

## MD5 Kolissions-Generator auf der Grundlage von Xiaoyun Wang

```
/* MD5 Collision Generator by Patrick Stach <pstach@stachliu.com>
 * Implementation of paper by Xiaoyun Wang, et all.
 *
 * A few optimizations to make the solving method a bit more
deterministic
 *
 * Usage:
 *     ./md5coll or ./md5coll IV0 IV1 IV2 IV3
 *
 * Requires being built as 32 bit (unsigned int as 32 bit)
 *
 * Any derivative works or references must cite the authors.
 */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>

#define F(x, y, z) (z ^ (x & (y ^ z)))
#define G(x, y, z) F(z, x, y)
#define H(x, y, z) (x ^ y ^ z)
#define I(x, y, z) (y ^ (x | ~z))

#define RL(x, y) (((x) << (y)) | ((x) >> (32 - (y))))
#define RR(x, y) (((x) >> (y)) | ((x) << (32 - (y))))

unsigned int A0, B0, C0, D0;
unsigned int A1, B1, C1, D1;

unsigned int IV[4] = { 0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476 };
unsigned int Q0[65], Q1[65];
unsigned int X0[32], X1[32];

void block1(void)
{
    size_t i, max;

block1_again:
    for(;;)
    {
        /* C1 */
        Q0[ 3] = random() & ~0x00800040;
        Q1[ 3] = Q0[ 3];

        /* B1 */
        Q0[ 4] = (random() | 0x80080800) & ~(0x00800040 | 0x0077f780);
        Q0[ 4] |= (Q0[ 3] & 0x0077f780);
        Q1[ 4] = Q0[ 4];

        /* A2 */
        Q0[ 5] = (random() | 0x88400025) & ~0x02bffc0;
        Q1[ 5] = Q0[ 5] - 0x00000040;

        /* D2 */
        Q0[ 6] = (random() | 0x027fbc41) & ~(0x888043a4 | 0x7500001a);
```

```

Q0[ 6] |= (Q0[ 5] & 0x7500001a);
Q1[ 6] = Q0[ 6] - 0x7f800040;

/* C2 */
Q0[ 7] = (random() | 0x03fef820) & ~0xfc0107df;
Q1[ 7] = Q0[ 7] - 0x07800041;

X0[ 6] = RR(Q0[ 7] - Q0[ 6], 17) - F(Q0[ 6], Q0[ 5], Q0[ 4])
      - Q0[ 3] - 0xa8304613;
X1[ 6] = RR(Q1[ 7] - Q1[ 6], 17) - F(Q1[ 6], Q1[ 5], Q1[ 4])
      - Q1[ 3] - 0xa8304613;
if(X0[ 6] != X1[ 6])
    continue;

/* B2 */
Q0[ 8] = (random() | 0x01910540) & ~0xfe0eaabf;
Q1[ 8] = Q0[ 8] - 0x00827fff;

X0[ 7] = RR(Q0[ 8] - Q0[ 7], 22) - F(Q0[ 7], Q0[ 6], Q0[ 5])
      - Q0[ 4] - 0xfd469501;
X1[ 7] = RR(Q1[ 8] - Q1[ 7], 22) - F(Q1[ 7], Q1[ 6], Q1[ 5])
      - Q1[ 4] - 0xfd469501;
if(X0[ 7] != X1[ 7])
    continue;

/* A3 */
Q0[ 9] = (random() | 0xfb102f3d) & ~(0x040f80c2 | 0x00001000);
Q0[ 9] |= (Q0[ 8] & 0x00001000);
Q1[ 9] = Q0[ 9] - 0x8000003f;

X0[ 8] = RR(Q0[ 9] - Q0[ 8], 7) - F(Q0[ 8], Q0[ 7], Q0[ 6])
      - Q0[ 5] - 0x698098d8;
X1[ 8] = RR(Q1[ 9] - Q1[ 8], 7) - F(Q1[ 8], Q1[ 7], Q1[ 6])
      - Q1[ 5] - 0x698098d8;
if(X0[ 8] != X1[ 8])
    continue;

/* D3 */
Q0[10] = (random() | 0x401f9040) & ~0x80802183;
Q1[10] = Q0[10] - 0x7ffff000;

X0[ 9] = RR(Q0[10] - Q0[ 9], 12) - F(Q0[ 9], Q0[ 8], Q0[ 7])
      - Q0[ 6] - 0x8b44f7af;
X1[ 9] = RR(Q1[10] - Q1[ 9], 12) - F(Q1[ 9], Q1[ 8], Q1[ 7])
      - Q1[ 6] - 0x8b44f7af;
if(X0[ 9] != X1[ 9])
    continue;

/* C3 */
Q0[11] = (random() | 0x000180c2) & ~(0xc00e3101 | 0x00004000);
Q0[11] |= (Q0[10] & 0x00004000);
Q1[11] = Q0[11] - 0x40000000;

X0[10] = RR(Q0[11] - Q0[10], 17) - F(Q0[10], Q0[ 9], Q0[ 8])
      - Q0[ 7] - 0xffff5bb1;
X1[10] = RR(Q1[11] - Q1[10], 17) - F(Q1[10], Q1[ 9], Q1[ 8])
      - Q1[ 7] - 0xffff5bb1;
if(X0[10] != X1[10])
    continue;

```

```

/* B3 */
Q0[12] = (random() | 0x00081100) & ~(0xc007e080 | 0x03000000);
Q0[12] |= (Q0[11] & 0x03000000);
Q1[12] = Q0[12] - 0x80002080;

X0[11] = RR(Q0[12] - Q0[11], 22) - F(Q0[11], Q0[10], Q0[ 9])
        - Q0[ 8] - 0x895cd7be;
X1[11] = RR(Q1[12] - Q1[11], 22) - F(Q1[11], Q1[10], Q1[ 9])
        - Q1[ 8] - 0x895cd7be;
if((X0[11] ^ X1[11]) != 0x00008000)
    continue;

/* A4 */
Q0[13] = (random() | 0x410fe008) & ~0x82000180;
Q1[13] = Q0[13] - 0x7f000000;

X0[12] = RR(Q0[13] - Q0[12], 7) - F(Q0[12], Q0[11], Q0[10])
        - Q0[ 9] - 0x6b901122;
X1[12] = RR(Q1[13] - Q1[12], 7) - F(Q1[12], Q1[11], Q1[10])
        - Q1[ 9] - 0x6b901122;
if(X0[12] != X1[12])
    continue;

/* D4 */
Q0[14] = (random() | 0x000be188) & ~0xa3040000;
Q1[14] = Q0[14] - 0x80000000;

X0[13] = RR(Q0[14] - Q0[13], 12) - F(Q0[13], Q0[12], Q0[11])
        - Q0[10] - 0xfd987193;
X1[13] = RR(Q1[14] - Q1[13], 12) - F(Q1[13], Q1[12], Q1[11])
        - Q1[10] - 0xfd987193;
if(X0[13] != X1[13])
    continue;

/* C4 */
Q0[15] = (random() | 0x21008000) & ~0x82000008;
Q1[15] = Q0[15] - 0x80007ff8;

X0[14] = RR(Q0[15] - Q0[14], 17) - F(Q0[14], Q0[13], Q0[12])
        - Q0[11] - 0xa679438e;
X1[14] = RR(Q1[15] - Q1[14], 17) - F(Q1[14], Q1[13], Q1[12])
        - Q1[11] - 0xa679438e;
if((X0[14] ^ X1[14]) != 0x80000000)
    continue;

/* B4 */
Q0[16] = (random() | 0x20000000) & ~0x80000000;
Q1[16] = Q0[16] - 0xa0000000;

X0[15] = RR(Q0[16] - Q0[15], 22) - F(Q0[15], Q0[14], Q0[13])
        - Q0[12] - 0x49b40821;
X1[15] = RR(Q1[16] - Q1[15], 22) - F(Q1[15], Q1[14], Q1[13])
        - Q1[12] - 0x49b40821;
if(X0[15] != X1[15])
    continue;
break;
}

