Red Grupo de Trabajo H. Ohta
Petición de Comentarios: 2994 M. Matsui
Categoría: Informativo Mitsubishi Electric Corporation
11 2000

Una descripción del algoritmo de cifrado MISTY1

Condición de este Memo

Este memorándum proporciona información para la comunidad de Internet. Lo hace no especifica un estándar de Internet de cualquier tipo. La distribución de este memo es ilimitada.

Aviso de copyright

Abstracto

Este documento describe un sistema de cifrado de clave secreta MISTY1, que es cifrado de bloques con una clave de 128 bits, un bloque de 64 bits y un número variable de rondas. Documenta la descripción del algoritmo incluyendo clave programación de parte y parte aleatoria de datos.

1 . Introducción

Este documento describe un sistema de cifrado de clave secreta MISTY1, que es cifrado de bloques con una clave de 128 bits, un bloque de 64 bits y un número variable de rondas. Está diseñado sobre la base de la teoría de la demostrable la seguridad contra el criptoanálisis diferencial y lineal, y por otra parte que se da cuenta de cifrado de alta velocidad en las plataformas de hardware, así como en entornos de software. Como el resultado de pesaje fuerza y ??velocidad, 8 rondas de MISTY1 se recomienda y se utiliza en la mayoría de los casos.

Nuestra implementación demuestra que MISTY1 con ocho rondas puede cifrar un flujo de datos en el modo CBC a una velocidad de 57Mbps y 40Mbps en Pentium II/266MHz y PA-7200/120MHz, respectivamente. Por su hardware rendimiento, hemos elaborado un prototipo de LSI por un proceso de 0,8- micras CMOS puerta-array y confirmó una velocidad de 512Mbps.

2 . Descripción del algoritmo

Algoritmo [1] se puede dividir en dos partes, a saber, "la programación de la tecla . parte "y" parte aleatoria de datos "Pieza programación se lleva a 128 - llave y de entrada de bits genera una clave de 128 bits ampliado. Datos asignaron al azar

RFC 2994 MISTY1 noviembre 2000 toma parte un conjunto de datos de entrada de 64 bits y lo mezcla, es decir, el cifrado. Si los datos de parte asignaron al azar se procesan en orden inverso, los datos mezclados se transforman en datos de entrada, a saber, de descifrado.

2.1 Terminología

Algunos operadores se utilizan en este documento para describir el algoritmo. El operador `+ 'indica además de complemento a dos. El operador '*' indica multiplicación. El operador '/' se obtiene el cociente y el operador '%' se obtiene el resto de la división. El operador `& 'indica la operación AND bit a bit. El operador `| 'indica bit a bit OR inclusivo operación. El `^ 'indica operador bit a bit OR exclusiva operación. El operador '<<' indica un funcionamiento desviación a la izquierda bit a bit. El operador '>>' indica operación de desplazamiento a la derecha bit a bit.

2.2 Tecla Programación Parte

Pieza programación consta de las siguientes operaciones. para i = 0, ..., 7 hacer EK [i] = K [i * 2] * 256 + K [i * 2 1], para i = 0, ..., 7 Cómo comenzar EK [i + 8] = FI (EK [i], EK [(i +1)% 8]); EK [i 16] = EK [i 8] y 0x1ff; EK [i 24] = EK [i 8] >> 9; extremo K es una tecla de entrada, y cada

elemento de K, a saber, K [i], tiene una de 8 bits de la clave, respectivamente. EK denota una clave ampliada, y cada elemento de EK, a saber EK [i], tiene una de 16 bits de la clave ampliada. Los datos de entrada de K [0], ..., K [15] se copian en EK [0], ..., EK [7]. Se produce clave Ampliado de EK [0], ..., EK [7] utilizando la función FI, y se almacena en EK [8], ..., EK [15]. Función FI se describe en la siguiente sección.

