UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROTECCIÓN EN UNA TRANSMISIÓN DE VOZ UTILIZANDO TÉCNICAS DE ENCRIPTACIÓN

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE

INGENIERO ELECTRÓNICO

PRESENTADO POR:

WILHEM BELISARIO RODRÍGUEZ VEGA

PROMOCIÓN
1989 – 2

LIMA – PERÚ
2005

Dedico el presente:

A mis padres, mi esposa y mis hijos.

Gracias por todo.

PROTECCIÓN EN UNA TRANSMISIÓN DE VOZ UTILIZANDO TÉCNICAS DE ENCRIPTACIÓN

SUMARIO

El presente informe ilustra las diferentes maneras en que la información, especialmente la voz, puede ser protegida. El primer capítulo describe como se puede proteger la información a través de la encriptación de datos y sus diferentes estándares. De manera paralela se revisa los métodos que existen para revertir el proceso de encriptación llamado Criptoanálisis. Ambas teorías deben ser dominadas por el diseñador de un método de encriptación, para que el resultado final no sea vulnerable a ataques por criptoanálisis. Una vez que se tiene el método de encriptación, se describe las ventajas que se tienen si se implementa en software ó en hardware. En el segundo capítulo trata sobre la voz, como es producida y las formas de identificar los diferentes sonidos. El tercer capítulo revisamos el proceso de la adquisición de datos en sus diferentes etapas. Se describe las opciones que tenemos para adquirir datos, con las facilidades que nos brinda Matlab. En el cuarto y último capítulo describimos algunas formas de encriptación de voz y desarrollamos nuestro método de encriptación de voz utilizando como herramienta la computadora y el Matlab. Finalmente ilustramos los cambios que se dan en la señal de voz en el proceso de encriptación así como el proceso inverso.

ÍNDICE

PROLOGO

El uso de Internet esta cada vez mas difundido y al alcance de todos, este medio nos permiten tener acceso a diferentes fuentes de información así como mayores facilidades en nuestra vida diaria. Por ejemplo, operaciones con nuestra cuenta bancaria, pagos de servicio, comprar con tarjeta de crédito, conversar, entre otros. Todas estas facilidades pueden ser vistas o escuchadas por terceros a fin de aprovecharse de las claves que se usan, correos que se transmiten, conversaciones, números de tarjetas de crédito, sin que nos percatemos de ello, con la finalidad de beneficiarse. Esto no sólo sucede a nivel de usuarios finales, también puede suceder entre gobiernos, fuerzas armadas, grupos políticos, industrias, bancos, etc.. Por este motivo, es importante proteger la información que transmitimos o recibimos.

CAPITULO I

DESCRIPCIÓN DE LA ENCRIPTACIÓN DE DATOS

1.1 Introducción

El término Criptología viene del griego criptos = oculto y logos = tratado, ciencia. Este término se emplea para denominar a dos disciplinas opuestas y a la vez complementarias: Criptografía y Criptoanálisis. La Criptografía se ocupa del estudio y desarrollo de técnicas matemáticas que proveen la seguridad de la información que cubre confidencialidad, integridad, y autenticación. Por otro lado, el Criptoanálisis, es el estudio de técnicas matemáticas orientadas a revertir las técnicas criptográficas y por lo general los servicios de la seguridad de información. El desarrollo de una técnica criptográfica implica el conocimiento de todas las técnicas de criptoanálisis.

La Criptografía nace en un principio como una necesidad de proteger la información que transmitian los ejercitos y diplomáticos, sin importar aún la comprobación de otros servicios como por ejemplo la autenticidad de los remitentes ó la integridad de la información.

El uso de las técnicas criptográficas ha cambiado y progresado en los últimos años y se está haciendo cada vez mas un requisito en la seguridad de la información. Esto debido a la proliferación de las comunicaciones, los equipos de computo y el uso de

Internet, que implica exponer la información en estos medios y por lo tanto genera la necesidad de que la información transmitida sea protegida. Hoy en día no solo las instituciones del estado lo requieren, también las instituciones privadas (ejemplo: para proteger sus proyectos, fórmulas y o estrategias de ventas) y personas particulares que no quieren ver violada su privacidad.

1.2 Evolución del encriptado de datos

La encriptación aparece casi en paralelo con la escritura y se remonta hasta 4000 años. En los últimos años la encriptación ha jugado un papel importante en las comunicaciones de datos, voz, imágenes y videos. Sin embargo los primeros que desarrollaron y usaron esta técnica fueron los militares, el servicio diplomático y los gobiernos en general. El encriptado se ha usado como una herramienta para proteger secretos y estrategias. La proliferación de las computadoras y los sistemas de comunicación en los años 1960, trajo consigo una demanda del sector privado de proteger su información y proveer servicios de seguridad.

En 1976, Diffie y Hellman plantean el concepto de la "llave pública" y provee una nuevo e ingenioso método para intercambiar llaves, la seguridad esta basada en el Problema de Logaritmo Discreto. En 1977 se da a conocer el standard DES, iniciales del ingles "Data Encryption Standard". En 1978 Rivest, Shamir y Adleman presentan el primer esquema de encriptado y firmas con llave pública, conocido como RSA. El esquema RSA esta basado en otro problema matemático, la dificultad de factorizar grandes números enteros. Este problema matemático, revitalizó el encriptado, esforzándose en encontrar métodos eficientes de factorización. En 1980 se vio mejores avances en esta área, pero ninguno de ellos dejo el sistema RSA

inseguro. Otra clase de poderosa y practico esquema de llave pública, fue encontrada por ElGamal en 1985. Estos son también basados en un problema logarítmico discreto.

Uno de los más significativos contribuciones proporcionadas por el encriptado de llave pública es la firma digital. En 1991 el primer standard para firmas digitales (ISO/IEC 9796) fue aceptado. Este está basado en el esquema de llave pública del RSA. En 1994 el Gobierno de Estados Unidos de America, adoptó el Estándar de Firma Digital, un mecanismo basado en el esquema de llave publica de ElGamal.

La búsqueda de nuevos esquemas de llave pública, mejoró los mecanismos de encriptado existentes y las pruebas de seguridad continúan a un ritmo acelerado.

1.3 Seguridad de la información y encriptación

Para entender el encriptado debemos entender lo que es seguridad de la información. La seguridad de la información se manifiesta de diferentes maneras de acuerdo a la situación y requerimientos. Además de quien esta envuelto, en un grado u otro, todas las etapas por las que atraviesa esta deben ser seguras o se debe tener la certeza de que así es. Algunos de estos objetivos los listamos a continuación:

Privacidad o confidencialidad: Mantiene la información secreta para todos excepto para quienes están autorizados.

Integridad de los datos: Esta no debe alterarse por quien no esta autorizado o por medios desconocidos.

Validación de identificación: Comprobación de la identidad.

Validación de mensaje: Comprobar el origen de los datos. Conocido como "validación del origen de la data"

Firma: Asociado con la identidad.

Autorización: Acuerdo, con otra entidad, para que oficialmente certifique una acción.

Validación: Provee tiempo limite de autorización para usar o manipular las fuentes de información.

Control de acceso: Acceso restringido a las fuentes a entidades privilegiadas.

Certificación: Endoso de información por una entidad confiable.

Registro de creación: Grabado de la fecha de creación o existencia de la información.

Testificación: Verificación de la creación ó existencia de la información por una entidad que no es la creadora.

Recibo: Validación de que la información ha sido recibida.

Confirmación: Validación de que el servicio ha sido prestado.

Propietario: Proveer a una entidad el derecho legal de usar o transferir la fuente a otro.

Anonimato: Proteger la identidad e una entidad envuelta en un proceso.

No repudio: Prevención de la negativa a acuerdos o acciones previstas.

Revocación: Retiro de la certificación o autorización.

Por años se han creado protocolos y mecanismos que están relacionados con la seguridad de la información. Con frecuencia los objetivos de la seguridad de la información no solo se logran a través de algoritmos matemáticos y protocolos,

también requieren técnicas y permanencias de leyes que permitan alcanzar el resultado deseado.

En un principio la información era escrita en papeles y entregadas a su destino. Hoy en día, a pesar de que la forma de grabar información no ha cambiado dramáticamente, la información grabada en medios magnéticos se transmiten vía sistemas de telecomunicaciones, algunas de ellas inalámbricas.

1.4 Criptoanálisis

1.4.1 Definición

Son técnicas orientadas a encontrar las claves usadas por determinado sistema criptográfico ó encontrar los textos originales o textos claros. La persona que diseña un criptosistema debe tener en cuenta todas las técnicas que existen para romper las claves o encontrar los textos claros, de tal manera que los algoritmos que se desarrollen bloqueen los posibles ataques.

No existe una uniformidad en las formas en que se puede atacar un criptosistema. Esto dependerá del algoritmo que se esté usando así como las condiciones de seguridad física que pueda brindar un sistema de comunicación.

Los algoritmos de encriptación que actualmente se usan son casi irrompibles. Si se quisiera romper la seguridad en un cripsosistema se deberá evaluar que tan valiosa es la información para que los costos computacionales que se requieran no superen el costo de información, además, que el tiempo de vigencia de la información no sea inferior al tiempo que se tarde en desencriptar la información. Por ejemplo, el algoritmo DES utiliza una clave de 56 bits es decir:

El número de claves $2^{56} = 7.2 \times 10^{16}$ claves.

Suponiendo que 1 prueba demora 1μs entonces 2^{56} tomarán 2248 años para probar todas las claves.

En algunos casos se deberá evaluar otras formas de acceder a la información, como por ejemplo espionaje.

Una manera de proteger mas aún la información es realizando un proceso de compresión, eliminando así información redundante, que puede dar pistas al criptoanalísta. La información debidamente comprimida se asemeja a un ruido aleatorio, impidiendo de esta manera toda posibilidad de análisis.

El compresor que se use debe ser especializado para la encriptación ya que los comercialmente conocidos como el ZIP, WinRAR tienen formatos conocidos, como las cabeceras y simplemente están hechos para reducir el espacio que ocupa cierta información.

1.4.2 Teoría, redundancia y entropia de una fuente de información.

Las posibles formas de análisis se basan en estadisticas, redundancia de información y entropía de la información. Por ejemplo los distintos idiomas están caracterizados por tener cierta frecuencia con la que se repite determinada letra, así las letras mas frecuentes en español son "A" y "E", estas dos letras representan el 8% del abecedario, sin embargo en un texto de 10000 letras su incidencia es del 26%.

El criptoanálisis utiliza también la redundancia en la información que se envía, esta redundancia se refiere a información innecesaria y repetida, por ejemplo los acentos, espacios en blanco y algunas letras que pueden por su sonido ser reemplazadas por otras (c, q, v, h), los géneros y números, tiempos de los verbos etc.

La cantidad de información es opuesta con la redundancia. Según la ley de Abramson, la información puede ser medida en unidades de información por la siguiente relación:

$$I(E)= \log 1/P(E) \quad \text{unidades de información}$$

Sea E un suceso con la probabilidad de ocurrencia P(E).

La entropía de una fuente de información se define como la cantidad media de información por símbolo emitido. Esta dada por la siguiente fórmula:

$$H(S) = \sum P(S_i).\log_2 1/P(S_i). \text{ bits/símbolo}$$

Donde $P(S_i)$ es la probabilidad de ocurrencia del símbolo S_i

1.4.3 Criptoanálisis Diferencial

La mayoría de los ataques se basan en estadísticas. En este método se trata de comparar y analisar grandes cantidades de parejas de texto claro y sus respectivos criptogramas. El proceso se realiza comparando la diferencia módulo 2 entre los textos claros, la diferencia que se produce cuando los textos claros están en proceso y finalmente la diferencia cuando están encriptados con la misma clave.

Es suficiente escoger pares de texto en claro con una cierta diferencia. Entonces, usando las diferencias en los textos encriptados resultantes, se van asignando probabilidades a las distintas claves posibles. Según se van analizando más y más pares de texto encriptado, una clave surgirá como la más posible. Finalmente se hallará la clave secreta.

1.4.4 Criptoanálisis Lineal

Se basa en aproximaciones lineales para poder describir el funcionamiento de un encriptador en bloque (mas adelante veremos los métodos de encriptado en flujo y en bloque). Se suman módulo 2 varios bits de texto claro y texto encriptado, obteniéndose un solo bit. Esta operación se realiza muchas veces utilizando la misma clave, luego se comparan estos resultados. Se realizan estas pruebas con diferentes textos claros usando otras claves calculando la probabilidad de coincidencia p de ambos bits. Si el resultado de la probabilidad de coincidencia es:

$$p \neq ½$$

es posible recuperar la clave a través de un ataque de texto conocido. Tiene una mayor probabilidad de éxito cuanto mayor sea la diferencia

$$| p - ½ |$$

así como mayor sea la cantidad de textos claros y sus correspondientes textos encriptados

1.4.5 Criptoanálisis por fallas de hardware

Este ataque esta basado en provocar fallas en el hardware de encriptado de tal manera que retransmita algún texto claro logrando obtener 2 o mas parejas de textos claros con sus respectivos textos encriptados. Con estos datos se pueden realizar comparaciones y recuperar la clave.

Estas falla pueden ser provocadas mediante averías temporales o permanentes del hardware o por falta de energía o a través de impulsos en el sistema de energía o el reloj ó dañar la etapa de recepción de tal manera que se obligue al emisor la retransmisión del mensaje.

1.5 Criptosistemas en Hardware y Software

La mayor parte de los criptosistemas están desarrollados en software debido a que muchas transacciones no necesitan que sean en tiempo real, sin embargo, el avance de la tecnología hace que todos los sistemas sean cada vez mas rápidos y por lo tanto mas eficientes. En la búsqueda de la eficiencia es que se esta difundiendo cada vez mas el uso de hardware en criptosistemas. Estos ya están difundidos en equipos de comunicación que usan especialmente los militares. En un futuro no muy lejano es posible que todos los dispositivos de comunicaciones incluyan circuitos que permitan la encriptación automática de datos y voz. Por ejemplo entre centrales telefónicas, routers,

1.5.1 Hardware en criptosistemas

Como ya mencionamos los criptosistemas pueden ser implementados en hardware y software o una combinación de ambos. La implementación se realiza en base a la velocidad de procesamiento requerida y costos.

Los desarrollos de estos sistemas en software están diseñados y codificados en lenguajes de programación tales como C, C++, Java y assembler además de estar desarrollados para correr en procesadores de propósito general, procesadores digitales de señales y tarjetas inteligentes, se debe tomar en cuenta que los desarrollos en software no son caros.

Los procesadores de propósito general pueden cubrir satisfactoriamente las necesidades de un usuarios individuales, por consiguiente la mayoría de los desarrollos existentes se basan en software. Para poder alcanzar velocidades mejores en criptografía se debe recurrir a implementaciones de hardware específicas.

Los desarrollos en hardware estan diseñadas y codificadas en lenguajes de máquina, tales como VHDL y Verilog HDL.

Entre las implementaciones mas cercanas a diseño por hardware se tienen los ASIC (Application Specific Integrated Circuits) y FPGA (Field Programmable Gate Arrays).

Un ASIC es diseñado en base a su aplicación, el diseño demanda tiempo y demanda muchas horas hombre. El diseño final se envía a fabricar y en general es costoso. Un defecto en el diseño es fatal ya que no se pueden corregir una vez que estos dispositivos estan configurados con la consecuencia de elevar costos y perdida de tiempo. Por este motivo los diseñadores necesitaban un prototipo rápido y que sea menos costoso, de allí que nace la idea de la fabricación e los dispositivos FPGA.

Los FPGA ofrecen las siguientes características: pueden ser comprados en cualquier tienda especializada en circuitos electrónicos y reconfigurado para diferentes aplicaciones. Cada reconfiguración es inmediata. Los FPGA están fabricados con miles de pequeños bloques conocidos como Bloques Lógicos Programables, (CLB del ingles Configurable Logic Blocks). CLBs estan conectados usando conectores programables y algunas familias de los FPGA tienen blocks de memoria dedicada.

A pesar de que los FPGA fueron inventados para realizar pruebas previas que después se convertirían en ASIC, estos rara vez llegan a explotarse a plenitud. Una de las principales diferencias es que el ASIC es hasta 10 veces mas rápido que un equivalente en FPGA, sin embargo la capacidad de reprogramar de los FPGA los convierte un unos componentes versátiles. En criptografía es útil tener la capacidad de intercambiar algoritmos de encriptación y en esto son de utilidad los FPGA.

También brindan la posibilidad de corregir posibles errores o mejorar un producto ya existente agregando mayores funciones.

El costo del diseño basado en FPGA es mucho menor que en los basados en ASIC.

Un factor importante en la implementación de un criptosistema es su nivel de seguridad. Ningún algoritmo de seguridad ayuda si existe un ataque en la implementación. Las implementaciones en software son las mas comunes, sin embargo, ofrecen menor seguridad. Es extremadamente dificil asegurar que no se filtre la información. Por ejemplo se pueden usar pequeños programas, como virus para recolectar cierta información que pueda ayudar a desencriptar un mensaje, sin que el usuario logre detectar. Otra forma es buscando en la memoria RAM que usó el programa de encriptación, rastros de claves o mensajes.

Las implementaciones de hardware son mas fáciles de proteger y es relativamente facil diseñar circuitos que no permitan ataques a pesar de que pueda existir un acceso físico, sin embargo, una forma de ataque puede ser reemplazando el circuito integrado que maneja la criptografia, reemplazarlo para que siga realizando la misma función que el anterior, con subrutinas adicionales que permitan enviar información a los atacantes.

Los diseños basados en FPGA no ofrecen protección contra la manipulación física. Los bits pueden ser fácilmente leídos y aprovechados para romper los mecanismos de seguridad. Por otro lado estos dispositivos tienen la función de lectura anterior, que es útil cuando se realiza un debug, el cual permite leer los contenidos de los registros y memoria junto con la configuración del bus de datos. De esta manera también se puede acceder a información crítica como la llave de encriptación.

Los diseños basados en ASIC son de cierta forma fáciles de proteger. El chip ASIC puede ser diseñado contra manipulación. Esto significa que el chip no puede dejar filtrar información. Una de las formas en que se puede atacar es analizando el consumo de energía. Por otro lado el chip puede tener formas de autenticarse que no puedan implementarse en otro chip, con lo que se hace resistente a filtrado de información por reemplazo de chip.

	ASICs	FPGA	SOFTWARE
Velocidad	muy rápido	rápido	menos rápido
Proceso de desarrollo			
Costo de diseño	muy costoso	moderado	no costoso
Ciclo de diseño	extenso	moderado	corto
Herramientas de diseño	costoso	moderado	no costoso
Mantenimiento y mejoras	costoso	no costoso	no costoso
Características criptográficas			
Resistencia a la manipulación	robusto	limitado	débil
Protección de la llave	robusto	limitado	débil
Versatilidad del algoritmo	no	Si	Si

Tabla 1.1: Cuadro comparativo entre ASIC, FPGA y Software

Como podemos ver cada tipo de implementación tiene sus ventajas y desventajas. La implementación por software es atractiva mientras no este en juego la velocidad del proceso y el nivel de seguridad no sea elevado. Si se requiere de mayor seguridad una buena solución es implementar el sistema con FPGA cuyo nivel de seguridad se compara con los ASIC siempre y cuando no se tenga acceso físico a los chips.

1.6 Parámetros de la implementación por hardware

Cada circuito está caracterizado por los siguientes parámetros: Velocidad de operación y área. La velocidad de procesamiento en criptografía normalmente esta dado por la velocidad de salida del chip, esta no siempre brinda la información de la velocidad debido a que otros parámetros están involucrados como es la latencia.

1.6.1 Velocidad de salida

Esta definida por el número de bits procesados en un segundo. Normalmente la velocidad de procesamiento del encriptado es igual al desencriptado. La velocidad de proceso de un block de datos es normalmente fijo y conocido, al menos que se implemente algunos artificios para variar estos tiempos. Sin embargo, esto no es deseable ya que la relación entre los datos procesados y el tiempo que se demora en procesar pueden dar pistas para que se filtre información en base a un ataque basado en tiempos.

1.6.2 Latencia

Esta definida como el tiempo que se requiere para completar el proceso de un block de datos y es normalmente expresado en número de ciclos de reloj. Es el tiempo entre el momento que ingresan los datos al sistema de encriptación y al momento en el que sale del sistema.

1.6.3 Área

Depende de la tecnología que se use. Con ASIC el área es expresada en términos de número de transistores o puertas lógicas o el tamaño del bloque en μm^2, ambos relacionadas con el costo del diseño.

En FPGA el costo está relacionado con el número de bloques lógicos configurables.

1.7 Métodos de encriptación

1.7.1 Método del encriptación en Flujo

Verman utilizó una clave de igual longitud que el texto claro, siendo esta clave una secuencia aleatoria, la misma que se utilizará una sola vez. Este método sin embargo no es práctico ya que resulta complicado enviar la clave tanto al emisor como receptor por un canal seguro. En la práctica se utiliza el método de encriptación en flujo cuyo esquema fundamental se presenta en la figura 1.7.1.

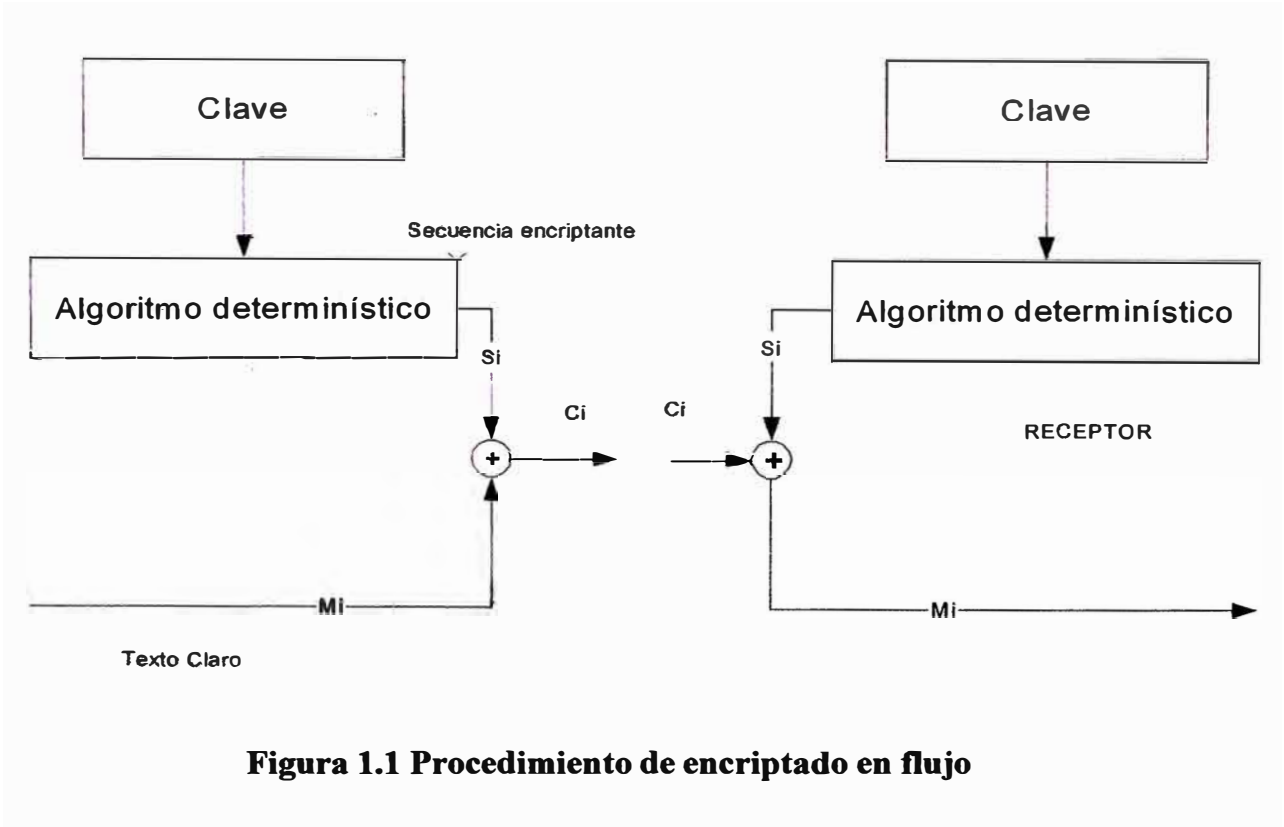


Figura 1.1 Procedimiento de encriptado en flujo

El emisor utiliza un algoritmo determinístico ó público generando una secuencia binaria Si cuyos elementos se suman módulo 2 con los correspondientes bits del texto claro Mi, resultando el texto encriptado Ci. Este último se envía por el canal público. En el lado de recepción se utiliza el mismo algoritmo y clave para desencriptar el mensaje Ci.. La clave secreta varia entre 120 y 250 bits, esta se hace llegar a los usuarios a través de claves públicas, una vez en su destino se procede al esquema de ecriptado en flujo.

1.7.2 Métodos de encriptación en bloque

Este es el método mas usado, se le llama en bloque porque el texto claro es encriptado usando porciones de texto claro ó bloques, estos tienen las siguientes características:

Cada símbolo se encripta de manera dependiente de los adyacentes

Cada bloque ce encripta de manera independiente

Dos mensajes con la misma clave dan como resultado 2 mensajes encriptados iguales.

No es preciso desencriptar desde el primer bloque que llegue a recepción, se puede hacer esta labor desde cualquier bloque.

1.8 Técnicas de encriptación

Las técnicas de criptografía moderna se pueden clasificar en dos según el tipo de clave utilizado:

Criptografía simétrica.

Criptografía de clave pública o asimétrica.

1.8.1 Criptografía simétrica

Es el sistema de criptografía más antiguo. Se utiliza desde los tiempos de Julio César hasta la actualidad. Se caracteriza por usar la misma clave para encriptar y desencriptar

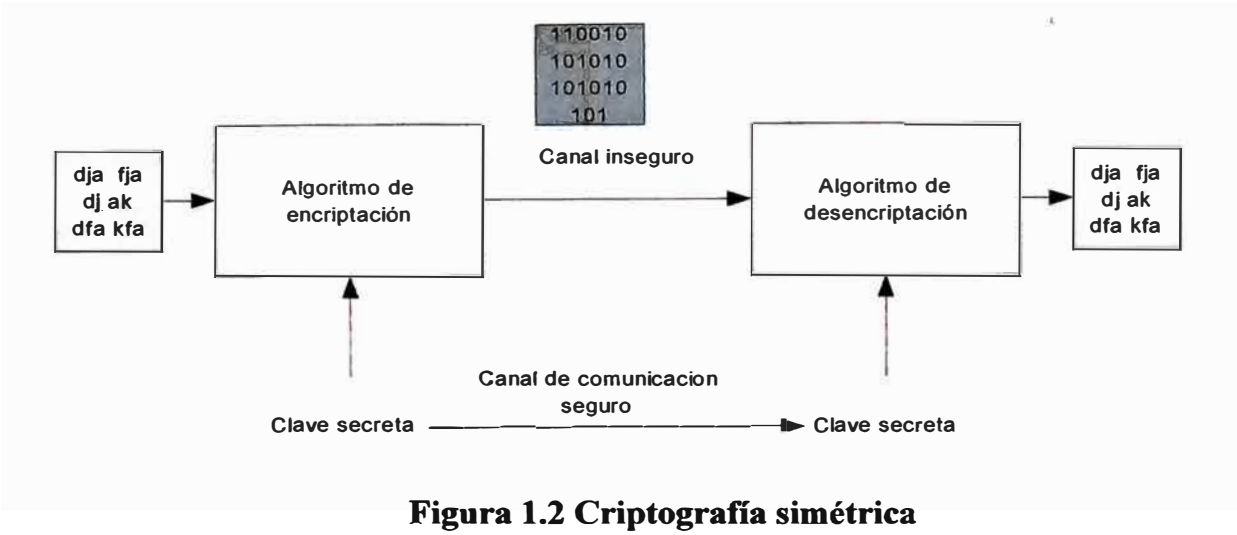


Figura 1.2 Criptografía simétrica

Toda la seguridad está basada en la privacidad de esta clave secreta, llamada simétrica porque es la misma para el emisor y el receptor. El emisor del mensaje genera una clave y después la transmite mediante un canal seguro a todos los usuarios autorizados a recibir sus mensajes. La distribución de claves es un gran problema para los sistemas simétricos, hoy en día se resuelve mediante sistemas asimétricos montados únicamente para transmitir claves simétricas.