```

```
#define LOOP_11 300
```

```

for(i = 0; i < LOOP_11; i++)
{
    /* A5 */
    Q0[17] = random() & ~(0x80020000 | 0x00008008);
    Q0[17] |= (Q0[16] & 0x00008008);
    Q1[17] = Q0[17] - 0x80000000;

    X0[ 1] = RR(Q0[17] - Q0[16], 5) - G(Q0[16], Q0[15], Q0[14])
            - Q0[13] - 0xf61e2562;
    X1[ 1] = RR(Q1[17] - Q1[16], 5) - G(Q1[16], Q1[15], Q1[14])
            - Q1[13] - 0xf61e2562;
    if(X0[ 1] != X1[ 1])
        continue;

    /* D5 */
    Q0[18] = RL(G(Q0[17], Q0[16], Q0[15]) + Q0[14]
                + X0[ 6] + 0xc040b340, 9) + Q0[17];
    if((Q0[18] & 0xa0020000)
        != (0x00020000 | (Q0[17] & 0x20000000)))
    {
        continue;
    }
    Q1[18] = RL(G(Q1[17], Q1[16], Q1[15]) + Q1[14]
                + X1[ 6] + 0xc040b340, 9) + Q1[17];
    if((Q0[18] ^ Q1[18]) != 0x80000000)
        continue;

    /* C5 */
    Q0[19] = RL(G(Q0[18], Q0[17], Q0[16]) + Q0[15]
                + X0[11] + 0x265e5a51, 14) + Q0[18];
    if(Q0[19] & 0x80020000)
        continue;
    Q1[19] = RL(G(Q1[18], Q1[17], Q1[16]) + Q1[15]
                + X1[11] + 0x265e5a51, 14) + Q1[18];
    if(Q0[19] - Q1[19] != 0x7ffe0000)
        continue;

    /* B5 */
    Q0[20] = random() & ~0x80000000;
    Q1[20] = Q0[20] - 0x80000000;

    X0[ 0] = RR(Q0[20] - Q0[19], 20) - G(Q0[19], Q0[18], Q0[17])
            - Q0[16] - 0xe9b6c7aa;
    X1[ 0] = RR(Q1[20] - Q1[19], 20) - G(Q1[19], Q1[18], Q1[17])
            - Q1[16] - 0xe9b6c7aa;
    if(X0[ 0] != X1[ 0])
        continue;

    Q0[ 1] = RL(F(IV[1], IV[2], IV[3]) + IV[0]
                + X0[ 0] + 0xd76aa478, 7) + IV[1];
    Q1[ 1] = Q0[ 1];

    Q0[ 2] = RL(F(Q0[ 1], IV[1], IV[2]) + IV[3]
                + X0[ 1] + 0xe8c7b756, 12) + Q0[ 1];
    Q1[ 2] = Q0[ 2];

    X0[ 2] = RR(Q0[ 3] - Q0[ 2], 17) - F(Q0[ 2], Q0[ 1], IV[1])
            - IV[2] - 0x242070db;
    X1[ 2] = X0[ 2];
}

```

```

X0[ 3] = RR(Q0[ 4] - Q0[ 3], 22) - F(Q0[ 3], Q0[ 2], Q0[ 1])
      - IV[1] - 0xc1bdceee;
X1[ 3] = X0[ 3];

X0[ 4] = RR(Q0[ 5] - Q0[ 4],  7) - F(Q0[ 4], Q0[ 3], Q0[ 2])
      - Q0[ 1] - 0xf57c0faf;
X1[ 4] = RR(Q1[ 5] - Q1[ 4],  7) - F(Q1[ 4], Q1[ 3], Q1[ 2])
      - Q1[ 1] - 0xf57c0faf;
if((X0[ 4] ^ X1[ 4]) != 0x80000000)
    continue;

X0[ 5] = RR(Q0[ 6] - Q0[ 5], 12) - F(Q0[ 5], Q0[ 4], Q0[ 3])
      - Q0[ 2] - 0x4787c62a;
X1[ 5] = RR(Q1[ 6] - Q1[ 5], 12) - F(Q1[ 5], Q1[ 4], Q1[ 3])
      - Q1[ 2] - 0x4787c62a;
if(X0[ 5] != X1[ 5])
    continue;

/* A6 */
Q0[21] = RL(G(Q0[20], Q0[19], Q0[18]) + Q0[17]
      + X0[ 5] + 0xd62f105d,  5) + Q0[20];
if((Q0[21] & 0x80020000) != (Q0[20] & 0x00020000))
    continue;
Q1[21] = RL(G(Q1[20], Q1[19], Q1[18]) + Q1[17]
      + X1[ 5] + 0xd62f105d,  5) + Q1[20];
if((Q0[21] ^ Q1[21]) != 0x80000000)
    continue;

/* D6 */
Q0[22] = RL(G(Q0[21], Q0[20], Q0[19]) + Q0[18]
      + X0[10] + 0x02441453,  9) + Q0[21];
if(Q0[22] & 0x80000000)
    continue;
Q1[22] = RL(G(Q1[21], Q1[20], Q1[19]) + Q1[18]
      + X1[10] + 0x02441453,  9) + Q1[21];
if((Q0[22] ^ Q1[22]) != 0x80000000)
    continue;

/* C6 */
Q0[23] = RL(G(Q0[22], Q0[21], Q0[20]) + Q0[19]
      + X0[15] + 0xd8a1e681, 14) + Q0[22];
if(Q0[23] & 0x80000000)
    continue;
Q1[23] = RL(G(Q1[22], Q1[21], Q1[20]) + Q1[19]
      + X1[15] + 0xd8a1e681, 14) + Q1[22];
if(Q0[23] != Q1[23])
    continue;

/* B6 */
Q0[24] = RL(G(Q0[23], Q0[22], Q0[21]) + Q0[20]
      + X0[ 4] + 0xe7d3fbc8, 20) + Q0[23];
Q1[24] = RL(G(Q1[23], Q1[22], Q1[21]) + Q1[20]
      + X1[ 4] + 0xe7d3fbc8, 20) + Q1[23];
if(Q0[24] != Q1[24])
    continue;

/* A7 */
Q0[25] = RL(G(Q0[24], Q0[23], Q0[22]) + Q0[21]
      + X0[ 9] + 0x21e1cde6,  5) + Q0[24];
Q1[25] = RL(G(Q1[24], Q1[23], Q1[22]) + Q1[21]

```

```

        + X1[ 9] + 0x21e1cde6, 5) + Q1[24];
if(Q0[25] != Q1[25])
    continue;

    /* D7 */
    Q0[26] = RL(G(Q0[25], Q0[24], Q0[23]) + Q0[22]
        + X0[14] + 0xc33707d6, 9) + Q0[25];
    Q1[26] = RL(G(Q1[25], Q1[24], Q1[23]) + Q1[22]
        + X1[14] + 0xc33707d6, 9) + Q1[25];
    if(Q0[26] != Q1[26])
        continue;

    /* C7 */
    Q0[27] = RL(G(Q0[26], Q0[25], Q0[24]) + Q0[23]
        + X0[ 3] + 0xf4d50d87, 14) + Q0[26];
    Q1[27] = RL(G(Q1[26], Q1[25], Q1[24]) + Q1[23]
        + X1[ 3] + 0xf4d50d87, 14) + Q1[26];
    if(Q0[27] != Q1[27])
        continue;

    break;
}
if(i >= LOOP_11)
    goto block1_again;

#define LOOP_12 0x20000000

for(i = 0; i < LOOP_12; i++)
{
    /* B5 */
    Q0[20] ^= (1 << (random() % 31));
    Q1[20] = Q0[20] - 0x80000000;

    X0[ 0] = RR(Q0[20] - Q0[19], 20) - G(Q0[19], Q0[18], Q0[17])
        - Q0[16] - 0xe9b6c7aa;
    X1[ 0] = RR(Q1[20] - Q1[19], 20) - G(Q1[19], Q1[18], Q1[17])
        - Q1[16] - 0xe9b6c7aa;
    if(X0[ 0] != X1[ 0])
        continue;

    Q0[ 1] = RL(F(IV[1], IV[2], IV[3]) + IV[0]
        + X0[ 0] + 0xd76aa478, 7) + IV[1];
    Q1[ 1] = Q0[ 1];

    Q0[ 2] = RL(F(Q0[ 1], IV[1], IV[2]) + IV[3]
        + X0[ 1] + 0xe8c7b756, 12) + Q0[ 1];
    Q1[ 2] = Q0[ 2];
    X0[ 2] = RR(Q0[ 3] - Q0[ 2], 17) - F(Q0[ 2], Q0[ 1], IV[1])
        - IV[2] - 0x242070db;
    X1[ 2] = X0[ 2];

    X0[ 3] = RR(Q0[ 4] - Q0[ 3], 22) - F(Q0[ 3], Q0[ 2], Q0[ 1])
        - IV[1] - 0xc1bdceee;
    X1[ 3] = X0[ 3];

    X0[ 4] = RR(Q0[ 5] - Q0[ 4], 7) - F(Q0[ 4], Q0[ 3], Q0[ 2])
        - Q0[ 1] - 0xf57c0faf;
    X1[ 4] = RR(Q1[ 5] - Q1[ 4], 7) - F(Q1[ 4], Q1[ 3], Q1[ 2])
        - Q1[ 1] - 0xf57c0faf;
    if((X0[ 4] ^ X1[ 4]) != 0x80000000)

