Diseño de un sistema encriptador para mensajes telefónicos

René Rodríguez-Ávila, Alfonso Gutiérrez.

Instituto Politécnico Nacional, Centro de Investigación en Computación
Av. Juan de Dios Batíz s/n, casi Esq. Miguel Othón de Mendizábal,
Unidad Profesional "Adolfo López Mateos", Edificio CIC.
Col. Nueva Industrial Vallejo 07738, México D.F.
citvrene@hotmail.com

Resumen. Este sistema pretende proteger conversaciones realizadas a través de líneas telefónicas (señales analógicas de hasta 4 KHz de frecuencia) a partir de la implementación de un algoritmo de encriptación/descencriptación RSA en un DSP; Primeramente la señal entra al interfase de comunicación, pasa por un filtro antialias con una frecuencia de corte de 3.5 KHz, después se convierte de analógico a digital con un DAC de 8 bits después se procesa por el DSP de la 5ª generación de Texas Instruments TMS32050, conectado a un módem de 56 Kbps para realizar la llamada al receptor, la señal se hace pasar por un filtro de reconstrucción con una frecuencia de corte de 3.5 KHz para obtener la señal original y sea escuchada por el receptor.

1 Introducción

La Criptografía según el Diccionario de la Real Academia, proviene del griego kriptos, que significa oculto, y grafos, que significa escritura, y su definición es: "Arte de escribir con clave secreta o de un modo enigmático".

Hace años que la Criptografía dejó de ser un arte para convertirse en una técnica, o más bien un conglomerado de técnicas, que tratan sobre la protección u ocultamiento frente a observadores no autorizados de la información. Entre las disciplinas que engloba cabe destacar la Teoría de la Información, la Teoría de Números o Matemática Discreta, que estudia las propiedades de los números enteros, y la Complejidad Algorítmica [1].

Definiremos un criptosistema como una quíntupla (M, C, K, E, D), donde:

- M representa el conjunto de todos los mensajes sin cifrar que pueden ser enviados (lo que se denomina texto claro, o plaintext).
- C representa el conjunto de todos los posibles mensajes cifrados, o criptogramas.
- K representa el conjunto de claves que se pueden emplear en el criptosistema.

- E es el conjunto de transformaciones de cifrado o familia de funciones que aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente Ek para cada valor posible de la clave k.
- D es el conjunto de transformaciones de descifrado, análogo a E.

Todo criptosistema ha de cumplir la siguiente condición:

$$Dk\ (Ek\ (m)) = m,$$

es decir, que si tenemos un mensaje m, lo ciframos empleando la clave k y luego desciframos empleando la misma clave, obtenemos de nuevo el mensaje original m.

Existen dos tipos fundamentales de criptosistemas:

Criptosistemas simétricos o de clave privada. Son aquellos que emplean la misma clave k tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser empleados en comunicaciones la clave k debe estar tanto en el emisor como en receptor, lo cual nos lleva preguntarnos cómo transmitir la clave de forma segura.

Criptosistemas asimétricos o de llave pública, emplean una doble clave (kp, kP) donde kp se conoce como clave privada y kP se conoce como clave pública. Una de ellas sirve para la transformación E de cifrado y la otra para la transformación D de descifrado. En muchos casos son intercambiables, esto es, si empleamos una para cifrar la otra sirve para descifrar y viceversa, y en general se basan en plantear atacante problemas matemáticos difíciles de resolver. En la práctica muy pocos algoritmos son realmente útiles. El más popular por su sencillez es RSA, que ha sobrevivido a multitud de ataques [2].

El Algoritmo RSA de entre todos los algoritmos asimétricos, quizá sea el más sencillo de comprender e implementar, sus claves sirven tanto para codificar como para autentificar. Debe su nombre a sus tres inventores: Ronald Rivest, Adi Shamir Leonard Adleman, y estuvo bajo patente de los Laboratorios RSA hasta el 20 de septiembre de 2000. Sujeto a múltiples controversias, desde su nacimiento nadie ha conseguido probar o rebatir su seguridad, pero se le tiene como uno de los algoritmos asimétricos más seguros.

RSA se basa en la Teoría de Números, que entre otras técnicas, estudia la dificultad para factorizar números grandes. Las claves pública y privada se calculan a partir un número que se obtiene como producto de dos primos grandes. El atacante enfrentará, si quiere recuperar un texto claro a partir del criptograma y la llave pública, a un problema de factorización[3].

Para poder implementar este algoritmo nos remontamos a la Teoría o Aritmética Modular.

