CANDADO ELÉCTRONICO HARDWARE EN EL DESCIFRADO DE DATOS PARA TRANSFERENCIA DE INFORMACIÓN

Jesus Pimentel Cruz 1 Patricia Pérez Romero2 Israel Rivera Zárate3

Centro de Innovación y Desarrollo Tecnológico en Cómputo1,2, 3

{jpimente1, promerop2, irivera 3}@ipn.mx

1. Resumen

Este documento muestra el diseño de un circuito electrónico basado en microcontrolador así como el código fuente desarrollados respectivamente para conformar un sistema de seguridad en el manejo de información de una máquina de "bordados electrónicos". El circuito electrónico se instala en el puerto paralelo de la PC

2. Planteamiento

Las máquinas bordadoras de alto rendimiento han sido diseñadas para el bordado convencional en continuo o en bastidor individual. Las diversas variantes ofrecidas permiten realizar un bordado creativo y económico. **Ver figura 1**.

Figura 1. Maquina de bordado

Los posibles anchos del campo de bordado varían entre 162.42 y 1200 mm. Para motivos en continuo está garantizada una conexión en rapport. Sistemas inteligentes de tensado y de bastidores optimizan los tiempos de preparación en la producción, también bordando material semi-confeccionado.

Para llevar acabo el bordado se requiere el empleo de un programa que genere a partir de diseños gráficos archivos que contengan la información necesaria para el manejo de las máquinas, tal información incluye colores, hilos, cabezales, puntadas y otras acciones de control del movimiento sobre las telas. **Ver figuras 2 y 3.**

Figura 2. Bordado del diseño gráfico

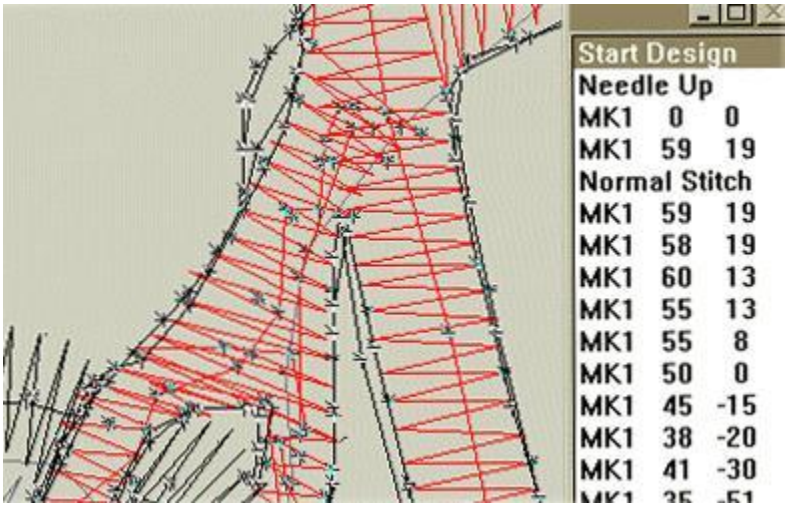

Figura 3. Aproximación visual de las puntadas

La mayoría de empresas emplean el programa PesView que es un pequeño y rápido Visor con cajas de diálogo para Win95/98. Capaz de desplegar los siguientes formatos de archivos de bordado:

- Brother *.PES
- Brother *.PEC
- Pfaff *.PCS
- Tajima *.DST
- Janome *SEW

Capaz de convertir estos archivos a los siguientes formatos:

- Pfaff *.PCS
- Tajima *.DST

Este programa Puede hacer zoom/rotar los diseños para probar un diseño (Esto funciona en Prevista. No se podrán rotar/agrandar y salvar los diseños). Para Pffaf: PesView solo checa un diseño si tiene más de 16 colores y no hará la conversión si eso ocurre. No hace el chequeo del número de puntadas o tamaño del bastidor. **Ver figura 4.**

Figura 4. Pantalla de captura del programa Pesview

Las empresas de este ramo manifiestan continuamente la necesidad de resguardar sus diseños para protegerlos de su uso indebido con fines de lucro o competencia por

parte de personal no autorizado, así mismo indica la necesidad de validar al personal operario de las PC de diseño gráfico.