```

```

        continue;

X0[ 5] = RR(Q0[ 6] - Q0[ 5], 12) - F(Q0[ 5], Q0[ 4], Q0[ 3])
        - Q0[ 2] - 0x4787c62a;
X1[ 5] = RR(Q1[ 6] - Q1[ 5], 12) - F(Q1[ 5], Q1[ 4], Q1[ 3])
        - Q1[ 2] - 0x4787c62a;
if(X0[ 5] != X1[ 5])
    continue;

/* A6 */
Q0[21] = RL(G(Q0[20], Q0[19], Q0[18]) + Q0[17]
            + X0[ 5] + 0xd62f105d, 5) + Q0[20];
if((Q0[21] & 0x80020000) != (Q0[20] & 0x00020000))
    continue;
Q1[21] = RL(G(Q1[20], Q1[19], Q1[18]) + Q1[17]
            + X1[ 5] + 0xd62f105d, 5) + Q1[20];
if((Q0[21] ^ Q1[21]) != 0x80000000)
    continue;

/* D6 */
Q0[22] = RL(G(Q0[21], Q0[20], Q0[19]) + Q0[18]
            + X0[10] + 0x02441453, 9) + Q0[21];
if(Q0[22] & 0x80000000)
    continue;
Q1[22] = RL(G(Q1[21], Q1[20], Q1[19]) + Q1[18]
            + X1[10] + 0x02441453, 9) + Q1[21];
if((Q0[22] ^ Q1[22]) != 0x80000000)
    continue;

/* C6 */
Q0[23] = RL(G(Q0[22], Q0[21], Q0[20]) + Q0[19]
            + X0[15] + 0xd8a1e681, 14) + Q0[22];
if(Q0[23] & 0x80000000)
    continue;
Q1[23] = RL(G(Q1[22], Q1[21], Q1[20]) + Q1[19]
            + X1[15] + 0xd8a1e681, 14) + Q1[22];
if(Q0[23] != Q1[23])
    continue;

/* B6 */
Q0[24] = RL(G(Q0[23], Q0[22], Q0[21]) + Q0[20]
            + X0[ 4] + 0xe7d3fbc8, 20) + Q0[23];
Q1[24] = RL(G(Q1[23], Q1[22], Q1[21]) + Q1[20]
            + X1[ 4] + 0xe7d3fbc8, 20) + Q1[23];
if(Q0[24] != Q1[24])
    continue;

/* A7 */
Q0[25] = RL(G(Q0[24], Q0[23], Q0[22]) + Q0[21]
            + X0[ 9] + 0x21e1cde6, 5) + Q0[24];
Q1[25] = RL(G(Q1[24], Q1[23], Q1[22]) + Q1[21]
            + X1[ 9] + 0x21e1cde6, 5) + Q1[24];
if(Q0[25] != Q1[25])
    continue;

/* D7 */
Q0[26] = RL(G(Q0[25], Q0[24], Q0[23]) + Q0[22]
            + X0[14] + 0xc33707d6, 9) + Q0[25];
Q1[26] = RL(G(Q1[25], Q1[24], Q1[23]) + Q1[22]
            + X1[14] + 0xc33707d6, 9) + Q1[25];

```

```

if(Q0[26] != Q1[26])
    continue;

/* C7 */
Q0[27] = RL(G(Q0[26], Q0[25], Q0[24]) + Q0[23]
    + X0[ 3] + 0xf4d50d87, 14) + Q0[26];
Q1[27] = RL(G(Q1[26], Q1[25], Q1[24]) + Q1[23]
    + X1[ 3] + 0xf4d50d87, 14) + Q1[26];
if(Q0[27] != Q1[27])
    continue;

/* B7 */
Q0[28] = RL(G(Q0[27], Q0[26], Q0[25]) + Q0[24]
    + X0[ 8] + 0x455a14ed, 20) + Q0[27];
Q1[28] = RL(G(Q1[27], Q1[26], Q1[25]) + Q1[24]
    + X1[ 8] + 0x455a14ed, 20) + Q1[27];
if(Q0[28] != Q1[28])
    continue;

/* A8 */
Q0[29] = RL(G(Q0[28], Q0[27], Q0[26]) + Q0[25]
    + X0[13] + 0xa9e3e905, 5) + Q0[28];
Q1[29] = RL(G(Q1[28], Q1[27], Q1[26]) + Q1[25]
    + X1[13] + 0xa9e3e905, 5) + Q1[28];
if(Q0[29] != Q1[29])
    continue;

/* D8 */
Q0[30] = RL(G(Q0[29], Q0[28], Q0[27]) + Q0[26]
    + X0[ 2] + 0xfcefa3f8, 9) + Q0[29];
Q1[30] = RL(G(Q1[29], Q1[28], Q1[27]) + Q1[26]
    + X1[ 2] + 0xfcefa3f8, 9) + Q1[29];
if(Q0[30] != Q1[30])
    continue;

/* C8 */
Q0[31] = RL(G(Q0[30], Q0[29], Q0[28]) + Q0[27]
    + X0[ 7] + 0x676f02d9, 14) + Q0[30];
Q1[31] = RL(G(Q1[30], Q1[29], Q1[28]) + Q1[27]
    + X1[ 7] + 0x676f02d9, 14) + Q1[30];
if(Q0[31] != Q1[31])
    continue;

/* B8 */
Q0[32] = RL(G(Q0[31], Q0[30], Q0[29]) + Q0[28]
    + X0[12] + 0x8d2a4c8a, 20) + Q0[31];
Q1[32] = RL(G(Q1[31], Q1[30], Q1[29]) + Q1[28]
    + X1[12] + 0x8d2a4c8a, 20) + Q1[31];
if(Q0[32] != Q1[32])
    continue;

/* A9 */
Q0[33] = RL(H(Q0[32], Q0[31], Q0[30]) + Q0[29]
    + X0[ 5] + 0xfffa3942, 4) + Q0[32];
Q1[33] = RL(H(Q1[32], Q1[31], Q1[30]) + Q1[29]
    + X1[ 5] + 0xfffa3942, 4) + Q1[32];
if(Q0[33] != Q1[33])
    continue;

/* D9 */

