Trabajo Práctico de Implementación

Stegowav

Criptografía y seguridad

Grupo: ‘Hola mundo’

Integrantes:

- Nardini, Gonzalo Martín 54387
- Perez Cuñarro, Javier 49729
- Sakuda, María Eugenia 53191

Fecha de entrega: Lunes 27 de Junio del 2016

Desarrollo

1. Para la implementación del programa stegowav se pide que la ocultación comience en la primer muestra del archivo de audio. ¿Sería mejor empezar en otra ubicación? ¿Por qué?

Es una buena opción comenzar en la primer muestra del archivo de audio porque permite tener más espacio que si se comenzara en un punto intermedio. También podría ser posible escribir en los headers del archivo, pero habría que ser sumamente cuidadosos de dónde se escribe para no dañar la configuración y que aún siga siendo reproducible.

2. Esteganografiar un mismo archivo en un .wav con cada uno de los tres algoritmos, y comparar los resultados obtenidos. Hacer un cuadro comparativo de los tres algoritmos estableciendo ventajas y desventajas.

Para este punto se tomó como portador el archivo “wavfile.wav” incluido en la carpeta ejemplos_informe dentro de data y se ocultó un archivo .zip con un conjunto de imágenes. Las dimensión del archivo .wav original es de 1.245.228 bits y el comprimido varía entre 32.509 y 1.983.544 bits, dependiendo el método utilizado.

	LSB1	**LSB4**	**LSBE**
	test_lsb1_min.wav	test_lsb4_min_com.wav	test_lsbe.wav
Tamaño máximo que puede ocultar	622.592 bits	2.490.368 bits	35.121bits
Bits aprovechados	32.509 bits	32.509 bits	32.509 bits
Ruido	Se percibe especialmente en los silencios.	Bastante, acumulado al inicio, lo cual lo hace notorio por poco tiempo.	Está distribuído a lo largo de todo el archivo. Se percibe menos que en el LSB1
Porcentaje utilizado	5,22%	1,3%	92,56%

Tabla 1: tabla comparativa entre los métodos LSB1, LSB4 y LSBE para un mismo archivo oculto

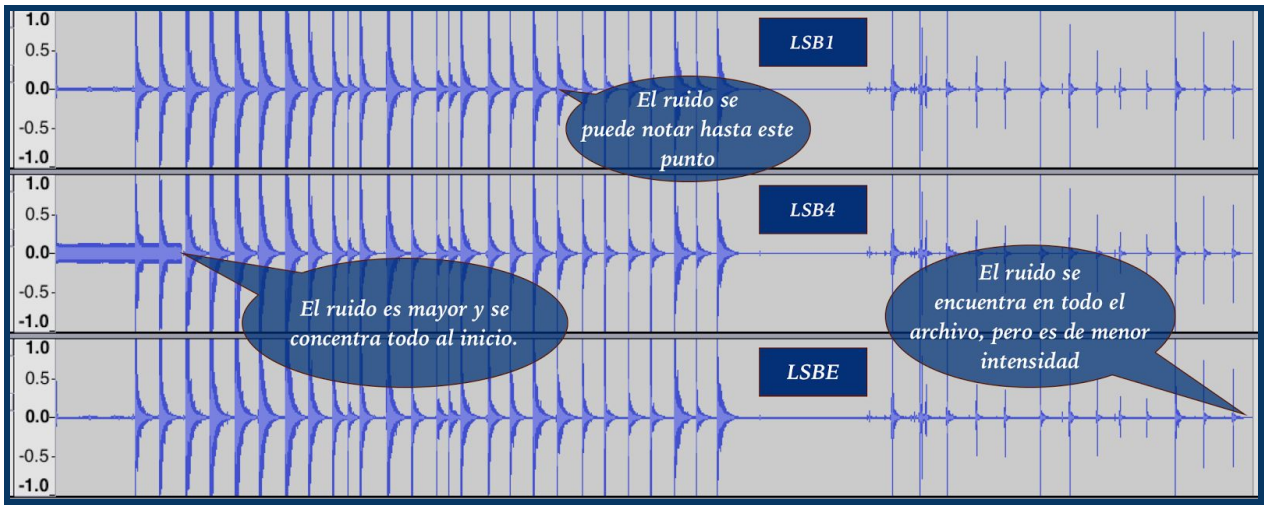

Imagen 1: Gráfico de las frecuencias para las tres muestras (LSB1, LSB4 y LSBE) ocultando 32.509 bits de información