2.3 Datos parte asignaron al azar

Datos parte asignaron al azar utiliza dos tipos de funciones, que se denominan FO función y función de FL. Función FO llama a otra función, es decir, FI. La parte clave de expansión también utiliza la función de FI. Función FI utiliza dos cajas-S, es decir, S7, S9. Cada función se describe como sigue. Función FO toma dos parámetros. Se trata de un conjunto de datos de entrada de ancho de 32 bits, es decir FO_IN. El otro es un índice de EK, a saber, k. Y FO devuelve un conjunto de datos de ancho de 32 bits, es decir FO_OUT. Ohta y Matsui Informativo [Página 2]


```
FO (FO_IN, k)
comenzar
    var t0, t1 como entero de 16 bits;
    t0 = FO_IN >> 16;
    t1 = FO_IN y 0xffff;
    t0 = t0 ^ EK [k];
    t0 = FI (t0, EK [(k 5) 8% 8]);
    t0 t1 = t0 ^;
    t1 = t1 ^ EK [(k 2)% 8];
    t1 = FI (t1, EK [(k +1)% 8 8]);
    t1 = t0 t1 ^;
    t0 = t0 ^ EK [(k 7)% 8];
    t0 = FI (t0, EK [(k +3)% 8 8]);
    t0 t1 = t0 ^;
    t1 = t1 ^ EK [(k +4)% 8];
    FO_OUT = (t1 << 16) | t0;
    volver FO_OUT;
final.
```

Función FI toma dos parámetros. Se trata de un conjunto de datos de entrada de ancho de 16 bits,
a saber FI_IN. La otra es una parte de EK, a saber FI_KEY, que es
También ancho de 16 bits. Y FI devuelve un conjunto de datos de ancho de 16 bits, es decir, FI_OUT.

```
FI (FI_IN, FI_KEY)
comenzar
    var d9 como 9 bits sin signo;
    var d7 como 7 bits sin signo;
    d9 = FI_IN >> 7;
    d7 = FI_IN y 0x7f;
    d9 = S9TABLE [D9] ^ d7;
    d7 = S7TABLE [d7] ^ d9;
    (D7 d7 = y 0x7f;)
    d7 d7 = ^ (FI_KEY >> 9);
    d9 d9 = ^ (FI_KEY y 0x1ff);
    d9 = S9TABLE [D9] ^ d7;
    FI_OUT = (d7 << 9) | d9;
    volver FI_OUT;
final.
```