```

```

Q0[34] = RL(H(Q0[33], Q0[32], Q0[31]) + Q0[30]
           + X0[ 8] + 0x8771f681, 11) + Q0[33];
Q1[34] = RL(H(Q1[33], Q1[32], Q1[31]) + Q1[30]
           + X1[ 8] + 0x8771f681, 11) + Q1[33];
if(Q0[34] != Q1[34])
    continue;

/* C9 */
Q0[35] = RL(H(Q0[34], Q0[33], Q0[32]) + Q0[31]
           + X0[11] + 0x6d9d6122, 16) + Q0[34];
Q1[35] = RL(H(Q1[34], Q1[33], Q1[32]) + Q1[31]
           + X1[11] + 0x6d9d6122, 16) + Q1[34];
if((Q0[35] ^ Q1[35]) != 0x80000000)
    continue;

/* B9 */
Q0[36] = RL(H(Q0[35], Q0[34], Q0[33]) + Q0[32]
           + X0[14] + 0xfde5380c, 23) + Q0[35];
Q1[36] = RL(H(Q1[35], Q1[34], Q1[33]) + Q1[32]
           + X1[14] + 0xfde5380c, 23) + Q1[35];
if((Q0[36] ^ Q1[36]) != 0x80000000)
    continue;

/* A10 */
Q0[37] = RL(H(Q0[36], Q0[35], Q0[34]) + Q0[33]
           + X0[ 1] + 0xa4beea44,  4) + Q0[36];
Q1[37] = RL(H(Q1[36], Q1[35], Q1[34]) + Q1[33]
           + X1[ 1] + 0xa4beea44,  4) + Q1[36];
if((Q0[37] ^ Q1[37]) != 0x80000000)
    continue;

/* D10 */
Q0[38] = RL(H(Q0[37], Q0[36], Q0[35]) + Q0[34]
           + X0[ 4] + 0x4bdecfa9, 11) + Q0[37];
Q1[38] = RL(H(Q1[37], Q1[36], Q1[35]) + Q1[34]
           + X1[ 4] + 0x4bdecfa9, 11) + Q1[37];
if((Q0[38] ^ Q1[38]) != 0x80000000)
    continue;

/* C10 */
Q0[39] = RL(H(Q0[38], Q0[37], Q0[36]) + Q0[35]
           + X0[ 7] + 0xf6bb4b60, 16) + Q0[38];
Q1[39] = RL(H(Q1[38], Q1[37], Q1[36]) + Q1[35]
           + X1[ 7] + 0xf6bb4b60, 16) + Q1[38];
if((Q0[39] ^ Q1[39]) != 0x80000000)
    continue;

/* B10 */
Q0[40] = RL(H(Q0[39], Q0[38], Q0[37]) + Q0[36]
           + X0[10] + 0xbebfb7c0, 23) + Q0[39];
Q1[40] = RL(H(Q1[39], Q1[38], Q1[37]) + Q1[36]
           + X1[10] + 0xbebfb7c0, 23) + Q1[39];
if((Q0[40] ^ Q1[40]) != 0x80000000)
    continue;

/* A11 */
Q0[41] = RL(H(Q0[40], Q0[39], Q0[38]) + Q0[37]
           + X0[13] + 0x289b7ec6,  4) + Q0[40];
Q1[41] = RL(H(Q1[40], Q1[39], Q1[38]) + Q1[37]
           + X1[13] + 0x289b7ec6,  4) + Q1[40];

```

```

if((Q0[41] ^ Q1[41]) != 0x80000000)
    continue;

/* D11 */
Q0[42] = RL(H(Q0[41], Q0[40], Q0[39]) + Q0[38]
    + X0[ 0] + 0xeaa127fa, 11) + Q0[41];
Q1[42] = RL(H(Q1[41], Q1[40], Q1[39]) + Q1[38]
    + X1[ 0] + 0xeaa127fa, 11) + Q1[41];
if((Q0[42] ^ Q1[42]) != 0x80000000)
    continue;

/* C11 */
Q0[43] = RL(H(Q0[42], Q0[41], Q0[40]) + Q0[39]
    + X0[ 3] + 0xd4ef3085, 16) + Q0[42];
Q1[43] = RL(H(Q1[42], Q1[41], Q1[40]) + Q1[39]
    + X1[ 3] + 0xd4ef3085, 16) + Q1[42];
if((Q0[43] ^ Q1[43]) != 0x80000000)
    continue;

/* B11 */
Q0[44] = RL(H(Q0[43], Q0[42], Q0[41]) + Q0[40]
    + X0[ 6] + 0x04881d05, 23) + Q0[43];
Q1[44] = RL(H(Q1[43], Q1[42], Q1[41]) + Q1[40]
    + X1[ 6] + 0x04881d05, 23) + Q1[43];
if((Q0[44] ^ Q1[44]) != 0x80000000)
    continue;

/* A12 */
Q0[45] = RL(H(Q0[44], Q0[43], Q0[42]) + Q0[41]
    + X0[ 9] + 0xd9d4d039,  4) + Q0[44];
Q1[45] = RL(H(Q1[44], Q1[43], Q1[42]) + Q1[41]
    + X1[ 9] + 0xd9d4d039,  4) + Q1[44];
if((Q0[45] ^ Q1[45]) != 0x80000000)
    continue;

/* D12 */
Q0[46] = RL(H(Q0[45], Q0[44], Q0[43]) + Q0[42]
    + X0[12] + 0xe6db99e5, 11) + Q0[45];
Q1[46] = RL(H(Q1[45], Q1[44], Q1[43]) + Q1[42]
    + X1[12] + 0xe6db99e5, 11) + Q1[45];
if((Q0[46] ^ Q1[46]) != 0x80000000)
    continue;

/* C12 */
Q0[47] = RL(H(Q0[46], Q0[45], Q0[44]) + Q0[43]
    + X0[15] + 0x1fa27cf8, 16) + Q0[46];
Q1[47] = RL(H(Q1[46], Q1[45], Q1[44]) + Q1[43]
    + X1[15] + 0x1fa27cf8, 16) + Q1[46];
if((Q0[47] ^ Q1[47]) != 0x80000000)
    continue;

/* B12 */
Q0[48] = RL(H(Q0[47], Q0[46], Q0[45]) + Q0[44]
    + X0[ 2] + 0xc4ac5665, 23) + Q0[47];
if((Q0[48] ^ Q0[46]) & 0x80000000)
    continue;
Q1[48] = RL(H(Q1[47], Q1[46], Q1[45]) + Q1[44]
    + X1[ 2] + 0xc4ac5665, 23) + Q1[47];
if((Q0[48] ^ Q1[48]) != 0x80000000)
    continue;

```

```

/* A13 */
Q0[49] = RL(I(Q0[48], Q0[47], Q0[46]) + Q0[45]
  + X0[ 0] + 0xf4292244, 6) + Q0[48];
if((Q0[49] ^ Q0[47]) & 0x80000000)
  continue;
Q1[49] = RL(I(Q1[48], Q1[47], Q1[46]) + Q1[45]
  + X1[ 0] + 0xf4292244, 6) + Q1[48];
if((Q0[49] ^ Q1[49]) != 0x80000000)
  continue;

/* D13 */
Q0[50] = RL(I(Q0[49], Q0[48], Q0[47]) + Q0[46]
  + X0[ 7] + 0x432aff97, 10) + Q0[49];
if(!((Q0[50] ^ Q0[48]) & 0x80000000))
  continue;
Q1[50] = RL(I(Q1[49], Q1[48], Q1[47]) + Q1[46]
  + X1[ 7] + 0x432aff97, 10) + Q1[49];
if((Q0[50] ^ Q1[50]) != 0x80000000)
  continue;

/* C13 */
Q0[51] = RL(I(Q0[50], Q0[49], Q0[48]) + Q0[47]
  + X0[14] + 0xab9423a7, 15) + Q0[50];
if((Q0[51] ^ Q0[49]) & 0x80000000)
  continue;
Q1[51] = RL(I(Q1[50], Q1[49], Q1[48]) + Q1[47]
  + X1[14] + 0xab9423a7, 15) + Q1[50];
if((Q0[51] ^ Q1[51]) != 0x80000000)
  continue;

/* B13 */
Q0[52] = RL(I(Q0[51], Q0[50], Q0[49]) + Q0[48]
  + X0[ 5] + 0xfc93a039, 21) + Q0[51];
if((Q0[52] ^ Q0[50]) & 0x80000000)
  continue;
Q1[52] = RL(I(Q1[51], Q1[50], Q1[49]) + Q1[48]
  + X1[ 5] + 0xfc93a039, 21) + Q1[51];
if((Q0[52] ^ Q1[52]) != 0x80000000)
  continue;

/* A14 */
Q0[53] = RL(I(Q0[52], Q0[51], Q0[50]) + Q0[49]
  + X0[12] + 0x655b59c3, 6) + Q0[52];
if((Q0[53] ^ Q0[51]) & 0x80000000)
  continue;
Q1[53] = RL(I(Q1[52], Q1[51], Q1[50]) + Q1[49]
  + X1[12] + 0x655b59c3, 6) + Q1[52];
if((Q0[53] ^ Q1[53]) != 0x80000000)
  continue;

/* D14 */
Q0[54] = RL(I(Q0[53], Q0[52], Q0[51]) + Q0[50]
  + X0[ 3] + 0x8f0ccc92, 10) + Q0[53];
if((Q0[54] ^ Q0[52]) & 0x80000000)
  continue;
Q1[54] = RL(I(Q1[53], Q1[52], Q1[51]) + Q1[50]
  + X1[ 3] + 0x8f0ccc92, 10) + Q1[53];
if((Q0[54] ^ Q1[54]) != 0x80000000)
  continue;