Con base en lo anterior, se determina el diseño de:

I. Sistema de cifrado de datos para seguridad en el almacenamiento y transferencia de información
II. Desarrollo de un candado electrónico para validar y autorizar el uso de información

3. Sistema de cifrado de datos para seguridad en el almacenamiento y transferencia de información

3.1 Introducción

Los mensajes por encriptar, conocidos como texto llano, son transformados por una función parametrizada por una clave. El resultado del proceso de encriptación, conocido como texto cifrado, se transmite a continuación. Suponemos que el enemigo, o intruso, escucha y copia con exactitud todo el texto cifrado. Sin embargo, a diferencia del destinatario original, el intruso no conoce la clave de desencriptación y no puede desencriptar con facilidad el texto cifrado. **Ver figura 5.**

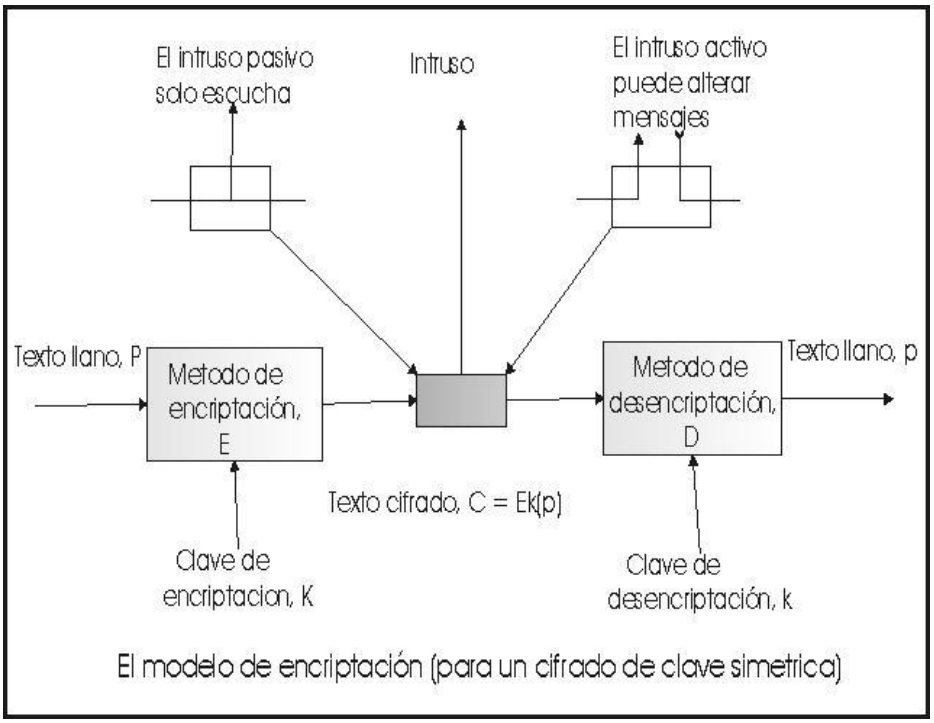

Figura 5. Modelo de encriptación

El arte de descifrar los mensajes (criptoanálisis) y el arte de crear los cifrados (criptografía) se conocen en conjunto como criptología.

En la actualidad la criptografía es fundamental para múltiples campos de la informática. Cuando se realiza una compra por Internet, los datos de la tarjeta de crédito se transmiten cifrados para que no caigan en manos no deseadas.

3.2 Cifrado polialfabético

Son sistemas más sofisticados que intentan hacer fracasar el análisis estadístico haciendo que la letra-origen se convierta en una letra cifrada en función de la posición que ocupe dentro del mensaje. Obviamente, tendremos que usar varios alfabetos. Uno de los más interesantes es el cifrado Vigenére, inventado por Blaise Vingeneré.

