



Comunicación Encubierta

Proyecto final

Integrantes: Andrés Calderón
Roberto Rivera

Profesor: Alejandro Hevia

Auxiliar: Bryan Ortiz

Ayudante: David Contreras

Fecha de entrega: 9 de Diciembre de 2024
Santiago, Chile

I. Introducción

La comunicación encubierta, *Covert Communication* en inglés, es un método de intercambio de información que esconde la existencia misma de la comunicación. Es fundamentalmente distinta de las técnicas convencionales de criptografía vistas en el curso, que más bien apuntan a proteger el contenido de la comunicación, mientras que sigue siendo claro que la comunicación está ocurriendo. El objetivo de la comunicación encubierta es entonces prevenir que un adversario que intente espiar la comunicación siquiera se entere que esta está ocurriendo.

Para ello suele ocultarse el mensaje a comunicar dentro de otra información que no llame la atención, esto le permite al remitente y receptor mantener el contenido de su comunicación confidencial ante terceros. En el mundo de la tecnología digital, esto puede realizarse ocultando los mensajes cifrados dentro de imágenes, archivos de audio, videos, entre otros. Si bien de forma convencional los sistemas criptográficos se idean para comunicación entre pares, para este trabajo se diseñó e implementó una solución con enfoque de chat global.

La comunicación encubierta constituye un enfoque desafiante y distintivo dentro de la criptografía y la ciberseguridad, pues la pregunta fundamental ya no es qué se está comunicando entre usuarios, sino más bien “¿Se está comunicando algo siquiera a través del sistema?”

II. Solución e Implementación

El problema planteado exigía como requerimientos desarrollar una aplicación que implementase comunicación encubierta, de tal forma que un adversario que monitorease la conversación no pudiese decidir con seguridad si un mensaje secreto ha sido transmitido o no. Además, aún si el adversario confirmase la existencia de tal comunicación secreta, esta debe permanecer confidencial e íntegra/autenticada. Para desarrollar dicha aplicación, se optó por utilizar el *framework* de desarrollo web *Django*.

La solución consiste en un chat global en forma de tablero de imágenes, en el que cada usuario registrado tendrá unas claves pública y privada asociadas. Los usuarios observarán su clave privada por única vez al momento de registrarse, por lo que es de su exclusiva responsabilidad guardar esta de forma segura, y serán capaces de subir imágenes de forma ilimitada al tablero, con la posibilidad de esconder en dichas imágenes un mensaje secreto para algún destinatario registrado en el sistema.

Estos mensajes se cifrarán haciendo uso de la clave pública del destinatario, y solo podrán ser descifrados por este cuando ingrese su clave privada en el formulario correspondiente y accione el botón “Decodificar”. Para mantener la integridad y autenticidad se concatena un MAC creado a partir del mensaje encriptado y la clave pública del remitente, de modo que se cumple con el esquema *Encrypt-then-Mac*.

El modelo de la aplicación web define dos clases, una de usuarios llamada *User* que hereda de la clase por defecto *AbstractUser* de *Django*, y guarda además como atributo la clave pública asociada. Así mismo, se tiene la clase para las imágenes *Image*, que guarda la ruta de la imagen, la marca de tiempo de creación de la imagen, y un campo extra utilizado para guardar el largo del mensaje secreto.

La lógica de encriptación, autenticación y encubrimiento de la comunicación se lleva acabo utilizando las funciones descritas dentro del módulo `covert.py` del proyecto. En particular, se definen las siguientes funciones:

- `generar_hmac`: genera un MAC en base al mensaje cifrado y la clave pública del emisor, de modo que se utiliza el esquema *Encrypt-then-Mac*.