```

```

/* C14 */
Q0[55] = RL(I(Q0[54], Q0[53], Q0[52]) + Q0[51]
    + X0[10] + 0xffeff47d, 15) + Q0[54];
if((Q0[55] ^ Q0[53]) & 0x80000000)
    continue;
Q1[55] = RL(I(Q1[54], Q1[53], Q1[52]) + Q1[51]
    + X1[10] + 0xffeff47d, 15) + Q1[54];
if((Q0[55] ^ Q1[55]) != 0x80000000)
    continue;

/* B14 */
Q0[56] = RL(I(Q0[55], Q0[54], Q0[53]) + Q0[52]
    + X0[ 1] + 0x85845dd1, 21) + Q0[55];
if((Q0[56] ^ Q0[54]) & 0x80000000)
    continue;
Q1[56] = RL(I(Q1[55], Q1[54], Q1[53]) + Q1[52]
    + X1[ 1] + 0x85845dd1, 21) + Q1[55];
if((Q0[56] ^ Q1[56]) != 0x80000000)
    continue;

/* A15 */
Q0[57] = RL(I(Q0[56], Q0[55], Q0[54]) + Q0[53]
    + X0[ 8] + 0x6fa87e4f,  6) + Q0[56];
if((Q0[57] ^ Q0[55]) & 0x80000000)
    continue;
Q1[57] = RL(I(Q1[56], Q1[55], Q1[54]) + Q1[53]
    + X1[ 8] + 0x6fa87e4f,  6) + Q1[56];
if((Q0[57] ^ Q1[57]) != 0x80000000)
    continue;

/* D15 */
Q0[58] = RL(I(Q0[57], Q0[56], Q0[55]) + Q0[54]
    + X0[15] + 0xfe2ce6e0, 10) + Q0[57];
if((Q0[58] ^ Q0[56]) & 0x80000000)
    continue;
Q1[58] = RL(I(Q1[57], Q1[56], Q1[55]) + Q1[54]
    + X1[15] + 0xfe2ce6e0, 10) + Q1[57];
if((Q0[58] ^ Q1[58]) != 0x80000000)
    continue;

/* C15 */
Q0[59] = RL(I(Q0[58], Q0[57], Q0[56]) + Q0[55]
    + X0[ 6] + 0xa3014314, 15) + Q0[58];
if((Q0[59] ^ Q0[57]) & 0x80000000)
    continue;
Q1[59] = RL(I(Q1[58], Q1[57], Q1[56]) + Q1[55]
    + X1[ 6] + 0xa3014314, 15) + Q1[58];
if((Q0[59] ^ Q1[59]) != 0x80000000)
    continue;

/* B15 */
Q0[60] = RL(I(Q0[59], Q0[58], Q0[57]) + Q0[56]
    + X0[13] + 0x4e0811a1, 21) + Q0[59];
if(Q0[60] & 0x02000000)
    continue;
Q1[60] = RL(I(Q1[59], Q1[58], Q1[57]) + Q1[56]
    + X1[13] + 0x4e0811a1, 21) + Q1[59];
if((Q0[60] ^ Q1[60]) != 0x80000000)
    continue;

```

```

/* A16 */
Q0[61] = RL(I(Q0[60], Q0[59], Q0[58]) + Q0[57]
            + X0[ 4] + 0xf7537e82, 6) + Q0[60];
A0 = IV[0] + Q0[61];
Q1[61] = RL(I(Q1[60], Q1[59], Q1[58]) + Q1[57]
            + X1[ 4] + 0xf7537e82, 6) + Q1[60];
A1 = IV[0] + Q1[61];
if((A0 ^ A1) != 0x80000000)
    continue;

/* D16 */
Q0[62] = RL(I(Q0[61], Q0[60], Q0[59]) + Q0[58]
            + X0[11] + 0xbd3af235, 10) + Q0[61];
D0 = IV[3] + Q0[62];
if(D0 & 0x02000000)
    continue;
Q1[62] = RL(I(Q1[61], Q1[60], Q1[59]) + Q1[58]
            + X1[11] + 0xbd3af235, 10) + Q1[61];
D1 = IV[3] + Q1[62];
if((D0 - D1) != 0x7e000000)
    continue;

/* C16 */
Q0[63] = RL(I(Q0[62], Q0[61], Q0[60]) + Q0[59]
            + X0[ 2] + 0x2ad7d2bb, 15) + Q0[62];
C0 = IV[2] + Q0[63];
if((C0 & 0x86000000) != ((D0 & 0x80000000) | 0x02000000))
    continue;
Q1[63] = RL(I(Q1[62], Q1[61], Q1[60]) + Q1[59]
            + X1[ 2] + 0x2ad7d2bb, 15) + Q1[62];
C1 = IV[2] + Q1[63];
if((C0 - C1) != 0x7e000000)
    continue;

/* B16 */
Q0[64] = RL(I(Q0[63], Q0[62], Q0[61]) + Q0[60]
            + X0[ 9] + 0xeb86d391, 21) + Q0[63];
B0 = IV[1] + Q0[64];
if((B0 & 0x86000020) != (C0 & 0x80000000))
    continue;
Q1[64] = RL(I(Q1[63], Q1[62], Q1[61]) + Q1[60]
            + X1[ 9] + 0xeb86d391, 21) + Q1[63];
B1 = IV[1] + Q1[64];
if((B0 - B1) != 0x7e000000)
    continue;

break;
}
if(i >= LOOP_12)
    goto block1_again;
return;
}

const unsigned int mask22[30] = {
    0x00000001, 0x00000002, 0x00000004, 0x00000008,
    0x00000010, 0x00000020, 0x00000040, 0x00000080,
    0x00000100, 0x00000200, 0x00000400, 0x00000800,
    0x00001000, 0x00002000, 0x00004000, 0x00008000,
    0x00010000, 0x00020000, 0x00040000, 0x00080000,

```

```

    0x00100000, 0x00200000, 0x00400000, 0x00800000,
    0x01000000, 0x02000000, 0x04000000, 0x08000000,
    0x10000000, 0x40000000
};

void block2(void)
{
    size_t i;

block2_again:
    for(;;)
    {
        /* A1 */
        Q0[ 1] = (random() | 0x84200000) & ~0x0a000820;
        Q1[ 1] = Q0[ 1] - 0x7e000000;

        X0[16] = RR(Q0[ 1] - B0, 7) - F(B0, C0, D0)
                - A0 - 0xd76aa478;
        X1[16] = RR(Q1[ 1] - B1, 7) - F(B1, C1, D1)
                - A1 - 0xd76aa478;
        if(X0[16] != X1[16])
            continue;
        break;
    }

    for(i = 0; i < 10; i++)
    {
        /* D1 */
        Q0[ 2] = (random() | 0x8c000800) & ~(0x02208026 |
0x701f10c0);
        Q0[ 2] |= (Q0[ 1] & 0x701f10c0);
        Q1[ 2] = Q0[ 2] - 0x7dffffe0;

        X0[17] = RR(Q0[ 2] - Q0[ 1], 12) - F(Q0[ 1], B0, C0)
                - D0 - 0xe8c7b756;
        X1[17] = RR(Q1[ 2] - Q1[ 1], 12) - F(Q1[ 1], B1, C1)
                - D1 - 0xe8c7b756;
        if(X0[17] != X1[17])
            continue;
        break;
    }
    if(i >= 10)
        goto block2_again;

    for(i = 0; i < 10; i++)
    {
        /* C1 */
        Q0[ 3] = (random() | 0xbelf0966) & ~(0x40201080 | 0x00000018);
        Q0[ 3] |= (Q0[ 2] & 0x00000018);
        Q1[ 3] = Q0[ 3] - 0x7dfef7e0;

        X0[18] = RR(Q0[ 3] - Q0[ 2], 17) - F(Q0[ 2], Q0[ 1], B0)
                - C0 - 0x242070db;
        X1[18] = RR(Q1[ 3] - Q1[ 2], 17) - F(Q1[ 2], Q1[ 1], B1)
                - C1 - 0x242070db;
        if(X0[18] != X1[18])
            continue;
        break;
    }
    if(i >= 10)