Estos sistemas sólo permiten confidencialidad y no autenticación ni firma digital. Para mantener la confidencialidad delante de un criptoanalista, el algoritmo debe cumplir las siguientes condiciones:

Conocido el criptograma no se puede desencriptar el texto ni adivinar la clave.

Conocido el texto y el criptograma es más caro (en tiempo y/o dinero) desencriptar la clave que el valor de la información.

Para la segunda condición siempre existe el sistema de "prueba y ensayo" para encontrar la clave, es decir, probar todas las claves posibles hasta encontrar la que descifra el criptograma. La seguridad respecto a este tipo de ataque depende de la longitud de la clave.

Los algoritmos simétricos encriptan bloques de texto, el tamaño de los bloques puede ser constante o variable según el tipo de algoritmo. Tienen 4 formas de funcionamiento:

Electronic CodeBook (ECB). Se encriptan los bloques de texto por separado.

Cipher Block Chainning (CBC). Los bloques de criptograma se relacionan entre ellos mediante funciones OR-EXCLUSIVA.

Cipher FeedBack (CFB). Se realiza una OR-EXCLUSIVA entre caracteres o bits aislados del texto y las salidas del algoritmo. El algoritmo utiliza como entrada los criptogramas.

Output FeedBack (OFB). Igual que el CFB, se realiza una OR-EXCLUSIVA entre caracteres o bits aislados del texto y las salidas del algoritmo. Pero éste utiliza como entradas sus propias salidas, por lo tanto no depende del texto, es un generador de números aleatorios.

Los algoritmos simétricos son más sencillos que los asimétricos, por ese motivo los **procesos son más simples y rápidos**. Los algoritmos más utilizados son:

DES (*Data Encryption Standard*). El más utilizado y más antiguo

IDEA (International Data Encryption Algorithm). Se utiliza mucho en sistemas

nuevos europeos.

RC5. Algoritmo adoptado por *Netscape*, no está probada completamente su seguridad.

Rijndael. En el año 2000 fue adoptado como el algoritmo estándar del NIST, con el nombre de AES. Esto significa que será el algoritmo más utilizado en los próximos años.

1.8.2 Criptografía de clave pública o asimétrica

En 1976 *Diffie* y *Hellman* publicaron el artículo "*New directions in cryptography*". En él proponían un nuevo tipo de criptografía basado en utilizar claves distintas para encriptar y desencriptar, una de ellas se hace pública y la otra es privada de cada usuario. Así todos los usuarios de la red tienen acceso a las claves públicas, pero únicamente ellos conocen su clave privada. Estas ideas supusieron la revolución de la criptología: se podía utilizar para confidencialidad (como los sistemas simétricos), autenticación y firma digital, además de solucionar el problema de la distribución de claves simétricas.

Para cada tipo de servicio se encripta de manera diferente:

Confidencialidad. El emisor encripta el texto con la clave pública del receptor y el receptor lo desencripta con su clave privada. Así cualquier persona puede enviar un mensaje encriptado, pero sólo el receptor, que tiene la clave privada, y el emisor, que lo ha creado, pueden desencriptar el contenido

Autenticación. Se encripta el mensaje o un resumen de éste mediante la clave privada y cualquier persona puede comprobar su procedencia utilizando la clave

pública del emisor. El mensaje es auténtico porque sólo el emisor verdadero puede encriptar con su clave privada

Firma digital. Igual que la autenticación pero siempre se encripta el resumen del mensaje, cuyo criptograma es la firma del emisor. Así el emisor no puede negar la procedencia ya que se ha encriptado con su clave privada. Por otro lado, el receptor no puede modificar el contenido porque el resumen sería diferente y se vería que no coincide con la desencriptación de la firma. Pero el receptor sí puede comprobar que el resumen coincide con la firma desencriptada para ver si es auténtico La firma digital lleva implícita la autenticación.

Se puede realizar sistemas completos con autenticación o firma y confidencialidad.

Los algoritmos asimétricos están basados en funciones matemáticas fáciles de resolver pero muy complicadas de realizar la inversa, por ejemplo, la potencia y el logaritmo. Estas funciones son útiles para criptografía si la inversa es fácil de calcular conociendo un número concreto, la clave privada. Así la clave privada y pública están relacionadas matemáticamente, pero esta relación debe ser suficientemente compleja para que el criptoanalista no la pueda encontrar. Debido a esto, las claves privadas y públicas no las elige el usuario sino que las calcula un algoritmo y, normalmente, son muy largas.

Un algoritmo de clave pública debe cumplir:

Conocido el criptograma no se puede desencriptar el texto ni adivinar la clave.

Conocido el texto y el criptograma es más caro (en tiempo y/o dinero) desencriptar la clave que el valor de la información.

Conocida la clave pública y el texto no se puede generar un criptograma encriptado con clave privada.

En estos sistemas también funciona el criptoanálisis de "prueba y ensayo" y se puede aplicar las mismas suposiciones que en algoritmos simétricos. Aparte de este método, también hay algoritmos matemáticos para obtener la clave privada a partir de la pública pero, si el algoritmo es bueno, éstos son más caros que el valor de la información. Para complicar estos sistemas de criptoanálisis se utilizan claves muy largas.

El inconveniente de estos sistemas es la dificultad de implementación y la lentitud de proceso.

La ventaja es que implementan servicios de autenticación y firma, y además no tienen problemas con distribución de claves: la clave pública puede ser visible por cualquiera y la privada no se transmite nunca. El algoritmo más utilizado es el RSA (iniciales de sus creadores Rivest-Shamir- Adleman).

Únicamente para firma digital también se utiliza el algoritmo DSS (Digital Signature Standard) que ha sido adoptado como estándar por el NIST.

Para distribuir claves simétricas también se utiliza el algoritmo Diffie-Hellman, pero no sirve para confidencialidad, autenticación ni firma digital.

1.9 Funciones Hash

1.9.1 Definición

Las funciones Hash sirven para comprimir un texto de cualquier longitud en un bloque de longitud fija, el cual es pequeño comparado con el original. Su principal aplicación es en autenticación y firma digital. Con esto se logra:

No tener que encriptar todo el texto en los servicios de autenticación y firma digital, ya que este proceso es lento con los algoritmos asimétricos. El resumen sirve para comprobar si la clave privada del emisor es auténtica, no es necesario encriptar todo el texto si no se quiere confidencialidad (Figura 1.3).

Para poder comprobar automáticamente la autenticidad. Si se encripta todo el texto, al desencriptar sólo se puede comprobar la autenticidad mirando si el resultado es inteligible, evidentemente este proceso debe realizarse de forma manual. Utilizando un resumen del texto, se puede comprobar si es auténtico comparando el resumen realizado en el receptor con el desencriptado.

Para comprobar la integridad del texto, ya que si ha sido dañado durante la transmisión o en recepción no coincidirá el resumen del texto recibido con la desencriptación.

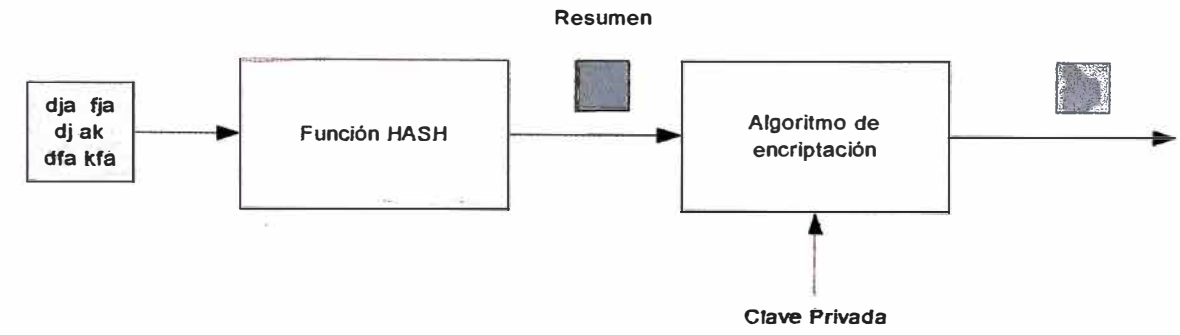


Figura 1.3 Firma digital con funciones Hash.

Las funciones Hash son públicas e irreversibles. No encriptan, sólo comprimen los textos en un bloque de longitud fija. Son diferentes de las funciones clásicas de compresión de textos, como ZIP, Huffman, V-42, etc.., que son reversibles e intentan

eliminar la redundancia de los textos manteniendo el significado. Las funciones Hash no son reversibles, es decir, no se puede recuperar el texto desde el resumen, pero deben cumplir las siguientes condiciones:

Transformar un texto de longitud variable en un bloque de longitud fija.

Ser irreversibles.

Conocido un mensaje y su función Hash debe ser imposible encontrar otro mensaje con la misma función Hash. Esto se debe cumplir para evitar que los criptoanalistas firmen un mensaje propio como si fueran otra persona.

Es imposible inventar dos mensajes cuya función Hash sea la misma.

Los algoritmos más utilizados son:

- **MD5**. Inventado en 1992 por Rivest. La longitud del bloque es de 128 bits. Es de libre circulación.

  SHA. Inventado en 1994 por la agencia americana NIST. La longitud del bloque es de 160 bits.

1.9.2 Utilización de claves

Se puede añadir una clave a la codificación de una función Hash. Está clave debe ser la misma para el emisor y para el receptor, por lo tanto, es un sistema de criptografía simétrica. Permite realizar autenticación e integridad pero no firma electrónica porque utiliza criptografía simétrica. Se comprueba la autenticación al codificar con la misma clave y obtener la misma función Hash y la integridad por las características de la función Hash. (Ver Figura 4)

Su principal ventaja es la rapidez, codificar una función Hash es más rápido y sencillo que encriptar con clave pública. Se utiliza mucho en autenticación e integridad de paquetes de información de un protocolo ON-LINE, donde no se permiten retardos en cada paquete, un ejemplo es el famoso protocolo SSL. El principal inconveniente es la debilidad del algoritmo, es un sistema de criptografía débil, su seguridad se basa en las reglas de las funciones Hash comentadas en el anterior apartado.

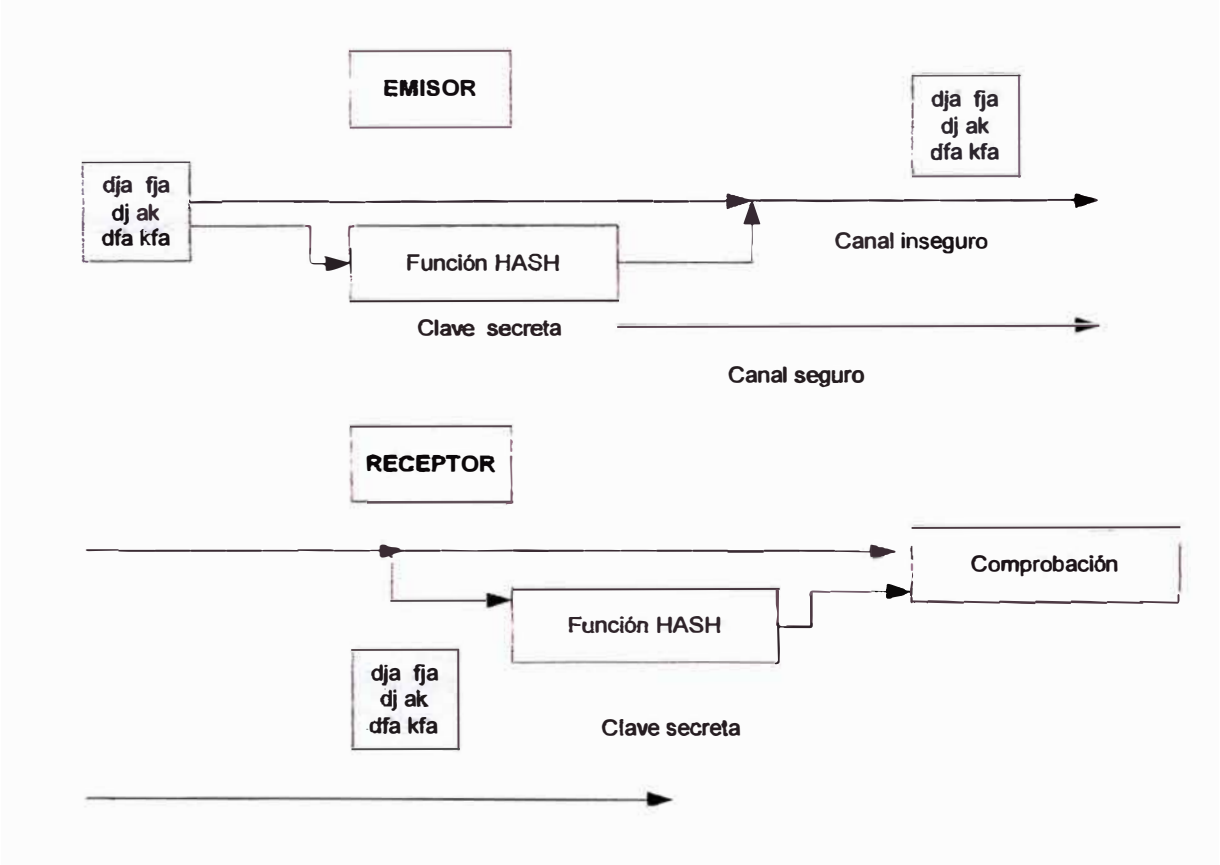


Figura 1.4 Funciones Hash con clave.

En los protocolos donde se busca rapidez, la debilidad se compensa con la poca información enviada en cada paquete. Así el esfuerzo para romper el algoritmo es menor que con un algoritmo asimétrico pero para obtener información de valor se deben descodificar muchos paquetes.

Un problema es utilizar una clave simétrica que se debe transmitir por un canal seguro.

1.10 Concurso de algoritmos del NIST (AES).

El primer concurso para un sistema de encriptación estándar fue lanzado en 1973 por la NBS (antecesora del NIST), y lo ganó en 1977 el DES. Este algoritmo se ha mantenido hasta hace poco como estándar de los organismos oficiales de los EE.UU. y como algoritmo simétrico más utilizado por las empresas privadas.

Actualmente existe una base instalada muy grande, todos los sistemas operativos lo incorporan para encriptar sus sistemas de seguridad. No se ha podido romper de una manera analítica pero se ha quedado anticuado en dos sentidos:

La clave es demasiado corta, 56 bits, para ser resistente a ataques de "prueba y ensayo". Durante muchos años la potencia de cálculo de los procesadores hacía que probar las 256 posibilidades de claves fuera más lento que el tiempo de vida de cualquier información. Pero se ha visto que la velocidad de los procesadores está aumentando muy rápidamente debido a los avances en electrónica digital. además desde la aparición de Internet se pueden conectar millones de máquinas trabajando en paralelo sobre el mismo proceso. Así se considera que en pocos años probar 256 claves será cuestión de días.

El sistema no permite longitud de clave variable. Se ha demostrado que los algoritmos que permiten elegir la longitud de clave son mucho más prácticos.

Las ventajas son las siguientes:

El usuario del sistema puede resolver el compromiso entre velocidad de proceso y seguridad eligiendo la longitud de la clave sin cambiar el algoritmo.

El sistema se adapta a los avances en velocidad de proceso. Cuando aumenta la velocidad de cálculo de las máquinas se utiliza una clave más larga, que mantiene la eficiencia del usuario y la dificultad de análisis con las nuevas máquinas.

Permite crear limitaciones legales por longitud de clave a los EE.UU., así el algoritmo es libre pero la utilización está limitada por el tamaño clave usada.

El NIST reprobó el DES desde 1994 hasta diciembre de 1998. Pero en 1997 decidió no volver a utilizarlo y, por lo tanto, convocar un concurso para buscar un nuevo sistema. El nombre de sistema es AES (Advanced Encryption Standard) y el del algoritmo utilizado AEA (Advanced Encryption Algorithm).

Las propuestas fueron presentadas antes de junio de 1998 y después se realizó una primera ronda para eliminar candidatos. En agosto de 1998 se publicó la lista de los 15 algoritmos candidatos. En agosto de 1999 el NIST publicó la lista de los 5 algoritmos que pasaron la primera ronda:

MARS

RC6

Rijndael

Serpent

TwoFish

En agosto del 2000 se publicó como ganador al algoritmo Rijndael de Joan Daemen y Vincent Rijmen.

1.10.1 Rijndael (AES)

Este algoritmo fue el ganador del concurso lanzado por el NIST para conseguir el nuevo sistema de encriptación simétrico que ha de sustituir al DES. El sistema ganador del concurso y, por lo tanto este algoritmo, se denomina AES (Advanced Encryption Standard). Aunque el NIST es una organización que afecta únicamente a la administración de los EE.UU., es muy probable que se convierta en el estándar mundial como ya pasó con el DES. Está diseñado por Joan Daemen y Vincent Rijmen, una curiosidad en un estándar americano es que todo el diseño se ha hecho en Bélgica. El diseño se realizó buscando 3 objetivos:

- La máxima resistencia a ataques.
- Velocidad y código compacto para varias plataformas informáticas.
- Simplicidad de diseño.

Estos últimos objetivos se añaden al primero utilizado en los otros concursos, 2 criterios prácticos para facilitar la producción a escala mundial. Permite 3 tamaños de bloque y 3 tamaños de clave, estos pueden ser: 128, 192 o 256 bits. El tamaño del bloque no condiciona el tamaño de la clave.

El algoritmo utiliza 3 transformaciones llamadas capas:

Capa de mezcla lineal. Crea la difusión del algoritmo, como las permutaciones.

Capa no lineal. Como las cajas S del DES. Realiza la confusión, o sea, las sustituciones.

Capa de suma de clave. ExOR. Para mezclar la clave transformada con los resultados de las iteraciones.

En cada iteración se realiza una aplicación de cada capa. El número de iteraciones probado como seguro depende de la longitud de la clave o el bloque. Así siempre se toma la longitud mayor entre la clave o el bloque y se aplica la siguiente fórmula:

- 128 bits son 10 iteraciones.
- 192 bits son 12 iteraciones.
- 256 bits son 14 iteraciones.

El algoritmo resultante es rápido y sencillo de implementar en cualquier plataforma. Además su seguridad está probada por los múltiples test realizados en el concurso del NIST. Igualmente por ser el estándar mundial, todas las comunidades científicas dedicarán grandes esfuerzos a probar su seguridad. Así probablemente será el algoritmo simétrico más utilizado.

1.11 Autoridades de certificación (CA)

El sistema de niveles sirve para grupos pequeños de usuarios donde siempre hay un enlace entre ellos, aunque sea por una cadena de confianzas de muchas personas. Pero tiene dos inconvenientes:

No es útil para los millones de usuarios de Internet, no pueden certificarse todos entre si.

No es útil para sistemas judiciales. Si se tiene que comprobar la procedencia de una firma y, por lo tanto, de la clave pública, con PGP se han de seguir largas cadenas de usuarios.

Para solucionar estos problemas se han creado las Autoridades de Certificación (CA). Son entidades públicas o privadas cuya función es ofrecer confianza en los certificados que firman. Generan claves públicas y certificados para usuarios bajo

demanda, además de dar a conocer sus claves públicas para las comprobaciones. Los usuarios se deben identificar personalmente para pedir un certificado a una CA. Es un sistema parecido al carnet de identidad, donde el Estado, como entidad de confianza, genera un documento que los bancos y las empresas consideran fiable.

Para descentralizar la gestión de CAs está previsto crear una estructura jerárquica a nivel mundial. Las CAs locales son certificadas por otras de nivel superior hasta llegar a la principal que es de confianza en todo el mundo. Así se consigue que la confianza sea mundial, para la red Internet sin fronteras, y que la gestión pueda ser local, para los procesos judiciales y facilitar el proceso de identificación personal.

CAPITULO II

LA VOZ

2.1 Descripción

Los sonidos del habla son producidos por una corriente de aire vibrante, que es formada y modificada de diferentes formas en la cavidad bucal. Existen muchos parámetros para identificar y clasificar los sonidos del habla. Estos parámetros indican de dónde viene la corriente de aire, en qué dirección se está moviendo, cómo se inician las vibraciones, qué sucede específicamente al ser alterada mientras se mueve y resuena, y qué partes del aparato vocal están interactuando para originar el cambio. Estos parámetros, llamados características fonéticas, son la base para la identificación de los sonidos.

2.2 Descripción de las cuerdas vocales

Las cuerdas vocales se localizan en la laringe (a veces también llamada caja resonante) y funcionan de forma diferente para diferentes sonidos. La principal diferencia consiste en que las cuerdas estén juntas o separadas y en que estén vibrando o no. Aquí sólo consideraremos dos estados diferentes de las cuerdas vocales: el de los sonidos sonoros, en el cual las cuerdas vocales están juntas y

vibrando: y el de los sonidos sordos. en el cual las cuerdas vocales están normalmente separadas e inmóviles. Los ejemplos de sonidos sonoros incluyen b. d. y g. Entre los sonidos sordos se incluyen p. t. y k. Podemos agrupar estos seis sonidos en pares que son idénticos en todos sus rasgos, excepto por el rasgo de sonoridad: b y p; d y t; g y k. El primer sonido de cada par es sonoro y el segundo es sordo.

2.3 Mecanismos de fonación y dirección del flujo de aire

Un flujo de aire en movimiento puede proporcionar la energía necesaria para producir un sonido del habla, a ésta nos referimos con la expresión mecanismo de fonación. Cada mecanismo de fonación (de los cuáles hay tres diferentes posibilidades) implica una cavidad en el aparato vocal, que cambia de tamaño debido a movimientos musculares complejos o contracciones, con los que se inicia el movimiento del aire. El Cuadro 2.1 muestra los tres mecanismos de fonación. la cavidad y el movimiento inicial pertinente a cada uno. El o los iniciadores del mecanismo de fonación, la o las partes específicas del cuerpo que se mueven para crear la corriente de aire, figuran en la tercera columna. La columna final describe el movimiento de cada iniciador. Los nombres en el Cuadro 2.1 le ayudarán a identificar las partes del aparato vocal.

Mecanismo de Fonación	Cavidad	Iniciador	Movimiento Inicial
pulmonar	pulmonar	Costillas	movimiento descendente y hacia adentro
		Diafragma	movimiento ascendente
glotal	faringe	laringe con obstrucción de la glotis	movimiento ascendente y descendente
		paredes de la faringe	contracción o expansión
velar	Oral	parte posterior de la lengua	cierre contra la parte posterior de la cavidad bucal (velo).
		cuerpo de la lengua	movimiento descendente

Cuadro 2.1 Mecanismo de fonación

Para cada sonido del habla, la corriente de aire se mueve ya sea hacia adentro o hacia afuera. Una corriente de aire hacia afuera, que se realiza normalmente en todos los sonidos del español, se llama egresiva. Una corriente de aire hacia adentro se llama ingresiva. La distinción sólo es importante para el aire glotal, que puede ser ingresivo o egresivo. El aire pulmonar utilizado para producir sonidos del habla siempre es egresivo, y el aire velar siempre es ingresivo.

2.4 Cierre velar y las cavidades donde resuenan los sonidos

Una analogía simple del aparato vocal, nos indica que es un conjunto de minúsculas cavidades en los cuales los sonidos resuenan. Las dos cavidades más comúnmente utilizadas para distinguir los sonidos del habla son la boca (cavidad bucal) y la nariz (cavidad nasal). La tercera cavidad (cavidad faríngea) está en la garganta sobre la laringe. También existe una cavidad labial entre la parte delantera de los dientes y los labios.

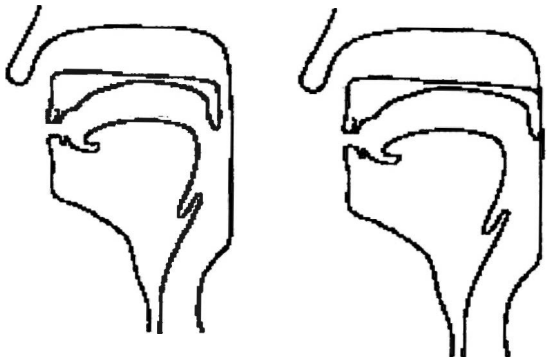
A diferencia de las habitaciones de una casa, la mayoría de estas cavidades cambian libremente de forma y tamaño, porque la mayoría de sus paredes están constituidas de músculos muy flexibles. (Debido a que las paredes de la cavidad nasal son hueso

y cartílago, su dimensión y forma están determinadas.) La forma y el tamaño de una cavidad en la cual la corriente de aire se mueve y vibra, afectan el sonido.

Los diferentes sonidos del habla se producen cuando las cavidades cambian de tamaño y forma, alterando con ello las resonancias en el tracto bucal. (Con resonancia se aluda a la frecuencia de las ondas sonoras producidas por la libre vibración del aire.) Ocurre lo mismo que en un edificio: habitaciones de diferentes formas y tamaños producen diferentes ecos.

Cuando está abierto el paso entre la cavidad nasal y oral, se dice que hay una abertura velar. La parte superior del paladar suave (el velo), se encuentra frente a la pared faríngea. Función como una puerta destinada a cerrar el paso.

Cuando hay un cierre velar (es decir que el velo se levanta hacia la pared faríngea, cerrando la abertura a la cavidad nasal), el aire no puede entrar a la cavidad nasal y entra solamente a la cavidad bucal. La Figura 2.1 nos muestra un corte lateral de la cabeza, lengua, etc., ilustrando esas posiciones velares.



Abertura Velar Cierre Velar

Figura 2.1

Los sonidos producidos por ondas sonoras que resuenan en la cavidad bucal se llaman sonidos orales. Los sonidos producidos por ondas sonoras que resuenan en la cavidad nasal (para los cuales el velo está abierto) se llaman sonidos nasales. Cuando el aire pasa simultáneamente a través de la boca y de la nariz, las ondas de algunos sonidos resuenan en ambas cavidades, la oral y la nasal. Estos sonidos (para los cuales existe una abertura velar) son conocidos como sonidos nasalizados.

Una analogía más cercana al funcionamiento real del tracto bucal. es la de un tubo cuya amplitud varía. Así como la analogía de las habitaciones discutida anteriormente resultó muy adecuada, la imagen de un tubo con diferentes amplitudes para producir diferentes sonidos explica que los sonidos del habla tengan diferentes resonancias. Si hay una abertura velar, la ramificación nasal del tubo está participando en la producción del sonido; si hay un cierre velar, la ramificación nasal no está participando en la producción del sonido.

Considere los siguientes pares de sonidos: b y m; d y n. En cada par, el tracto bucal tiene aproximadamente la misma forma para cada par de sonidos excepto por el cierre velar. Para b y d, que son sonidos orales, hay un cierre velar; para m y n, que son sonidos nasales, hay una abertura velar.