- **verificar_hmac:** verifica que el MAC del mensaje recibido sea igual al que resulta de crear un MAC usando el mensaje y la clave pública del supuesto emisor.
- **generar_claves:** genera un par de claves RSA, pública y privada, que se utilizarán después en el cifrado y descifrado de mensajes.
- **encriptar_mensaje:** cifra un mensaje utilizando la clave pública del receptor, de tal forma que solo este, utilizando su clave privada, podrá descifrarlo.
- **desencriptar_mensaje:** descifra un mensaje cifrado utilizando la clave privada, asegurando que solo el destinatario objetivo pueda acceder al contenido del mensaje.
- **ocultar_mensaje_imagen:** inserta un mensaje cifrado dentro de una imagen usando técnicas de esteganografía (ocultación de información dentro de un objeto), alterando los bits menos significativos de los píxeles de dicha imagen (*LSB steganography*).
- **extraer_mensaje_imagen:** recupera un mensaje oculto de una imagen procesada previamente, leyendo los bits menos significativos según la longitud especificada.

Respecto a las vistas de la aplicación, en el archivo `views.py` se tienen funciones que gestionan la lógica de la aplicación, procesan las solicitudes de los usuarios y devuelven las respuestas respectivas. Para ello se implementan las siguientes funcionalidades:

- **convert_to_png:** convierte una imagen cargada por el usuario a formato PNG, haciendo uso de la librería *Pillow* para convertir la imagen a formato RGBA y luego guardarla en el formato deseado, esto, para que no sea posible identificar si hubo una comunicación por el tipo de archivo subido.
- **index:** página por defecto de la aplicación, siendo esta el chat global, se necesita haber iniciado sesión para ingresar.
- **register:** muestra un formulario que permite a un nuevo usuario registrarse en la aplicación. Una vez envía este formulario (POST), se genera su par de claves pública y privada RSA, guardándose la primera automáticamente y redirigiendo al usuario a la página principal.
- **login2:** permite a los usuarios iniciar sesión. Si el usuario proporciona credenciales válidas, se autentica y dirige a la página principal.
- **logout2:** permite a los usuarios cerrar sesión. Al ser llamada, el usuario es desconectado y se redirige a la página de inicio de sesión.
- **home:** muestra la página principal, donde los usuarios pueden ver las imágenes con mensajes ocultos y subir nuevas imágenes, siguiendo la lógica de “tablero de imágenes global” descrita anteriormente.

Si el usuario realiza una solicitud GET, se muestran las imágenes en el tablero, y los mensajes ocultos que se le han enviado si ingresa su clave privada en el formulario respectivo, en caso de que además hayan aprobado la verificación del MAC. Esto último añade una capa más de seguridad, pues en caso de que un adversario haya intervenido la comunicación y manipulado el mensaje, este no será mostrado.

Por otro lado, si el usuario realiza una solicitud POST, puede subir una imagen con un mensaje oculto en ella, el mensaje entonces se cifra con la clave pública del destinatario. En caso de no especificarse un destinatario, simplemente se sube la imagen al tablero sin mensaje oculto alguno.

Por último, Si se presiona el botón “refrescar” se recarga la página eliminando los mensajes desencriptados o la clave privada que se muestra al registrarse.

Así, se tienen unas vistas diseñadas para manejar la interacción del usuario con la aplicación de comunicación encubierta, haciendo uso adecuado de las funciones criptográficas implementadas en el módulo `covert.py`.

Estas vistas gestionan el registro de nuevos usuarios, inicio y cierre de sesión, y les permiten interactuar con el sistema de mensajes ocultos en imágenes, subiendo los suyos propios referenciando a un usuario destinatario, o revelando aquellos que ha recibido haciendo uso de su clave privada, la cuál le es entregada una única vez al registrarse en el sistema y es de su responsabilidad mantener, de forma similar a como lo hace `ssh` en *Linux*, guardando dichas credenciales en archivos dentro del dispositivo.

III. Resultados

Para este apartado, se mostrará un flujo completo de uso de la aplicación, simulando un ejemplo de comunicación efectiva entre dos usuarios, tanto explicando cada paso como mostrando imágenes apropiadas.