```

```

        goto block2_again;

for(i = 0; i < 10; i++)
{
    /* B1 */
    Q0[ 4] = (random() | 0xba040010) & ~(0x443b19ee | 0x00000601);
    Q0[ 4] |= (Q0[ 3] & 0x00000601);
    Q1[ 4] = Q0[ 4] - 0x7dffffe2;

    X0[19] = RR(Q0[ 4] - Q0[ 3], 22) - F(Q0[ 3], Q0[ 2], Q0[
1])
        - B0 - 0xc1bdceee;
    X1[19] = RR(Q1[ 4] - Q1[ 3], 22) - F(Q1[ 3], Q1[ 2], Q1[
1])
        - B1 - 0xc1bdceee;
    if(X0[19] != X1[19])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* A2 */
    Q0[ 5] = (random() | 0x482f0e50) & ~0xb41011af;
    Q1[ 5] = Q0[ 5] - 0x7ffffcbf;

    X0[20] = RR(Q0[ 5] - Q0[ 4], 7) - F(Q0[ 4], Q0[ 3], Q0[
2])
        - Q0[ 1] - 0xf57c0faf;
    X1[20] = RR(Q1[ 5] - Q1[ 4], 7) - F(Q1[ 4], Q1[ 3], Q1[
2])
        - Q1[ 1] - 0xf57c0faf;
    if((X0[20] ^ X1[20]) != 0x80000000)
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* D2 */
    Q0[ 6] = (random() | 0x04220c56) & ~0x9a1113a9;
    Q1[ 6] = Q0[ 6] - 0x80110000;

    X0[21] = RR(Q0[ 6] - Q0[ 5], 12) - F(Q0[ 5], Q0[ 4], Q0[
3])
        - Q0[ 2] - 0x4787c62a;
    X1[21] = RR(Q1[ 6] - Q1[ 5], 12) - F(Q1[ 5], Q1[ 4], Q1[
3])
        - Q1[ 2] - 0x4787c62a;
    if(X0[21] != X1[21])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

```

```

for(i = 0; i < 10; i++)
{
    /* C2 */
    Q0[ 7] = (random() | 0x96011e01) & ~(0x083201c0 | 0x01808000);
    Q0[ 7] |= (Q0[ 6] & 0x01808000);
    Q1[ 7] = Q0[ 7] - 0x88000040;

    X0[22] = RR(Q0[ 7] - Q0[ 6], 17) - F(Q0[ 6], Q0[ 5], Q0[ 4])
            - Q0[ 3] - 0xa8304613;
    X1[22] = RR(Q1[ 7] - Q1[ 6], 17) - F(Q1[ 6], Q1[ 5], Q1[ 4])
            - Q1[ 3] - 0xa8304613;
    if(X0[22] != X1[22])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* B2 */
    Q0[ 8] = (random() | 0x843283c0) & ~(0x1b810001 | 0x00000002);
    Q0[ 8] |= (Q0[ 7] & 0x00000002);
    Q1[ 8] = Q0[ 8] - 0x80818000;

    X0[23] = RR(Q0[ 8] - Q0[ 7], 22) - F(Q0[ 7], Q0[ 6], Q0[ 5])
            - Q0[ 4] - 0xfd469501;
    X1[23] = RR(Q1[ 8] - Q1[ 7], 22) - F(Q1[ 7], Q1[ 6], Q1[ 5])
            - Q1[ 4] - 0xfd469501;
    if(X0[23] != X1[23])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* A3 */
    Q0[ 9] = (random() | 0x9c0101c1) & ~(0x03828202 | 0x00001000);
    Q0[ 9] |= (Q0[ 8] & 0x00001000);
    Q1[ 9] = Q0[ 9] - 0x7fffffbf;

    X0[24] = RR(Q0[ 9] - Q0[ 8], 7) - F(Q0[ 8], Q0[ 7], Q0[ 6])
            - Q0[ 5] - 0x698098d8;
    X1[24] = RR(Q1[ 9] - Q1[ 8], 7) - F(Q1[ 8], Q1[ 7], Q1[ 6])
            - Q1[ 5] - 0x698098d8;
    if(X0[24] != X1[24])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* D3 */
    Q0[10] = (random() | 0x878383c0) & ~0x00041003;
    Q1[10] = Q0[10] - 0x7ffff000;

```

```

X0[25] = RR(Q0[10] - Q0[ 9], 12) - F(Q0[ 9], Q0[ 8], Q0[ 7])
      - Q0[ 6] - 0x8b44f7af;
X1[25] = RR(Q1[10] - Q1[ 9], 12) - F(Q1[ 9], Q1[ 8], Q1[ 7])
      - Q1[ 6] - 0x8b44f7af;
if(X0[25] != X1[25])
    continue;
break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* C3 */
    Q0[11] = (random() | 0x800583c3) & ~(0x00021000 | 0x00086000);
    Q0[11] |= (Q0[10] & 0x00086000);
    Q1[11] = Q0[11] - 0x80000000;

    X0[26] = RR(Q0[11] - Q0[10], 17) - F(Q0[10], Q0[ 9], Q0[ 8])
      - Q0[ 7] - 0xffff5bb1;
    X1[26] = RR(Q1[11] - Q1[10], 17) - F(Q1[10], Q1[ 9], Q1[ 8])
      - Q1[ 7] - 0xffff5bb1;
    if(X0[26] != X1[26])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* B3 */
    Q0[12] = (random() | 0x80081080) & ~(0x0007e000 | 0x7f000000);
    Q0[12] |= (Q0[11] & 0x7f000000);
    Q1[12] = Q0[12] - 0x80002080;

    X0[27] = RR(Q0[12] - Q0[11], 22) - F(Q0[11], Q0[10], Q0[ 9])
      - Q0[ 8] - 0x895cd7be;
    X1[27] = RR(Q1[12] - Q1[11], 22) - F(Q1[11], Q1[10], Q1[ 9])
      - Q1[ 8] - 0x895cd7be;
    if((X0[27] ^ X1[27]) != 0x00008000)
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* A4 */
    Q0[13] = (random() | 0x3f0fe008) & ~0x80000080;
    Q1[13] = Q0[13] - 0x7f000000;

    X0[28] = RR(Q0[13] - Q0[12], 7) - F(Q0[12], Q0[11], Q0[10])
      - Q0[ 9] - 0x6b901122;
    X1[28] = RR(Q1[13] - Q1[12], 7) - F(Q1[12], Q1[11], Q1[10])
      - Q1[ 9] - 0x6b901122;
    if(X0[28] != X1[28])
        continue;
    break;
}

```

```

}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* D4 */
    Q0[14] = (random() | 0x400be088) & ~0xbf040000;
    Q1[14] = Q0[14] - 0x80000000;

    X0[29] = RR(Q0[14] - Q0[13], 12) - F(Q0[13], Q0[12], Q0[11])
        - Q0[10] - 0xfd987193;
    X1[29] = RR(Q1[14] - Q1[13], 12) - F(Q1[13], Q1[12], Q1[11])
        - Q1[10] - 0xfd987193;
    if(X0[29] != X1[29])
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

for(i = 0; i < 10; i++)
{
    /* C4 */
    Q0[15] = (random() | 0x7d000000) & ~0x82008008;
    Q1[15] = Q0[15] - 0x7fff7ff8;

    X0[30] = RR(Q0[15] - Q0[14], 17) - F(Q0[14], Q0[13], Q0[12])
        - Q0[11] - 0xa679438e;
    X1[30] = RR(Q1[15] - Q1[14], 17) - F(Q1[14], Q1[13], Q1[12])
        - Q1[11] - 0xa679438e;
    if((X0[30] ^ X1[30]) != 0x80000000)
        continue;
    break;
}
if(i >= 10)
    goto block2_again;

#define LOOP_21 1000

for(i = 0; i < LOOP_21; i++)
{
    /* B4 */
    Q0[16] = (random() | 0x20000000) & ~0x80000000;
    Q1[16] = Q0[16] - 0xa0000000;

    X0[31] = RR(Q0[16] - Q0[15], 22) - F(Q0[15], Q0[14], Q0[13])
        - Q0[12] - 0x49b40821;
    X1[31] = RR(Q1[16] - Q1[15], 22) - F(Q1[15], Q1[14], Q1[13])
        - Q1[12] - 0x49b40821;
    if(X0[31] != X1[31])
        continue;

    /* A5 */
    Q0[17] = RL(G(Q0[16], Q0[15], Q0[14]) + Q0[13]
        + X0[17] + 0xf61e2562, 5) + Q0[16];
    if((Q0[17] & 0x80028008) != (Q0[16] & 0x00008008))
        continue;
    Q1[17] = RL(G(Q1[16], Q1[15], Q1[14]) + Q1[13]
        + X1[17] + 0xf61e2562, 5) + Q1[16];