2.5 Modo de articulación

Los sonidos del habla implican una corriente de aire vibrante a la cual le sucede algo mientras avanza. Una cosa que puede sucederle es que el paso de la corriente de aire sea obstruido (bloqueado), en cierto grado, y en alguna parte o partes del mecanismo vocal. Esto altera también la dimensión y tamaño de las cavidades de

resonancia. El modo de articulación de un sonido describe el grado de obstrucción a la corriente de aire y el tipo de cierre que produce esa obstrucción.

Si para determinado sonido la corriente de aire se bloquea totalmente, se dice que hay bloqueo total. Tales sonidos, que son consonantes, se los conoce como oclusivos (ocasionalmente como explosivos), por ejemplo, p, t, y k.

Si la corriente de aire no es bloqueada totalmente, se produce una turbulencia audible. La corriente de aire para un sonido caracterizado por una turbulencia audible, se dice que está parcialmente obstruida, y el sonido se llama fricativo. (El término fricativo se asemeja a la palabra fricción, y podemos pensar en una fricativa como sonido con fricción audible.) Los ejemplos de fricativas incluyen s, f, y z. Todas las fricativas son además consonantes.

Si la corriente de aire no es bloqueada en gran medida, sino que es simplemente formada o dirigida por cierta parte de la lengua y quizás por los labios, se dice que es una corriente de aire ligeramente obstruida. Los ejemplos de estos sonidos incluyen a las vocales a, i, y o.

Para los tres sonidos, t, s, y l, el aparato vocal tiene la misma forma, excepto en dos aspectos. Primero, para t y s las cuerdas vocales no vibran mientras que para l sí vibran. Segundo, son diferentes en la manera en que la punta y los costados de la lengua se dirigen hacia arriba para impedir el paso del aire, lo cual determina su modo de la articulación; para t (por su modo de articulación, una oclusiva), la corriente de aire es obstruida totalmente por el cierre completo de los lados y de la punta de la lengua contra los dientes; para s (una fricativa), es obstruida completamente por el cierre total de los lados y parcial por la punta de la lengua contra la parte superior de la boca; y para l, la corriente de aire se impide levemente,

aunque la punta de la lengua se levanta hacia la parte superior de la boca para formar y para dirigir la corriente de aire, los costados de la lengua sólo se levantan levemente, dando por resultado un cierre parcial que no produce fricción entre ellos y la parte superior de la boca.

2.6 Articulador

El aparato vocal está formado por muchos elementos distintos, los mismos que se muestran en la Figura 2.2, "Partes del aparato vocal". Hay dos tipos de articuladores. El término articulador activo, se refiere a una parte movible del aparato vocal que impide o dirige la salida del aire. Ya hemos descrito los sonidos según el grado de cierre que impide la salida del aire durante su producción. Un articulador activo es el que se mueve para causar una obstrucción. Puede ser el labio inferior, alguna parte de la lengua, o algo más en el aparato vocal.

El otro tipo de articulador es el articulator pasivo. Ésta es la parte más fija de la boca, a la que un articulador activo toca o a la que se aproxima mucho para articular un sonido; es el caso del labio superior, los dientes superiores y varias partes de la región superior de la boca.

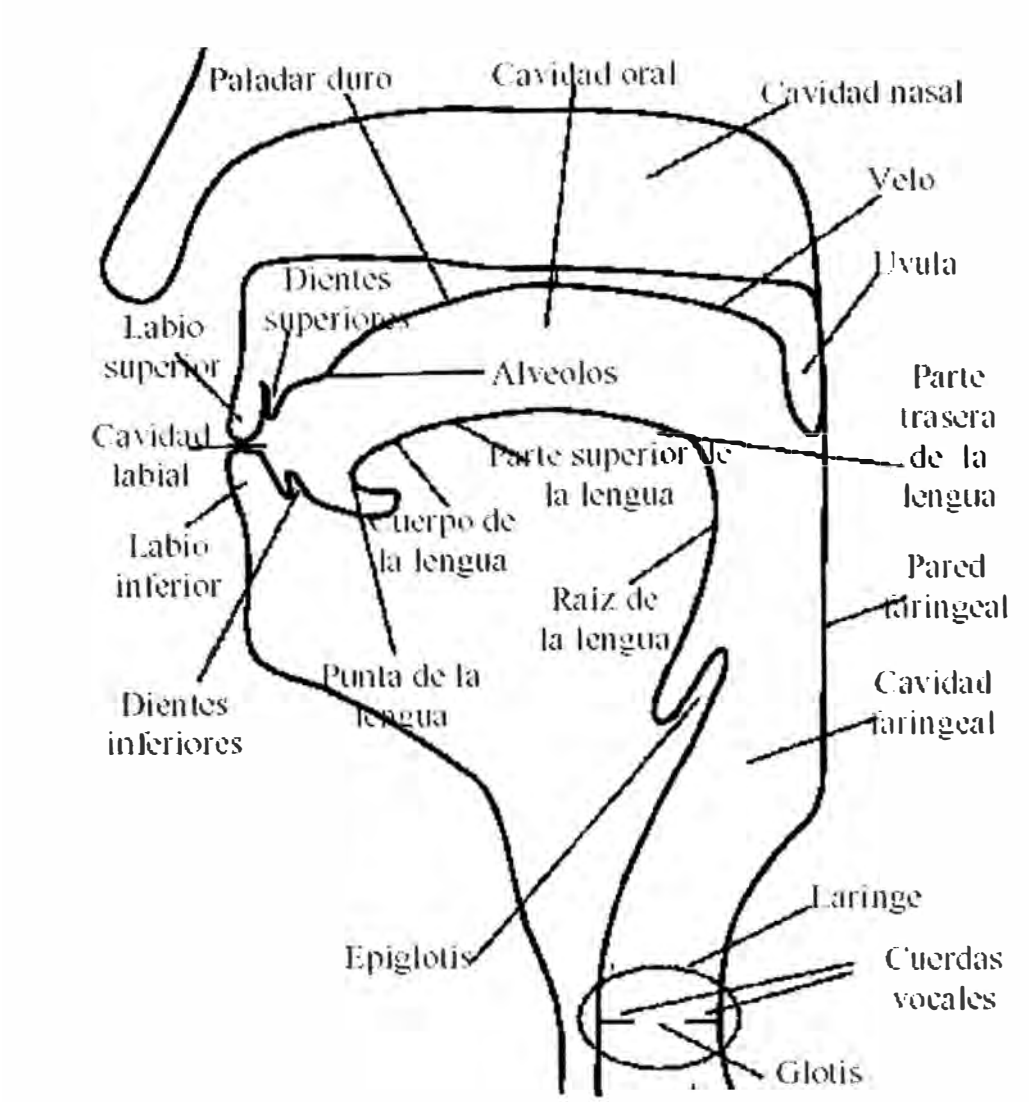


Figura 2.2

2.7 Partes del aparato vocal

La Figura 2.3 muestra las direcciones en que se mueven los articuladores activos, así como los articuladores pasivos con los cuales interactúan. Por ejemplo, podemos comparar los diferentes movimientos de la lengua implicados en la producción de "d" y "g" y las diferencias en los movimientos de sus labios y dientes para "m" y "f". Puesto que la mayoría de los articuladores pasivos están situados a lo largo de la superficie superior de la cavidad bucal, a veces se los conocen como articuladores superiores. En contraste, la mayoría de los articuladores activos están situados en la

cavidad bucal, relativamente más abajo que los correspondientes articuladores pasivos; algunas veces los articuladores activos son llamados articuladores inferiores. Generalmente, cuando usted vea el término articulador sin la especificación que hemos discutido, diferenciando entre activo y pasivo, puede asumir que es un articulador activo.

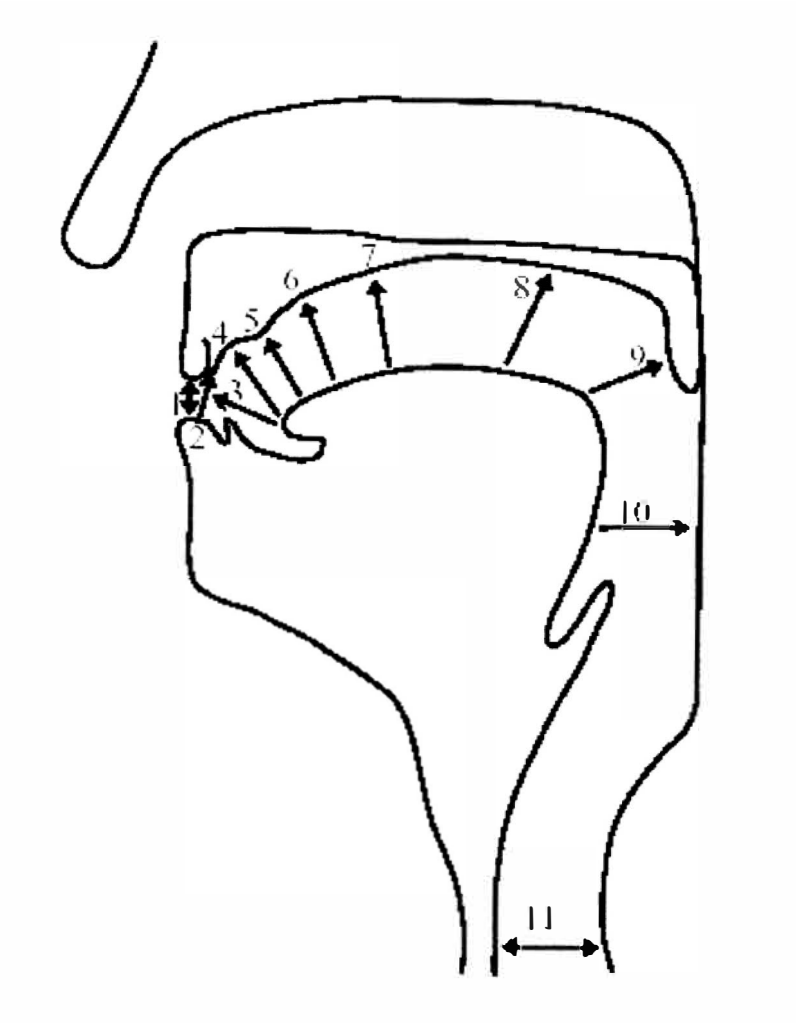


Figura 2.3

2.8 Zona de Articulación

Los sonidos también se describen según su zona de articulación. Para entender el término "zona de articulación" se tiene que considerar primero lo que no es. La zona de articulación no es el lugar en donde el articulador activo se pone en contacto con otra parte del aparato vocal (por ejemplo, alvéolos o velo); tal punto se llama articulador pasivo. Así, la "zona de articulación" es realmente la relación (la "sobre posición" o "acercamiento") entre los articuladores activos y pasivos cuando forman o impiden el paso de la corriente de aire. El Cuadro 2.2 contiene una lista parcial de las zonas de articulación y de los articuladores para cada una de ellas.

La única diferencia entre la pronunciación de las palabras pasa, tasa y casa es la zona de articulación de los sonidos iniciales: p es bilabial t es dental, y k ("c") es velar.

El Cuadro 2.3 muestra los articuladores activos que son utilizados para cada una de las principales zonas de la articulación. Observe que algunos de los articuladores activos pueden tener una relación con más de un articulador pasivo. Por ejemplo, la punta de la lengua puede articular un sonido entre los dientes (el sonido castellano z), detrás de los dientes superiores tocándolos (los sonidos del español t, n, l), o levemente detrás, a lo largo de los alvéolos, es decir, la parte dura del paladar (los sonidos ingleses t, n, l).

Estas tres relaciones se llaman interdental, dental y alveolar, respectivamente.

	Zona de articulación	Articulador activo	Articulador pasivo
1.	Bilabial	labio inferior	labio superior
2.	Dental	punta de la lengua	dientes
3..	interdental	punta de la lengua	detrás y arriba de los dientes
4.	Dental	punta de la lengua	detrás y arriba de los dientes
5.	Alveolar	punta de la lengua	alvéolos
6.	palato-alveolar	cuerpo de la lengua	detrás de los alvéolos
7.	palatal	parte superior de la lengua	paladar duro
8.	velar	parte posterior de la lengua	frente del paladar blando; velo
9.	uvular	parte posterior de la lengua	parte posterior del paladar
10.	faringea	raiz de la lengua	parte posterior de la pared faringea
11.	glotal	cuerdas vocales	(ninguno)

Cuadro 2.2

Articulador activo	Zona de articulación
Labios	bilabial – labiodental
Punta de la lengua / apex	interdental – dental alveolar
Cuerpo de la lengua	palato – alveolar
Parte superior de la lengua	palatal
Parte posterior de la lengua	velar – uvular
Raíz de la lengua	faringea
Cuerdas vocales	glotal

Cuadro 2.3

2.9 Procesamiento de la señal de voz

Los sonidos consisten en cambios de presión a través del tiempo y a frecuencias que podemos escuchar. Estos sonidos pueden ser digitalizados utilizando un

micrófono o cualquier otro medio que convierte la presión del aire a pulsos eléctricos. La voz es un subconjunto de los sonidos generados por el tracto vocal.

En el preprocesamiento de la señal se extraen las características de la misma que se pueden utilizar en un reconocedor. Esta señal se divide en segmentos y se obtiene una característica acústica distintiva de cada segmento. con estas características obtenidas, se construye un conjunto de vectores que constituyen la entrada al siguiente modulo. Una de las representaciones mas usadas en los coeficientes Linear Predictive Coding (LPC) y los coeficientes Mel-Frecuency Cepstrum Coefficients (MFCC).

Una manera de representar el sonido es graficándolo en forma de onda. El eje horizontal representa el tiempo y el vertical la amplitud, una limitación de este hecho es que no describe explícitamente el contenido de la señal de voz en términos de sus propiedades.

Una representación mas adecuada para el análisis de la señal son los espectrogramas. Un espectrograma es una representación de la señal de voz de acuerdo a las variaciones de la energía, con respecto al tiempo y frecuencia. El espectrograma contiene mucha información y revela las características acústicas especificas del habla. Los espectrogramas son útiles para un análisis visual de la señal. Sin embargo un reconocedor debe extraer de la señal acústica solo la información que requiere para poder reconocer una frase. para ello la señal de muestra a cierta frecuencia, se cuantiza y posteriormente se crean vectores característicos. Estos últimos son las que son utilizadas por el reconocedor. En la etapa del reconocimiento se traduce la señal de entrada a su texto correspondiente. Este proceso se puede llevar a cabo de diversas

formas utilizando enfoques como Redes Neuronales Artificiales (RNA) y Modelos Ocultos de Markov (HMM), entre otros.

CAPITULO III

ADQUISICIÓN DE DATOS

3.1 Sistema de adquisición de datos

El sistema de adquisición de datos se realizará utilizando el programa MATLAB. El propósito es capturar un fenómeno físico para realizar mediciones, estudios, estadísticas, etc.

Hardware del sistema de adquisición

Lo principal en un sistema de adquisición de datos reside en el hardware, su función principal es convertir la señal análoga en señal digital y viceversa.

Transductores

Dispositivo que se encarga de convertir la forma de energía de entrada en otra energía a la salida. Por ejemplo un micrófono es un transductor, convierte la energía del sonido (niveles de presión) en energía eléctrica. Un parlante realiza una transformación inversa.

Hardware de acondicionamiento de señal

Muchas veces los transductores no son compatibles con la etapa de la adquisición de datos, para superar esto, se debe acondicionar la señal ya sea amplificándola o filtrándola.

La computadora.

Este equipo proporciona el procesamiento de la señal, un reloj, un bus para transferir la data y memoria o capacidad de disco para almacenar la data.

Software

Permite interactuar al usuario con los datos adquiridos por el hardware.

Los componentes de la adquisición de datos y su relación con otros se muestran en la figura 3.1.

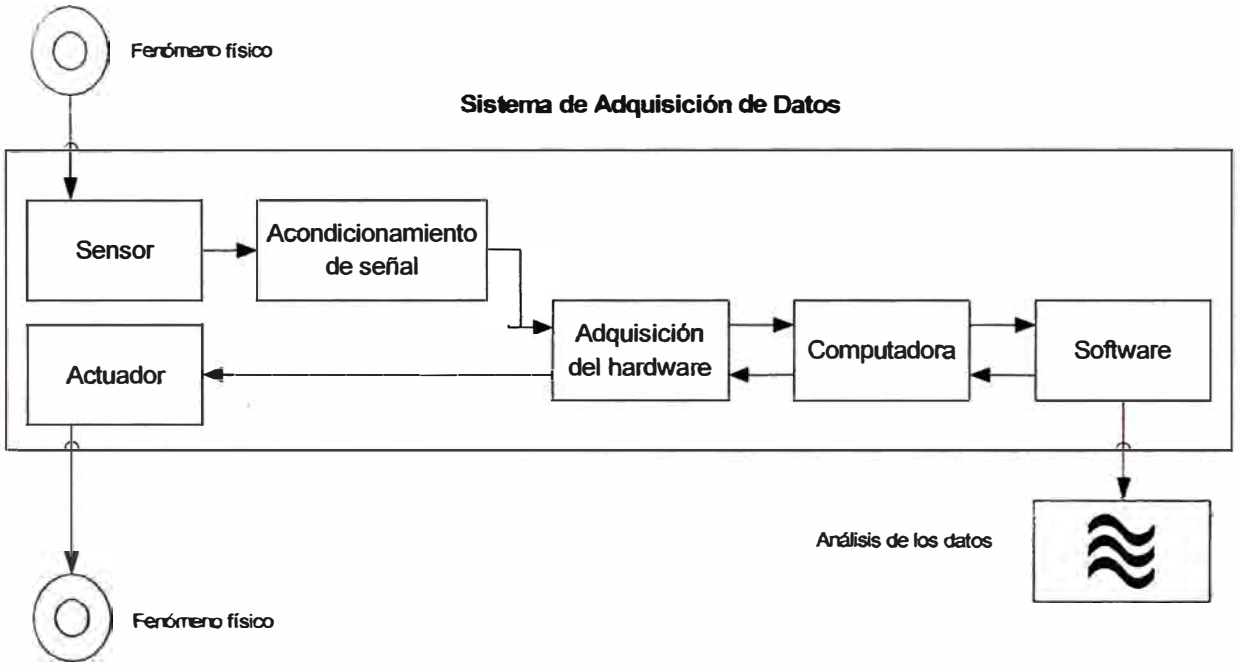


Figura 3.1 Esquema del Sistema de Adquisición de Datos

La figura ilustra las dos principales características de un sistema de adquisición de datos:

La señal ingresa a través del sensor, y se convierte en bits para que pueda ser procesado por la computadora, es el caso de adquirir datos con un micrófono, amplificarlo y digitalizarlo con una tarjeta de sonido y cargarlo en la computadora para luego ser procesado y analizado.

Los datos de la computadoras son convertidos en una señal análoga para que sea aceptado por un actuador.

3.2 Hardware de adquisición de datos

Este hardware puede ser interno o externo a una computadora, este es caracterizado por los subsistemas que posee Figura 3.2. Un subsistema es un componente del sistema de adquisición de datos que realiza una tarea especializada. Estos incluyen:

- Entrada análoga
- Salida análoga
- Entrada/salida digital
- Contador/cronómetro

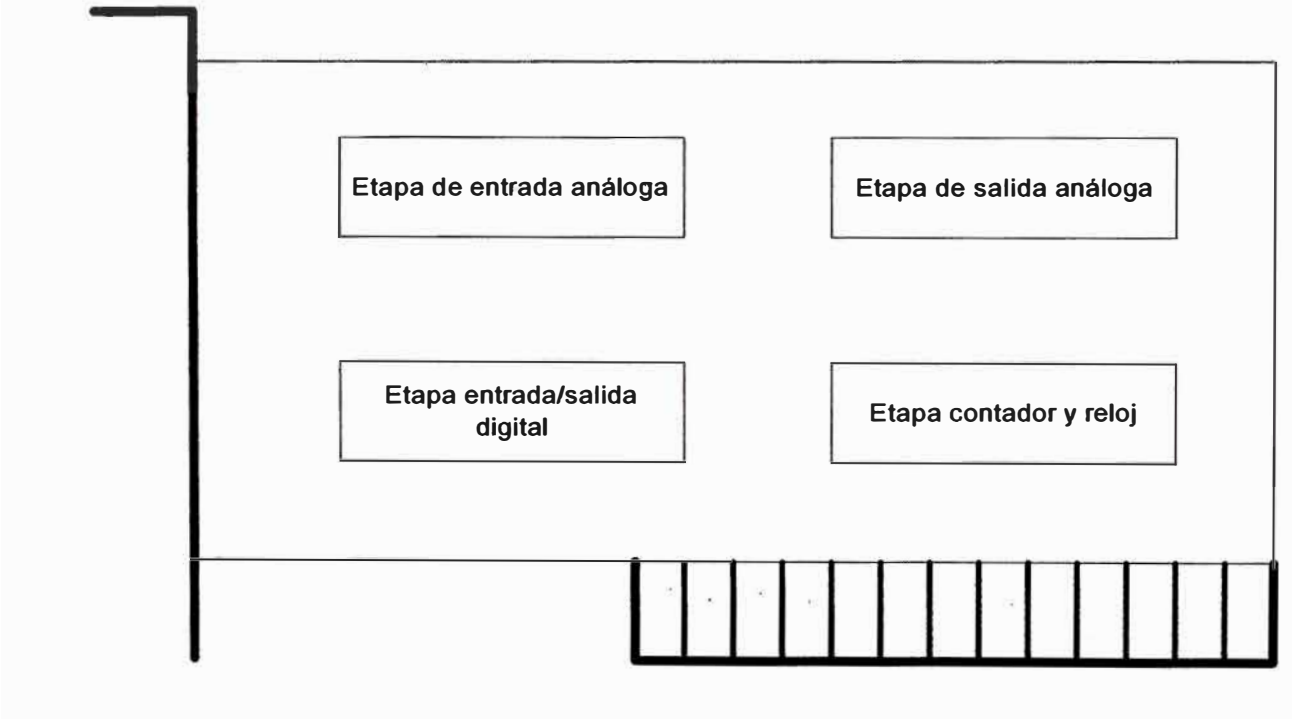


Figura 3.2 Etapas en la tarjeta de adquisición de datos

Estos dispositivos son llamados tarjetas multifunción

3.2.1 Subsistema entrada análoga

Se encarga de convertir la señal análoga que viene del sensor en bits que puedan leer la computadora. Es común que esta etapa ofrezca diferentes niveles de resolución, 12 o 16 bits. Estos subsistemas también son conocidos como AI subsistemas. convertidores A/D o ADCs.

3.2.2 Subsistema de salida análoga

Se encarga de convertir la señal digital almacenada en la computadora en una señal análoga. Estos sistemas se conocen también como subsistemas AO, convertidores D/A o DACs.

3.2.3 Subsistema digital entrada/salida

Estos son diseñados para recibir y transmitir valores digitales (niveles lógicos) de y hacia la computadora. Estos valores son típicamente manejados ya sea como simples bits o líneas, o como puertos, los que típicamente tienen 8 líneas. Mientras las mas populares tarjetas de adquisición de datos incluyen algunas capacidades de I/O digitales, estos están limitados a operaciones simples, para que se realicen operaciones digitales I/O avanzadas es necesario tener una tarjeta dedicada especial.

3.2.4 Subsistema contador / cronómetro

Estos (C/T) son usados para contar los eventos, frecuencia y periodo de medida y generación del tren de pulsos.

3.3 Sensores

Un sensor convierte un fenómeno físico en una señal que puede ser adquirida por el hardware. Existen 2 tipos de sensores basados en la salida que producen: sensores digitales y sensores análogos

Los sensores digitales producen una señal de salida que es una representación de la señal de entrada, y tiene valores discretos en magnitud medidos a tiempos discretos. Un sensor digital debe emitir niveles lógicos de magnitud a tiempos discretos. Un sensor digital debe emitir niveles lógicos que son compatibles con el receptor digital. Algunos niveles standard incluyen TTL y emisores lógicos acoplados (ECL). Estos sensores digitales pueden ser interruptores y codificadores de posición.

Sensores análogos producen una señal de salida que es directamente proporcional a la señal de entrada. y es continua en magnitud y en tiempo. La mayor parte de variables físicas como la temperatura, presión, aceleración en la práctica son continuos y son rápidamente medidos con un sensor análogo. Por ejemplo, la temperatura de un sistema de enfriamiento de un automóvil y la aceleración producida por una persona al nadar varían continuamente.

El sensor que se use depende del fenómeno que se va a medir.

Algunos sensores análogos comunes y la variable física que se mide se muestran en la Tabla 3.1

Sensor	Variable Física
Tacómetro	Aceleración
Micrófono	Presión
Medidor de presión	Presión
Dispositivo de resistencia de temperatura	Temperatura
Medidores de tensión	Fuerza
Termocupla	Temperatura

Tabla 3.1 Relación de sensores y las variables físicas que miden

Existen dos características importantes al momento de elegir un sensor, la primera la salida del sensor y segunda el ancho de banda.

3.3.1 Salida del sensor

La salida del sensor puede ser una señal analógica o una señal digital, y la salida variable es normalmente un voltaje y ocasionalmente la salida puede ser una corriente.

3.3.2 Señales de corriente.

La corriente es frecuentemente usada para transmitir señales en ambientes ruidosos ya que son menos afectadas. El rango de la señal de corriente es de 4-20mA o 0-20mA. La señal del primer rango tiene la ventaja que a pesar de tener un valor mínimo, siempre habrá una señal de corriente medible, mientras que la ausencia de esta indica que existe un problema de cableado.

Antes de la conversión del subsistema de entrada análoga. la señal de corriente normalmente es convertida a señal de voltaje por una resistencia sensora de corriente. La resistencia debe ser de alta precisión, 0.03% o 0.01% dependiendo de la resolución del hardware. Adicionalmente, la señal de voltaje debe coincidir con el rango de la señal de entrada del dispositivo analógico de entrada..

3.3.3 Señales de Voltaje.

La señal mas común es una señal de voltaje. Los principales aspectos a tomar en cuenta son:

Amplitud

Si la señal es del orden de pocos milivoltios, se deberá amplificar. Si la señal es muy mayor que el rango del sistema de adquisición de datos (normalmente ±10V), se deberá dividir la señal usando una red de resistencias.

La amplitud esta relacionada con la sensibilidad (resolución) del hardware.

Frecuencia

Cuando se adquiere una señal se debe decidir cual será la frecuencia máxima a medir.

La mayor componente de frecuencia de la señal determina la frecuencia de muestreo. Si se tiene mas de una señal de entrada pero sólo una entrada análoga al sistema de adquisición de datos, entonces se deberá considerar en conjunto que la frecuencia de muestreo crece en proporción a el número de entradas. Altas frecuencias pueden estar presentes como ruido, los cuales pueden ser eliminados a través de filtros antes que sea digitalizada.

Si el muestreo de la señal de entrada es al menos el doble del componente de frecuencia mas alta. entonces la señal será únicamente caracterizada. Sin embargo este muestreo puede no seguir la forma de onda correctamente. Para una señal rápida, se necesitará una frecuencia de muestreo entre 10 a 20 veces mas rápida para obtener una figura mas precisa de la forma de onda. Para señales de frecuencias bajas, se necesitará considerar el tiempo mínimo para un significativo cambio en la señal. La frecuencia esta relacionada al ancho de banda de la medición.

Duración

Si se está almacenando datos en la memoria o disco duro de la computadora, la duración determinará los recursos que se requieren. Otro factor a considerar es el formato en el que se va a almacenar dicha información, así la información almacenada en formato ASCII ocupara mas espacio que si se almacena en formato binario.