S7TABLE y S9TABLE denotar el S-cajas S7 y S9, respectivamente, en términos de admiran notación mesa. He aquí la descripción de S7TABLE y S9TABLE en notación hexadecimal.


```
S7TABLE:
     0 1 2 3 4 5 6 7 8 9 abcdef
00: 1b 32 33 5a 3b 10 17 54 72 73 5b 1a 6b 2c 66 49
10: 1f 24 13 6c 37 2e 3f 4a 5d 0f 40 56 25 51 04 1c
20: 0b 46 20 0d 7b 35 44 42 2b 1e 41 14 4b 79 15 6f
30: 0e 55 09 36 74 0C 67 53 28 0a 7e 38 02 07 60 29
40: 19 12 65 2f 30 39 08 68 78 5F 2a 4c 64 45 75 3d
50: 59 48 03 57 7c 4f 62 1d 3c 21 5e 27 6a 70 4d 3a
```

```
60: 01 6d 6e 63 18 77 23 05 26 76 00 31 7F 7a 2d 61
70: 50 22 11 06 47 16 52 4E 71 3e 69 43 34 58 5c 7d

S9TABLE:
       0 1 2 3 4 5 6 7 8 9 abcdef
000: 1c3 0CB 153 19f 1E3 0E9 0fb 035 181 0B9 117 133 009 1eb 02d 0D3
010: 0C7 037 14a 07e 0EB 164 193 1d8 0A3 02c 01d 11e 055 1A2 163 118
020: 14b 152 1d2 00f 02b 13a 0E5 030 111 138 063 18e 0E3 0C8 1F4 01b
030: 001 09d 0f8 1a0 16d 1F3 01c 07d 146 0D1 082 183 1ea 12d 19e 0f4
040: 1d3 0DD 1e2 1e0 0 º C 128 059 091 011 12f 026 0DC 0B0 18c 1F7 10F
050: 0e7 16c 0B6 0f9 0d8 151 101 14C 103 0B8 154 017 071 1ae 12b 00c
060: 047 058 07f 1a4 134 129 084 15d 19d 07c 1B2 1A3 048 051 023 1ca
070: 03b 13d 1A7 165 042 192 0DA 0CE 0C1 06b 09F 1f1 12c 184 0FA 196
080: 1E1 169 17d 031 180 10a 094 1da 186 13e 11c 060 175 067 119 1CF
090: 065 068 099 150 008 007 17c 0B7 024 019 127 0 dB 0DE 0E4 1A9 052
0A0: 109 090 19c 1c1 028 1B3 135 16a 176 0DF 1E5 188 0c5 16e 1de 1b1
0B0: 0C3 1df 036 0EE 1ee 0F0 093 049 09a 1B6 069 081 125 00b 05e 0B4
0C0: 1C7 149 174 03e 13b 08e 1B7 1c6 0AE 010 095 1EF 04s 0f2 1FD 085
0D0: 0fd 0f6 0A0 16f 083 156 08a 09b 13c 107 167 098 1D0 1E9 003 1FE
0e0: 0BD 122 089 0d2 18F 012 033 06a 142 0ED 170 11b 0E2 14f 158 131
0F0: 147 05D 113 1cd 079 161 1A5 179 09e 1B4 0CC 022 132 01a 0E8 004
100: 187 197 039 1ed 1BF 1D7 027 18b 09c 0C6 0D0 14e 06c 034 1F2 06e
110: 0CA 025 0BA 0fe 191 013 106 02f 1ad 172 1db 0C0 10b 1d6 0F5 1EC
120: 10d 076 114 075 1ab 10c 1E4 159 054 11f 04b 0C4 1BE 0F7 029 0A4
130: 00E 1F0 077 086 04d 17a 08b 0B3 171 0BF 10e 104 097 160 168 15b
140: 0d7 0bb 066 1ce 0FC 092 1c5 06F 016 04a 0A1 139 0AF 0f1 190 00a
150: 1aa 143 056 17b 18d 166 0d4 1fb 14d 194 19a 087 1F8 123 0A7 1B8
160: 141 03c 1F9 140 155 02a 11a 1a1 0d5 198 126 061 1af 12e 157 1dc
170: 072 18a 0AA 096 115 045 0EF 07b 08d 145 053 05f 178 0B2 02e 020
180: 1d5 03f 1C9 1E7 1ac 044 038 014 0B1 0B5 05a 16b 0AB 182 1c8 1d4
190: 018 177 064 0cf 06d 100 199 130 005 120 15a 1bb 1bd 0e0 04F 0d6
1a0: 13f 12a 1c4 015 006 0ff 19b 0A6 043 088 050 121 073 1e8 15f 17e
1B0: 0bC 0C2 0C9 173 189 1F5 074 1cc 1e6 1A8 195 01f 041 00d 1ba 032
1C0: 03d 1d1 080 0A8 057 1B9 162 148 0d9 105 062 07a 021 112 108 1FF
1D0: 1C0 0A9 11d 1B0 1A6 0CD 0f3 05c 05b 1D9 102 144 1F6 0ad 0A5 03a
1E0: 1cb 136 17f 046 0E1 01E 1dd 0e6 137 1FA 185 08c 08F 040 1B5 0BE
1F0: 078 000 110 0AC 15e 124 002 1bc 0A2 0EA 070 1fc 116 15c 04c 1c2
```


Función FL toma dos parámetros. Se trata de un conjunto de datos de 32 bits, es decir, FL_IN. El otro es un índice de EK, a saber, k. Y FL devuelve un 32 - datos de ancho de bit, es decir, FL_OUT.

```
FL (FL_IN, k)
comenzar
    var d0, d1 como entero de 16 bits;
    d0 = FL_IN >> 16;
    d1 = FL_IN y 0xffff;
    si (k es un número par) y luego
        d1 d1 = ^ (d0 y EK [k / 2]);
        d0 d0 = ^ (d1 | EK [(k / 2 +6) 8% 8]);
    más
        d1 = d1 ^ (d0 y EK [((k-1) / 2 2) 8% 8]);
        d0 d0 = ^ (d1 | EK [((k-1) / 2 4) 8%]);
    endif
    FL_OUT = (d0 << 16) | d1;
    volver FL_OUT;
final.
```