```

```

        if((Q0[17] ^ Q1[17]) != 0x80000000)
            continue;

        /* D5 */
        Q0[18] = RL(G(Q0[17], Q0[16], Q0[15]) + Q0[14]
            + X0[22] + 0xc040b340, 9) + Q0[17];
        if((Q0[18] & 0xa0020000)
            != ((Q0[17] & 0x20000000) | 0x00020000))
        {
            continue;
        }
        Q1[18] = RL(G(Q1[17], Q1[16], Q1[15]) + Q1[14]
            + X1[22] + 0xc040b340, 9) + Q1[17];
        if((Q0[18] ^ Q1[18]) != 0x80000000)
            continue;

        /* C5 */
        Q0[19] = RL(G(Q0[18], Q0[17], Q0[16]) + Q0[15]
            + X0[27] + 0x265e5a51, 14) + Q0[18];
        if(Q0[19] & 0x80020000)
            continue;
        Q1[19] = RL(G(Q1[18], Q1[17], Q1[16]) + Q1[15]
            + X1[27] + 0x265e5a51, 14) + Q1[18];
        if((Q0[19] - Q1[19]) != 0x7ffe0000)
            continue;

        /* B5 */
        Q0[20] = RL(G(Q0[19], Q0[18], Q0[17]) + Q0[16]
            + X0[16] + 0xe9b6c7aa, 20) + Q0[19];
        if(Q0[20] & 0x80000000)
            continue;
        Q1[20] = RL(G(Q1[19], Q1[18], Q1[17]) + Q1[16]
            + X1[16] + 0xe9b6c7aa, 20) + Q1[19];
        if((Q0[20] ^ Q1[20]) != 0x80000000)
            continue;

        /* A6 */
        Q0[21] = RL(G(Q0[20], Q0[19], Q0[18]) + Q0[17]
            + X0[21] + 0xd62f105d, 5) + Q0[20];
        if((Q0[21] & 0x80020000) != (Q0[20] & 0x00020000))
            continue;
        Q1[21] = RL(G(Q1[20], Q1[19], Q1[18]) + Q1[17]
            + X1[21] + 0xd62f105d, 5) + Q1[20];
        if((Q0[21] ^ Q1[21]) != 0x80000000)
            continue;
        break;
    }
    if(i >= LOOP_21)
        goto block2_again;

#define LOOP_22 0x4000000

    for(i = 0; i < LOOP_22; i++)
    {
        /* B4 */
        Q0[16] ^= mask22[random() % 30];
        Q1[16] = Q0[16] - 0xa0000000;

        X0[31] = RR(Q0[16] - Q0[15], 22) - F(Q0[15], Q0[14],
Q0[13])

```

```

        - Q0[12] - 0x49b40821;
X1[31] = RR(Q1[16] - Q1[15], 22) - F(Q1[15], Q1[14],
Q1[13])
        - Q1[12] - 0x49b40821;
if(X0[31] != X1[31])
    continue;

/* A5 */
Q0[17] = RL(G(Q0[16], Q0[15], Q0[14]) + Q0[13]
    + X0[17] + 0xf61e2562, 5) + Q0[16];
if((Q0[17] & 0x80028008) != (Q0[16] & 0x00008008))
    continue;
Q1[17] = RL(G(Q1[16], Q1[15], Q1[14]) + Q1[13]
    + X1[17] + 0xf61e2562, 5) + Q1[16];
if((Q0[17] ^ Q1[17]) != 0x80000000)
    continue;

/* D5 */
Q0[18] = RL(G(Q0[17], Q0[16], Q0[15]) + Q0[14]
    + X0[22] + 0xc040b340, 9) + Q0[17];
if((Q0[18] & 0xa0020000)
    != ((Q0[17] & 0x20000000) | 0x00020000))
{
    continue;
}
Q1[18] = RL(G(Q1[17], Q1[16], Q1[15]) + Q1[14]
    + X1[22] + 0xc040b340, 9) + Q1[17];
if((Q0[18] ^ Q1[18]) != 0x80000000)
    continue;

/* C5 */
Q0[19] = RL(G(Q0[18], Q0[17], Q0[16]) + Q0[15]
    + X0[27] + 0x265e5a51, 14) + Q0[18];
if(Q0[19] & 0x80020000)
    continue;
Q1[19] = RL(G(Q1[18], Q1[17], Q1[16]) + Q1[15]
    + X1[27] + 0x265e5a51, 14) + Q1[18];
if((Q0[19] - Q1[19]) != 0x7ffe0000)
    continue;

/* B5 */
Q0[20] = RL(G(Q0[19], Q0[18], Q0[17]) + Q0[16]
    + X0[16] + 0xe9b6c7aa, 20) + Q0[19];
if(Q0[20] & 0x80000000)
    continue;
Q1[20] = RL(G(Q1[19], Q1[18], Q1[17]) + Q1[16]
    + X1[16] + 0xe9b6c7aa, 20) + Q1[19];
if((Q0[20] ^ Q1[20]) != 0x80000000)
    continue;

/* A6 */
Q0[21] = RL(G(Q0[20], Q0[19], Q0[18]) + Q0[17]
    + X0[21] + 0xd62f105d, 5) + Q0[20];
if((Q0[21] & 0x80020000) != (Q0[20] & 0x00020000))
    continue;
Q1[21] = RL(G(Q1[20], Q1[19], Q1[18]) + Q1[17]
    + X1[21] + 0xd62f105d, 5) + Q1[20];
if((Q0[21] ^ Q1[21]) != 0x80000000)
    continue;

```

```

/* D6 */
Q0[22] = RL(G(Q0[21], Q0[20], Q0[19]) + Q0[18]
    + X0[26] + 0x02441453, 9) + Q0[21];
if(Q0[22] & 0x80000000)
    continue;
Q1[22] = RL(G(Q1[21], Q1[20], Q1[19]) + Q1[18]
    + X1[26] + 0x02441453, 9) + Q1[21];
if((Q0[22] ^ Q1[22]) != 0x80000000)
    continue;

/* C6 */
Q0[23] = RL(G(Q0[22], Q0[21], Q0[20]) + Q0[19]
    + X0[31] + 0xd8a1e681, 14) + Q0[22];
if(Q0[23] & 0x80000000)
    continue;
Q1[23] = RL(G(Q1[22], Q1[21], Q1[20]) + Q1[19]
    + X1[31] + 0xd8a1e681, 14) + Q1[22];
if(Q0[23] != Q1[23])
    continue;

/* B6 */
Q0[24] = RL(G(Q0[23], Q0[22], Q0[21]) + Q0[20]
    + X0[20] + 0xe7d3fbc8, 20) + Q0[23];
Q1[24] = RL(G(Q1[23], Q1[22], Q1[21]) + Q1[20]
    + X1[20] + 0xe7d3fbc8, 20) + Q1[23];
if(Q0[24] != Q1[24])
    continue;

/* A7 */
Q0[25] = RL(G(Q0[24], Q0[23], Q0[22]) + Q0[21]
    + X0[25] + 0x21e1cde6, 5) + Q0[24];
Q1[25] = RL(G(Q1[24], Q1[23], Q1[22]) + Q1[21]
    + X1[25] + 0x21e1cde6, 5) + Q1[24];
if(Q0[25] != Q1[25])
    continue;

/* D7 */
Q0[26] = RL(G(Q0[25], Q0[24], Q0[23]) + Q0[22]
    + X0[30] + 0xc33707d6, 9) + Q0[25];
Q1[26] = RL(G(Q1[25], Q1[24], Q1[23]) + Q1[22]
    + X1[30] + 0xc33707d6, 9) + Q1[25];
if(Q0[26] != Q1[26])
    continue;

/* C7 */
Q0[27] = RL(G(Q0[26], Q0[25], Q0[24]) + Q0[23]
    + X0[19] + 0xf4d50d87, 14) + Q0[26];
Q1[27] = RL(G(Q1[26], Q1[25], Q1[24]) + Q1[23]
    + X1[19] + 0xf4d50d87, 14) + Q1[26];
if(Q0[27] != Q1[27])
    continue;

/* B7 */
Q0[28] = RL(G(Q0[27], Q0[26], Q0[25]) + Q0[24]
    + X0[24] + 0x455a14ed, 20) + Q0[27];
Q1[28] = RL(G(Q1[27], Q1[26], Q1[25]) + Q1[24]
    + X1[24] + 0x455a14ed, 20) + Q1[27];
if(Q0[28] != Q1[28])
    continue;

