = []
for i in k:
  lst.append(i)
k = c.get_key('HASH')
```

Cifrar y Descifrar

```
enc = c.encrypt('hello world'.encode(), recipients=list_keys, sign=False)
dec = c.decrypt(enc[0])
```

En Ruby

Gema

```
gem install gpgme
```

Comandos de Ruby

ri	Ayuda en terminal.
irb	Consola REPL
#help	Comando de ayuda dentro de irb.

Cifrar y Descifrar

```
require 'gpgme'
c = GPGME::Crypto.new :armor => true
d = c.encrypt ``hola mundo'',
  :recipients => ['christian.gimenez@fi.uncoma.edu.ar']
d2 = c.decrypt d.to_s
```

C'est Fini

Consejos de Última Hora

- Siempre dejar disponible sus claves públicas.

C'est Fini

Consejos de Última Hora

- Siempre dejar disponible sus claves públicas.
- Siempre estén prevenidos: TOR, SSH, HTTPS, etc.

C'est Fini

Consejos de Última Hora

- Siempre dejar disponible sus claves públicas.
- Siempre estén prevenidos: TOR, SSH, HTTPS, etc.
- Siempre cifren su información sensible.

C'est Fini

Consejos de Última Hora

- Siempre dejar disponible sus claves públicas.
- Siempre estén prevenidos: TOR, SSH, HTTPS, etc.
- Siempre cifren su información sensible.
- Siempre traten de compartirse claves cuando se encuentren personalmente.

C'est Fini

Consejos de Última Hora

- Siempre dejar disponible sus claves públicas.
- Siempre estén prevenidos: TOR, SSH, HTTPS, etc.
- Siempre cifren su información sensible.
- Siempre traten de compartirse claves cuando se encuentren personalmente.
- Nunca deben dar información personal identificatoria sea parcial o total a desconocidos.

C'est Fini

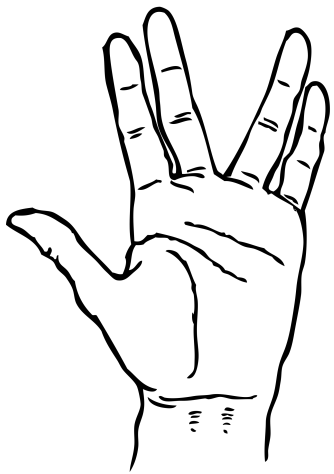
Consejos de Última Hora

- Siempre dejar disponible sus claves públicas.
- Siempre estén prevenidos: TOR, SSH, HTTPS, etc.
- Siempre cifren su información sensible.
- Siempre traten de compartirse claves cuando se encuentren personalmente.
- Nunca deben dar información personal identificatoria sea parcial o total a desconocidos.
- Nunca deberían aceptar información de importancia si no está firmada digitalmente.

C'est Fini

¡Muchas Gracias!

"Larga vida y prosperidad"



¡Gracias!

¡Gracias por su participación!

Vulcan Hand por qubodup: https://openclipart.org/detail/201741/vulcan-hand
Waving man por liftarn: https://openclipart.org/detail/181805/waving-man
Ambos obtenidos desde OpenClipArt y con licencia CC-0.

Ce N'est Pas Fini!

Ah, ¿no se terminó?... Ufff, qué cosas... Me quedé sin presentación...

Ce N'est Pas Fini!

Ah, ¿no se terminó?... Ufff, qué cosas... Me quedé sin presentación...

¡Siempre hay algo para charlar!

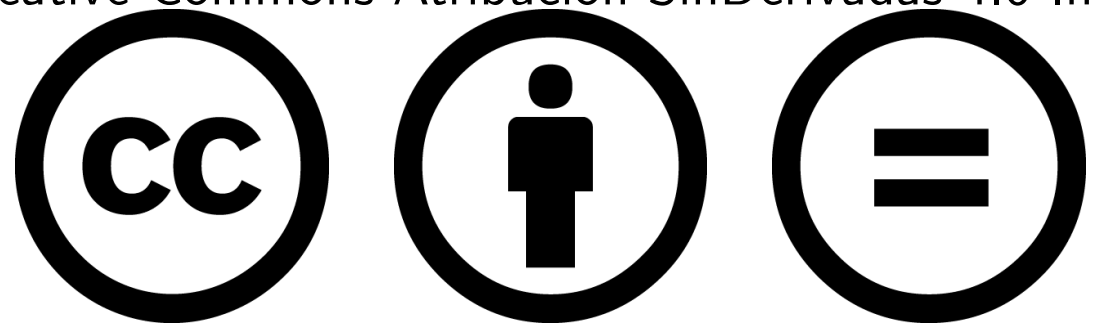
- TOR, FreeNet, ZeroNet, GNUnet, Bitmessage.
- XMPP + OpenPGP, OMEMO, OTR.
- Redes sociales libres y descentralizadas: Diáspora, Friendica, etc.
- Herramientas: Kleopatra, SSH/STunel, Cryptsetup y LUKS.
- Criptomonedas y blockchain.
- Criptografía post-quantum.

Emojiu1f605 por laughingman11 desde OpenClipart (Licencia: CC-0) https://openclipart.org/detail/253500/emojiu1f605

Licencia de Esta Presentación

Excepto en los lugares que se ha indicado lo contrario:

Introducción a la Criptografía por Christian Gimenez se distribuye bajo una Licencia Creative Commons Atribución-SinDerivadas 4.0 Internacional.



CC-By-ND

Excepto en los lugares que se ha indicado lo contrario:
Esta obra está licenciada bajo la Licencia Creative Commons Atribución-SinDerivadas 4.0 Internacional. Para ver una copia de esta licencia, visite http://creativecommons.org/licenses/by-nd/4.0/.