3.3.4 Ancho de banda del sensor

El ancho de banda esta dado por el rango de frecuencias presentes en la señal a medir, por esto el ancho de banda del sensor debe ser compatible con el ancho de banda a medir.

Se puede desear que el sensor tenga el mayor ancho de banda posible, asegurándose que el sistema de medición sea capaz de responder linealmente a todo el rango de frecuencias de la señal a medir. Sin embargo, si mayor es el ancho de banda del sensor, mayor será el cuidado de eliminar componentes de frecuencias no deseadas.

3.4 Acondicionamiento de señal

Los sensores de señal son con frecuencia incompatibles con el hardware del sistema de adquisición de datos. Para superar esta incompatibilidad, la señal del sensor debe ser acondicionada. El tipo de señal que se requiere acondicionar, depende del sensor que se esta usando. Por ejemplo, una señal puede ser muy pequeña y puede requerir amplificación, o puede contener frecuencias no deseadas y requiera filtrado.

Las formas mas frecuentes de acondicionamiento de una señal incluye:

Amplificación

Filtrado

Aislamiento Eléctrico

Multiplexado

Fuente de Alimentación

Amplificación

Las señales pequeñas (menores a 100 mili voltios), usualmente requieren amplificación. Las señales grandes pueden también requerir de amplificación, dependiendo del rango de entrada de la etapa análoga.

Filtrado

El filtrado quita el ruido de la señal de interés. Un filtro de ruido es usado en señales de baja frecuencia como la temperatura, para atenuar altas señales de frecuencia que puede reducir la precisión de la medida.

Señales que varían rápidamente, como las vibraciones, requieren un tipo diferente de filtro, conocido como filtro para evitar solapamiento, este tipo de filtro no permite altas frecuencias que pueden alterar las mediciones.

Aislamiento eléctrico

Si la señal de interés contiene altos voltajes, puede dañar la computadora, entonces la señal debe ser eléctricamente aislada de la computadora por seguridad.

También se puede requerir aislamiento eléctrico para asegurarse de que las lecturas del hardware del sistema de adquisición de datos no sea afectado por diferencias de potenciales. Por ejemplo, cuando el dispositivo de hardware y el sensor de la señal están con referencias a tierra, se pueden producir problemas si hay diferencias entre cada una de las referencias traduciéndose en errores en la medición. Si se usa aislamiento eléctrico, se elimina las diferencias de voltaje con respecto a diferentes referencias de tierra y se asegura que la señal es exactamente representada.

Multiplexado

Una técnica común de medir varias señales con un solo dispositivo de medida es el multiplexado.

Los dispositivos de acondicionamiento de señal, frecuentemente proveen multiplexado para usarse con señales de baja variación como la temperatura. Esto es en adición a cualquier componente de multiplexaje interno de la tarjeta DAQ. El convertidor A/D, muestrea un canal, cambia al siguiente canal y muestrea, y

así sucesivamente. Debido a que el mismo convertidor A/D esta muestreando varios canales. la eficiencia del muestreo de cada canal individual es inversamente proporcional a el número de canales a muestrear.

Se debe tener cuidado cuando se usa multiplexores para que cada señal muestreada tenga el tiempo suficiente para establecerse.

Fuente de Alimentación

Algunos sensores requieren de una excitación para funcionar. Por ejemplo un sensor de presión, un dispositivo resistivo de temperatura requieren de una fuente externa de voltaje. El módulo de acondicionamiento de señal para estos sensores normalmente proveen la alimentación necesaria.

3.5 La computadora

La computadora provee el procesador, reloj, bus de transferencia de datos y memoria y espacio de disco duro para almacenar la data.

El procesador controla cuan rápido la data es aceptada por el convertidor. El reloj provee información sobre el tiempo a adquirir data. Sabiendo que no es suficiente grabar la lectura del sensor, sino también en que momento se debe medir.

La data es transferida del hardware a la memoria a través del acceso directo a memoria (DMA) o interrupciones. DMA es un controlado por el hardware y consecuentemente es rápido. Las interrupciones pueden ser lentas por el tiempo de espera entre que la tarjeta pida un servicio de interrupción y cuando la computadora responda.

El promedio máximo de adquisición esta determinado también por la velocidad del bus de la computadora.

3.6 Software

Independiente de que hardware se usa, se debe enviar información al hardware y recibir información del hardware. Se envía la información de la configuración a el hardware tal como la frecuencia de muestreo, y recibe información del hardware tal como data, mensajes de status y mensajes de error. Se podría también implementar el hardware con información de tal manera que se pueda integrar con otro hardware y con recursos de la computadora. Esta información se intercambia usando el software, ver la siguiente figura.

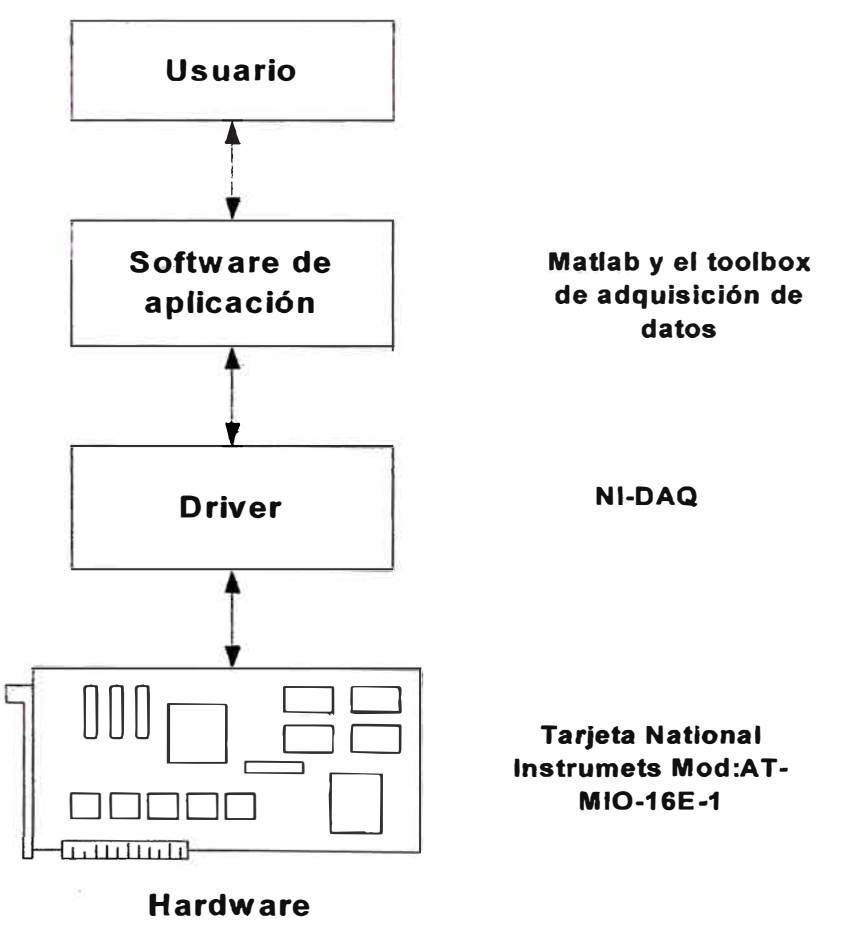


Figura 3.3 Relación entre el usuario, driver, aplicación y hardware

Hay dos tipos de software:

Driver.

Aplicación

Por ejemplo cuando se usa alguna tarjeta de adquisición de datos (AT-MIO-16E-1), la relación entre el usuario, el driver, la aplicación y el hardware se muestra en la Figura 3.3

3.6.1 Driver

Para el dispositivo de adquisición, hay un software driver asociado que debe usarse. El driver permite acceder y controlar las capacidades del hardware. Permite entre otras cosas:

Adquirir la data y obtener la data fuera de la tarjeta

Controlar la frecuencia a la cual la data es adquirida

Integrar el hardware de adquisición de datos con los recursos de la computadora tal como interrupción al procesador, DMA y memoria.

Integra el hardware de adquisición de datos con el hardware de acondicionamiento de la señal

Acceder a múltiples etapas de una tarjeta de adquisición de datos.

Acceder a múltiples tarjetas de adquisición de datos.

3.6.2 Aplicación

La aplicación provee una interfase a el driver. La aplicación básicamente permite:

Brinda información como el número de muestras adquiridas.

Genera eventos

Administra la data guardada en la memoria de la computadora

Acondiciona una señal

Diagrama la data

Con algunas aplicaciones se puede analizar la data. Matlab y el "Data Adquisition Toolbox" proveen estas capacidades y mas.

3.7 Etapa de la entrada análoga

Los dispositivos de adquisición de datos contienen etapas que convierten las señales de los sensores en números que la computadora puede leer. Estos dispositivos son llamados "Analog input subsystems" (AI etapas, A/D convertidores o ADCs). Después que la señal es digitalizada, se puede analizar, guardar en la memoria del sistema o almacenar en el disco duro.

La función de la etapa de entrada análoga es muestrear y cuantificar la señal análoga usando uno o mas canales. Se puede decir que un canal es un camino por el cual la señal del sensor viaja. Es común que una etapa de entrada análoga tenga 8 o 16 entradas disponibles. Después que la data es muestreada y cuantificada, se debe transferir a la memoria.

Las señales analógicas con continuas en tiempo y amplitud. Los muestreos toman porciones de la señal en tiempos discretos, mientras que la cuantización divide el voltaje (o corriente) en amplitudes discretas.

3.7.1 Muestreo

El muestreo toma porciones de la señal del sensor en tiempos discretos. Para la mayoría de las aplicaciones, el intervalo de tiempo entre muestras es constante, a no ser que se use un reloj externo.

Para la mayoría de convertidores, el muestreo es ejecutado por un circuito de muestra y espera (S/H). Un circuito S/H consiste en una señal del buffer almacenada seguida por un interruptor electrónico conectado a un condensador. La operación de un circuito S/H sigue los siguientes pasos:

1. En un determinado instante del muestreo, el switch conecta el buffer y condensador a la entrada.
2. El condensador es cargado al voltaje de entrada.
3. La carga es mantenida hasta que el convertidor A/D digitaliza la señal
4. Para múltiples conectados a un convertidor A/D, el paso previo es repetido para cada entrada del canal.
5. Todo el proceso es repetido para el siguiente instante de muestreo.

Un multiplexor, un circuito S/H y un convertidor A/D son ilustrados en la siguiente sección.

El hardware puede ser dividido en dos principales categorías basados en como se toman muestras de la señal: hardware de escaneo, el cual toma muestras de la señal de entrada secuencialmente, y el hardware de espera y muestreo simultaneo (simultaneous sample and hold SS/H), el cual toma muestras de todas las señales al mismo tiempo. Estos dos tipos de hardware son explicados a continuación.

3.7.2 Hardware del escáner

El hardware del escáner toma muestras de una sola señal de entrada, convierte la señal en valores digitales y luego repite el proceso por cada canal usado. En otras palabras, se toman muestras de cada canal secuencialmente. El escaneo ocurre cuando cada entrada en un grupo se toma muestras una vez.

Como se muestra, la mayoría de los dispositivos de adquisición de datos tienen un convertidor A/D que es multiplexado a múltiples entradas de canal.

Así, si se usa múltiples canales, estos canales no pueden ser muestreados simultáneamente y un vacío de tiempo existe entre muestras consecutivas de canales. Este vacío de tiempo es conocido como proceso de canal. Se puede pensar del proceso de canal como el tiempo que le toma a la etapa de entrada análoga, en tomar la siguiente muestra de un canal.

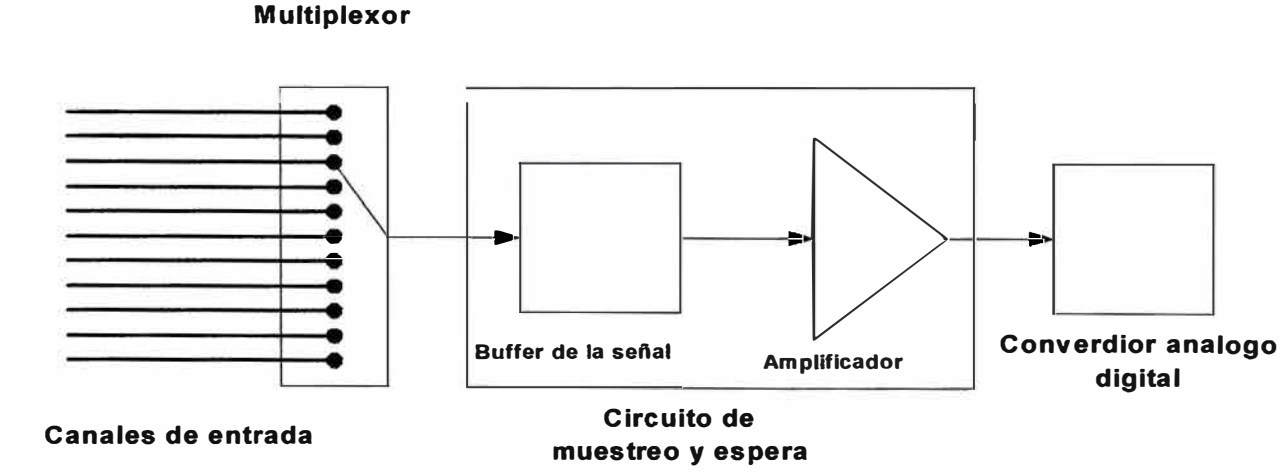


Figura 3.4 Convertidor analogo digital de multiples canales

Adicionalmente, el promedio máximo de toma de muestras al que el hardware esta graduado se aplica a un canal. Por este motivo, la graduación de la toma de muestras por canal esta dado por la siguiente fórmula:

A– Graduación máxima de la tarjeta

B= Número de canales escaneados

Graduación máxima de toma de muestras por canal = A/B

Generalmente, se puede tener esta graduación máxima sólo en condiciones ideales. En la práctica, la graduación de la toma de muestras depende de varias características de la etapa de entrada análoga, incluyendo el tiempo de instalación y la ganancia, así como el proceso del canal. El periodo de muestreo y el canal de proceso para una configuración con varios canales usando hardware de escaneo se muestra a continuación.

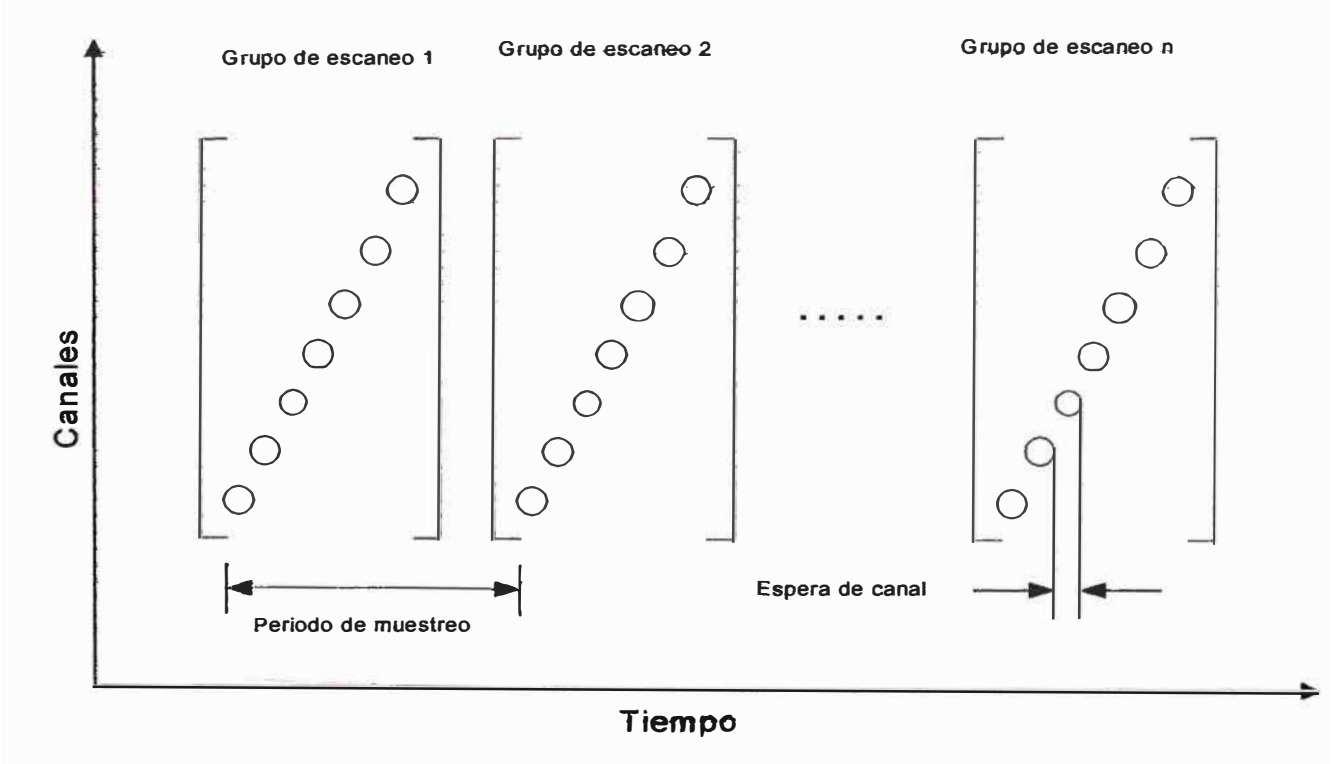


Figura 3.5 Diagrama del Periodo de Muestreo y Espera de Canal

Si no es permitido el canal de proceso en la aplicación , se debe usar el hardware que permita el muestreo simultaneo de canales. El muestreo simultaneo y memoria del hardware son discutidos en la próxima sección.

3.7.3 Muestreo simultaneo y memoria del hardware

Muestreo simultaneo y memoria del hardware muestrea todas las señales de entrada al mismo tiempo, memoriza los valores hasta que el convertidor A/D digitaliza todas las señales. Para sistemas avanzados, pueden implementarse separadamente el convertidor A/D para cada uno de los canales.

El periodo de muestreo para una configuración multicanal usando SS/H hardware es mostrado a continuación, en el podemos observar que no existe el tiempo de canal de proceso.

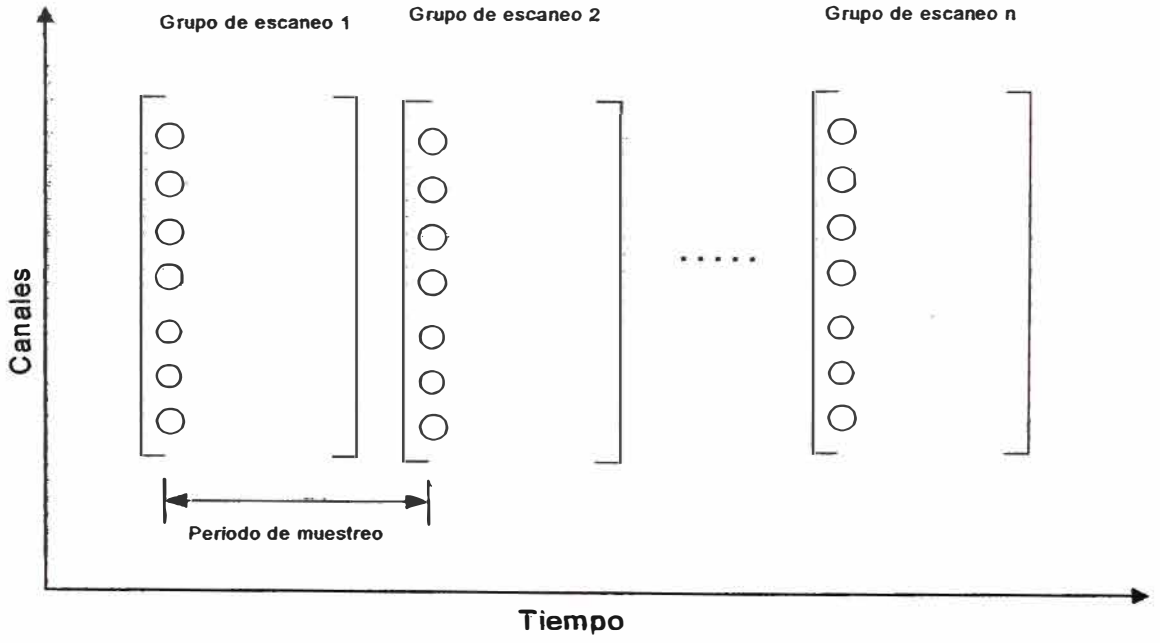


Figura 3.6 Diagrama del Muestreo Simultaneo

3.8 Cuantificación

La toma de una muestra se realiza en un instante de tiempo. Cuando se toma la muestra, el valor en voltaje que representa dicha muestra se convierte en un número

binario, de esta manera la computadora puede leerlo y procesarlo. La conversión de este valor de voltaje (amplitud) a un número binario es llamado cuantificación.

Durante la cuantificación. el convertidor A/D usa una cantidad limitada de valores igualmente espaciados para representar una señal análoga. La cantidad de valores diferentes está determinado por el número de bits usados para la conversión. Los mas modernos convertidores usan 12 o 16 bits. Típicamente, el convertidor selecciona el valor digital que esta mas cerca del valor muestreado. En la figura 3.7 se muestra una onda sinusoidal de 1Hz de frecuencia cuantificada con un convertidor de 3 bits

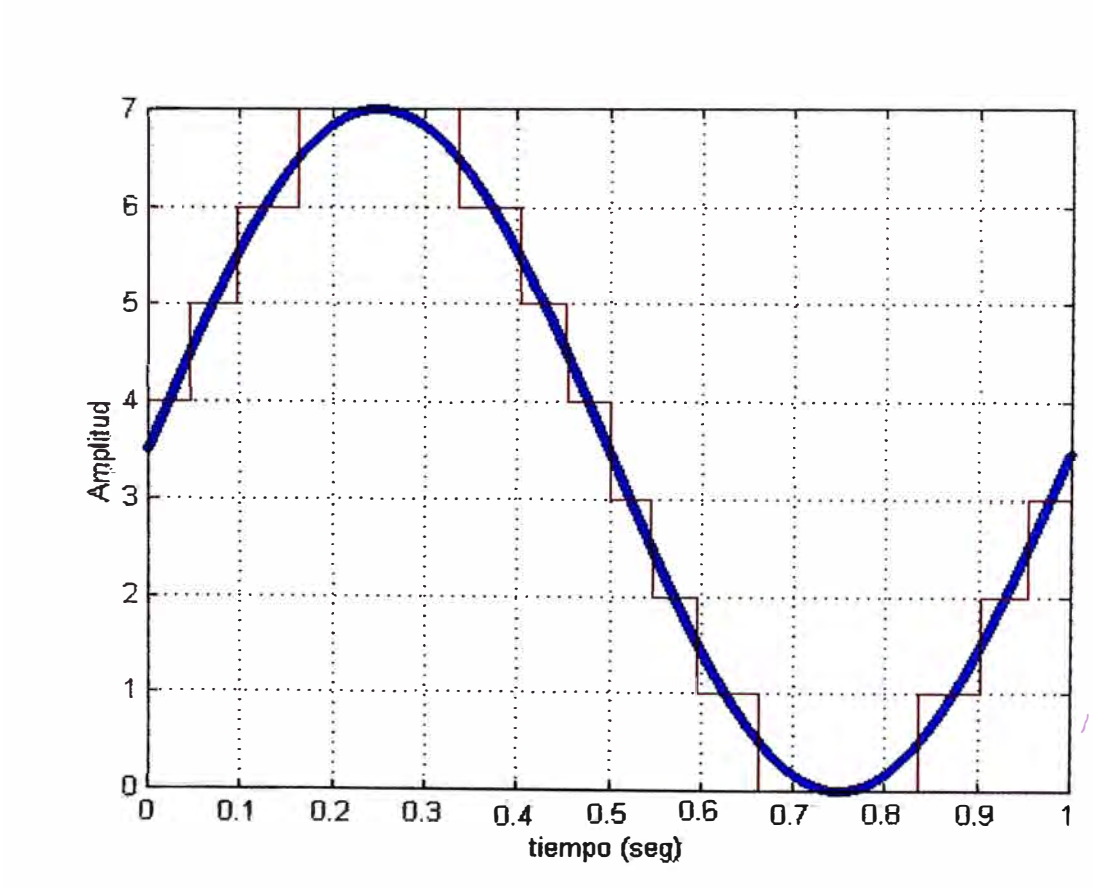


Figura 3.7 Resultado de cuantificación

El número de valores de la cuantificación esta dado por $2^3 = 8$, el mayor valor esta dado por $111 = 2^2 + 2^1 + 2^0 = 7$, y el menor valor esta dado por $000 = 0$.

3.8.1 Error de Cuantificación

La cuantificación de una señal continua esta asociado a un error. Idealmente, el máximo error de cuantificación es ±0.5 sobre el bit menos significativo, y sobre el máximo rango de entrada, el error promedio de cuantificación es cero.

Como se muestra en la figura 3.8, el error de cuantificación para la señal sinusoidal anterior, es calculada sustrayendo la señal original.

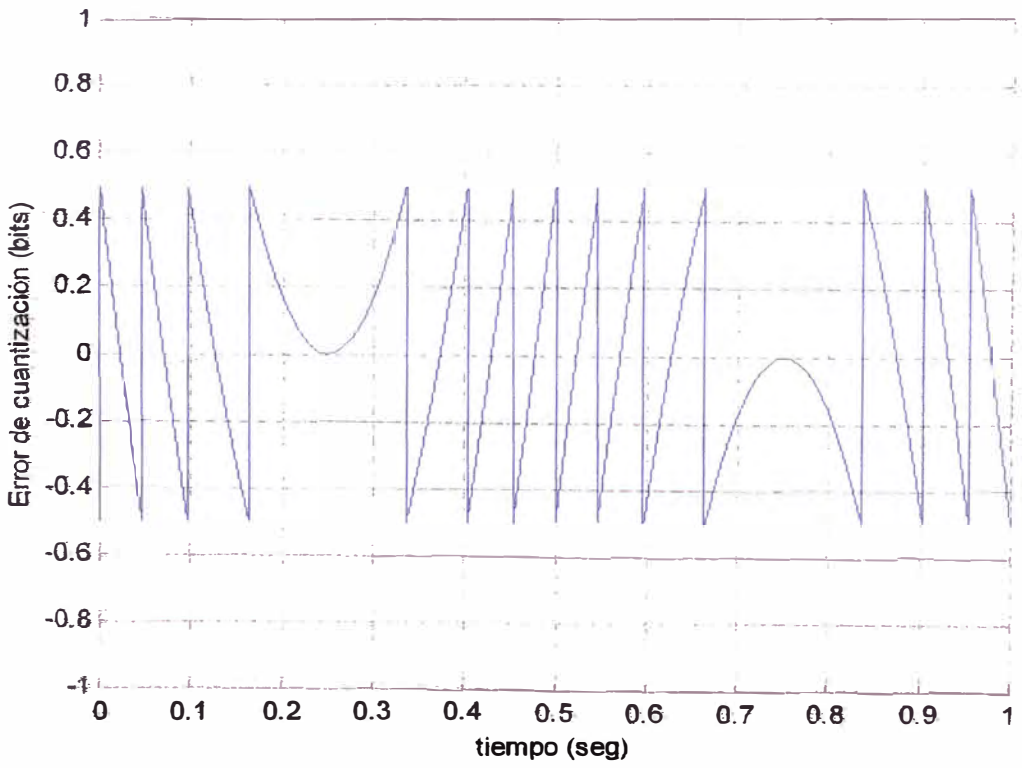


Figura 3.8 Error de cuantización

3.9 Configuración del Canal

Se puede configurar el canal de 2 maneras

Diferencial

Terminación simple

La elección dependerá si la señal de entrada esta con una referencia a tierra ó es flotante.