La *Aritmética Modular* es una parte de las matemáticas extremadamente útil criptografía, ya que permite realizar cálculos complejos y plantear problemas

interesantes, manteniendo siempre una representación numérica compacta y definida, puesto que sólo maneja un conjunto finito de números enteros [4].

El *Algoritmo de Euclides* que es utilizado para desarrollar el algoritmo RSA permite obtener de forma eficiente el máximo común divisor de dos números [5] y se basa en el Teorema de él mismo que dice:

Teorema de Euclides. Si un número primo divide a un producto, divide a uno de los factores[6].

El *Algoritmo Extendido de Euclides* también puede ser empleado para calcular inversas las cuales se requieren para elaborar el algoritmo de encriptación y es una ampliación del de Euclides que posee el mismo orden de complejidad, y que se obtiene simplemente al tener en cuenta los cocientes además de los restos en cada paso [7].

El *Algoritmo de Exponenciación Rápida* se utiliza para calcular el módulo de dos números enteros largos, y es una herramienta de la aritmética modular muy poderosa ya que si tuviéramos que evaluar el módulo de dos enteros largos usando el método convencional que es multiplicar n veces el número al que se desea obtener el módulo, implicaría realizar demasiadas operaciones, además de obtener números que quizá ni una computadora podría soportar, por tanto este algoritmo simplifica dichas operaciones [8].

A continuación se muestra el algoritmo de la exponención rápida:

```
int exp_rapida(int a, int b) {
int z,x,resul;
z=b;
x=a;
resul=1;
while (z>0) {
    if (z%2==1)
          resul=resul*x; }
x=x*x;
z=z/2;
}
```

Un ejemplo de cómo aplicar el algoritmo es el siguiente:

Calcular el valor de 210368 mod 187.

Tabla 1. Cálculos realizados por el algoritmo para calcular 210368 mod 187.

No. de iteraciones	resultado	z	x
1	r = 1	z = 10368	x = 2
2	r = 1	z = 5184	x = 4
3	r = 1	z = 2592	x = 8
4	r = 1	z = 1296	x = 16
5	r = 1	z = 648	x = 32
6	r = 1	z = 324	x = 64
7	r = 1	z = 162	x = 128
8	r = 1	z = 81	x = 256 (mod 187) = 69
9	r = 69	z = 40	x = 4761 (mod 187) = 86
10	r = 69	z = 20	x = 7396 (mod 187) = 103
11	r = 69	z = 10	x = 10609 (mod 187) = 137
12	r = 69	z = 5	x = 18769 (mod 187) = 69
13	r = 4761 (mod 187) = 86	z = 2	x = 7396 (mod 187) = 86
14	r = 86	z = 1	x = 10609 (mod 187) = 103
15	r = 8858 (mod 187) = 69		

El sistema consta de 5 partes: interfase comunicación que consta de un micrófono una bocina por la que el usuario mandara el mensaje y escuchara la respuesta, la etapa de filtrado para el evitar el efecto aliansing y para reconstruir la señal analógica, el DAC que es la etapa de conversión analógica a digital, un Procesador Digital de Señales (Digital Signal Processor o DSP) que hará el procesamiento de la señal aplicando el algoritmo de encriptación RSA, y por último el módem el cual efectuara la llamada telefónica y hará el acoplamiento hacia la línea telefónica, el sistema se muestra en la figura 1.

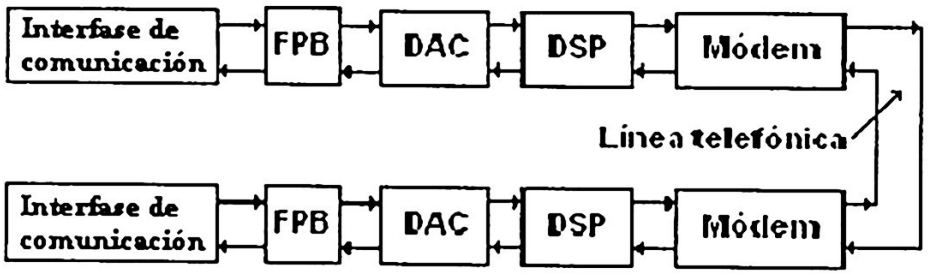

Fig. 1. Diagrama del sistema transmisor (encriptador)/receptor (descencriptador).

2 Metodología

En sustitución de la señal de voz (o telefónica) se utilizaron archivos de audio tipo WAV, con el uso de Matlab hemos pasado los datos de estos archivos de audio muestreados a 8 KHz, con 8 bits, monoaural con una duración de 60 s a archivos de tipo texto para probar el algoritmo de encriptación y reconstrucción de la señal (descencriptación).