Por ejemplo: Vamos a usar como clave la palabra MANO. Ahora obtendremos cinco alfabetos corridos (el original más cuatro letras de la clave). Cada uno empezará por esa letra. **Ver tabla 1.**

M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ

Tabla 1. Ejemplo de cifrado polialfabético con la clave MANO

Nos hemos ahorrado un alfabeto, porque el segundo coincide con el original al comenzar por la letra A. Imaginemos el mensaje original, y apliquemos la clave. **Ver tabla 2.**

T	U		E	S	P	I	A		H	A		D	E	S	E	R	T	A	D	O
M	A		N	O	M	A	N		O	M		A	N	O	M	A	N	O	M	A

Tabla 2. Texto plano con la clave MANO

Y buscamos a que letra cifrada equivale cada letra original, pero en el alfabeto que corresponde a la letra de la clave. O sea, que en la palabra “TU”, la letra “T” equivale a la “M”, luego habrá que buscar la cifrada en el alfabeto que empieza por M. La letra cifrada será pues la “F”.

Así el mensaje cifrado será

FU QHBIÑ SO DRESRMRO

El cifrado Vingenére es la base de los cifrados polialfábeticos. A partir del el se desarrollaron variaciones. Algunas de ellas muy inteligentes.

Una variación fue el cifrado de Gronsfeld. Este sistema utilizaba diez alfabetos asignándoles números del 0 al 9. El correspondiente al 0 comenzará Por A y el correspondiente al 9 por J, obviamente. El resto es igual que el cifrado Vingenére.

3.3. Código de Cifrado y Descifrado

De acuerdo con la base teórica expuesta en el apartado 3.2 se desarrollaron un par de programas para el cifrado y descifrado polialfabético respectivamente. Este código realiza las siguientes acciones:

1. Recibe como entrada un archivo de bordado de formatos:

- Brother *.PES
- Brother *.PEC
- Pfaff *.PCS
- Tajima *.DST
- Janome *SEW

2. Codifica la información del archivo de entrada mediante una palabra clave en un nuevo formato con extensión CFR y elimina el archivo de bordado.

3. Para obtener el archivo original se requiere el archivo con extensión CFR y la clave.

Cabe mencionar que para hacer uso de los códigos se requiere el uso de una clave de acceso así como de tener instalado el circuito electrónico (candado) en el puerto paralelo de la PC. Ver códigos fuente en el apéndice A.

4. Desarrollo del candado electrónico para validar y autorizar el uso de información

Debido a la importancia en el acceso a los programas que codifican los archivos gráficos se decidió restringir su uso a personal autorizado; el cual debe poseer por una parte una clave personal asignada y por otra contar con un circuito electrónico que valide el acceso (candado).

Con esto en mente se desarrolló un circuito basado en el microcontrolador PIC16F84 el cual se instala en el puerto paralelo de la PC desde donde por medio de un programa recibe la información de la clave del usuario y la valida contra un conjunto definido en su programación interna.

4.1 Diseño del Circuito

El circuito impreso es una componente del sistema, y afecta tanto el comportamiento de éste, como su reproductibilidad industrial y a la confiabilidad del producto final. En general un análisis de las características de la señal requiere del estudio de la teoría de líneas de transmisión para determinar principalmente las reflexiones generadas por efecto de desacoplamientos entre las impedancias de la fuente, de la línea y de la carga.

Por lo tanto los puntos a considerar son: la distribución de las señales de control así como de la alimentación eléctrica seguido de un análisis de las características de la señal por medio de simulación por computadora.

4.1.1 Empleo del paquete ORCAD ESP 386+

La herramienta de software utilizada fue el programa ORCAD ESP 386+ el cual se instala en máquinas compatibles con el procesador Intel 80386 y que opera en modo DOS. Este programa requiere recursos mínimos del sistema tales como disco duro de 100 MB y 32 MB de RAM.

Esta herramienta permite la captura del Diagrama Esquemático por una parte y el Diseño del PCB por otra. **Ver figura 6.**

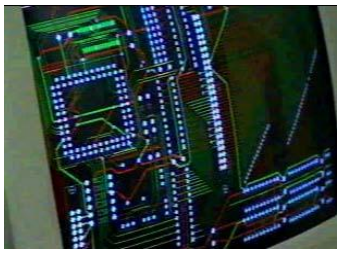

Figura 6. Pantalla del paquete ORCAD 386

Debido a las especificaciones eléctricas solicitadas por la Empresa se decidió realizar una tarjeta de doble cara con la distribución de pistas que permita la conducción de las líneas de control de forma independiente en cada cara como se aprecia en los Diagramas 3 y 4 respectivamente.