Una señal flotante usa una referencia de tierra aislada y no conectada a la tierra, en este caso del sistema de adquisición de datos. Como resultado, la señal de entrada y el hardware no están conectados a una misma referencia, lo cual puede causar que la señal de entrada exceda el rango de valores valido. Para superar este problema se debe conectar la señal a la tierra del hardware de adquisición de datos. Como ejemplos de señales flotantes tenemos a las termocuplas.

3.9.1 Entradas diferenciales

Cuando se configure el hardware para una entrada diferencial, hay 2 cables asociados con cada señal de entrada – uno por la entrada de la señal y otro por la señal de referencia (retorno). La medida es la diferencia en el voltaje entre ambos cables, el cual ayuda a reducir el ruido y cualquier voltaje que es común entre ambos terminales.

Como se muestra en la figura 3.9, la señal de entrada es conectado al terminal positivo (+) del amplificador y la señal de retorno es conectado al terminal negativo (-) del amplificador. El amplificador tiene un tercer terminal que permite a estas señales estar con una referencia a tierra.

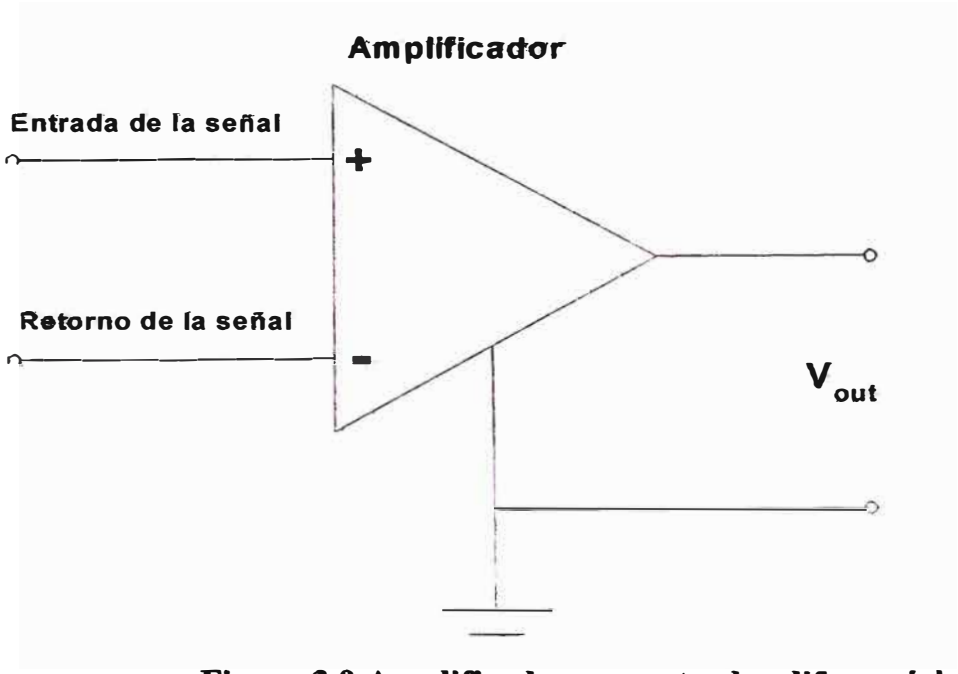


Figura 3.9 Amplificador con entradas diferenciales

Se recomienda que se use entradas diferenciales bajo las siguientes condiciones:

La señal de entrada es de bajo nivel (menor a 1 voltio)

La distancia a conectar la señal es mayor de 3 mts.

La señal de entrada requiere una referencia a tierra diferente.

La señal viaja a través de un medio muy ruidoso.

3.9.2 Entrada de terminales simples

Cuando se configura el hardware para una entrada simple. existe un terminal asociado con cada entrada de señal, y cada entrada de señal es conectada a la misma referencia (tierra).Las mediciones en este caso son mas susceptibles al ruido que las mediciones diferenciales a causa de las diferencias en los caminos de las señales.

Como se muestra en la figura 3.10, la señal de entrada es conectada al terminal positivo del amplificador y la tierra es conectado al terminal negativo del amplificador.

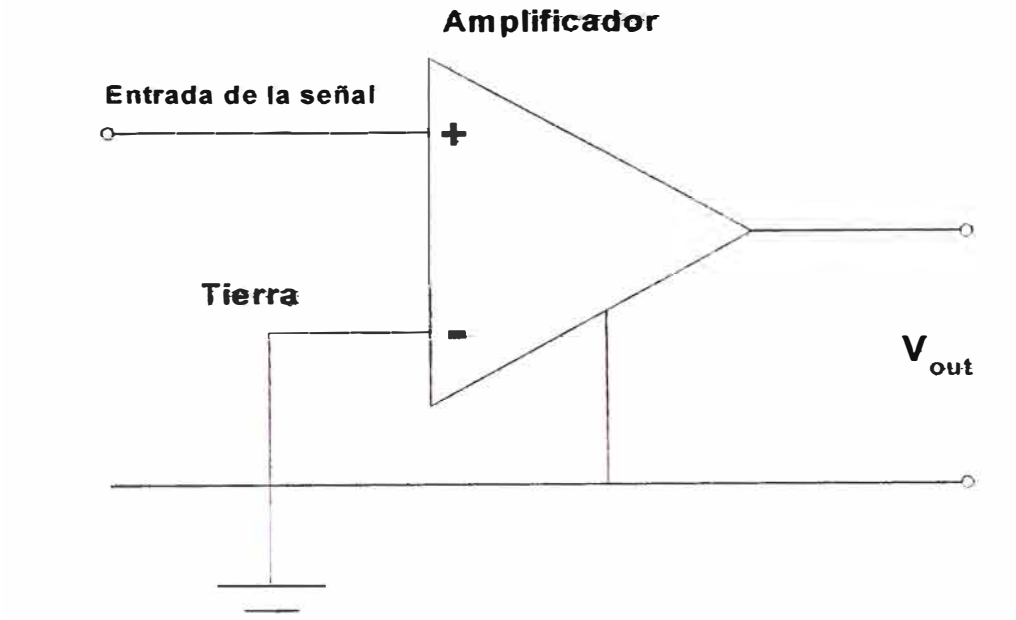


Figura 3.10 Amplificador con tierra común en la entrada y salida

Se sugiere que se use entradas con terminales simples bajo las siguientes condiciones.

La señal de entrada es de alto nivel (mayor a 1 voltio)

La distancia a conectar la señal es menor a 3 metros.

La señal de entrada puede compartir una referencia común con otra señal.

Se debe usar entradas diferenciales para cualquier señal que no cubran las recomendaciones anteriores.

3.10 Transferencia de los datos del hardware a la memoria

Para transferir los datos adquiridos del hardware a la memoria se pueden seguir los siguientes pasos:

La data adquirida es guardada en el hardware, primero en ingresar primero en salir (FIFO)

La data se transfiere del buffer FIFO a la memoria del sistema usando interrupciones o DMA

Estos pasos ocurren automáticamente. Normalmente, lo que se requiere del usuario, es una configuración inicial del hardware cuando se instala.

El Buffer FIFO

Es usado temporalmente para guardar la data adquirida. La data es temporalmente guardada hasta que se transfiere a la memoria del sistema. El proceso de trasferencia de datos de y hacia una entrada analoga FIFO es como sigue:

El buffer FIFO guarda los datos nuevos adquiridos a un muestreo constante.

Antes de que el buffer FIFO se llene, el software inicia la extracción de las muestras. Por ejemplo, una interrupción es generada cuando el FIFO esta medio lleno, e indica al software que extraiga las muestras tan rapido como pueda

Dado que el servicio de interrupciones esta programado por el controlador DMA, este puede tomar unos cuantos milisegundos , adicionalmente la data es guardada en el FIFO para un posterior recuperación. Para FIFO mas grande, tiempos de espera mas largos pueden ser tolerados.

Las muestras son trasferidas a la memoria del sistema a través del bus del sistema. Después que las muestras son transferidas, el software esta disponible para ejecutar otra tarea hasta que la próxima interrupción ocurra.

3.11 Interrupciones

El mas lento pero común método de mover los datos adquiridos a la memoria del sistema es por la generación de una señal de interrupción por la tarjeta. Esta señal puede ser generada cuando una muestra es adquirida o cuando múltiples muestras

son adquiridas. El proceso de transferencia de datos a la memoria del sistema via interrupciones es como sigue:

Cuando la data esta lista a ser transferida, el CPU detiene cualquier cosa que este ejecutando y corre una rutina de interrupción que guarda los registros actuales de la maquina y luego instruye a los registros a que accedan a la tarjeta.

La data es extraida de la tarjeta y colocada en la memoria del sistema

Los registros de la máquina son guardados y el CPU retorna al proceso original

En realidad el movimiento de la data es rápida, pero hay mucho procesamiento y demora guardando, instruyendo y reestableciendo la información de los registros. De esta manera, dependiendo del sistema, la transferencia de datos por interrupción puede no ser una buena elección cuando la velocidad de muestreo es mayor a los 5 Khz.

3.12 Acceso Directo a Memoria (DMA)

Acceso Directo a Memoria (DMA) es un sistema en el que las muestras son automáticamente guardadas en la memoria del sistema, mientras que el procesador esta realizando otra tarea. El proceso de trasferencia de datos vía DMA es como sigue:

Cuando la data esta lista a transferir, la tarjeta ordena al controlador DMA que coloque la data en la memoria del sistema tan pronto como sea posible.

Tan pronto como el CPU esta listo (lo cual es muy rápido), este detiene su interacción con la tarjeta de adquisición de datos y el controlador del DMA mueve la data directamente a la memoria.

El controlador de DMA esta listo para la siguiente muestra apuntando a la siguiente posición de memoria.

Los pasos previos son repetidos indefinidamente. con la data moviéndose a una localización de memoria continuamente. No acción entre el CPU y la tarjeta es requerida.

La computadora puede trabajar con diferentes canales DMA. Dependiendo de tu aplicación, se puede usan uno o mas de estos canales. Por ejemplo, simultáneamente una entrada y una salida con una tarjeta de sonido requiere un canal DMA para la entrada y otro DMA para la salida.

3.13 Calidad en la medición

Para la mayoría de las aplicaciones de adquisición de datos, se requiere que la señal sea medida a determinada velocidad, esta velocidad depende del tipo de fenómeno a evaluar, si es una señal que varía rápidamente una velocidad de muestreo elevada será necesaria y viceversa.

Para realizar mediciones de alta calidad debemos seguir las siguientes reglas:

Maximizar la precisión y exactitud.

Reducir el ruido

Los rangos del sensor y del A/D deben coincidir.

3.14 Exactitud y Precisión

Estos dos factores dependen del sistema de adquisición de datos y puede estar limitado por factores tales como la resolución de la tarjeta y/o el ruido ambiental.

En general la exactitud de una medición queda determinado por cuan cerca esta del valor verdadero, por consiguiente, indica cuan correcta es una medición. La precisión de una medición, se refleja en cuan exacto es el resultado sin tener una referencia de lo que significa este resultado. La precisión relativa indica incertidumbre en la medición como una fracción del resultado. Ambos factores los podemos apreciar en el siguiente gráfico:

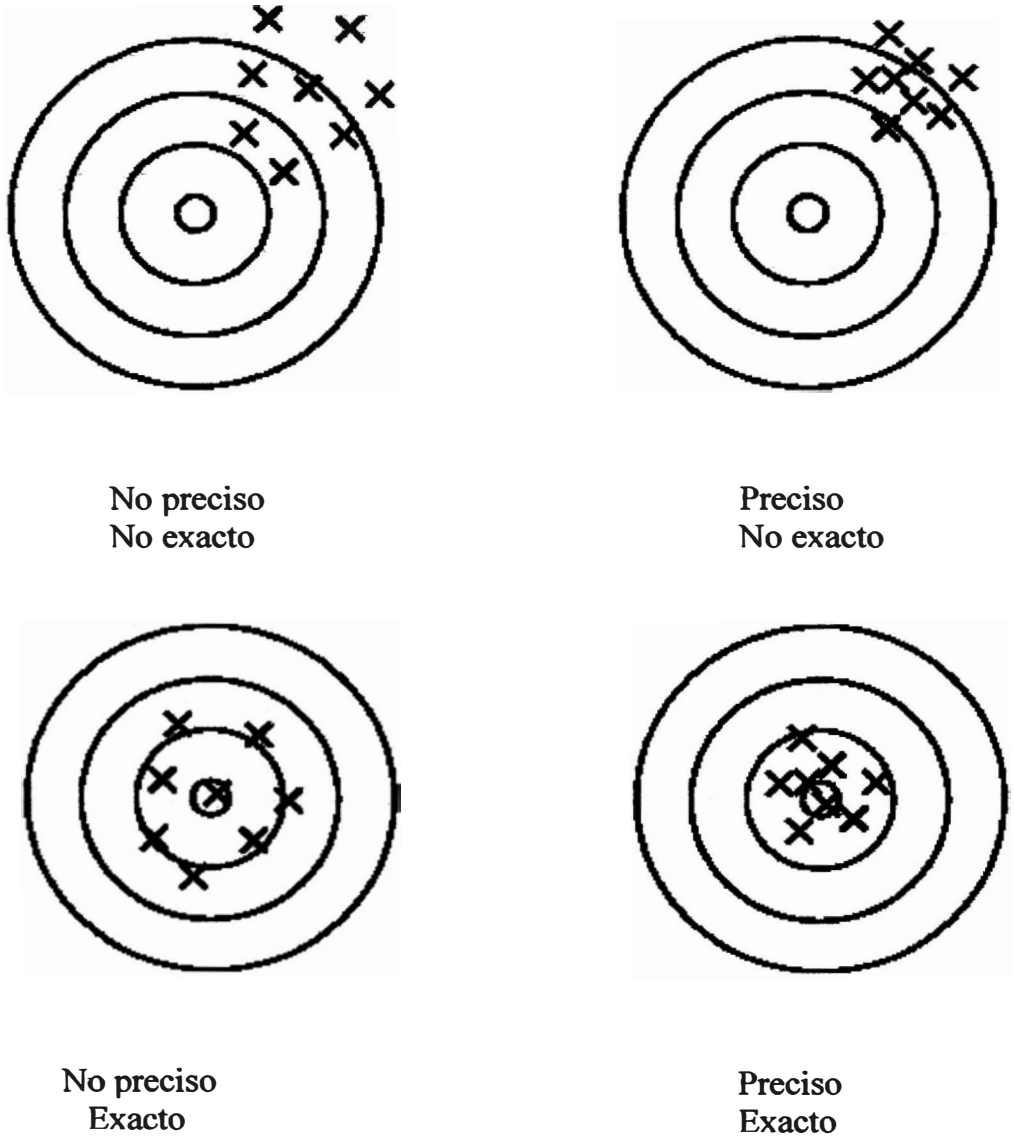


Figura 3.11 Diagramas de Precision y Exactitud

3.14.1 Exactitud

La exactitud esa definida como un relación entre la cantidad medida y la cantidad del verdadero valor. Cada componente que aparece en el camino de la señal análoga afecta la exactitud y funcionamiento del sistema.

Para el hardware de la adquisición de datos, la exactitud esta expresada como un porcentaje o fracción del menor bit significativo. Bajo condiciones ideales, la exactitud es típicamente ±0.5 LSB. Por consiguiente, un convertidor de 12 bits tiene solamente 11 bits utilizables.

3.14.2 Precisión

El número de bits usados para representar una señal análoga determina la precisión (resolución) del dispositivo. Cuanto mayor es la cantidad de bits que maneja la tarjeta, la mayor precisión se obtendrá en la medición. Una alta resolución divide el rango de entrada en una mayor cantidad de divisiones permitiendo la detección de menores valores de voltaje. Una baja precisión, disminuye la resolución del dispositivo dividiendo el rango de entrada en menores divisiones aumentando el valor de voltaje detectable.

La precisión del sistema de adquisición de datos es normalmente determinado por el convertidor A/D, y es especificado por el número de bits usados para representar la señal. La mayoría de tarjetas usan 12 o 16 bits. La precisión de la medición esta dado por:

$$\text{Precisión} = 1 / 2^{\text{número de bits}} \qquad 3.1$$

La precisión en voltaje esta dado por:

$$\text{Precisión} = \text{Rango de Voltaje} / 2^{\text{número de bits}} \qquad 3.2$$

Por ejemplo, si se esta usando un convertidor A/D de 12 bits configurado para un rango de 10 voltios, entonces:

$$\text{Precisión} = 10 \text{ voltios} / 2^{12} \qquad 3.3$$

Esto significa que el convertidor puede detectar diferencias de voltaje del nivel de 0.00244 volts (2.44 mV).

3.15 Relación entre Rango, Ganancia y Precisión

Cuando se configure el rango de entrada y ganancia de la entrada analógica del sistema, el resultado final debe maximizar la resolución de la medida y minimizar la posibilidad de sobrepasar el rango. El rango real de entrada esta dado por la siguiente fórmula:

$$\text{Rango real de entrada} = \text{Rango de entrada} / \text{Ganancia} \qquad 3.4$$

La relación entre ganancia, rango real de entrada y precisión para una señal polar y bipolar teniendo un rango de entrada de 10V se muestra en la tabla 3.2.

Como se muestra en la tabla, la ganancia afecta la precisión de la medida. Si se selecciona una ganancia que disminuye el rango real de entrada, entonces la precisión se incrementa. Inversamente, si se selecciona una ganancia que disminuye el rango real de entrada, entonces la precisión disminuye. Esto es porque el rango real de entrada varia pero el número de bits del convertidor A/D permanece fijo.

Con las Herramientas de Adquisición de Datos no se necesita especificar el rango y ganancia, sólo se debe especificar el rango real de entrada deseado.

Rango de entrada	Ganancia	Rango real entrada	Precisión (12 bits A/D)
0 a 10 V	1.0	0 a 10 V	2.44 mV
	2.0	0 a 5 V	1.22 mV
	5.0	0 a 2 V	0.488 mV
	10.0	0 a 1 V	0.244 mV
-5 a 5 V	0.5	-10 a 10 V	4.88 mV
	1.0	-5 a 5 V	2.44 mV
	2.0	-2.5 a 2.5 V	1.22 mV
	5.0	-1.0 a 1.0 V	0.488 mV
	10.0	-0.5 a 0.5 V	0.244 mV

Tabla 3.2 Relación entre Rango de entrada, ganancia y precisión

3.16 Ruido

Ruido es considerado como una medición que no ha de ser considerado como parte del fenómeno de interés. Ruido puede ser generado dentro de los componentes eléctricos del amplificador de entrada (ruido interno), o puede ser añadido a la señal cuando viaja a través de los cables al amplificador (ruido externo). Se mencionan algunas técnicas que se pueden usar para reducir el efecto del ruido.

3.16.1 Extrayendo el ruido Interno

El ruido interno aparece de los efectos térmicos en el amplificador. Normalmente los amplificadores generan unos microvoltios de ruido interno, el cual limita la

resolución de la señal a estos niveles. La cantidad de ruido adicionado a la señal depende del ancho de bando el amplificador.

Para reducir el ruido interno, se debe seleccionar un amplificador con un ancho de banda que coincida con el de la señal de entrada.

3.16.2 Eliminando el ruido externo

El ruido externo tiene varias fuentes. Por ejemplo, muchos experimentos de adquisición de datos están sujetos al ruido de 60 Hz. generado por las fuentes de alimentación. Este tipo de ruido aparece como una interferencia sinusoidal en el circuito de medida. Otro fuente común de interferencia son los flourecentes, estas fuentes generan un arco que es el doble de la línea eléctrica (120Hz.)

El ruido es adicionado al circuito de adquisición de estas fuentes externas porque la señal funciona como antena, capturando la actividad eléctrica del medio. La mayor parte de este ruido es común a la señal de ambos terminales. Para retirar la mayor parte de este ruido se debe:

Configurar los canales de entrada en modo diferencial.

Usar los cables de la señal tan cortos como sea posible

Mantener los cables de señal tan lejos como sea posible del medio donde exista actividad eléctrica.

3.16.3 Filtrado

El Filtrado también reduce el ruido de la señal. Para muchas aplicaciones de adquisición de datos, un filtro pasa bajo es beneficioso. Como el nombre lo sugiere, un filtro pasa bajo deja pasar las bajas frecuencias y atenúa los componentes altos. La

frecuencia de corte del filtro debe ir acorde con las frecuencias presentes en la señal de interés y el muestreo usado por el convertidor A/D.

Un filtro pasa bajo que es usado para prevenir que frecuencias elevadas introduzcan distorsión en la señal digitalizada es conocido como filtro antisolapamiento si la frecuencia de corte ocurre a la frecuencia de Nyquist. Esto es, el filtro remueve frecuencias mayores que la mitad de la frecuencia de muestreo. Estos filtros generalmente tienen una frecuencia de corte precisa que el normal de los filtros pasa bajos usados para condicionar una señal. Los filtros antisolapamiento son especificados de acuerdo a la velocidad de muestreo del sistema y debería haber uno por cada señal de entrada.

3.17 Relación entre el Rango del Sensor y el rango del Convertidor A/D

Cuando la data del sensor es digitalizada por un convertidor A/D, se debe tener cuidado con los siguientes casos.

El rango esperado de la data producida por el sensor. Este rango depende del fenómeno físico que se esta midiendo y el rango de salida del sensor.

El rango del convertidor A/D. Para muchos dispositivos, el rango del hardware es especificado por la ganancia y polaridad

Se debe seleccionar el sensor y rango del hardware de tal manera que se obtenga la máxima precisión, y el rango dinámico de la señal de entrada debe estar cubierta.

Por ejemplo, suponga que se esta usando un micrófono con un rango dinámico de 20dB a 140 dB y una sensibilidad de salida de 50mV/Pa. Si se esta midiendo el ruido de la calle, entonces se debe esperar que el nivel de sonido no superará los 80 dB, el

cual corresponde a la magnitud de la presión de un sonido de 200 mPa y un voltaje de salida de un micrófono de 10mV. Bajo estas condiciones, se debe establecer el rango de entrada del sistema de adquisición de datos para una amplitud máxima de señal de 10mV, o un poco mas.

3.18 Muestreo de la señal

Cada vez que se muestrea una señal, alguna información es perdida. La clave esta en muestrear a una velocidad tal que la señal de interés sea bien caracterizada y la cantidad de información perdida sea mínima.

Si la velocidad de muestreo es baja, entonces se produce el asolapamiento. Asolapamiento puede ocurrir en señales de variación rápida y lenta. Por ejemplo, supongamos que se mide la temperatura una vez por minuto. Si el sistema de adquisición esta recogiendo una señal ruido de 60 Hz, entonces ese ruido aparecerá como un nivel de ruido constante si se esta muestreando a 30 Hz.

Asolapamiento ocurre cuando la señal muestreada contiene componentes de frecuencia mayores que la mitad de la frecuencia de muestreo. Los componentes de frecuencia pueden originar del ruido, en tal caso se debe condicionar la señal usando un filtro. La regla para prevenir asolapamiento esta dada por el Teorema de Nyquist el cual dice

> Una señal análoga puede ser únicamente reconstruida, sin error, a partir de las muestras tomadas a iguales intervalos.
>
> La velocidad de muestreo debe ser igual o mayor que el doble del mayor componente de frecuencia de la señal. La frecuencia igual a la mitad de la velocidad de muestreo es llamada Frecuencia de Nyquist.

Sin embargo, si la señal de entrada esta mezclada con ruido, entonces el asolapamiento puede ocurrir.

Por ejemplo, suponga que el convertidor A/D se configura a una velocidad de muestreo de 4 Hz, y la señal de interés es de 1 Hz. Dado que la frecuencia de la señal es ¼ de la de muestreo, entonces de acuerdo al teorema de Nyquist, la señal se caracterizará completamente.

Sin embargo, si una señal de 5 Hz esta presente, entonces estas 2 señales no podrán distinguirse. En otras palabras, la señal de 1 Hz produce las mismas muestras que la de 5 Hz cuando la velocidad de muestreo es de 4 Hz. Ver figura 3.12

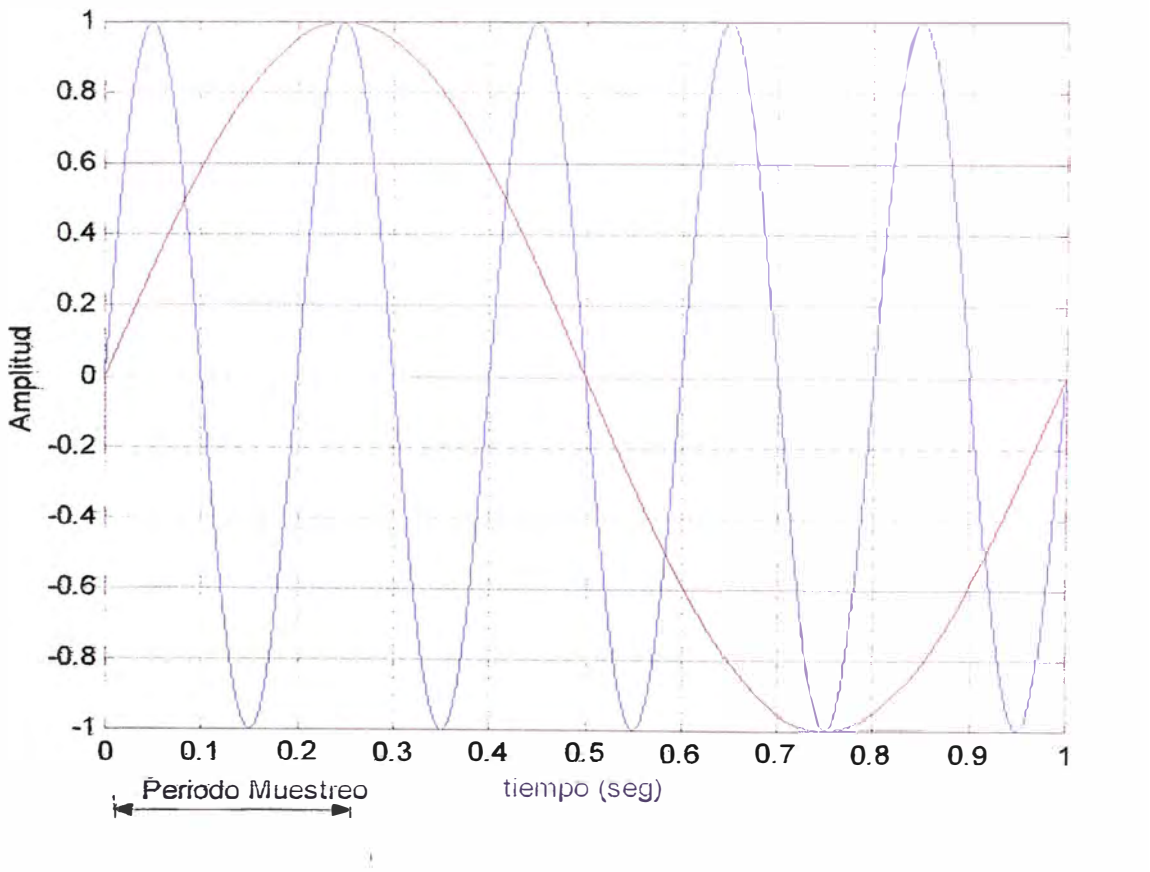


Figura 3.12 Señales de 1 Hz y 5 Hz con frecuencia de muestreo de 4 Hz

En la práctica, será necesario condicionar la señal filtrándola sobre los componentes mas altos de frecuencia.