- **Registro de usuario:** cada usuario que desee registrarse debe proveer de un nombre de usuario, correo electrónico y contraseña. Se aplican medidas mínimas para asegurar la correctitud y seguridad de la contraseña (frecuencia, largo, etcétera).

Registro de Usuario

Username: Required. 150 characters or fewer. Letters, digits and @/./+/_ only.

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

¿Ya tienes cuenta? [Inicia sesión aquí.](#)

Figura 1: Registro del usuario 'rob'

- **Recepción de clave privada:** una vez creado el usuario, este será redirigido a la página principal (home) donde recibirá en pantalla su clave privada. Como se ha mencionado previamente, esta es de uso y responsabilidad exclusiva del usuario, por lo que deberá guardarla apropiadamente.

Bienvenido rob al chat global

▼ Registro exitoso. Esta es tu clave privada (no se la entregues a nadie)

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAmd1pCbeMa1rbo3dIbeNG99QeD9WzHc:
FkA3b7wPQec1Y7TtazwCDDj9JWU74QpJLdR04QK5A
SmYlXg9EOTC4TH3fmg+eqQpTZXN0XNlSHk4d4:08HT:ZN24u
kFaklmy1SQ66fFlagAIdhAAdVYhL3qT9TVNwaBHHQaVeE
Ojw+atR2K6c2u36vNacY+eqQpTZY0Y046PDDV3pB8SCXNq
73AIDJFOc2YTr6XN3H8L83db8b+asG4fwEzQIDAQABoBADCd
AlZgWADCTGua99AR0FYzRmVU1e1ZkN85W68Nw5m9HYM4
dH1M6L2N9Q-Vf9v38E48CqjAWac+uHfQm+eqQpTATG0fgp5
pHON7p0hr+Y3Vfroz08L3CemZRe3808Bj2346pYbyT1eTXaAP
a+CoDFADaajZaA9AORNV65BLUW1r7Tj86qXN9j1eVTHUS
T3Cz2zSHUK7vQRIZ+uUChuanwV6EJZUT7EEDXZT+uPT7RgHj
9H8IDmCgYEA5SMq55w53kkl0epQdaGJOWMaaBefRufV0
zmakDM8L7awwYVt96C4ew7Kcm+lg8w7WZCQ2n6q6R8R1P7V
HgX0THgBY8VbD4p54R8gKIPN2YkTBSA609VTAT9YVY5eIK
a7KQp+CD7TugGLg0HQyfuYvKNE95oL9D5OGAD9EIEPRK3
d0X1dLSp4l+u0huaiV9H8T3AJV99+V7B9ZG69YyLLXgZPAT
2+R4M4pAWK4H89+3Sk8uG8ZD66cATVNT3CgYEAqCpSX59H
91umFCSKglw7FCTb9p9gOKK8d1dmi1C09DBagTHszZ2K9j
d6vav9f_BSL7ad4LLEkAmQ4KLTB5H+QoQpQWawW8K67U5
BxPe1asa1dpML8uFuAmM4CgYEA557m7Xa2DqB9VWU+zeEPK+
Thmz04Vp4hTYT4uZNG6g+gK3aR0876969eDMHJEU4MH8QJab
87L7ew462uRAd9pmuq+RCXN7432TH2JXvBRROG+Q7eq9efcy
148RQK9pQCLY5+uacj9VahOTQL23SkKc1oMC68V44U0Gqj
kN7uonQ2+u0dDmZB8Z7RK7KRCME4UDUG+G+5713V3qPw
sMXoaANZ4bJHJHCLGgpk8590a0N1pVzascyV4+zKODymf1A=
-----END RSA PRIVATE KEY-----
```

Guarda esta clave con cuidado, no se mostrará de nuevo.

No hay imágenes subidas todavía.

[Cerrar sesión](#)


Numero de usuario del receptor:

Mensaje:

Sube una imagen:

Ingresa tu clave privada (info):

Figura 2: El usuario recibe su clave privada (lado izquierdo)

- **Envío de mensaje:** si el usuario presiona el botón de refrescar el *home* () , verá que además del mensaje de bienvenida tiene algunas directrices sobre que es y como utilizar la aplicación web, además de que ya no puede visualizar su clave privada.

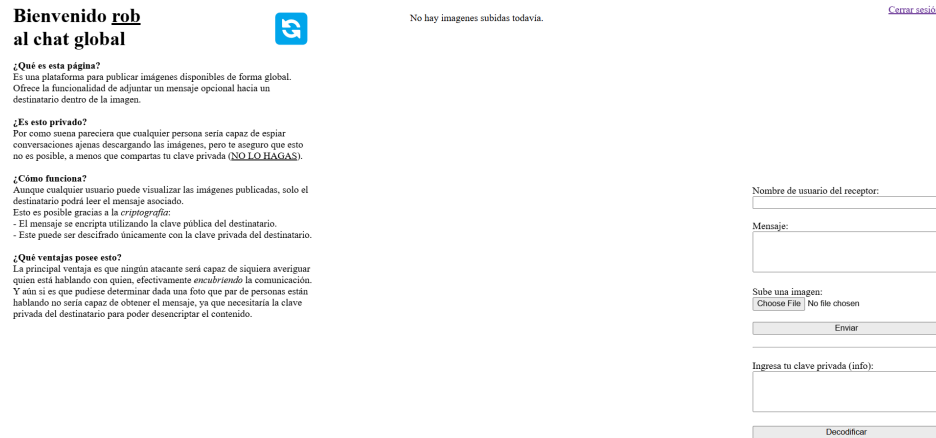


Figura 3: Página principal de la aplicación