En los diagramas se aprecia que en la cara de componentes se distribuyó la señal de control correspondiente al Rojo en tanto que en la cara de soldadura se distribuyó la señal correspondiente al Verde. **Ver figura 7.**

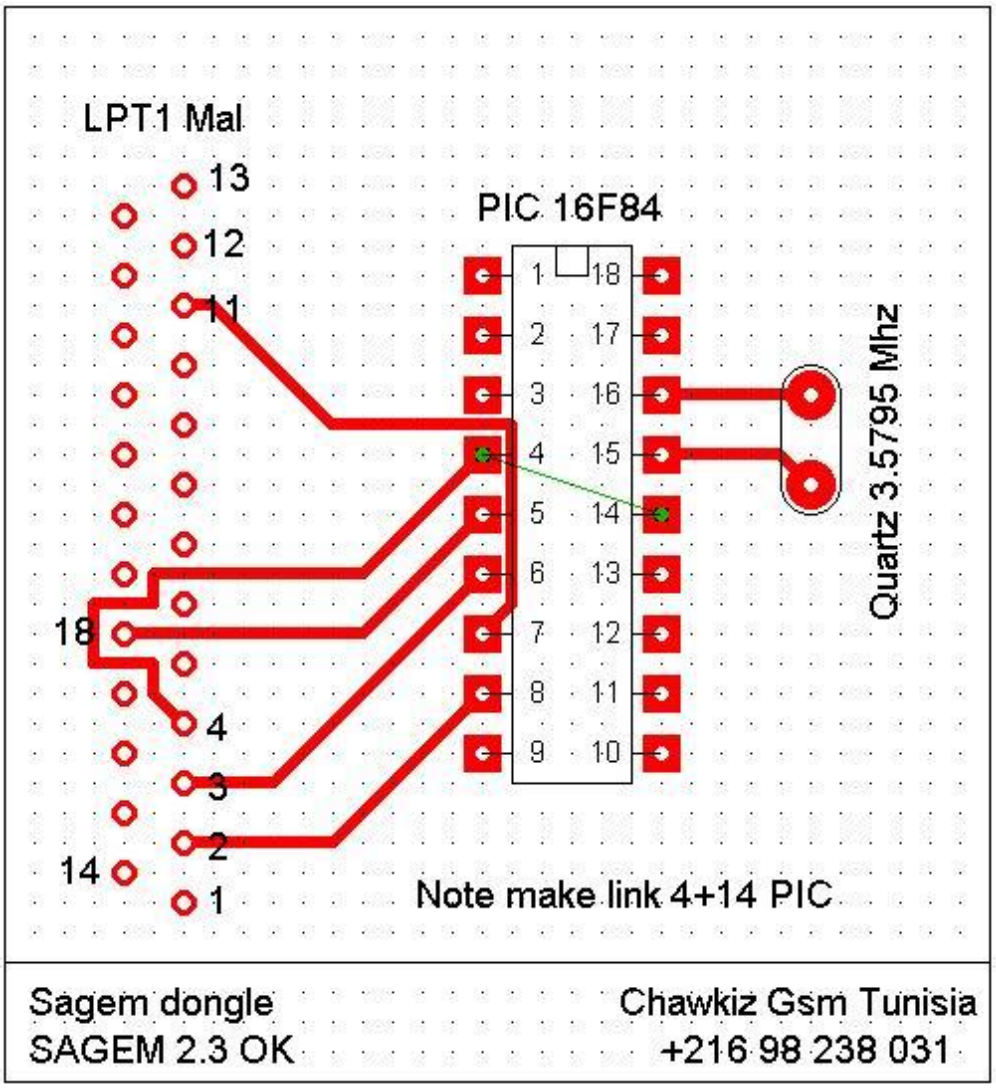

Figura 7. Diagrama de interconexión

De acuerdo con las especificaciones de montaje de componentes y ruteo de pistas, las características de diseño del PCB se resumen en la tabla 3

Pads CI	0.0750 pulg x 0.0750 pulg Drill 0.0450 pulg
Pads Resistencias	0.0750 pulg x 0.0750 pulg Drill 0.0400 pulg
Perforaciones para Montaje	0.1200 pulg
Guarda de la mascarilla antisoldante	0.0100 pulg
Guarda entre pistas de cobre	0.0150 pulg