Cuando el algoritmo se utiliza para el descifrado, se utiliza la función FLINV en lugar de la función de FL.

```
FLINV (FL_IN, k)
comenzar
    var d0, d1 como entero de 16 bits;
    d0 = FL_IN >> 16;
    d1 = FL_IN y 0xffff;
    si (k es un número par) y luego
        d0 d0 = ^ (d1 | EK [(k / 2 +6) 8% 8]);
        d1 d1 = ^ (d0 y EK [k / 2]);
    más
        d0 d0 = ^ (d1 | EK [((k-1) / 2 4) 8%]);
        d1 = d1 ^ (d0 y EK [((k-1) / 2 2) 8% 8]);
```

```
        endif
        FL_OUT = (d0 << 16) | d1;
        volver FL_OUT;
    final.
```

En la mayoría de los casos, los datos que asignaron al azar parte consta de 8 "rondas". Ronda contiene la llamada de la función FO. Además, redondo en número par incluye las llamadas de función de FL. Después de la ronda final, FLs son llama de nuevo. La descripción del detalle es el siguiente.

64-bit de texto plano P se divide en el extremo izquierdo de 32 bits D0 y el más a la derecha D1 32 bits.


```
    / / 0 redonda
    D0 = FL (D0, 0);
    D1 = FL (D1, 1);
    D1 = D1 ^ FO (D0, 0);
    / / 1 año
    D0 = D0 ^ FO (D1, 1);
    / / 2 vuelta
    D0 = FL (D0, 2);
    D1 = FL (D1, 3);
    D1 = D1 ^ FO (D0, 2);
    / / 3 ronda
    D0 = D0 ^ FO (D1, 3);
    / / 4 de vuelta
    D0 = FL (D0, 4);
    D1 = FL (D1, 5);
    D1 = D1 ^ FO (D0, 4);
    / / 5 ronda
    D0 D0 = ^ FO (D1, 5);
    / / 6 ronda
    D0 = FL (D0, 6);
    D1 = FL (D1, 7);
    D1 = D1 ^ FO (D0, 6);
    / / 7 ronda
    D0 D0 = ^ FO (D1, 7);
    / / Finales
    D0 = FL (D0, 8);
    D1 = FL (D1, 9);
```

64-bit de texto cifrado C se construye a partir de D0 y D1 de la siguiente operación.

```
    C = (D1 << 32) | D0;
```

Cuando los datos se asignaron al azar a parte se utiliza como operación de descifrado, lo que debería ser ejecutados en orden inverso. La descripción del detalle es el siguiente.

```
    D0 = C & 0xFFFFFFFF;
    D1 = C >> 32;
    D0 = FLINV (D0, 8);
    D1 = FLINV (D1, 9);
    D0 D0 = ^ FO (D1, 7);
    D1 = D1 ^ FO (D0, 6);
    D0 = FLINV (D0, 6);
    D1 = FLINV (D1, 7);
    D0 D0 = ^ FO (D1, 5);
    D1 = D1 ^ FO (D0, 4);
    D0 = FLINV (D0, 4);
    D1 = FLINV (D1, 5);
```



```
    D0 = D0 ^ FO (D1, 3);
    D1 = D1 ^ FO (D0, 2);
    D0 = FLINV (D0, 2);
```

```
    D1 = FLINV (D1, 3);
    D0 = D0 ^ FO (D1, 1);
    D1 = D1 ^ FO (D0, 0);
    D0 = FLINV (D0, 0);
    D1 = FLINV (D1, 1);
    P = (D0 << 32) | D1;
```

3 .