Entonces, en el formulario del extremo derecho podrá enviar un mensaje a otro usuario, adjuntando una imagen para que el cifrado respectivo se oculte en esta. En el ejemplo, el usuario 'rob' quiere enviarle un mensaje a su amigo 'caldecrack' que ya se encuentra registrado previamente. Es importante notar que el formulario superior corresponde a la subida de imágenes, y el inferior al descifrado que se explicará más adelante.

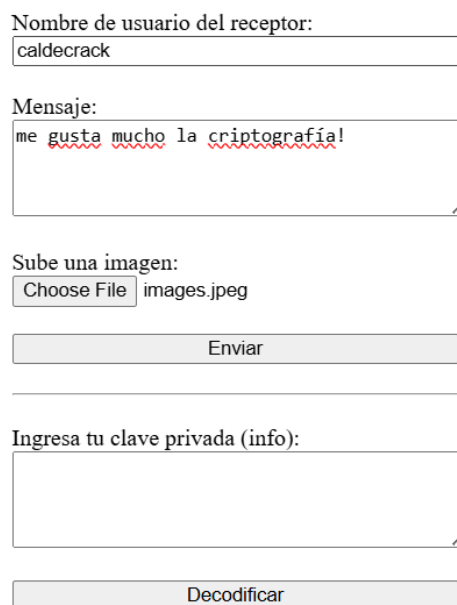


Figura 4: Formulario de envío de un mensaje (parte superior)

Una vez respondido el formulario, se encripta y oculta el mensaje en la imagen para subirla y esta será visible para todo usuario que visite la página.

En caso de que un adversario vea las imágenes en busca de intervenir/espiar una comunicación establecida entre usuarios, tan solo verá las imágenes del tablero, y no tendría como determinar la existencia de algún mensaje cifrado o siquiera de si hubo alguna comunicación, pues en principio no tiene acceso a las imágenes originales utilizadas antes de ser alteradas por el encubrimiento.

Bienvenido rob al chat global



Publicado el: Dec. 9, 2024, 7:15 p.m.

[Cerrar sesión](#)

¿Qué es esta página?

Es una plataforma para publicar imágenes disponibles de forma global. Ofrece la funcionalidad de adjuntar un mensaje opcional hacia un destinatario dentro de la imagen.

¿Es esto privado?

Por como suena pareciera que cualquier persona sería capaz de espiar conversaciones ajenas descargando las imágenes, pero te aseguro que esto no es posible, a menos que compartas tu clave privada (**SOLO HAGAS**).

¿Cómo funciona?

Aunque cualquier usuario puede visualizar las imágenes publicadas, solo el destinatario podrá leer el mensaje asociado. Esto es posible gracias a la *criptografía*.
- El mensaje se encripta utilizando la clave pública del destinatario.
- Este puede ser descifrado únicamente con la clave privada del destinatario.

¿Qué ventajas posee esto?

La principal ventaja es que ningún atacante será capaz de siquiera averiguar quien está hablando con quien, efectivamente *encubriendo* la comunicación. Y aún si es que pudiese determinar dada una foto que par de personas están hablando no sería capaz de obtener el mensaje, ya que necesitaría la clave privada del destinatario para poder descifrar el contenido.

Nombre de usuario del receptor:

Mensaje:

Sube una imagen:

[Choose File](#) No file chosen

[Enviar](#)

Figura 5: Es un Kirby!

- **Descifrado de mensajes:** supongamos ahora que ‘caldecrack’ se percata de que una imagen de Kirby ha sido subida al tablero, y sospecha que se trata de un mensaje de su amigo ‘rob’. Para descifrar este, basta que una vez iniciada su sesión responda el formulario de la esquina inferior derecha con su clave privada.

Nombre de usuario del receptor:

Mensaje:

Sube una imagen:

[Choose File](#) No file chosen

[Enviar](#)

Ingresa tu clave privada (info):