Tabla 3. Dimensiones de perforaciones y pistas utilizadas en el PCB

Obteniendo un prototipo como se ilustra en la figura 8 siguiente:

Figura 8. Dispositivo candado electrónico

5. Pruebas

5.1 Prueba del candado electrónico

Se ejecutó el programa de validación sin previa instalación del candado electrónico, obteniendo un mensaje de aviso y cancelación de las operaciones. Mediante la instalación oportuna del dispositivo (**Ver figura 9**) en el puerto paralelo y una nueva ejecución del programa se obtiene un mensaje de introducción de la clave personal.

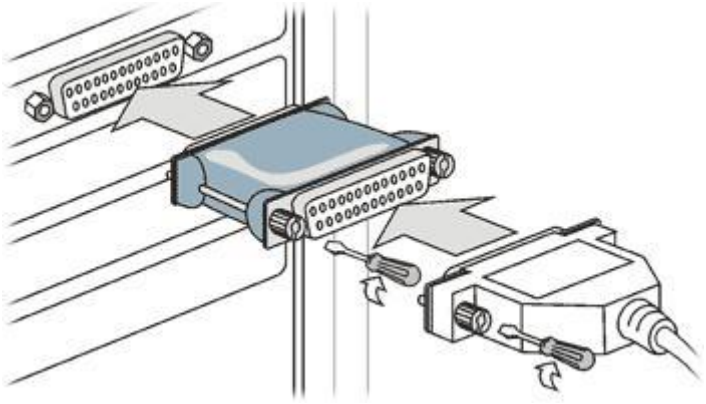

Figura 9. Diagrama de montaje del candado electrónico

En estas condiciones al ingresarse en forma errónea en tres ocasiones la clave de acceso, el programa termina las operaciones. Por el contrario. Al ingresarse en forma adecuada la clave de acceso el programa opera satisfactoriamente mostrando la pantalla principal que permite las operaciones de cifrado o descifrado a desarrollar.

5.2 Prueba de los programas de cifrado

Se seleccionó al azar un programa de bordado que sirviera como archivo de entrada para verificar la operación de los programas. Se determinó que el tamaño del buffer de datos resultó reducido para el tamaño del archivo, determinándose su modificación de 1 KB a 4 KB, pudiendo ampliarse hasta 64KB.

Por otra parte se observó que las claves menores de 6 caracteres o bien con caracteres repetitivos conducen a un bajo nivel de encriptación (entendiéndose como la ocultación de la información).

Del mismo modo, se observó que las claves de 8 caracteres o más y sin caracteres repetidos brindan un buen nivel de ocultación de la información.

6. Conclusiones

De acuerdo con los resultados observados durante la fase de pruebas, se puede establecer un nivel de satisfacción de la operación del sistema de seguridad.

Cabe mencionar que el software de cifrado es capaz de procesar archivos de bordado que no excedan el límite de 64 KB, por lo tanto considerando que los diseños gráficos

día con día se van realizando con una mayor cantidad de detalle y colorido, conduciendo al almacenamiento de archivos de gran tamaño; resulta recomendable en un futuro la realización de adaptaciones al software de cifrado que satisfagan tal demanda de información.