A pesar de que las muestras aparentan representar una señal sinusoidal con una frecuencia de ¼ de la velocidad de muestreo, la señal real puede ser cualquier onda con una frecuencia de:

$$(n \pm 0.25) \times (\text{Velocidad de muestreo})$$

donde n es cero o cualquier entero positivo. Para este ejemplo, la señal real puede estar a una frecuencia de 3 Hz., 5 Hz., 7 Hz., 9 Hz. y así sucesivamente. La relación 0.25 x (Velocidad de muestreo) puede ser otra frecuencia mas. En otras palabras, el asolapamiento ocurre cuando una frecuencia asume la identidad de otra frecuencia.

Si se muestrea una señal de entrada al menos 2 veces mas rápido que el mayor componente de frecuencia, entonces esta señal será únicamente caracterizada, pero esta velocidad no imitará la forma de onda . Como se muestra, para obtener una figura precisa, se necesita una velocidad de muestreo de 10 a 20 veces la mayor frecuencia.

Como se observa en la parte superior de la figura 3.13, una velocidad baja de muestreo produce una muestra de señal que aparenta ser una onda triangular. En la parte inferior de la figura 3.13, una alta fidelidad de la señal muestreada se produce cuando la velocidad de muestreo es mayor, en este caso la señal se asemeja a una onda sinusoidal.

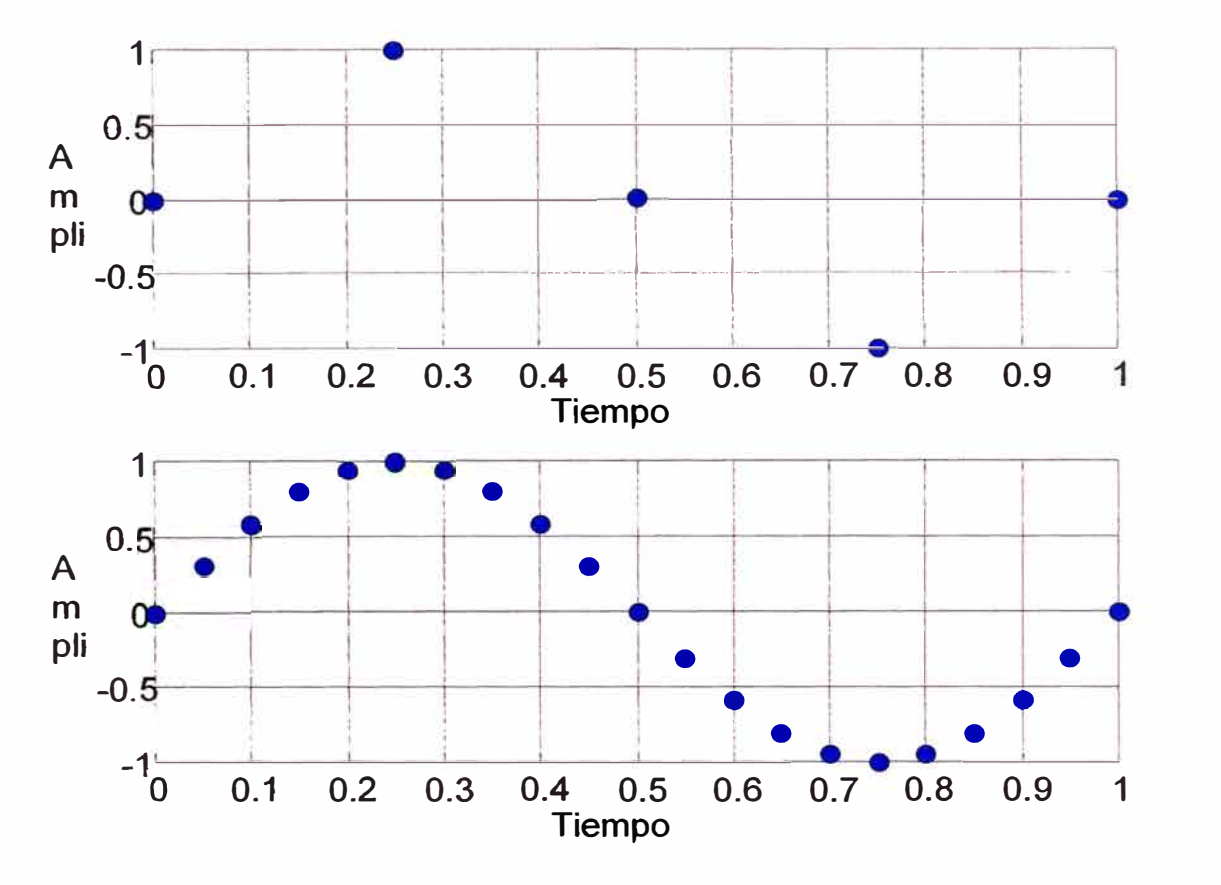


Figura 3.13 Muestreo de una señal con diferentes frecuencias de muestreo

3.19 Como se puede eliminar el asolapamiento

La principal consideración relacionada con el asolapamiento son la velocidad de muestreo del convertidor A/D y las frecuencias presentes en la señal muestreada. Para eliminar el asolapamiento se debe considerar:

Establecer un ancho de banda útil de la medición.

Seleccionar un sensor con el suficiente ancho de banda

Seleccionar un filtro análogo pasa bajo anti asolapamiento que pueda eliminar todas las frecuencias que excedan este ancho de banda.

Muestrear la señal a una velocidad por lo menos dos veces la frecuencia de corte superior del filtro.

CAPITULO IV

ENCRIPTACIÓN DE VOZ

4.1 Formas de encriptación de voz

La señal de voz tiene características definidas como son:

Rango de frecuencias

Amplitud

Frecuencia característica

Estructura armónica.

La encriptación de la señal de voz se puede realizar siguiendo diferentes técnicas las mismas que manipulan las características originales de la señal de voz.

4.1.1 Variación del espectro

Se puede sugerir las siguientes alternativas:

Inversión del espectro.

Manipulación y reordenamiento de partes de este espectro.

4.1.2 Variación de la amplitud

La señal se varía bajo cierto patrón en amplitud. Este sistema no es muy seguro ya que todavía puede ser entendible el mensaje.

4.1.3 Variación de muestras de la señal

Cuando se adquiere una señal, esta se realiza a determinada frecuencia de muestreo. Cada muestra forma parte de la señal original las mismas que pueden ser manipuladas en orden y/o amplitud, sin embargo estas variaciones deben seguir una secuencia de tal manera que en el lado receptor pueda desencriptarse.

4.1.4 Tratamiento de la señal de voz digitalizada

Cuando la voz se ha digitalizado se pueden aplicar los diferentes algoritmos que se disponen actualmente para poder encriptar la señal que esta como una secuencia de bits. Este sistema es el mas seguro y se aplica actualmente a en equipos de comunicación de uso militar, policial y gubernamental. Todos estos sistemas utilizan los DSP o procesadores digitales de señal con la finalidad de que el tiempo que se tarda en procesar la señal y encriptarla (retardo) no sea perceptible al escucharlo.

4.2 Materiales y Equipo a utilizarse.

La presente simulación se realizó utilizando como herramienta principal a la computadora, la que deberá estar equipada con una tarjeta de sonido, parlantes, micrófono. Sobre el sistema operativo se instaló MATLAB, el mismo que deberá contar con el toolbox de Adquisición de datos y comunicaciones, donde se encuentran las funciones que utilizamos en la simulación. Las características de la

computadora con la que se simuló este trabajo tiene un procesador Celerón de 1.3GHz. Memoria RAM de 192 Mb. y disco duro de 40 Gb.

4.3 Diagrama de bloques del sistema de encriptación de voz

En la figura 4.1 se muestran las diferentes etapas por la que atraviesa la señal de voz.

4.4 Etapa de adquisición de datos

En esta etapa intervienen componentes de hardware y software.

El hardware esta compuesto por el micrófono, tarjeta de sonido y computadora. El micrófono se encargará de transformar las vibraciones que produce la voz en señales eléctricas, estas a su vez son procesadas por la tarjeta de sonido almacenando los datos en la memoria RAM de la computadora.

En Matlab existe la Caja de herramientas de adquisición de datos que nos permite adquirir la señal con los parámetros que mas nos interese, esto es definiendo objetos. Esta característica nos permite definir y controlar los parámetros de la adquisición de datos como son: frecuencia de muestreo, tiempo en que se tomarán las muestras, cantidad de muestras que se tomarán, momento en que se iniciará la toma de muestras, la forma en que se iniciará la toma de muestras, la tarjeta de adquisición que se usará, en este caso la tarjeta de sonido. Podemos definir y tener la información del objeto "ai" de la siguiente manera:

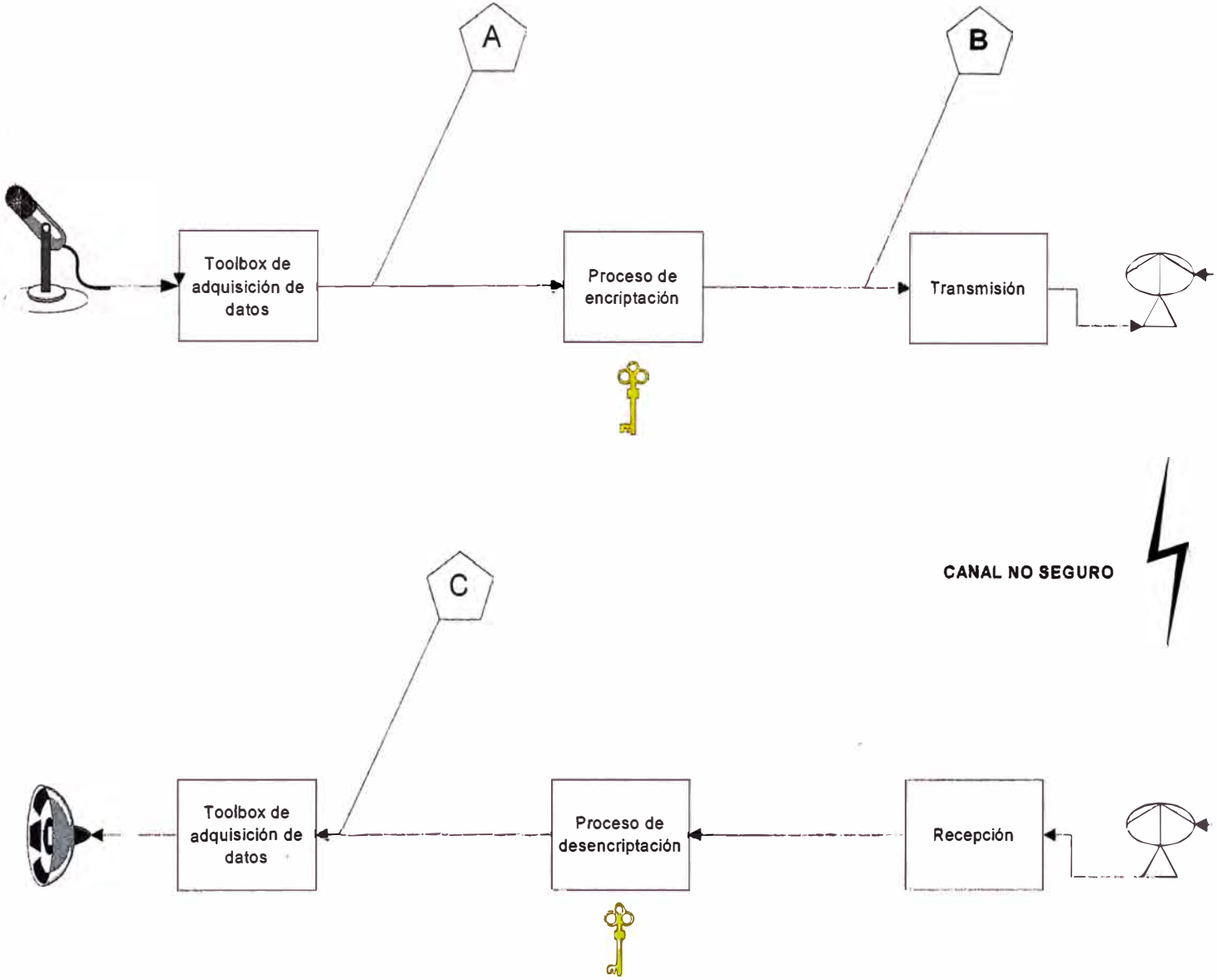


Figura 4.1 Diagrama de bloques del proceso transmisión de voz

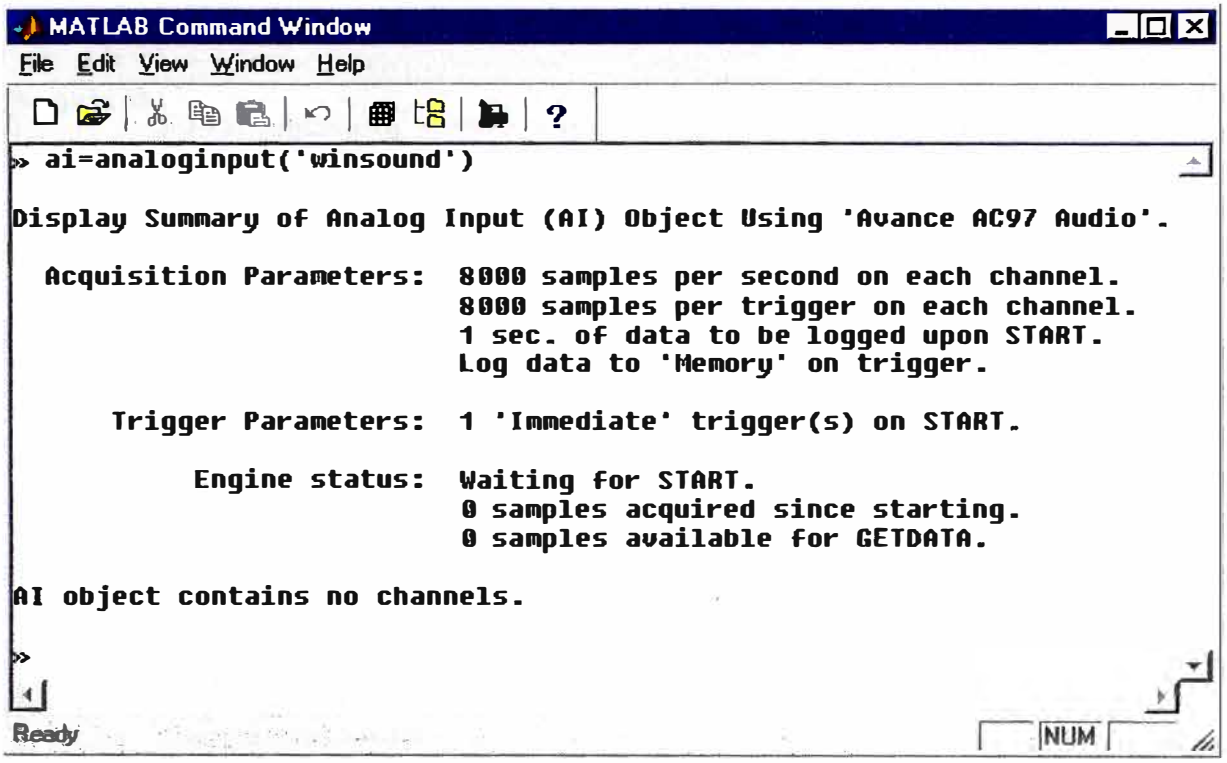


Figura 4.2 Información de la configuración de la tarjeta de sonido

Como podemos observar, existen parámetros los cuales ya se dan por defecto, tales como la frecuencia de muestreo, cantidad de muestras por canal y por tanto el tiempo que se tomará muestras, donde se guardarán los datos adquiridos y la forma en que se iniciará la toma de datos.

En el siguiente gráfico se puede observar los diferentes parámetros con los que podemos adquirir los datos. De todos estos parámetros nosotros aprovecharemos los referidos a la cantidad de muestras, tiempo de muestreo, frecuencia de muestreo y la forma en que se iniciará la toma de datos. Este último parámetro nos permite poner en pausa la rutina y reiniciarla presionando cualquier tecla, inmediatamente se inicia el proceso de guardado de las muestras para el posterior tratamiento de las mismas.

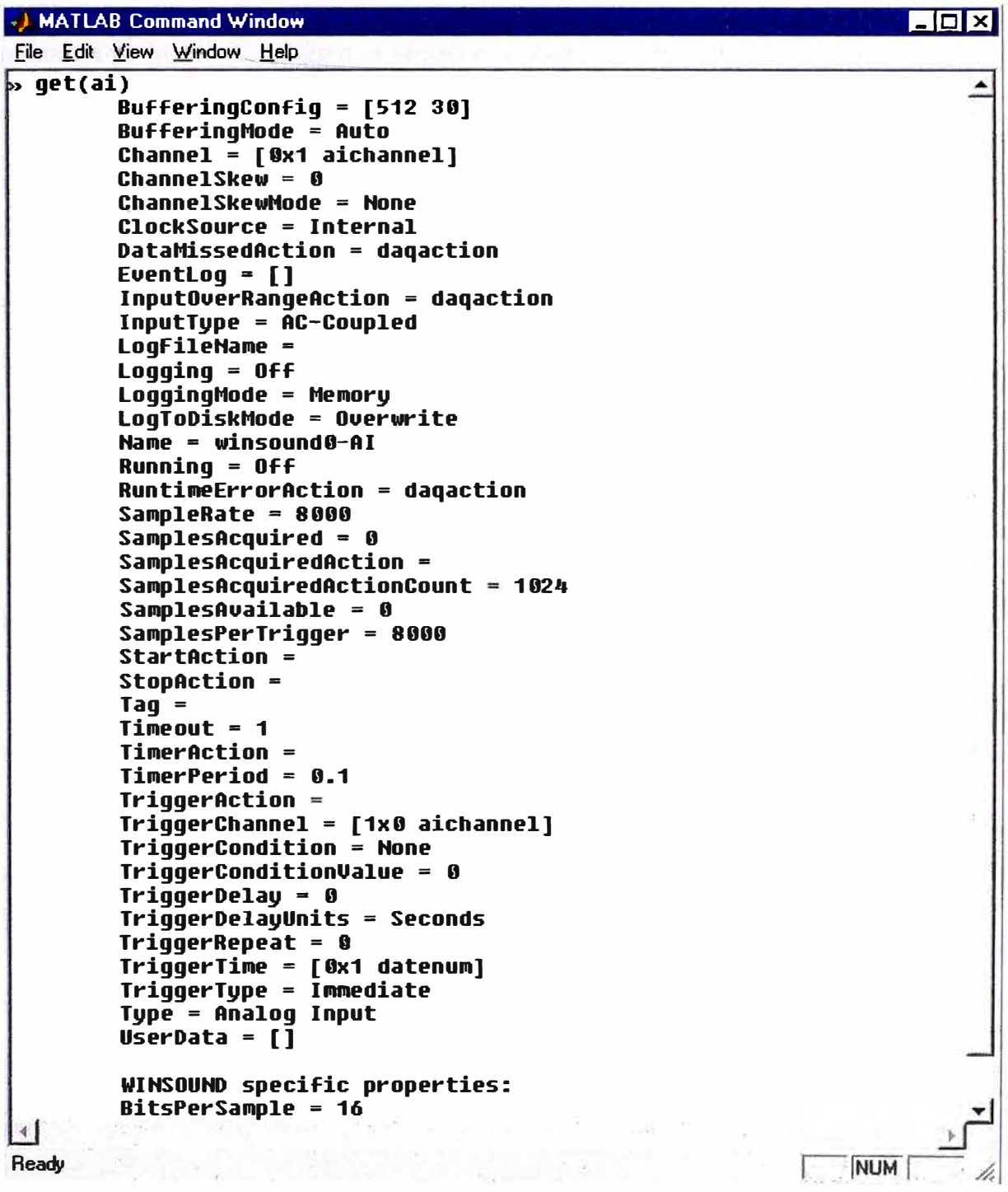


Figura 4.3 Parámetros configurables del objeto "ai"

Para poder trabajar con las muestras adquiridas, lo que está almacenado en la memoria de la computadora o el objeto que posee las muestras, debemos trasladarlas a una matriz columna donde podemos realizar el análisis correspondiente e iniciar el proceso de encriptación.

4.5 Algoritmo de encriptación de Voz

Para comprender bien el tratamiento que se le dará a la señal debemos aclarar que cuando la señal es adquirida, lo realizamos a través del muestreo de la señal. Primero determinamos la frecuencia de muestreo "f" y luego a intervalos de tiempo iguales a 1/f se toman las muestras. Estas muestras se almacenan en el objeto que definiremos para luego extraerlas a un vector, es decir cada una de las muestras tendrá un valor el cual ocupa una posición única en el vector. Esta posición deberá mantenerse con el objetivo de reconstruir posteriormente la señal. A este vector lo tratamos como una matriz, en nuestro caso el vector tiene una cantidad determinada de elementos que nos permite reorganizarla en una matriz cuadrada.

Las muestras que obtenemos almacenado en la memoria RAM, se encuentran normalizadas y cada una de los elementos del vector varían entre –1 +1. Para el procesamiento de datos en la computadora necesitamos trabajar con cantidades finitas. La computadora al entregarnos esta señal también ha debido de procesar internamente la señal, primero cuantizandola y posteriormente digitalizándola.

Para poder trabajar las muestras obtenidas y almacenadas en RAM, realizaremos el proceso de cuantización y a continuación la digitalización.

El secreto de nuestro algoritmo de encriptación no sólo residirá en la clave de encriptación sino también en el algoritmo que desarrollaremos.

DIAGRAMA DE FLUJO DEL PROCESO DE ENCRIPTADO

Figura 4.4 Diagrama Flujo Proceso Encriptado

Para nuestro propósito, el cual es simular la encriptación de la voz, trabajaremos en base a palabras, una palabra puede ser expresada de 1 a 2 segundos por lo que adquiriremos 2 segundos de datos. Esta palabra será la que será encriptada y posteriormente desencriptada.

Dentro del algoritmo hemos incluido puntos de análisis de la señal con el objetivo de poder analizar la variación que tiene la forma de la señal, el sonido que tiene la señal y el espectro de frecuencias. Los puntos en los cuales se compara el progreso de la señal son los siguientes:

- Señal de entrada
- Señal encriptada
- Señal de salida

Para poder calcular y visualizar el espectro de la señal, utilizaremos los algoritmos de la Transformada de Fourier Discreta (DFT) que se define como;

$$X_{DFT}[k] = \sum_{n=0}^{N-1} x[n] e^{-j 2\pi k n / N}, k = 0,1,2,\ldots, N-1 \qquad 4.1$$

En donde x[n]= Señal muestreada

El algoritmo DFT consume muchos recursos cuando se computa por lo que es necesario utilizar algoritmos que sean eficientes y no demoren mucho su calculo y no consuman recursos, para ello es necesario que la cantidad de muestras sea una potencia de dos ($N=2^M$), donde M es un entero. Con este requisito, que en nuestro caso se cumple, podemos recurrir al algoritmo de la Transformada Rápida de Fourier (FFT)

4.5.1 Adquisición de voz

El rango de frecuencias de la voz sobre la cual trabajaremos será entre los 300Hz y los 3.4KHz. Por el teorema de Nyquist:

$$f_m > 2f_s \quad 4.2$$

Donde f_m = frecuencia de muestreo

f_s = frecuencia de la señal

da como resultado que la frecuencia de muestreo debe ser como mínimo:

$$f_m = 6.8\text{KHz}. \quad 4.3$$

En nuestro caso y para efectos prácticos del algoritmo desarrollado elegiremos la frecuencia de muestreo de:

$$f_m = 8192\ \text{Hz}. \quad 4.4$$

En el algoritmo y como proceso de la encriptación que se ha desarrollado las muestras que adquiridas son colocadas en una matriz cuadrada, para que puedan ingresar el tamaño de la matriz n x n, donde n es un entero positivo, debe ser tal que:

$$n^2 = \text{número de muestras adquiridas} \quad 4.5$$

$$N = \text{número de muestras adquiridas} \quad 4.6$$

Para que se pueda cumplir la relación anterior de tal manera que n exista el número de muestras será de:

$$2 \times 8192 = 16384 = 128^2$$

entonces:

$$n = 128 \quad 4.7$$

En el algoritmo se incluye la visualización de los espectros de frecuencia de la señal antes de encriptar, después de encriptar y la señal desencriptada para los cual se

requiere que la cantidad de muestras a analizar sea una potencia de 2, requisito que cumple:

$$N = 16384$$

$$N = 2^{14} \tag{4.8}$$

Esto nos permite usar el algoritmo de la Transformada Rápida de Fourier (FFT).

4.5.2 Mezcla de muestras

Para producir la mezcla de posiciones en el vector generado por la adquisición de datos, utilizaremos algunos comandos que nos permitirán desordenar las posiciones con la garantía de que las posiciones originales se puedan reconstruir en el lado de la recepción.

Sea el siguiente vector el obtenido en la etapa de adquisición de datos:

$$\mathbf{data} = [a \quad b \quad c \quad d \quad e \quad f \quad g \quad h \quad i \quad j \quad k \quad l \quad m \quad n \quad ñ \quad o] \tag{4.9}$$

Primero invertimos las posiciones de las muestras

$$\text{data2} = [\text{o} \quad \text{ñ} \quad \text{n} \quad \text{m} \quad \text{l} \quad \text{k} \quad \text{j} \quad \text{i} \quad \text{h} \quad \text{g} \quad \text{f} \quad \text{e} \quad \text{d} \quad \text{c} \quad \text{b} \quad \text{a}] \tag{4.10}$$

Convertimos data2 en una matriz cuadrada

$$\mathbf{data3} = \begin{bmatrix} \mathbf{o} & \mathbf{ñ} & \mathbf{n} & \mathbf{m} \\ \mathbf{l} & \mathbf{k} & \mathbf{j} & \mathbf{i} \\ \mathbf{h} & \mathbf{g} & \mathbf{f} & \mathbf{e} \\ \mathbf{d} & \mathbf{c} & \mathbf{b} & \mathbf{a} \end{bmatrix} \tag{4.11}$$

Hallamos la matriz transpuesta de data3

data4 = data3'

$$data4 = \begin{bmatrix} o & l & h & d \\ ñ & k & g & c \\ n & j & f & b \\ m & i & e & a \end{bmatrix} \qquad 4.12$$

Transformamos data4 en un vector

$$data5 = [o \quad l \quad h \quad d \quad ñ \quad k \quad g \quad c \quad n \quad j \quad f \quad b \quad m \quad i \quad e \quad a] \qquad 4.13$$

En nuestro caso el vector columna tendrá 16832 elementos y el resultado aparentará una mezcla aleatoria.

4.5.3 Cuantización

Para procesar la señal debemos cuantizar las muestras, es decir, convertir las magnitudes que tiene cada una de las muestras en un número binario de tal manera que la computadora pueda procesarlo. El rango de valores que tienen las muestras adquiridas son entre –1 y +1. Para digitalizar primero cambiaremos el rango el mismo que depende de la cantidad de bits por muestra que deseamos obtener. En nuestro caso vamos a definir 256 escalones como partición del rango entre –1 +1 voltio. Es decir buscaremos que cada una de las muestras sea representada por 8 bits, lo que quiere decir que necesitaremos 2^8 niveles por las dos unidades de voltaje que tiene la señal adquirida.