Basados en la investigación anterior hemos desarrollado e implementado este algoritmo en un leguaje de programación de alto nivel "Delphi" ver. 4.0 para aplicar el sistema de criptografía RSA a los valores de la señal, el programa desarrollado permite introducir un archivo de texto con los datos de la señal que se desean encriptar o descencriptar, estos datos están en forma fraccionaria por lo que se trataron como datos tipo flotante.

El programa incluye una sección de acondicionamiento para la señal ya que el algoritmo utiliza enteros y nuestra información es de tipo fraccionario.

El acondicionamiento consistió en:

1. Obtener una señal positiva, esto debido a que la señal esta en el orden de ± 1Volt, se le sumo un 1 (1 Volts) teniendo ahora una señal de 0 a 2 Volts,
2. Convertir los datos fraccionarios a enteros, esto se hizo simulando la conversión digital; es decir; dependiendo el número de bits seleccionados para la conversión sé multiplico por la mitad de 2^{bits}, esto es similar a proceso de cuantificación del convertidor pues al realizar esta operación se les asigno valores dentro de los niveles de cuantificación que van de 0 a 2^{bits}. Por ejemplo: para 8 bits se tiene 256 niveles de cuantificación.

Prácticamente esto sería equivalente a tener una señal y procesarla por el DSP teniendo al final un archivo encriptado.

Para seleccionar el DAC óptimo requerimos de hacer pruebas con el programa para verificar la calidad de la señal reconstruida y que esta fuera comprendida; esta información se obtuvo usando nuevamente Matlab para convertir el archivo de tipo texto descencriptado a un archivo de audio tipo WAV y escuchar su contenido.

Posteriormente se seleccionó el DSP dependiendo del número de operaciones requeridas por el algoritmo RSA; otro factor a considerar es el tipo de operaciones que realiza, es decir, si son de punto fijo o flotante, como ya dijimos el algoritmo utiliza enteros por lo que queda claro que la familia del DSP debía ser de punto fijo.

3 Resultados

Al ejecutar el programa en Delphi, es decir, aplicar el algoritmo RSA a la información del archivo obtenemos como resultado un archivo de salida con los datos procesados o encriptados.

El programa RSA se muestra en la figura 3 que se muestra a continuación:

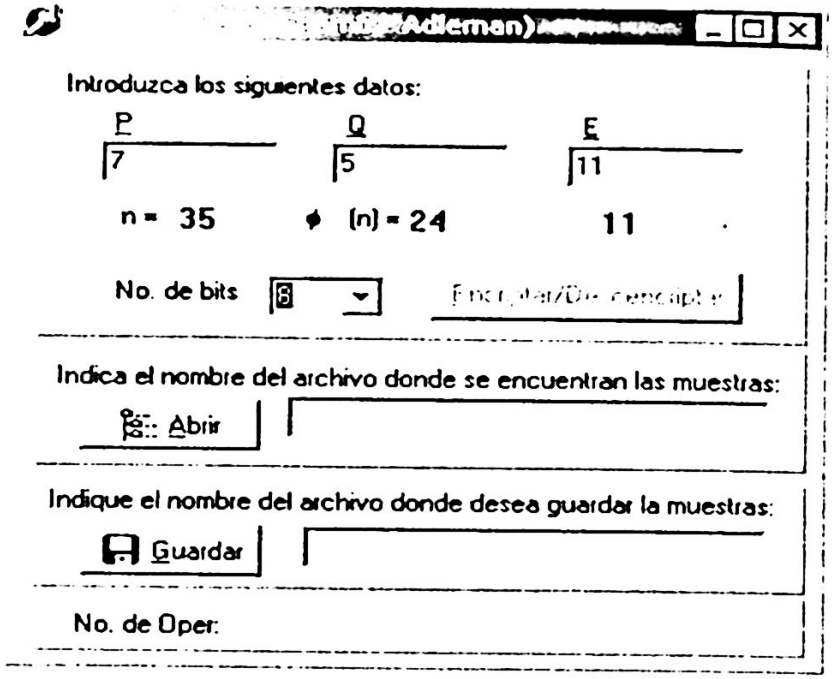

Fig. 3. Programa en Delphi para encriptar/descerncriptar archivos por medio del algoritmo RSA.

Los datos requeridos por el programa para realizar una encriptación son: P = primer número primo, Q =segundo numero primo, E = clave pública y el número de bits para la conversión digital; Por último se indica el archivo fuente que es donde se tienen los datos del contenido la señal y la dirección de la carpeta donde se guardará el resultado de la encriptación.

Debido a que el algoritmo sirve tanto para encriptar como para descencriptar, podemos después de encriptar la información introducir el archivo encriptado a entrada del programa y aplicar el algoritmo nuevamente para obtener la señal original.