Identificador de objeto

El modo de identificador de objeto para MISTY1 en Cipher Block Chaining (CBC) es la siguiente:

MISTY1-CBC OBJECT IDENTIFIER :: =
{Iso (1) miembro del cuerpo (2) CSAC (392)
Mitsubishi-Electric-empresa (200.011) isl (61) Seguridad (1)
Algoritmo (1) simétrica de cifrado algoritmo (1) misty1-cbc (1)}

MISTY1-CBC necesita vector de inicialización (IV) como como como otra algoritmos, tales como DES-CBC, DES-EDE3-CBC y así sucesivamente. Para determinar el valor de IV, MISTY1-CBC tiene parámetros como:

MISTY1-CBC parámetro :: = IV

donde IV :: = OCTET STRING - 8 octetos.

Cuando se utiliza este identificador de objeto, texto plano se rellena antes de cifrarlo. Al menos 1 octeto de relleno se añade al final de la texto plano para que la longitud de el texto en claro hasta el múltiplo de 8 octetos. El valor de estos octetos es el mismo que el número de octetos adjunta. (Por ejemplo, si se necesitan 5 octetos para rellenar, el valor es 0x05.)

4 .

Consideraciones de seguridad

El algoritmo, que se describe en este documento, está diseñado en consideración de la teoría de la seguridad demostrable contra diferencial criptoanálisis y el criptoanálisis lineal [2] [3] [4]. De acuerdo con la reciente resultado, cuando el algoritmo se compone de 8 rondas, tanto probabilidad diferencial característico y característica forro probabilidad es 2 ^ -140. Como referencia, las probabilidades de DES son 2 ^ - 62 y 2 ^ -46, respectivamente.

5 .

Cuestiones legales

La descripción del algoritmo se aplica para una patente en alguna países como PCT/JP96/02154. Sin embargo, el algoritmo es libremente disponibles para uso académico (no lucrativa). Además, el algoritmo se puede utilizar para uso comercial sin tener que pagar la cuota de la patente si contrato con Mitsubishi Electric Corporation. Para obtener más información, por favor contácteme en MISTY@isl.melco.co.jp.

6 .

Referencias

[1] M. Matsui, "Nuevo bloque Algoritmo de cifrado MISTY", Software Fast Cifrado - 4 º Workshop Internacional (FSE'97), LNCS 1267,

Springer Verlag, 1997, pp.54-68

[2] K. Nyberg y LR Knudsen, "seguridad demostrable contra un Diferencial Attack ", Revista de Criptología, Vol.8, N º 1, 1995, pp 27-37

[3] K. Nyberg, "Aproximación lineal de cifrado por bloques", Avances en Criptología - Eurocrypt'94, LNCS 950, Springer Verlag, 1995, pp.439-444

[4] M. Matsui, "Nueva Estructura de cifrado por bloques, con comprobables Seguridad contra criptoanálisis diferencial y lineal ", Fast Encryption Software - Tercer Taller Internacional, LNCS 1039, Springer Verlag, 1996, pp.205-218

7 .

De los autores Direcciones

Hidenori Ohta
Mitsubishi Electric Corporation, Information Technology R & D Center
5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japón

Teléfono: +81-467-41-2183
Fax: +81-467-41-2185
EMail: hidenori@iss.isl.melco.co.jp

Mitsuru Matsui
Mitsubishi Electric Corporation, Information Technology R & D Center
5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japón

Teléfono: +81-467-41-2181
Fax: +81-467-41-2185
EMail: matsui@iss.isl.melco.co.jp

Apéndice A . Ejemplo Datos de MISTY1

Aquí hay un ejemplo de texto cifrado MISTY1 cuando la llave y el texto plano se establece como valor siguiente.

Clave: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
Texto plano: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10
Texto cifrado: 8b 1d a5 f5 6a b3 d0 7c 04 b6 82 40 b1 3b e9 5d

En el ejemplo anterior, ya que el texto en claro tiene una longitud de 128 bits, MISTY1 se utiliza dos veces para cada 64 bits, a saber, el modo de BCE.

Siguiendo el ejemplo es el texto cifrado de MISTY1 en modo CBC.

Clave: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
IV: 01 02 03 04 05 06 07 08
Texto plano: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10
Texto cifrado: 46 1c 1e 87 9c 18 c2 7mo b9 ad f2 d8 0c 89 03 1f

Completo declaración de los Derechos de Autor

Reconocimiento

La financiación de la función del Editor RFC es actualmente el Internet Society.