```
-----BEGIN RSA PRIVATE KEY-----
MIEpAIBAAKCAQEAprSIRQwIHgIopGHm7fL9Qw
3/eeV4MLSfkc7ZND9Le2584y
MY7YQI7JfF15aB/Sx5UacA0CIPc7t1V9einni77
```

[Decodificar](#)

Figura 6: Ingreso de clave privada para descifrar mensajes (parte inferior)

Una vez enviado este formulario, en la página se decodificarán todos los mensajes enviados al usuario asociado a la clave privada enviada. En este caso, ‘caldecrack’ verá desaparecer la imagen de Kirby, y en su lugar aparecerá el mensaje que le enviaron.

De rob:
me gusta mucho la criptografía!

Publicado el: Dec. 9, 2024, 7:15 p.m.

Figura 7: Mensaje decodificado

- **No interferencia:** si ‘caldecrack’ responde a ‘rob’ con otro mensaje, ahora aparecerán dos imágenes para cualquier usuario que revise el tablero.

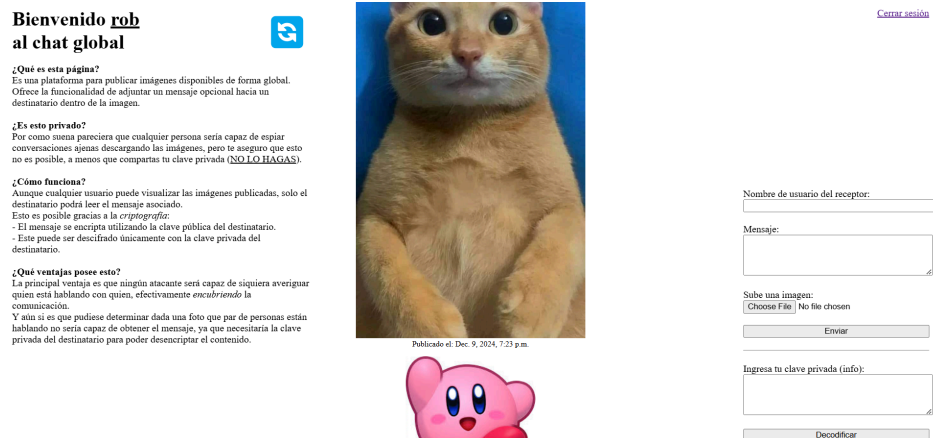


Figura 8: Tablero con dos imágenes

Cuando Rob ingrese su clave privada para revisar el mensaje que le envió su amigo, solo se decodificará este, dejando los mensajes enviados a otros usuarios aún cifrados y encubiertos en las imágenes respectivas.

De caldecrack:
a mí también, ojalá aprobar el curso

Publicado el: Dec. 9, 2024, 7:23 p.m.



Publicado el: Dec. 9, 2024, 7:15 p.m.

Figura 9: 'rob' observa el mensaje que recibió, pero no el que envió

Cabe destacar además que si un usuario subiese imágenes sin destinatario ni mensaje, esta simplemente se publicará en el tablero y nunca podrá ser descifrada. Esto asegura que un adversario no podrá distinguir entre aquellas imágenes que sí contienen algún mensaje oculto de aquellas que no, pues todas se subirán igualmente al tablero y se comportarán de la misma forma que una imagen que encubre algún mensaje pero cuya clave privada asociada al destinatario respectivo no ha sido ingresada.

Un pequeño detalle es que como se mencionó anteriormente, en el modelo *Image* se almacena el largo del mensaje secreto en bits, por lo que para este caso se genera un número aleatorio que sea múltiplo de 8 para que tampoco sea posible diferenciarlas de este modo.



Publicado el: Dec. 9, 2024, 7:32 p.m.

De rob:
a mí también, ojalá aprobar el curso

Publicado el: Dec. 9, 2024, 7:23 p.m.



Figura 10: Una imagen sin mensaje oculto

IV. Conclusión