Se hizo la prueba con el archivo de texto obtenido con Matlab con la información del archivo WAV que tenía una duración 60 s.

La gráfica de 500 muestras de la señal original y reconstruida se muestra en la figura

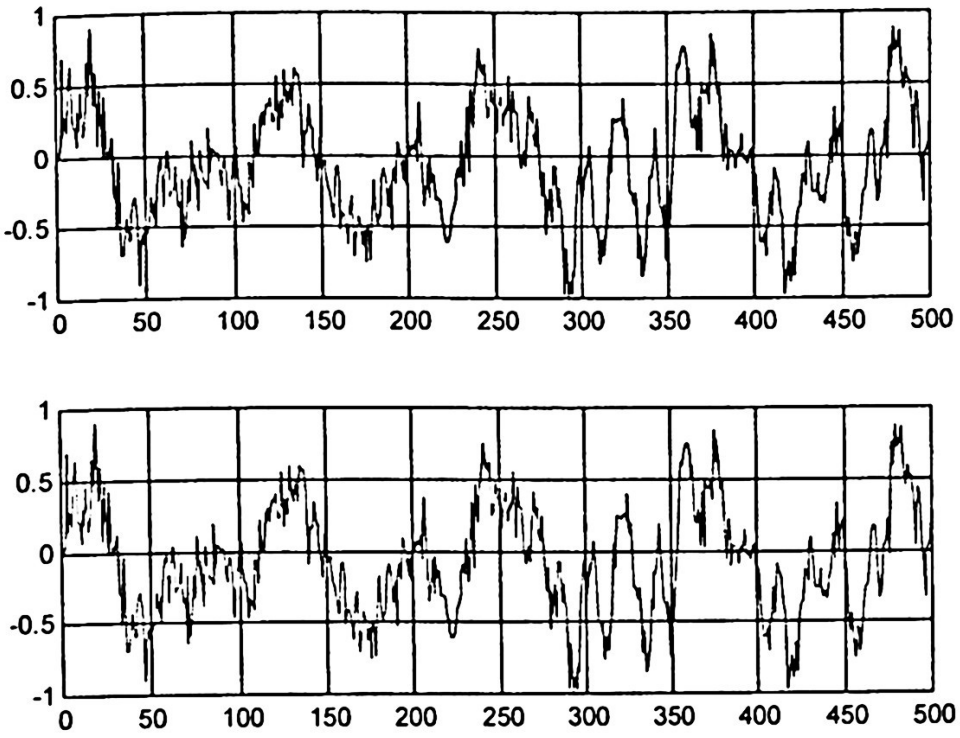

Fig. 4. Gráfica de la señal original y reconstruida.

Señal original.

Señal reconstruida.

La gráfica del error producido por el redondeo en las 500 muestras se observa en la figura 5.

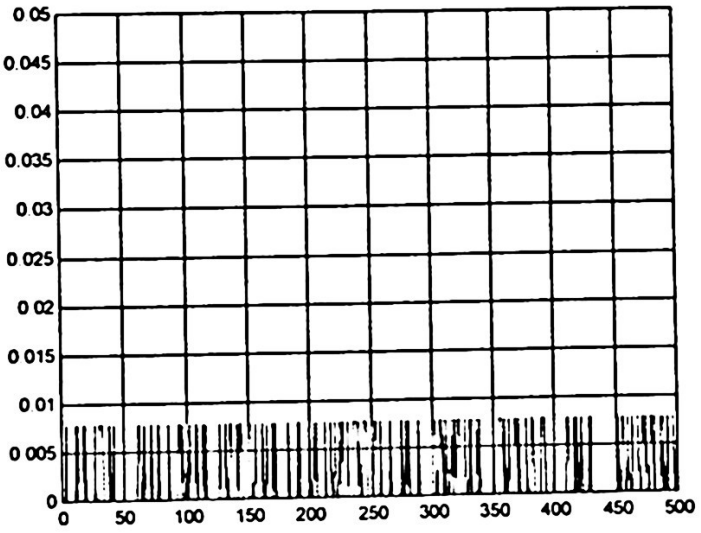

Fig. 5. Gráfica de las muestras con error de redondeo.

En las 500 muestras se presentaron 196 errores, esto quiere decir que 39.2% de los datos tuvieron errores de redondeo en promedio, este error es de 0.007843 que con respecto a la señal es poco significante respecto a sus valores reales.

La calidad de la señal reconstruida con los DAC's de 8 y 16 bits para un archivo de audio de 60 s de duración es casi la misma, es decir, la señal reconstruida en la que se uso el DAC de 8 bits y muy similar a la obtenida cuando se uso el DAC de 16 bits, de hecho, la diferencia es poco perceptible por lo que bastaría con utilizar un DAC de bits.