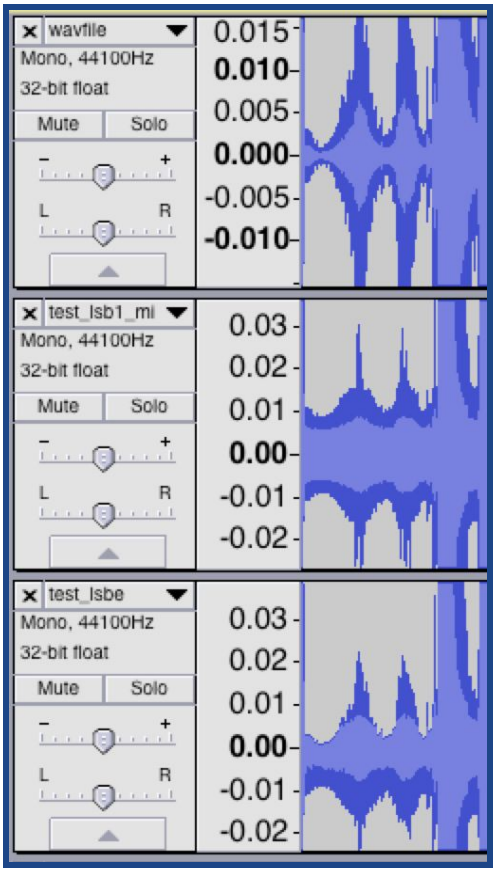

Gráfico 2: ampliación de los audios con LSB1 y LSBE junto al portador sin modificar.

Si bien a simple vista no se pueden observar las perturbaciones iniciales entre estos dos métodos y el archivo original, al realizar una ampliación se puede contemplar las diferencias en las frecuencias.

	LSB1		LSB4		LSBE	
Tamaño máximo que puede ocultar	622.592 bits		2.490.368 bits		35.121bits	
Bits aprovechados	40.397 bits	-	40.397 bits	1.983.544 bits	-	-
Ruido	Las perturbaciones no son grandes, pero se encuentran distribuídas hasta el final del archivo	-	Bastante pero acumulado al inicio.	Bastante ruido durante todo el archivo.	-	-
Porcentaje utilizado	6,4%	-	1,6%	79,64%	-	-

Tabla 3: comparación de los tres métodos con archivos de otros tamaños. (- en caso de que la información no pudiese ser contenida dentro del archivo)