Código de Cifrado Apéndice A

APENDICE A
I. CODIGO DE CIFRADO

```
/* Funciones de manejo de archivos       */
/* Programa de cifrado polialfabetico     */

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<dir.h>

unsigned char TLLano[4096],TCifrado[4096], Clave[256], ruta[256], ruta2[256];
char Archivo[256], Archivo2[256];

char *current_directory(char *path)
{
  strcpy(path, "X:\\");     /* fill string with form of response: X:\ */
  path[0] = 'A' + getdisk();    /* replace X with current drive letter */
  getcurdir(0, path+3);  /* fill rest of string with current directory */
  return(path);
}

int main (void)
{
  int Long=0, i, LongClave=0, j;
  FILE *in, *out;
  char curdir[MAXPATH];
  clrscr();
  system("dir");
```

```
  current_directory(curdir);
  printf("The current directory is %s\n", curdir);

  getch();
  printf("\n Escribe el nombre del archivo origen a cifrar  --");
  gets(Archivo);
  strcat(ruta, Archivo);
  if ((in = fopen(ruta, "rb"))== NULL)
  {
    fprintf(stderr, "No se pudo abrir el archivo.\n");
    return 1;
  }
  printf("\n Escribe el nombre archivo de destino  --");
  gets(Archivo2);
  strcat(ruta2, Archivo2);
  strcat(ruta2, ".cfr");
  printf("\n Ruta = %s", ruta2);
  getch();

  if ((out = fopen(ruta2, "wb"))== NULL)
  {
    fprintf(stderr, "No se pudo abrir el archivo.\n");
    return 1;
  }

/*while (!feof(in))
  fputc(fgetc(in), out);*/

  printf("\n Cadena = %s\n", TLLano);
  printf("\n Escribe la clave  --");
  gets(Clave);
  printf("\n Clave = %s\n", Clave);
  //Long = strlen(TLLano);
  LongClave=strlen(Clave);
  i=0;
  while(!feof(in))
  {
    TLLano[i]=fgetc(in);
    i++;
```

```
  }
  Long = i;
  printf("\n Longitud del Texto plano --> %d", Long);
  printf("\n Longitud de la  Clave    --> %d\n", LongClave);

  /* Cifrado de la informacion */
  j=0;
  for(i= 0; i < Long;i++)
  {
    if(j>=(LongClave-1))j=0;
    TCifrado[i] = TLLano[i] + Clave[j];
    j++;

  }
  for(i =0; i< Long; i++)
    fputc(TCifrado[i], out);

  fclose(in);
  fclose(out);

  printf("\n Cadena        = %s\n", TLLano);
  printf("\n Texto Cifrado   =\n %s", TCifrado);
  printf("\n\n Oprime una tecla para Terminar ...");

  getch();

  return 0;
}
```

Regresar

Código de Descifrado

II. CODIGO DE DESCIFRADO

```
/* Funciones de manejo de archivos       */
/* Programa de cifrado polialfabetico     */

#include<stdio.h>
#include<conio.h>
```

```
#include<string.h>
#include<stdlib.h>
#include<math.h>

unsigned char TCifrado[4096], Clave[256], ruta[256], ruta2[256];
char Archivo[256], Archivo2[256];;
unsigned char TLlano2[4096];

int main (void)
{
  int Long=0, i, LongClave=0, j;
  FILE *in, *out;
  clrscr();
  printf("\n Escribe el nombre del archivo origen a descifrar  --");
  gets(Archivo);
  strcat(ruta, Archivo);
  if ((in = fopen(ruta, "rb"))== NULL)
  {
    fprintf(stderr, "No se pudo abrir el archivo.\n");
    return 1;
  }
  printf("\n Escribe el nombre archivo de destino  --");
  gets(Archivo2);
  strcat(ruta2, Archivo2);
  strcat(ruta2, ".pla");
  printf("\n Ruta = %s", ruta2);
  getch();

  if ((out = fopen(ruta2, "wb"))== NULL)
  {
    fprintf(stderr, "No se pudo abrir el archivo.\n");
    return 1;
  }

/*while (!feof(in))
  fputc(fgetc(in), out);*/

  printf("\n Escribe la clave  --");
  gets(Clave);
  printf("\n Clave = %s\n", Clave);
```

```
  LongClave=strlen(Clave);
  i=0;
  while(!eof(in))
  {
    TCifrado[i]=fgetc(in);
    i++;
  }
  Long = i-1;
  LongClave=strlen(Clave);
  printf("\n Longitud del Texto plano --> %d", Long);
  printf("\n Longitud de la Clave    --> %d\n", LongClave);

    /* Descifrado de la informacion */

//   printf("\n Cadena           = %s\n", TLLano);
  printf("\n Texto Cifrado     = %s\n", TCifrado);
  j=0;
  for(i= 0; i < Long;i++)
  {
    if(j>=(LongClave-1))j=0;
    TLlano2[i] = TCifrado[i] - Clave[j];
    j++;

  }
  printf("\n Texto Descifrado   = ");
  for(i = 0; i < Long;i++)
  {
     printf("%c", TLlano2[i]);
  }

  getch();

  return 0;
}
```

Regresar