Definimos nuestra partición de la siguiente manera:

```
%Proceso de Cuantización
partition=[-1:1/127:1];
codebook=[0:1:255];
[index2,s2]=quantiz(s,partition,codebook);
```

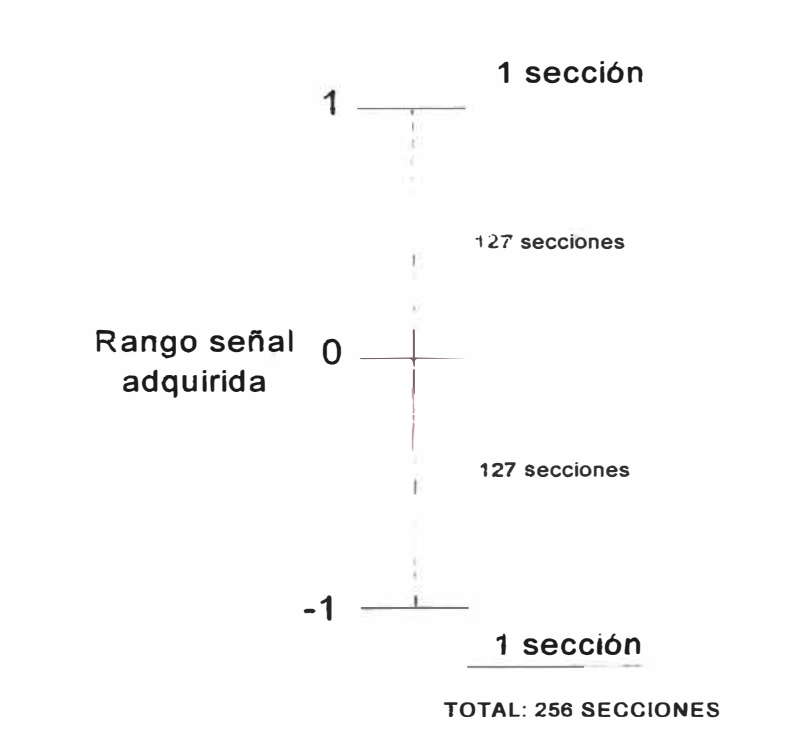


Figura 4.5 División en secciones de la señal adquirida

A cada una de estas secciones le corresponderá un valor desde 0 hasta 255, dependiendo de la amplitud de la muestra.

La precisión que obtendremos está dado por:

$$\text{Precisión} = \text{Rango de Voltaje} / 2^{\text{número de bits}} \qquad 4.14$$

En nuestro caso:

$$\text{Precisión} = 2 \text{ volts} / 2^8 = 0.0078125$$

$$\text{Precisión} = 0.0078125 \text{ volts.} \qquad 4.15$$

Nuestro proceso de cuantización puede detectar variaciones de voltaje del orden de los 7.8 mV

4.5.4 Digitalización

Ahora que tenemos cada una de las muestras con valores que pueden transformarse a números binarios utilizaremos comandos que nos permitirán obtener

valores convertidos en números binarios que van desde 00000000 hasta 11111111, los mismos que equivalen al rango que hemos definido anteriormente que va desde 0 hasta 255.

4.5.5 Llave de encriptación

La llave de encriptación se distribuye previamente entre las 2 personas que se comunicarán por un canal seguro. En nuestro caso generaremos una clave de 16 números que varían entre 0 y 255, para esto usaremos el comando "rand" del Matlab

```
k1=255*rand(1,16);%Los números aleatorios generados van entre 0 y 1,
%al multiplicarlos por 255
%con esta operación estamos generando valores que varian entre 0 y
%255
key=round(k1);%Redondeamos los valores antes obtenidos ya que sólo
%trabajamos con números enteros
```

Las operaciones que se realizan en el proceso de encriptación están relacionados a operaciones a nivel de bit, para esto se hará uso de la teoría de los campos finitos. Revisaremos previamente la teoría de los Campos Finitos y algunos ejemplos que nos ayudarán a comprender estas operaciones.

- Un Campo es un conjunto de elementos con 2 operaciones aritméticas definidas, por lo general son la suma y multiplicación. Todos los elementos del campo tienen un aditivo inverso y todos los elementos diferentes de cero tienen un multiplicativo inverso.
- Un campo es llamado finito si tiene un número finito de elementos. El mas común de los campos finitos usados en encriptación son los campos F_p (p es un número primo) y el campo F_2^m

- Un campo finito o Campo Galois denotado por $GF(q=p^n)$, es un campo con característica p y q elementos. En el cual p es primo y n entero positivo, y contiene un subcampo con p elementos. Este subcampo es llamado campo original.
- Los casos mas usados en encriptación se dan cuando: q=p, y $q=2^m$. El primer caso GF(p), se denomina campo primo y el segundo caso $GF(2^m)$ es conocido como campo finito de característica dos o simplemente campo binario, el cual usaremos en el presente trabajo.
- Un campo finito es un campo con un número finito de elementos. El número de elementos en el campo finito es llamado orden del campo. Los campos con el mismo orden se denominan isomorphic, quiere decir que ellos muestran exactamente la misma estructura algebraica diferenciándose solo en la representación de sus elementos.
- En el caso de la suma en un campo finito el resultado también pertenecera a este mismo campo finito. La suma de dos elementos esta dado por la suma Modulo 2, ejemplo: 1 + 1 = 0.

  Suponiendo que una muestra de la señal tenga el siguiente valor:

$$m = 01101101 \text{ (muestra)} \qquad 4.16$$

  Que la clave que va a afectar esta muestra sea:

$$k = 01011011 \qquad 4.17$$

  El resultado de la suma será: e = m + k

$$e = 01101101 + 01011011$$

$$e = 00110110 \text{ (muestra encriptada)} \qquad 4.18$$

En el lado de recepción se afectará la muestra encriptada transmitida por la misma clave resultando lo siguiente (comparar el resultado con 4.16):

$$d = e + k \qquad 4.19$$

$$d = 00110110 + 01011011$$

$$d = 01101101 \text{ (muestra desencriptada)} = m \qquad 4.20$$

Tomar en cuenta que estamos trabajando con la suma modulo 2, es decir:

1 + 1 = 0; 1 + 0 = 1; 0 + 1 = 1; 0 + 0 = 0

En nuestro caso, después de haber mezclado las muestras se define una llave, en forma de vector, de una longitud de 16 columnas con valores que van desde 0 hasta 255, esto se debe a que la clave debe pertenecer también al campo finito definido.

Se realiza una suma módulo 2 de esta llave a la señal que se ha digitalizado. Tomar en cuenta que el vector de datos tiene una longitud de 16384 elementos. El vector de la clave se sumará a los primeros 16 elementos del vector de datos, luego a los 16 siguientes elementos del vector de datos y así sucesivamente hasta terminar de afectar el vector de datos con la clave.

El objetivo de usar la suma módulo 2 es obtener los valores originales en el lado de recepción sumándole la misma clave.

En esta parte escuchamos la señal que ha sido encriptada obteniendo solamente un ruido.

4.6 Desencriptación.

De acuerdo al proceso de afectación de la señal por la clave y la mezcla de la misma el proceso de desencriptación sigue el siguiente proceso

DIAGRAMA DE FLUJO DEL PROCESO DE DESENCRIPTADO

Figura 4.5 Diagrama de flujo del proceso de desencritado

4.6.1 Llave de desencriptación

Dado que es un sistema simétrico la llave que se use será la misma que en el lado de transmisión. Aplicando esta llave y tomando en cuenta que las operaciones que se realizan son en base a los campos finitos se recuperan las muestras de datos originales.

4.6.2 Digitalización

Se convierte el vector obtenido anteriormente a nuestro sistema decimal, nuevamente el rango de valores obtenidos variará entre 0 y 255

4.6.3 Inversión del proceso de Mezclado

Se realiza el proceso inverso al mezclado con el siguiente proceso:

- Se transforma el vector columna en una matriz cuadrada de 128 x 128
- Se halla la matriz transpuesta
- La matriz transpuesta se convierte en un vector columna
- Se invierten las posiciones del vector columna

4.6.4 Normalización

Los valores que tenemos en el vector resultante del proceso anterior varían entre 0 y 255, para poder reproducir este vector se debe normalizar, es decir llevar nuevamente a través del proceso de cuantización a valores que oscilen entre –1 y +1

4.6.5 Reproducción

Se reproduce el vector resultante a través de la tarjeta de sonido.

4.7 Gráficos del proceso de encriptación de voz

A continuación mostramos los diferentes gráficos obtenidos del proceso seguido a la palabra "enterprise" con el programa de encriptación de voz en Matlab (Anexo B). Adicionalmente se incluye CD con el algoritmo desarrollado y pruebas con diferentes palabras o frases.

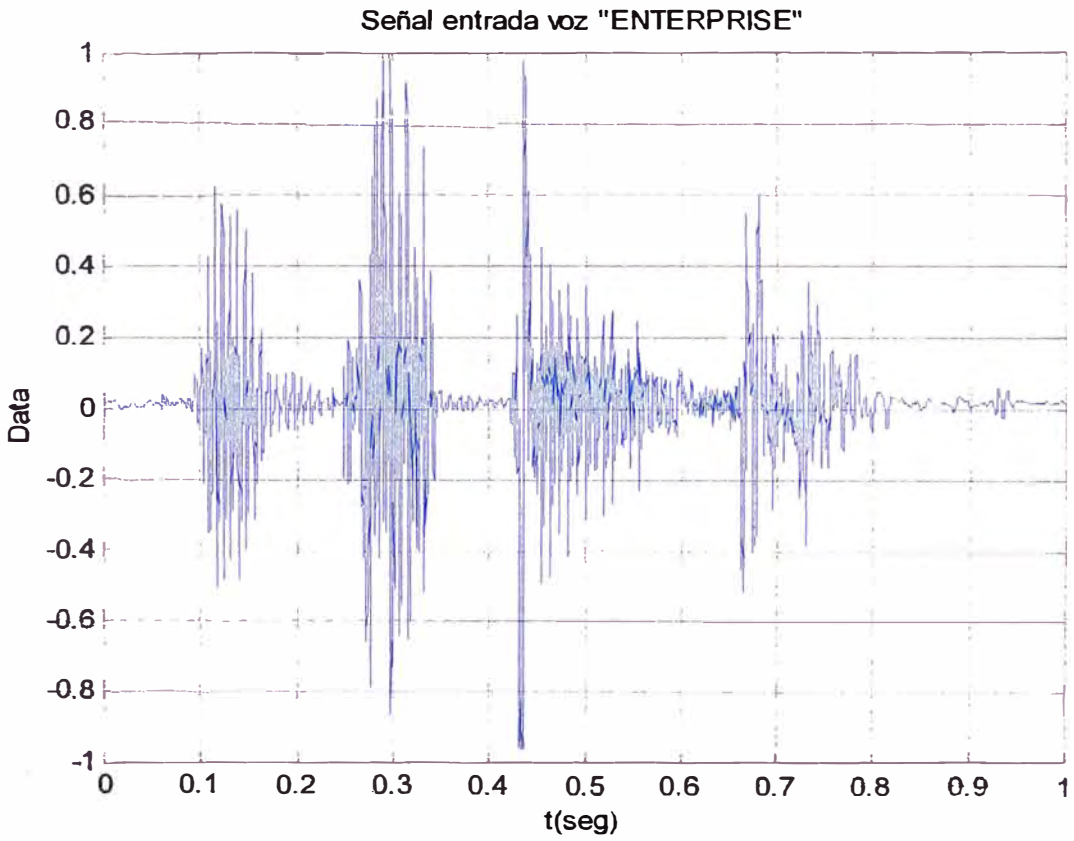


Figura 4.6 Señal entrada de voz "Enterprise"

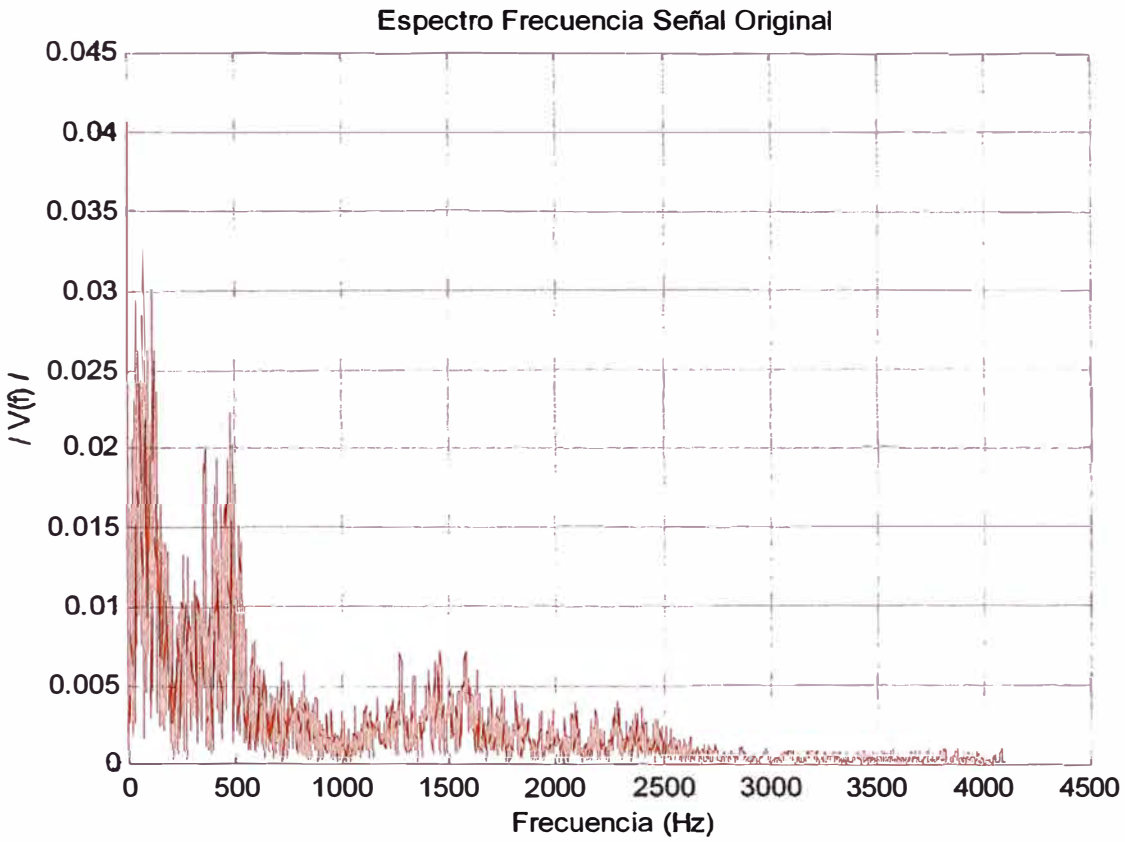


Figura 4.7 Espectro en Frecuencia Señal Original

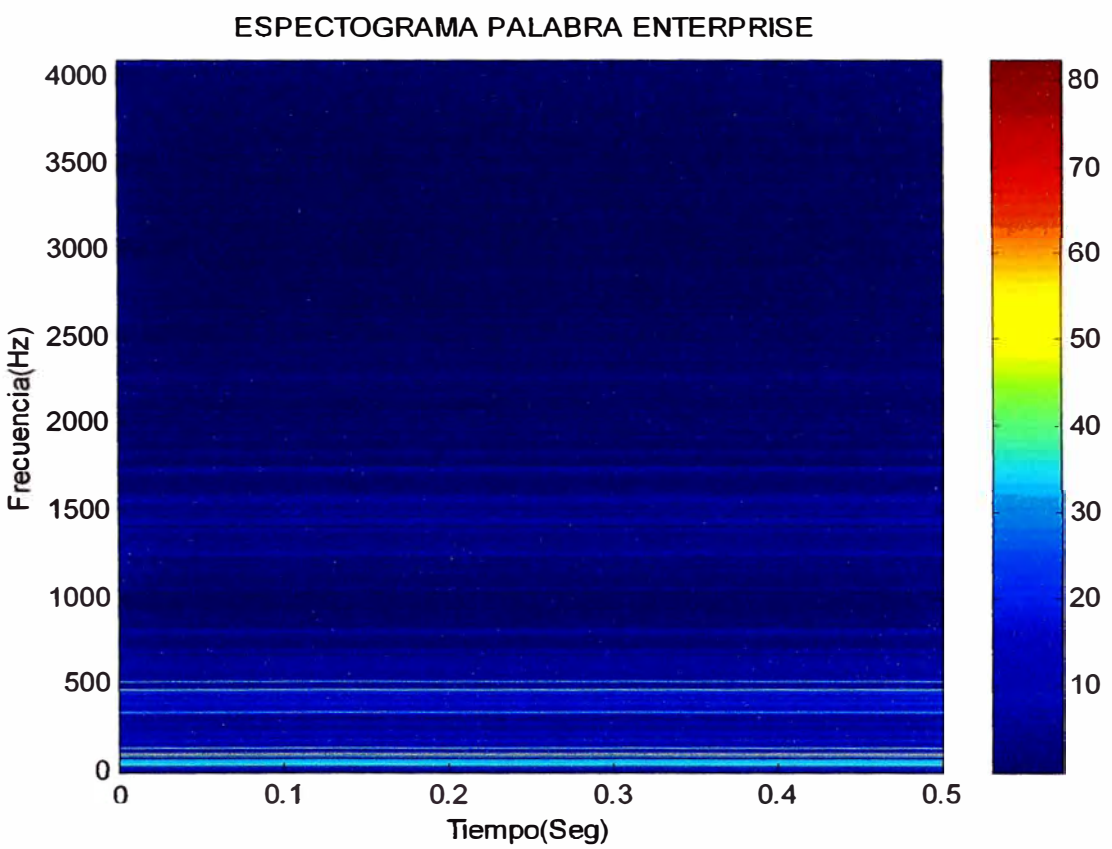


Figura 4.8 Espectrograma de señal original

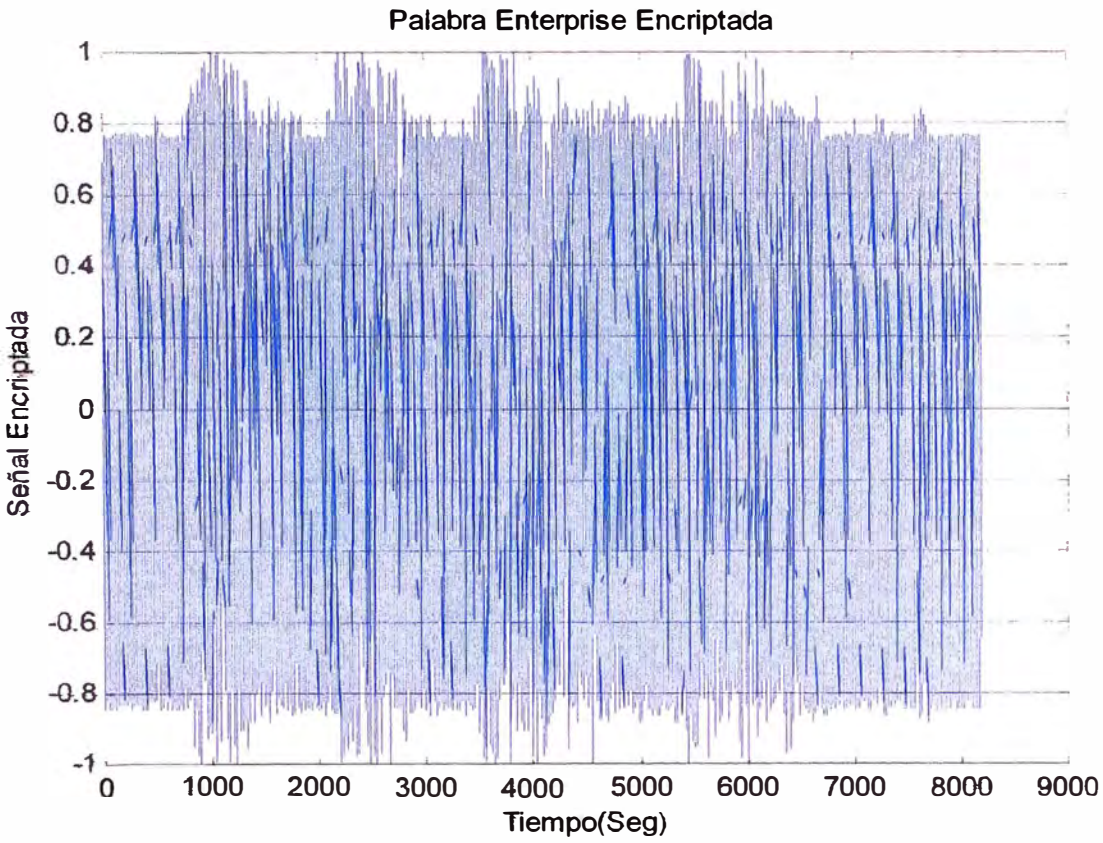


Figura 4.9 Señal de voz encriptada

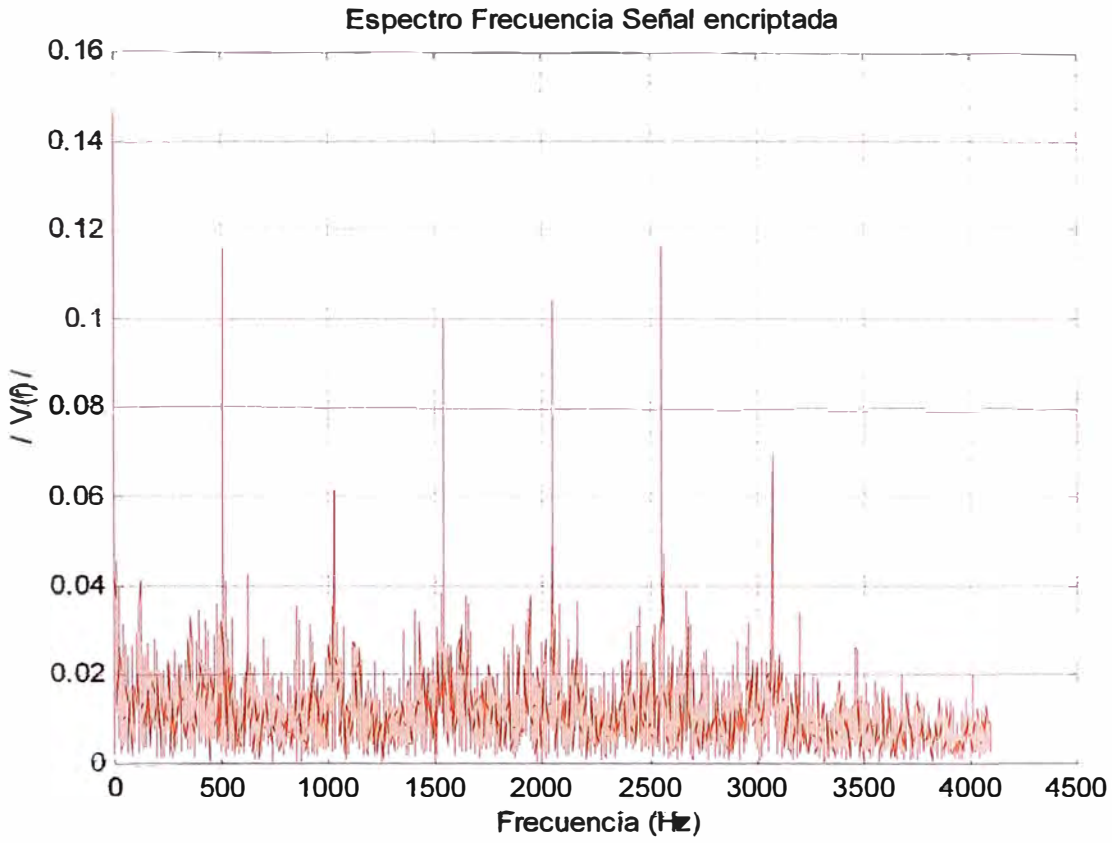


Figura 4.10 Espectro en frecuencia señal encriptada

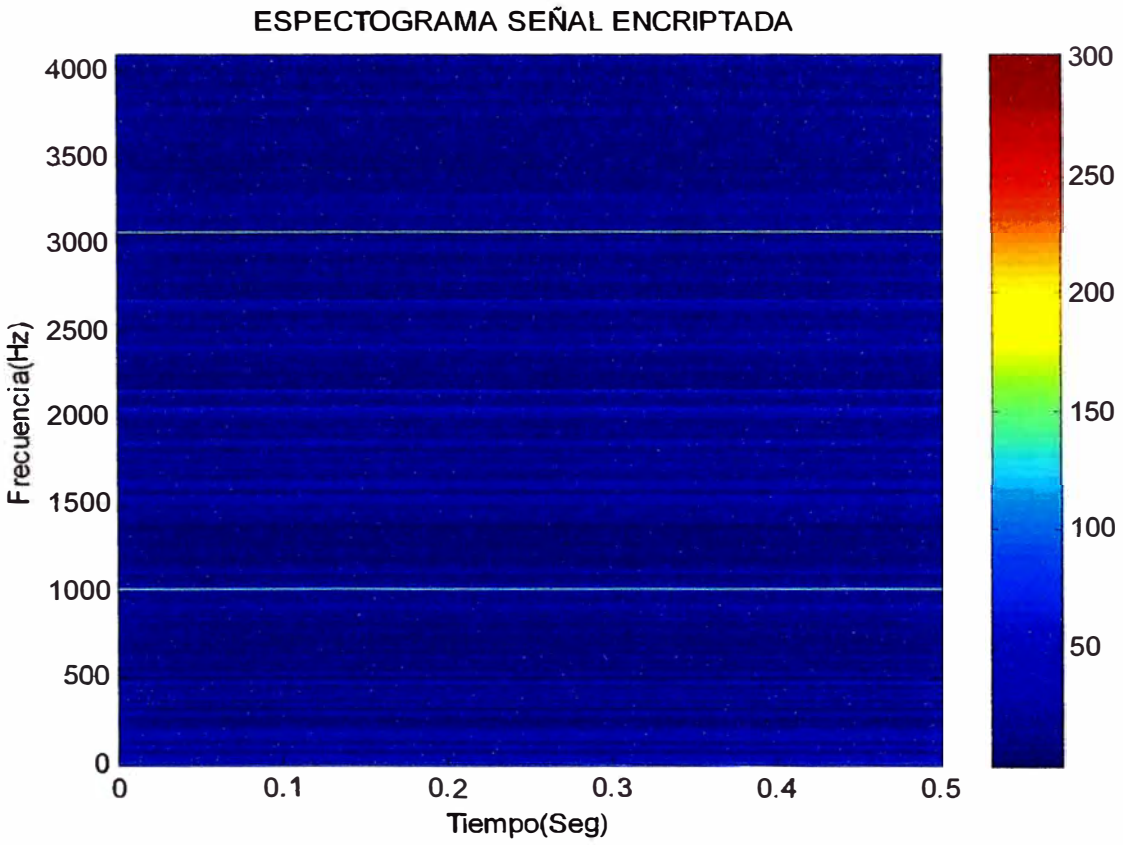


Figura 4.11 Espectrograma señal encriptada

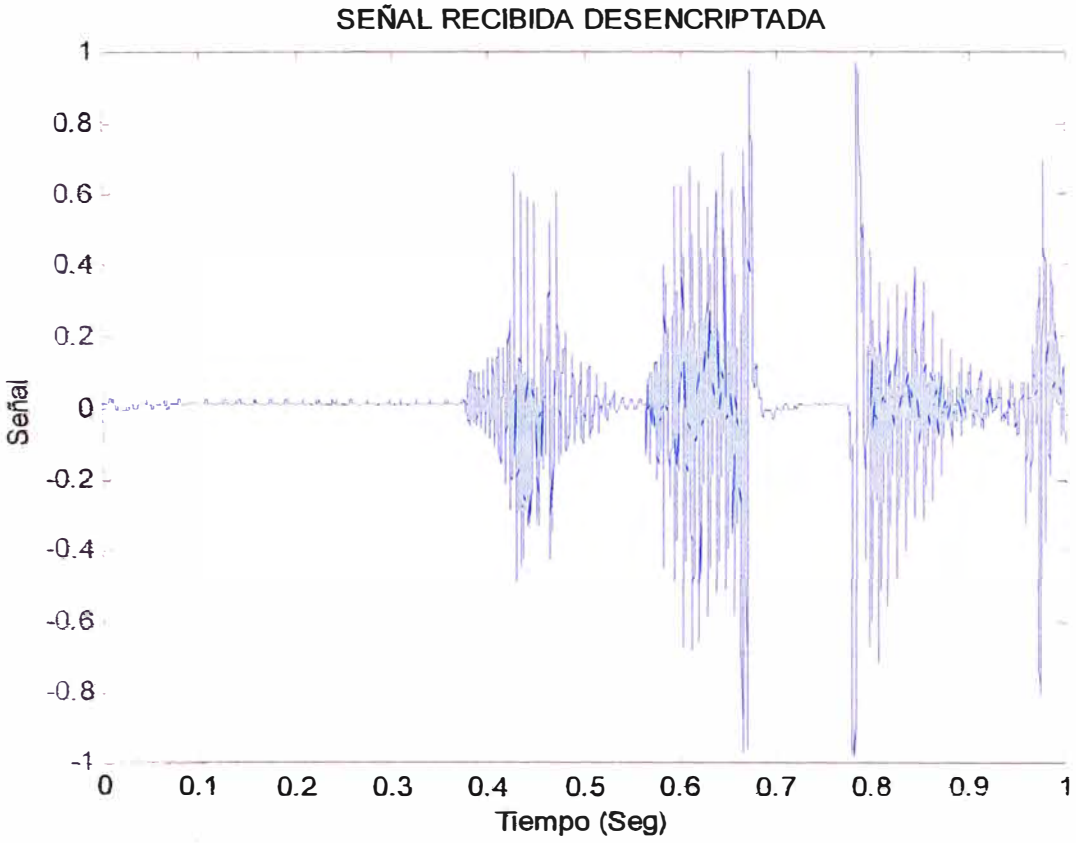


Figura 4.12 Señal desencriptada

CONCLUSIONES

1. La encriptación de voz, es una herramienta que será muy difundida en los próximos años. Por ahora esta tecnología, está siendo aprovechada por fuerzas militares y entidades gubernamentales. Sin embargo obligará a la mayoría a proteger su privacidad recurriendo a tecnología que pronto estará al alcance de todos. Aplicaciones en el Anexo C.
2. La encriptación de voz es una de las formas en que se aplica los estandares de encriptación de datos, como el AES. Se puede desarrollar en base a Procesadores digitales de señales (DSP), módulos que permitan brindar a los usuarios la privacidad en una conversación. Estos podrían implementarse para que sean conectados en los equipos telefónicos.
3. Si la velocidad es un requisito indispensable en la comunicación, se deberá desechar la idea de implementarlo en software. Para este caso se deberá considerar la implementación utilizando la tecnología FPGA ó ASIC, dependiendo también de la cantidad de información que se desea transmitir.
4. La teoría del procesamiento digital de señales nos brinda múltiples posibilidades de desarrollar equipos electrónicos complejos con sencillez.