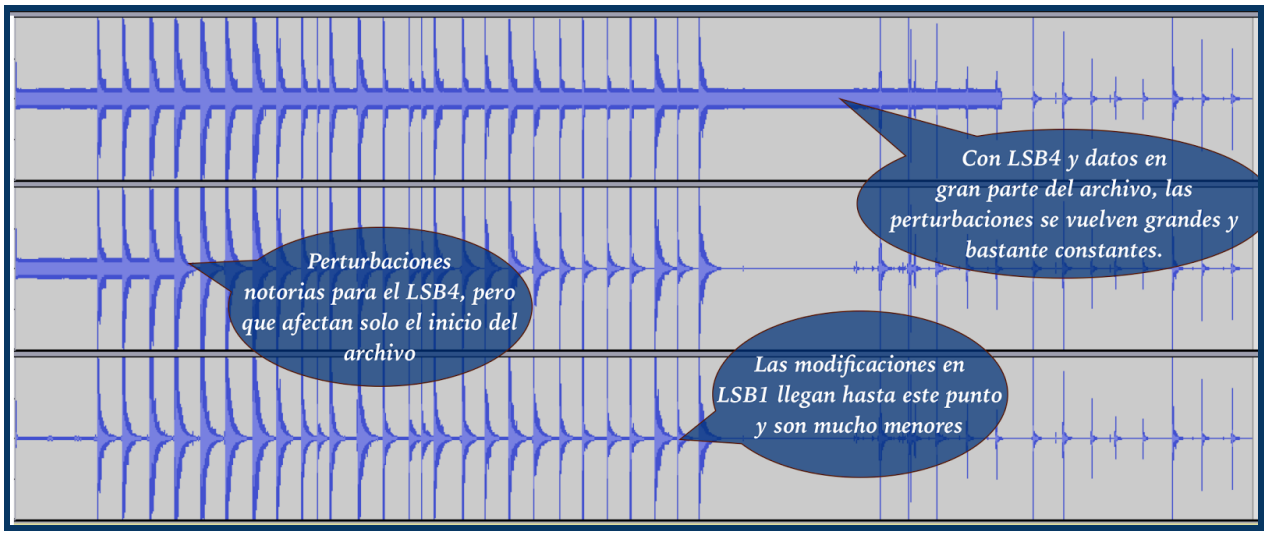

Imagen 3: Frecuencia del portador con información oculta para los métodos LSB1 y LSB4 y tamaños de 40.397 y 1.983.544 bits respectivamente.

3. Para la implementación del programa stegowav se pide que la extensión del archivo se oculte después del contenido completo del archivo. ¿por qué no conviene ponerla al comienzo, después del tamaño de archivo?

No conviene ponerla al comienzo por cuestiones de seguridad. Si la extensión del archivo aparece al principio (luego del tamaño del archivo) bastaría con realizar unas pocas pruebas para encontrar información legible (texto plano) verificando que el método seleccionado es el correcto. Una vez determinado el método resulta sencillo extraer el archivo completo. De todos modos, habiendo levantado el tamaño de la información escondida, y conociendo que se esconde la información con alguna variante de LSB, podría buscarse los dos samples posibles, para el LSB1 y el LSB4, e intentar levantar la extensión de ahí.

4. Explicar detalladamente el procedimiento realizado para descubrir qué se había ocultado en cada archivo y de qué modo.

Como sabíamos que 3 de los 4 archivos provistos por la cátedra usaban los métodos LSB1, LSB4 y LSBE y que solo uno de ellos estaba encriptado, primero intentamos obtener archivos ocultos en los 4 con todos los métodos posibles. Lo que obtuvimos fue una imagen de tipo png usando LSB4 en el archivo *vivalavida9a.wav* y un archivo pdf usando LSBE en *barcelona9b.wav*. El archivo pdf contenía instrucciones (explicadas con más detalle en el punto 6) que nos permitieron extraer de la imágen obtenida previamente la información de que, para obtener el archivo restante, había que desencriptar usando el algoritmo AES de 192 bits en modo CBC. Como el único método restante era LSB1, sabíamos también que teníamos que usarlo. La única información restante era la contraseña a utilizar, y como nos quedaban solo dos archivos supusimos que estaba en uno de ellos (y que del otro debíamos obtener información usando el resultado). Después de analizar la situación y hacer diferentes pruebas, decidimos abrir los archivos con un editor de texto y nos encontramos con que uno de ellos, *vivalavida.wav*, tenía la contraseña (*exitoso*) escrita al final. Con este descubrimiento logramos encontrar el archivo oculto en *barcelona9a.wav*, que era un fragmento de video.

5. ¿Qué se encontró en cada archivo?

vivalavida9a.wav	Archivo png con un juego de buscaminas.
vivalavida.wav	Un texto que decía “la password es exitoso”.
barcelona9b.wav	Un pdf con instrucciones. Proponía cambiar la extensión del archivo png a zip.
barcelona9a.wav	Una porción de un video .wmv que contenía una escena de la serie *Bones* de *Fox*.

6. Algunos mensajes ocultos tenían, a su vez, otros mensajes ocultos. Indica cuál era ese mensaje y cómo se había ocultado.

El archivo de audio *vivalavida9a.wav* contenía oculto usando el método LSB4 un archivo png con un juego de buscaminas. En principio no parecía muy útil, pero luego una pista nos dijo que le cambiemos la extensión a zip. Al hacerlo y descomprimir el archivo, nos encontramos con un archivo de texto (.txt) que contenía instrucciones de cómo leer la imágen para obtener resultados útiles. Al seguir las instrucciones logramos deducir que la forma de obtener información del archivo de audio que nos faltaba era usando el algoritmo de desencriptación AES de 192 bits con modo CBC. Con esto, sumado a lo descubierto en otro archivo (que la contraseña era *exitoso*) y que sabíamos de antemano que habían 3 archivos con información oculta usando LSB1, LSB4 y LSBE y solo nos restaba usar LSB1 logramos obtener el fragmento de video oculto.

7. Uno de los archivos ocultos era una porción de un video, donde se ve ejemplificado una manera de ocultar información ¿cuál fue el portador?