El hecho de usar un DAC de 16 bits reduciría el ancho de banda que necesita el módem para enviar los datos ya que por su naturaleza el módem utiliza 8 bits y si enviamos un dato 16 bits necesitaríamos ocupar el DAC 2 veces para enviar un dato .

Debido a lo anterior el DSP seleccionado debía ser uno que ejecutara 500,000 operaciones/s o más, el procesador TMS32050 cumple con esta especificación, este DSP de la marca Texas Instrumentes de la 5ª generación ejecuta una instrucción en 50 ns, procesa datos de punto fijo mismo que nos servirá porque los datos que procesaremos son enteros.

Discusión

Las pruebas indican que el desarrollo es viable por lo que se ha iniciado la implementación del algoritmo en el DSP y para ello estamos utilizando el software Code Compuser Studio de TI mostrado en la figura 2, este software permite programar los procesos utilizando Lenguaje "C" el cual es muy usado en la ingeniera en lugar del lenguaje ensamblador, el compilador del software traduce del lenguaje "C" a "ensamblador" las rutinas implementadas minimizando el tiempo para programación e implementación de rutinas o algoritmos.

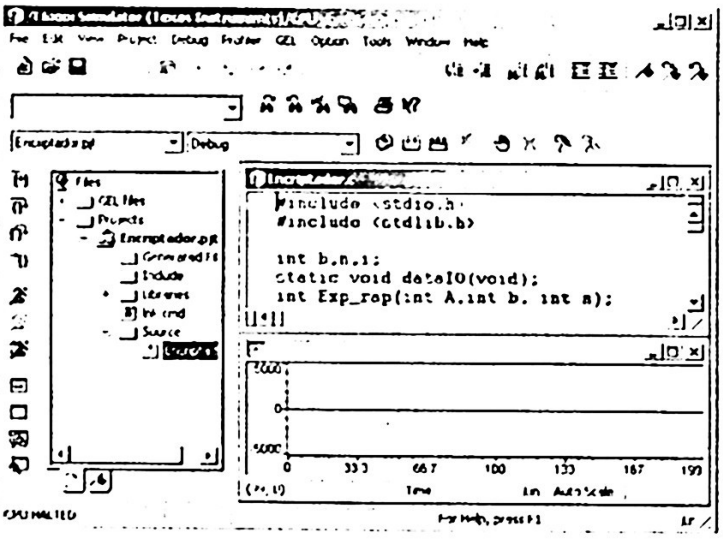

Fig. 2. Code Compuser Studio de TI última versión para ambiente Windows 2000 / XP.

Una vez implementado el algoritmo se harán pruebas con señales digitales (ya muestreadas) y posteriormente se adaptara el DAC el cual se encargara de convertir la señal de Analógica a Digital (DAC) y viceversa con los filtros antialias y de reconstrucción respectivamente, y por último el marcado telefónico por medio del módem y el aparato telefónico.

Como se dijo habrá una etapa de filtrado donde se implementará un filtro antialias que se construirá con la configuración de un filtro Butterworth pasabajas 60 db/dec con una frecuencia corte 3.5 KHz para obtener una señal lo más limpia posible de ruido, esto, basándonos en la frecuencia de la señal original, además de un filtro de reconstrucción de la señal analógica, con la configuración de un filtro Butterworth pasabajas de 60 db/dec con frecuencia de corte 3.5 KHz para obtener una señal analógica similar a la original o de entrada. Estos filtros se implementarán con elementos activos como Amplificadores Operacionales (OPAM's).

Referencias

1. Manuel José Lucena López
   Criptografía y Seguridad en Computadores
   Tercera Edición (Versión 1.14). Marzo de 2002
   e-mail: mlucena@ujaen.es
   Capítulo. 2.1, 29
2. Capítulo 2.2, 30
3. Capítulo 12.2, 143 - 144
4. Capítulo 5.1, 57
5. Capítulo 5.1.1, 58
6. Amparo Fúster, Dolores de la Guía
   Técnicas Criptográficas de protección de datos
   2ª. Edición Actualizada
   Alfaomega & Ra-Ma
   Apéndice A, 1, 240.
7. Manuel José Lucena López
   Criptografía y Seguridad en Computadores
   Tercera Edición (Versión 1.14). Marzo de 2002
   e-mail: mlucena@ujaen.es
   Capítulo 5.2.3, 63
8. Capítulo 5.4.1, 64
9. Apéndice C, Capítulo 5, problema 5.