Se logró implementar un sistema de comunicación encubierta efectivo mediante una aplicación web desarrollada en *Django*. Esta asegura que solo los usuarios poseedores de sus claves privadas podrán decodificar aquellos mensajes que reciban, y que para cualquier otro observador, sea otro usuario o un adversario intentando intervenir la comunicación, no se puedan distinguir las imágenes con mensajes cifrados ocultos de simples imágenes sin manipular, pues todas comparten el mismo espacio común del tablero.

El principal desafío de este proyecto fue el diseño del sistema de comunicación encubierta, pues si bien dentro de los primeros días se consiguió elaborar un *script* de *Python* que simulaba un flujo de emisión y recepción de una imagen con un mensaje oculto, el tener que escalar este para implementar una aplicación completamente funcional y utilizable por varios usuarios fue difícil, en particular, no se tenía claro cómo hacer uso de las claves, lo cual en el caso del *script* original era trivial pues solo se trataba de generarla una vez y utilizarla en la misma ejecución, pero en caso de la aplicación esta se generaría cada vez que se utilice, o se guardaría en la base de datos del sistema, lo cual puede no ser tan seguro.

Finalmente se optó por el enfoque descrito para solucionar el problema, asociando a cada usuario con su par de claves pública y privada. La idea del tablero se ajustó a este enfoque, y aseguró además la indistinguibilidad a ojos de un adversario entre imágenes sin manipular y aquellas que ocultan mensajes. Una posible mejora para este proyecto sería implementar un enfoque que

no dependa de que el usuario tenga que guardar su clave privada, pues en caso de perderla, este no podrá volver a acceder a los mensajes que le fueron enviados en el pasado, ni a los que se le pudiesen enviar en un futuro. Similarmente a esto, también se podría implementar un sistema de generación de claves pública y privada nuevas, pero esto presenta desafíos como si en ese caso se debiese descartar la clave pública anterior o utilizar la nueva en conjunto con las anteriores, de modo que requeriría realizar una investigación de si esto preserva seguridad y confidencialidad o no.

Además, en caso de masificarse el uso de la aplicación, podría implementarse un sistema de paginación y búsqueda por fechas de las imágenes en el tablero para que los usuarios no pierdan tiempo intentando descifrar una multitud de mensajes. Otra mejora podría adición de salas para distribuir la carga del chat global, un tema con esto sería investigar si el realizar esta división resultaría en una pérdida de confidencialidad, integridad y/o autenticación.