La manera de ocultar información fue esconder la porción del video en un archivo de audio .wav. La información estaba encriptada usando una contraseña y el modo de el algoritmo de encriptación AES de 192 bits en modo CBC.

8. ¿De qué se trató el método de estenografiado que no era LSB? ¿Es un método eficaz? ¿Por qué?

El método restante se trató de simplemente esconder el binario correspondiente al texto plano al final de la “data” del archivo .wav. En general, este método no parece ser muy eficaz ya que cualquier persona puede leerlo sin mucho esfuerzo al abrir el archivo con un editor de texto. Sin embargo, en este caso en particular, como estábamos concentrados en buscar formas más complicadas de encontrar información no se nos ocurrió probar con ese método hasta haber agotado todas las otras opciones de métodos LSB.

9. ¿Qué mejoras o futuras extensiones harías al programa stegowav?

Desde la sección de encriptación del trabajo, se le puede agregar SALT para el cálculo de la contraseña correspondiente para cada método. De esta forma, la contraseña que se utiliza en los métodos de encriptación se calculan en función de un ‘Or’ entre la contraseña del usuario y el salt, agregándole un grado de dificultad más.

Por otro lado, se está implementando sólo uno de los métodos de estenografía conocidos para ocultar información en audio. Si bien se están utilizando 3 variantes del mismo, que pueden ser mejores o peores según el fin con el que se desee utilizar, existen otros métodos ya conocidos que poseen otras características. Algunas de las posibles opciones a desarrollar son:

- **Codificación por paridad:** dispone de más bits para ocultar información que los métodos LSB. Como contra, sigue existiendo la presencia de ruido, aunque en menor cantidad así como también la posible pérdida de información en el caso de que se efectúe un conversión en el rango de la muestra.
- **Codificación por fase:** es más efectivo para que el ruido no sea perceptible para el oído humano, dado que manda la información como cambios de fase en la onda sonora. Esto trae como desventaja aparejada la poca capacidad para ocultar información (sólo en la muestra inicial).
- **Espectro ensanchado (SS) y manejo de eco:** junto con la codificación por fase tienen la característica de ser más robustos. Además, se caracterizan por permitir “esconder” mayor cantidad de información.

Documentación del funcionamiento

Compilación

Para poder compilar en linux, es necesario de tener instaladas las siguientes librerías de openssl. En caso de que no sea así, deberá instalarlas primero.

- openssl - apt-get install openssl
- libssl-dev - apt-get install libssl-dev

Luego sólo tiene que ejecutar el makefile desde la carpeta crypto-tpe.

Ejecución

Para ejecutar el programa, sólo debe correr stegowav con los siguiente parámetros según lo que desee realizar:

Para embeber información:

- **-embed**
- **-in file**
  Es el path del archivo que se quiere ocultar. El mismo debe empezar con /
- **-out file.wav**
  Es el path del archivo que se va a generar. El mismo debe empezar con / y tener como extensión .wav
- **-steg<LSB1|LSB4|LSBE>**
  Es el método que se utilizará para esconder la información. Se puede elegir entre LSB1, LSB4 y LSBE
- **-p file.wav**
  Es el path del archivo donde se va a ocultar la información El mismo debe empezar con / y ser un archivo de audio .wav

<u>Parámetros opcionales</u>

- **-pass contraseña**
  Es un parámetro necesario si se desea que la información escondida se encuentre además encriptada.
- **-a <aes128|aes192|aes256|des>**
  El default es aes128
- **-m <ecb|cfb|ofb|cbc>**
  El default es cbc

Para extraer información:

- **-extract**
- **-out file.wav**
  Es el path del archivo que se va a generar. El mismo debe empezar con / y tener como extensión .wav
- **-steg<LSB1|LSB4|LSBE>**
  Es el método que se utilizará para extraer la información. Se puede elegir entre LSB1, LSB4 y LSBE
- **-p file.wav**
  Es el path del archivo donde se va a ocultar la información El mismo debe empezar con / y ser un archivo de audio .wav

Parámetros opcionales

- **-pass contraseña**
  Es un parámetro necesario si se debe desencriptar la información extraída
- **-a <aes128|aes192|aes256|des>**
  El default es aes128
- **-m <ecb|cfb|ofb|cbc>**
  El default es cbc