5. Matlab es una herramienta práctica que permite simular casi toda clase de modelos eléctricos y electrónicos, entre ellos el procesamiento digital de señales.
6. La legislación nacional menciona la encriptación relacionada a las firmas y certificados digitales, más no en relación a comunicaciones encriptadas. Tener un marco legal nos permitirá desarrollar ordenadamente formas de transmisión de voz en forma segura.

ANEXO A

ANACRÓNICOS

DES	Data Encryption Standard
RSA	Rivest, Shamir and Adleman
ISO	International Standards Organisation
IEC	International Electrotechnical Commission
ZIP	Conmutación de Protección Automática
VHDL	Very High-level Design Language
ASIC	Application Specific Integrated Circuits
FPGA	Field Programmable Gate Arrays
CLB	Configurable Logic Blocks
RAM	Random-Access Memory
ECB	Electronic CodeBook
CBC	Cipher Block Chaining
CFB	Cipher FeedBack
OFB	Output FeedBack
IDEA	International Data Encryption Algorithm
DSS	Digital Signature Standard
NIST	National Institute of Standards & Technology
SSL	Secure Sockets Layer (Netscape; web security protocol)
AES	Advanced Encryption Standard
AEA	Advanced Encryption Algorithm
CA	Certification Authority
LPC	Linear Predictive Coding
MFCC	Mel-Frecuency Cepstrum Coefficients
RNA	Redes Neuronales Artificiales
HMM	Modelos Ocultos de Markov
A/D	Convertidor Análogo Digital
ADCs	Convertidor Análogo Digital
DACs	Convertidor Digital Análogo
I/O	Input /Output
C/T	Counter/Timer
TTL	Lógica Transistor Transistor
ECL	Emitter-Coupled Logic
ASCII	American Standard Code for Information Interchange
DAQ	Data Acquisition
DMA	Direct Memory Access
S/H	Sample & Hold
SS/H	Simultaneous Sample and Hold
FIFO	First In. First Out
CPU	Central Processing Unit

ANEXO B

Desarrollo del algoritmo de encriptación de voz en Matlab

```
% TRANSMISION DE VOZ POR CANAL SEGURO

clear all;      % Limpiar la Memoria de Variables y Funciones
clc;            % Limpiar la Ventana de Comandos
close all;      % Cerrar todas las Ventanas de Gráficas abiertas

ai=analoginput('winsound');%Se define el "objeto" ai(podemos manipular las
%caracteristicas de esta variable)
addchannel(ai,1);%El canal 1 es asignado al objeta ai.
ai.SampleRate = 8192; % Se define la frecuencia de muestreo del objeto "ai"
ai.SamplesPerTrigger=16384; % Se define 2 seg de muestreo (2 x 8192)
ai.TriggerType = 'Manual'; % Se define que la adquisición de datos se iniciará
%manualmente
start(ai);%Se inicializa el objeto ai
fprintf('   Presione una Tecla para grabar....\n');%Displaya en pantalla lo indicado
pause%se genera una pausa para preparar al locutor e inicie la grabación
%presionando una tecla
trigger(ai);%Se inicia el proceso de adquisición de datos y almacenado en el objeto
%ai
data = getdata(ai);%Se pasan los datos adquiridos del objeto ai a un vector
if max(data)<0.007874 then, end;%Si no hay datos adquiridos no se transmite nada
%y se detiene todo el proceso
t=ai.SamplesPerTrigger/ai.SampleRate;%Se define la variable t, el mismo que es
%igual al tiempo de adquisición
fs=ai.SampleRate;%Se define "fs" como frecuencia de muestreo
tt=0:1/fs:t-1/fs;   %Definimos un vector que simula el rango de Tiempo de captura de
%DATOS

%Visualización en Pantalla lo grabado en Memoria RAM
figure(1);%Definimos un primer gráfico
plot(tt,data);%Grafica los datos adquiridos
xlabel('t(seg)');%Rotula el eje x
ylabel('Data ');%Rotula el eje y
% abcd = sprintf('Información Grabada a fs=%6.2f Hz , durante t=%2.0f seg. , Total
% Muestras=%7.0f',fs,t,t*fs);
% set(get(gca,'title'),'String',abcd);
zoom on;%Permite agrandar una parte del gráfico señalado con el mouse
grid;%Grafica una cuadricula para poder usarla como referencia

%Escuchar lo Grabado en Memoria
fprintf('   Presione una Tecla para escuchar lo grabado en Memoria
RAM....\n');%Mensaje en pantalla
fprintf(2,' \n');   % 1 Línea de Espacio
pause;
sound(data);%Reproduce a traves de los parlantes los datos adquiridos
[f1,c1]=size(data);     % f1 : filas y c1 : columnas de Vector "data"

%Grabación en Disco Duro de la Información de sonido en formato WAV
%que está en Memoria RAM
```

```
wavwrite(data,fs,'enterprise');%Escribe en el archivo "enterprise"
% lo almacenado en el vector "data" con la frecuencia de muestreo fs
fprintf('    Número de Muestras Guardadas a Disco Duro: %6.0f ....\n',f1);
fprintf(2,' \n');   % 1 Línea de Espacio

%Recuperación de Información
[r,fm,nb]=wavread('enterprise');%extrae a un vector lo almacenado en el archivo
"enterprise.wav"
[f2,c2]=size(r);  % f2 : filas y c2 : columnas de Vector "r"
nt=f2;            % nt : Total de Muestras
fprintf('    Presione una Tecla para escuchar lo Recuperado de Disco Duro....\n');
fprintf(2,' \n');   % 1 Línea de Espacio
pause;
fprintf('    Número de Muestras recuperadas de Disco Duro : %6.0f ....\n',nt);
fprintf(2,' \n');   % 1 Línea de Espacio
sound(r);

%Conversión de Matriz a Vector para obtener el Espectro
fprintf('    Conversion de Matriz a Vector en Progreso .... !Por favor espere!\n');
fprintf(2,' \n');   % 1 Línea de Espacio
for k=1:1:nt,%Lazo para convertir a vector
  s(k)=r(k,1); % k,1 debido a que es un solo canal
end

N=ai.SamplesPerTrigger; %Cantidad de muestras. Potencia de 2(16384=2^14)

for k=1:1:N,
  vv(k)=s(k);  % "vv" contiene las "N" primeras Muestras
end


% Convolución de "vv" con la Ventana "Hamming(N)"
    ventana=hamming(N);%Se define la ventana de Hamming
    for k=1:1:N,
      v(k)=vv(k)*ventana(k); % "v(k)" es la Convolución resultante de ambas señales
    end;

%*************************************************
%  Aplicando la  FFT a las "N" Muestras extraídas
%   el cual tiene "N" Muestras en Total
%*************************************************
kv=4/N;                % constante al aplicar la ventana
fftv=abs(fft(v,N));      % "fft" es la Transformada de Fourier de "v"
                  % evaluado en "N" puntos
for k=1:1:N/2,
  fftvv(k)=fftv(k)*kv;  % "fftvv" contiene solo las "N/2" primeras Muestras
end

   f=0:fm/N:fm/2*(N-1)/N;      % Frecuencia de Análisis

figure(2);%Genera una ventana para gráficar lo especificado a continuación
```

```
plot(f,fftvv,'r');%Grafica el espectro de la señal
xlabel(' Frecuencia (Hz)');%Rotula el eje x
ylabel('  / V(f) / ');%Rotula el eje y
abcd = sprintf('Espectro Frecuencia Señal Original');%Genera una cadena y lo
asigna
set(get(gca,'title'),'String',abcd);%Titula el gráfico
zoom on;%Permite ampliar zonas del gráfico
grid;%Dibuja lineas de referencia en el gráfico
%
% Espectograma de la Señal Original
%
figure(3);%Genera una ventana para gráficar lo especificado a continuación
[b,Fre,Tiem]=specgram(data,N,fs,ventana);%Calcula es espectograma de la señal
que en este caso se encuentra
%en el vector "data", con N muestras a la frecuencia de muestreo definida y
%utilizando la ventana en este
%caso de hamming
imagesc(Tiem,Fre,abs(b)),axis xy;%Determina la escala que se usa en el gráfico de
%acuerdo a los datos
%de entrada "Tiem" y "Fre"
axis([0 0.5 0 max(Fre)]),axis xy;%Determina el mínimo y maximo de las escalas
colorbar('vert');%Muestra la barra de escala de colores en este caso en forma
%vertical
xlabel('Tiempo(Seg)'),ylabel('Frecuencia(Hz)');%Rotula los ejes del gráfico
abcd=sprintf('ESPECTOGRAMA PALABRA ENTERPRISE');%Genera una cadena y
%lo asigna
set(get(gca,'title'),'String',abcd);%Titula el gráfico
zoom on;%Permite ampliar zonas del gráfico


%*******************************************************
%          Inicio del proceso de Encriptación
%*******************************************************
%
%Proceso de Mezcla de posición de las muestras
%
s1a=flipud(s); %Se invierten las posiciones del vector columna, la primera posición
%pasa al final y viceversa
s1b=reshape(s1a,128,128); %El vector invertido es acomodado en una matriz
%cuadrada
s1c=s1b';%La matriz cuadrada es transpuesta
s1=reshape(s1c,1,16384);%Conversión de la matriz cuadrada en vector columna

%Proceso de Cuantización
partition=[-1:1/127:1];%Define la cantidad de pasos en los que la señal es dividida
%para el muestreo
codebook=[0:1:255];%Define los valores que se asignarán a cada uno de los pasos
%definidos
%la instrucción anterior
[index2,s2]=quantiz(s1,partition,codebook);%Realiza la operación de cuantización,
%es decir que a
```

```
%cada valor en el vector "s1" cuyo valor esta entre -1 +1 le asignará un valor del
vector definido
%por codebook.

%*******************************************************
%Ingreso de la llave
%*******************************************************
%Los valores que se ingresen en la llave deberán ser menores o iguales al máximo
%valor del vector cuantizado, es decir 255. Esto debido a que se trabajará con
suma módulo 2,
%de tal manera que no se altere el rango de trabajo
%
%*********************************************************
%Generación de la clave para encriptar
%*********************************************************
k1=255*rand(1,16):%Los números aleatorios generados van entre 0 y 1, al
%multiplicarlos por 255
%con esta operación estamos generando valores que varian entre 0 y 255
key=round(k1):%Redondeamos los valores antes obtenidos ya que sólo trabajamos
%con números enteros
%key=[150 90 30 200 59 63 8 59 68 49 250 55 99 44 23 175];Ya que las claves son
%previamente
%distribuidas a través de un canal seguro, podemos trabajar con un valor fijo
%definido ya que las
%transmisiones
%
%Alteración de la señal con la llave a nivel de bits
%Generamos un lazo en el cual se suma la llave a cada uno de los valores
definidos en el
%vector "s2" y en grupos de 16
f=1:%Asignamos un valor inicial a "f"
while f<=16384%Iniciamos el lazo que define que cuando el contador "f" llegue a
%16384 termine el ciclo.
  g=1:%Asignamos un valor inicial a "g"
  while g<=16%Iniciamos el segundo lazo que define que cuando el contador "g"
%llegue a 16 termine el ciclo.
    crip(f)=bitxor(s2(f),key(g));%Realiza la operación suma modulo 2.
    g=g+1;%incrementa el valor de g para que se trabaje con la siguiente posición
%del vector "key"
    f=f+1;%incrementa el valor de f para que se trabaje con la siguiente posición del
%vector "s2"
  end%Finaliza el lazo secundario referido a la llave
end%Finaliza el lazo principal referido al vector de datos "s2"
%
%*********************************************
%Proceso para reproducir la señal encriptada
%*********************************************
%El presente proceso es parte del proceso de desencriptación
%Realizamos el proceso de cuantización que traslada la escala de la señal
almacenada en "s2"
```

```
%la cual esta comprendida entre valores de 0 hasta 255 a la escala original, es
%decir a valores comprendidos entre -1 +1.
partition3=[1:1:254];%Se define los pasos para la señal "s2"
codebook3=[-1:1/127:1];%Define los valores que se asignarán a cada uno de los
%pasos definidos anteriormente
[index3,s3]=quantiz(crip,partition3,codebook3);%Realiza la operación de
cuantización, es decir que a
%cada valor en el vector "crip" cuyo valor esta entre 0 y 255 le asignará un valor del
%vector definido por codebook.

wavwrite(s3,fs,'encryp_voice');%graba el contenido en el archivo "encryp_voice"
sound(s3);%Reproduce el sonido almacenado en "s3"

%Grafica de la palabra Enterprise encriptada

figure(4);
plot(tt,s3);
xlabel('Tiempo(Seg)'); ylabel('Señal Encriptada');
abcd=sprintf('Palabra Enterprise Encriptada');
set(get(gca,'title'),'String',abcd);
zoom on;

%Espectro de Enterprise encriptada
figure(5);

% Convolución de "s3" con la Ventana "Hamming(N)"
   for k=1:1:N,
      s3v(k)=s3(k)*ventana(k); % "v(k)" es la Convolución resultante de ambas
%señales
   end;

%*************************************************
%  Aplicando la  FFT a las "N" Muestras extraídas
%   el cual tiene "N" Muestras en Total, apartir
%          de la posición "nx"
%*************************************************
kv=4/N;                % constante al aplicar la ventana
ffts3v=abs(fft(s3v,N));  % "fft" es la Transformada de Fourier de "v"
                  % evaluado en "N" puntos
for k=1:1:N/2,
  ffts3vv(k)=ffts3v(k)*kv;  % "fftvv" contiene solo las "N/2" primeras Muestras
end

   f=0:fm/N:fm/2*(N-1)/N;      % Frecuencia de Análisis

plot(f,ffts3vv,'r');
xlabel(' Frecuencia (Hz)');
ylabel('  / V(f) / ');
abcd = sprintf('Espectro Frecuencia Señal encriptada');
set(get(gca,'title'),'String',abcd);
zoom on;
```

```
% Espectograma de la Señal Encriptada
figure(6);
%
[b,Fre,Tiem]=specgram(s3,N,fs,ventana);
imagesc(Tiem,Fre,abs(b)),axis xy;
axis([0 0.5 0 max(Fre)]),axis xy;
colorbar('vert');
xlabel('Tiempo(Seg)'),ylabel('Frecuencia(Hz)');
abcd=sprintf('ESPECTOGRAMA SEÑAL ENCRIPTADA');
set(get(gca,'title'),'String',abcd);
zoom on;

%*********************************************
%Proceso de digitalización
%*********************************************
crip_digit=de2bi(crip);%Covierte los valores decimales a binarios para que sean
%transmitidos de esta manera

%*******************************************************
%     Proceso de recepción de la señal y desencriptación
%*******************************************************
%
%SE REALIZARAN LOS PASOS INVERSOS DEL PROCESO DE ENCRIPTACIÓN
PARA OBTENER LA SEÑAL ORIGINAL
%Primero conversión de digitos a decimal, luego aplicación de llave,
%reordenamiento del vector
%y finalmente normalización de la señal a valores entre -1 +1
%
crip_rx1=bi2de(crip_digit);%Convierte los valores binarios a decimales, los mismos
%que variarán entre 0 a 255
crip_rx=reshape(crip_rx1,1,16384);%Convierte a vector columna
%
%
%*************************************************************
%Aplicando la misma llave para desencriptar la señal
%*************************************************************
%
%Sumaremos la misma llave usada en el proceso de encriptación al vector
%encriptado, siguiendo la  misma rutina que en el de encriptación
f=1;
while f<=16384
  g=1;
  while g<=16
    de_crip(f)=bitxor(crip_rx(f),key(g));
    g=g+1;
    f=f+1;
  end
end
%
%
```

```
%************************************************************
%Proceso para reacomodo de las posiciones de la señal original
%************************************************************
%Realizaremos el proceso inverso del mezclado de la señal
dec1=reshape(de_crip,128,128);%Convertimos en una matriz cuadrada
dec2=dec1';%Hallamos la matriz transpuesta
dec3=reshape(dec2,1,16384);%Conversión a vector columna
de_crip1=flipud(dec3);%Invertimos las posiciones. La última posición regresa a la
primera y viceversa


%************************************************************
%Proceso para reproducción de señal desencriptada
%************************************************************
%
%Realizamos el proceso de normalización a través del proceso de cuantización.
%Los valores que tenemos  en el vector "de_crip" estan comprendidos entre 0 y
%255, para que la computadora pueda reproducirla  como sonido debemos
%normalizarla, es decir, trasladar el rango de 0 - 255 a el rango -1 +1.Realizaremos
%el mismo proceso realizado inicialmente
partition4=[1:1:254];%Definimos los pasos que tendrá el vector "de_crip"
codebook4=[-1:1/127:1];%Define los valores que se asignarán a cada uno de los
%pasos definidos anteriormente
[index4,s4]=quantiz(de_crip1,partition4,codebook4);%Realiza la operación de
%cuantización, es decir que a
%cada valor en el vector "de_crip" cuyo valor esta entre 0 y 255 le asignará un valor
%del vector definido por codebook entre -1 +1.

wavwrite(s4,fs,'de_encryp_voice');%Se graba el vector obtenido anteriormente a la
%frecuencia de muestreo
sound(s4);%Reproduce la señal almacenada en "s4"
figure(7);%Nueva ventana para un gráfico
plot(tt,s4);%Grafica la señal recepcionada
xlabel('Tiempo(Seg)'),ylabel('Señal');%Rotula los ejes del gráfico
abcd=sprintf('SEÑAL DESENCRIPTADA');%Genera un string para abcd
set(get(gca,'title'),'String',abcd);%Titula el gráfico
zoom on;%Permite ampliar el gráfico
grid;%Define lineas de referencia
```

ANEXO C

Aplicaciones de encriptación de voz

DSP 9000

Sistema de encriptado de radio militar

El DSP 9000 es una familia de sistemas de cifrado militar que provee seguridad estratégica de largo plazo para comunicaciones transmitidas sobre canales de banda angosta. El DSP 9000 está disponible en configuraciones para
estación base, mochila, microteléfono y de tablero de circuitos de implante. La interfase programable y un diseño que cumple con las especificaciones militares hacen que el DSP 9000 sea capaz de dar seguridad a virtualmente cualquier aplicación HF, VHF o UHF.

Aplicaciones con seguridad

HF-SSB, VHF y radio UHF Radioteletipo
Teléfonos de discado estándares
Datos de baja velocidad
Fax (Grupo I y Grupo II)
Cuadros telefónicos tácticos y teléfonos de campaña

Tecnología avanzada

El DSP 9000 utiliza una avanzada tecnología en todo su diseño. Un poderoso procesador de señales digitales provee gran potencia de computación que se usa para obtener una calidad de recuperación de voz excepcional y la seguridad del cifrado. Todos los parámetros de entrada/salida de audio son controlados por programa. Esto permite que un solo DSP 9000 sea prontamente instalado en una variedad de radios sin tener que modificar el equipo. Exclusivas características tales como versiones semidúplex y dúplex completa, sincronización dual, selección automática de cifrado de voz/datos y almacenamiento de un gran número de claves, también separan claramente al DSP 9000 de sus competidores. Además, el DSP 9000 es compatible

con el teléfono de seguridad CSD 3324E de TCC para permitir comunicaciones "desde la oficina hasta el campo".

Características

Seguridad criptográfica estratégica
Calidad de recuperación de voz excepcional
Modelos semidúplex y dúplex completo
Interfaces y configuración programable guiada por menúes
Diseñado y probado para cumplir con los estándares militares
Capacidad completa de control remoto para vehículos, embarcaciones y aeronaves
Modo de selección de llamada para conversaciones privadas
Sincronización manual y de PTT (oprimir para hablar)
Característica de efecto de inercia en la sincronización ("coast")
Manejo automático de claves
Configuraciones fijas, móviles y de mochila

CSD 3600

Teléfono de Seguridad

El CSD 3600 es un dispositivo de seguridad compacto que se instala entre la base del teléfono y el auricular. Al oprimir un botón, la llamada telefónica es protegida por el algoritmo criptográfico de alto grado de Bell Laboratories. La llamada puede ser desencriptada por un dispositivo receptor de TCC, incluyendo otra unidad CSD 3600, el teléfono celular de seguridad CSD 9300 o el teléfono ejecutivo de seguridad CSD 4100.

Características claves

Compatibilidad telefónica universal. Al encajar ciertos módulos en la parte posterior del CSD 3600, se logra la compatibilidad telefónica universal. Los usuarios pueden tener el teléfono de su preferencia y operar con líneas de teléfono y conmutadores análogos y digitales.

Calidad de voz. La claridad de voz durante la llamada encriptada permite el reconocimiento del que habla. La calidad de voz superior es el resultado de un algoritmo de codificación de voz sofisticado desarrollado especialmente para el CSD 3600 por Bell Laboratories.

Portabilidad. El CSD 3600 es tan pequeño como una calculadora de mano y es extremadamente liviano. Puede ser fácilmente cargado en un portafolios para ser usado en la casa o mientras viaja - incluso internacionalmente. El CSD 3600 se conecta al aparato de mano del teléfono y no a la red de teléfono. Como resultado, puede ser convenientemente usado alrededor del mundo sin aprobaciones de redes telefónicas globales.

Manejo de claves

El CSD 3600 genera y distribuye claves usando claves negociadas. Con la tecnología de negociar una clave, el cargado de claves manual y el mantenimiento no son necesarios, lo que permite la protección de comunicaciones inmediata. Cada vez que se oprime el botón de seguridad, el CSD 3600 genera una nueva clave. Esta clave es única para esa sesión y para las dos unidades CSD 3600 en comunicación. Para asegurar aún más la llamada, se muestran variables aleatorias derivadas de la clave única. Al comparar las variables, el transmisor y el receptor pueden detectar si la negociación de claves está siendo violada o engañada.

CSD 9300

Sistema celular de seguridad CSD 9300

El CSD 9300 es un sistema de teléfono celular de seguridad de alto nivel. Combina la tecnología más avanzada de encripción del CSD 3600 con un teléfono celular con la ventajas de que es transportable en un estuche sencillo. Tomando ventaja de la calidad de voz y la encripción pública de claves de Bell Laboratories, la llamada puede ser decifrada solamente por el sistema compatible receptor de TCC.

Facilidad de uso

Al igual que el CSD 3600, el CSD 9300 es fácil de usar. El usuario coloca la antena en una posición vertical, marca el número deseado y luego oprime el botón de seguridad para activar la seguridad. Una sola batería recargable permite hasta 45 minutos de tiempo de comunicación segura y clara y puede ser recargada automáticamente al enchufarlo en el encendedor de cigarrillos del auto o por medio de una fuente de energía CA.

Teléfono celular con todas las ventajasEl CSD 9300 es uno de los teléfonos celulares de más ventajas en el mercado. Entre sus características están: operación sin necesidad de usar las manos (sin seguridad); discado de un solo toque y de "scratchpad", discado de un repertorio de 99 números y/o NAM's múltiples; rotación de memoria; rediscado automático, capacidad de restricción de llamada; una pantalla grande y fácil de leer; y un jack auxiliar para la transmisión de datos.

BIBLIOGRAFÍA

[1] Amparo Fúster Sabater, Dolores de la Guía Martínez, Luis Hernández Encinas, Fausto Montoya Vitini, Jaime Muñoz Masqué "Técnicas Criptográficas de protección de datos" 2da Edición, Alfa Omega Grupo Editor, 2001

[2] Ashok Ambardar, "Procesamiento de señales analógicas y digitales" Thomson Learning, 2002.

[3] Alan V. Oppenheim, Ronald W. Schafer, "Tratamiento de señales en tiempo discreto" 2da Edición, Prentice Hall, 2000.

[4] Duane Hanselman , Bruce Littlefield, "Matlab, edición de estudiante", Prentice Hall, 1995

[5] Javier García de Jalón, José Ignacio Rodríguez, Alfonso Brazález, "Aprenda Matlab 5.3 como si estuviera en primero", Universidad de Navarra, 1999.

[6] www.matworks.com "Communications Toolbox User's Guide" 2002

[7] www.matworks.com "Data Acquisition Toolbox" 2002

[8] A. Menezes, P. Van Oorschot, S. Vanstone, "Handbook of Applied Cryptography" 1996

[9] Fonética Articulatoria: herramienta para analizar las lenguas del mundo, Summer Institute of Linguistics 2002

[10] www.nist.gov/aes Advanced Encryption standard Development Effort

[11] http://www.itl.nist.gov/fipspubs/ Federal Information Processing Standards Publications

[12] http://www.tccsecure.com/ Technical Communications Corporation