

UNIVERSIDAD NACIONAL DEL LITORAL

FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS



CAPTURA Y PROCESAMIENTO DIGITAL DE SEÑALES E IMÁGENES

Encriptación y Desencriptación de Imágenes por el Método Xor.

31 de Octubre de 2004

Integrantes

Langhi, Sebastián

Machtey, Damián

Ronsoni, Alejandro

Encriptación y desencriptación de imágenes por el método xor.

Resumen

Presentamos un algoritmo que permite la encriptación y desencriptación de imágenes mediante la generación de una imagen aleatoria del mismo tamaño que la original, luego se aplica la función lógica xor entre ambas para obtener la imagen encriptada. Mediante la misma operación entre imagen encriptada y la aleatoria se logra la desencriptación.

Introducción

La Criptología es la ciencia que estudia los distintos sistemas de cifrado destinados a ocultar el significado de mensajes a otras partes que no sean el emisor y el receptor de dicha información o tenerlos cubierto de miradas indiscretas. La encriptación es una herramienta que seguramente va a ver incrementado su nivel de utilización, a medida que el comercio electrónico continúe su expansión. Muchos servicios digitales, como televisión paga, vide conferencia, sistemas de imágenes médicas y militares, requieren de un sistema confiable de seguridad en la transmisión y almacenamiento digital de imágenes y videos. Con el progreso de Internet en el mundo actual, la seguridad de imágenes y videos digitales se ha convertido en algo cada ves mas importante. Para satisfacer la seguridad y privacidad mencionadas en varias aplicaciones, la encriptación de imágenes es muy importante para frustrar ataques mal intencionados de personas no autorizadas. Desde principios de 1990, se han realizado numerosos esfuerzos de investigación para solucionar los problemas de encriptación de imágenes.

En general el proceso de encriptación consiste en aplicar una transformación lineal $E=T(I, \text{clave})$; donde I es la imagen original, T es una transformación lineal dependiente en general de alguna clave y E es la imagen resultante encriptada. El proceso inverso se logra con una transformación inversa.



Encriptación y desencriptación de imágenes por el método xor.

Se pueden distinguir dos métodos de encriptación: por bloques y por unidad. En el primer caso la imagen es encriptada por bloque en otro del mismo tamaño. Los encriptadores por unidad basados en secuencias psuedo-aleatorias son controlados por una clave de encriptación. Los generadores de secuencias psuedo-aleatorias se basan en fórmulas matemáticas recurrentes donde actual depende del valor anterior. El primer valor tomado para comenzar la secuencia se denomina semilla y este puede ser generado tomando, por ejemplo, la hora actual o inicializarlo con un valor específico y esta última la propiedad que utilizan los encriptadores.

Materiales y métodos

El algoritmo utilizado se basa en la función lógica xor (ver Tabla I). Nuestro algoritmo toma una imagen y genera otra del mismo tamaño en forma aleatoria utilizando la propiedad de los generadores pseudos aleatorios de generar la misma secuencia a partir de determinada semilla dada. Esta semilla se calcula siempre mediante el mismo método, partiendo de una palabra clave (ver código I). Para ello buscamos que cualquier secuencia de caracteres de hasta una longitud determinada, genere una semilla que no pueda ser igual que la generada por otra secuencia.

Tabla I

| A | B | XOR |
|----------|----------|------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

```
shift1=0;
shift2=0;
value=13;
for i = 1:length(password)
    a=sprintf('%d',password(i));
    ch = str2num(a);
```

Encriptación y desencriptación de imágenes por el método xor.

```
value = bitxor(uint64(value),uint64(ch * 2 ^ shift1));  
value = bitxor(uint64(value),uint64(ch * 2 ^ shift2));  
shift1 = mod((shift1 + 7),19);  
shift2 = mod((shift2 + 13),23);  
end  
semilla= double(value);
```

CÓDIGO I

Una vez calculada la semilla se inicializa el generador con esta y se genera la imagen aleatoria (ver código II)

```
rand('seed',semilla);  
R=uint8(fix(rand(size(i))*256));
```

CÓDIGO II

Luego se procede a codificar la imagen deseada aplicando la función xor entre esta y la aleatoria. (ver código III)

```
Encriptada=bitxor(i,R);
```

CÓDIGO III

El algoritmo de desencriptación es el mismo que fue descrito anteriormente

Resultados

En la figura I se observa un ejemplo de una imagen sin encriptar y posteriormente encriptada utilizando el algoritmo propuesto.

Aquí se ve que es imposible “descifrar” cual es la imagen original, sin haberla desencriptado previamente.

Encriptación y desencriptación de imágenes por el método xor.

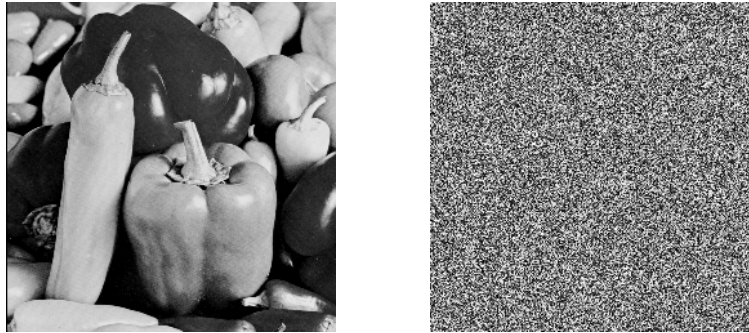


FIGURA I

Conclusiones

El algoritmo de encriptación presentado tiene la ventaja de ser muy simple su implementación. Respecto a la seguridad depende fundamentalmente de dos factores: en primer lugar la factibilidad de repetir el generador pseudo-aleatorio y en segundo lugar la capacidad de descubrir la semilla. Por ello sería importante que sólo el emisor y receptor conozcan tanto la semilla como el código del generador en la medida de lo posible. Cabe acotar que aún conociendo el código del generador utilizado es prácticamente imposible detectar cual fue la semilla inicializadora de la secuencia por el método de fuerza bruta o diccionario, ya que puede generar $2^{31} - 2$ secuencias diferentes.

Bibliografía consultada

- http://webs.ono.com/usr016/agika/6temas_relacionados/encriptar.htm
- Handbook of applied Cryptography. ISBN 0849385237