```

```

/* A8 */
Q0[29] = RL(G(Q0[28], Q0[27], Q0[26]) + Q0[25]
    + X0[29] + 0xa9e3e905, 5) + Q0[28];
Q1[29] = RL(G(Q1[28], Q1[27], Q1[26]) + Q1[25]
    + X1[29] + 0xa9e3e905, 5) + Q1[28];
if(Q0[29] != Q1[29])
    continue;

/* D8 */
Q0[30] = RL(G(Q0[29], Q0[28], Q0[27]) + Q0[26]
    + X0[18] + 0xfcefa3f8, 9) + Q0[29];
Q1[30] = RL(G(Q1[29], Q1[28], Q1[27]) + Q1[26]
    + X1[18] + 0xfcefa3f8, 9) + Q1[29];
if(Q0[30] != Q1[30])
    continue;

/* C8 */
Q0[31] = RL(G(Q0[30], Q0[29], Q0[28]) + Q0[27]
    + X0[23] + 0x676f02d9, 14) + Q0[30];
Q1[31] = RL(G(Q1[30], Q1[29], Q1[28]) + Q1[27]
    + X1[23] + 0x676f02d9, 14) + Q1[30];
if(Q0[31] != Q1[31])
    continue;

/* B8 */
Q0[32] = RL(G(Q0[31], Q0[30], Q0[29]) + Q0[28]
    + X0[28] + 0x8d2a4c8a, 20) + Q0[31];
Q1[32] = RL(G(Q1[31], Q1[30], Q1[29]) + Q1[28]
    + X1[28] + 0x8d2a4c8a, 20) + Q1[31];
if(Q0[32] != Q1[32])
    continue;

/* A9 */
Q0[33] = RL(H(Q0[32], Q0[31], Q0[30]) + Q0[29]
    + X0[21] + 0xfffa3942, 4) + Q0[32];
Q1[33] = RL(H(Q1[32], Q1[31], Q1[30]) + Q1[29]
    + X1[21] + 0xfffa3942, 4) + Q1[32];
if(Q0[33] != Q1[33])
    continue;

/* D9 */
Q0[34] = RL(H(Q0[33], Q0[32], Q0[31]) + Q0[30]
    + X0[24] + 0x8771f681, 11) + Q0[33];
Q1[34] = RL(H(Q1[33], Q1[32], Q1[31]) + Q1[30]
    + X1[24] + 0x8771f681, 11) + Q1[33];
if(Q0[34] != Q1[34])
    continue;

/* C9 */
Q0[35] = RL(H(Q0[34], Q0[33], Q0[32]) + Q0[31]
    + X0[27] + 0x6d9d6122, 16) + Q0[34];
Q1[35] = RL(H(Q1[34], Q1[33], Q1[32]) + Q1[31]
    + X1[27] + 0x6d9d6122, 16) + Q1[34];
if((Q0[35] ^ Q1[35]) != 0x80000000)
    continue;

/* B9 */
Q0[36] = RL(H(Q0[35], Q0[34], Q0[33]) + Q0[32]
    + X0[30] + 0xfde5380c, 23) + Q0[35];
Q1[36] = RL(H(Q1[35], Q1[34], Q1[33]) + Q1[32]

```

```

        + X1[30] + 0xfde5380c, 23) + Q1[35];
if((Q0[36] ^ Q1[36]) != 0x80000000)
    continue;

/* A10 */
Q0[37] = RL(H(Q0[36], Q0[35], Q0[34]) + Q0[33]
    + X0[17] + 0xa4beea44, 4) + Q0[36];
Q1[37] = RL(H(Q1[36], Q1[35], Q1[34]) + Q1[33]
    + X1[17] + 0xa4beea44, 4) + Q1[36];
if((Q0[37] ^ Q1[37]) != 0x80000000)
    continue;

/* D10 */
Q0[38] = RL(H(Q0[37], Q0[36], Q0[35]) + Q0[34]
    + X0[20] + 0x4bdecfa9, 11) + Q0[37];
Q1[38] = RL(H(Q1[37], Q1[36], Q1[35]) + Q1[34]
    + X1[20] + 0x4bdecfa9, 11) + Q1[37];
if((Q0[38] ^ Q1[38]) != 0x80000000)
    continue;

/* C10 */
Q0[39] = RL(H(Q0[38], Q0[37], Q0[36]) + Q0[35]
    + X0[23] + 0xf6bb4b60, 16) + Q0[38];
Q1[39] = RL(H(Q1[38], Q1[37], Q1[36]) + Q1[35]
    + X1[23] + 0xf6bb4b60, 16) + Q1[38];
if((Q0[39] ^ Q1[39]) != 0x80000000)
    continue;

/* B10 */
Q0[40] = RL(H(Q0[39], Q0[38], Q0[37]) + Q0[36]
    + X0[26] + 0xbefbfc70, 23) + Q0[39];
Q1[40] = RL(H(Q1[39], Q1[38], Q1[37]) + Q1[36]
    + X1[26] + 0xbefbfc70, 23) + Q1[39];
if((Q0[40] ^ Q1[40]) != 0x80000000)
    continue;

/* A11 */
Q0[41] = RL(H(Q0[40], Q0[39], Q0[38]) + Q0[37]
    + X0[29] + 0x289b7ec6, 4) + Q0[40];
Q1[41] = RL(H(Q1[40], Q1[39], Q1[38]) + Q1[37]
    + X1[29] + 0x289b7ec6, 4) + Q1[40];
if((Q0[41] ^ Q1[41]) != 0x80000000)
    continue;

/* D11 */
Q0[42] = RL(H(Q0[41], Q0[40], Q0[39]) + Q0[38]
    + X0[16] + 0xea127fa, 11) + Q0[41];
Q1[42] = RL(H(Q1[41], Q1[40], Q1[39]) + Q1[38]
    + X1[16] + 0xea127fa, 11) + Q1[41];
if((Q0[42] ^ Q1[42]) != 0x80000000)
    continue;

/* C11 */
Q0[43] = RL(H(Q0[42], Q0[41], Q0[40]) + Q0[39]
    + X0[19] + 0xd4ef3085, 16) + Q0[42];
Q1[43] = RL(H(Q1[42], Q1[41], Q1[40]) + Q1[39]
    + X1[19] + 0xd4ef3085, 16) + Q1[42];
if((Q0[43] ^ Q1[43]) != 0x80000000)
    continue;

```

```

/* B11 */
Q0[44] = RL(H(Q0[43], Q0[42], Q0[41]) + Q0[40]
    + X0[22] + 0x04881d05, 23) + Q0[43];
Q1[44] = RL(H(Q1[43], Q1[42], Q1[41]) + Q1[40]
    + X1[22] + 0x04881d05, 23) + Q1[43];
if((Q0[44] ^ Q1[44]) != 0x80000000)
    continue;

/* A12 */
Q0[45] = RL(H(Q0[44], Q0[43], Q0[42]) + Q0[41]
    + X0[25] + 0xd9d4d039, 4) + Q0[44];
Q1[45] = RL(H(Q1[44], Q1[43], Q1[42]) + Q1[41]
    + X1[25] + 0xd9d4d039, 4) + Q1[44];
if((Q0[45] ^ Q1[45]) != 0x80000000)
    continue;

/* D12 */
Q0[46] = RL(H(Q0[45], Q0[44], Q0[43]) + Q0[42]
    + X0[28] + 0xe6db99e5, 11) + Q0[45];
Q1[46] = RL(H(Q1[45], Q1[44], Q1[43]) + Q1[42]
    + X1[28] + 0xe6db99e5, 11) + Q1[45];
if((Q0[46] ^ Q1[46]) != 0x80000000)
    continue;

/* C12 */
Q0[47] = RL(H(Q0[46], Q0[45], Q0[44]) + Q0[43]
    + X0[31] + 0x1fa27cf8, 16) + Q0[46];
Q1[47] = RL(H(Q1[46], Q1[45], Q1[44]) + Q1[43]
    + X1[31] + 0x1fa27cf8, 16) + Q1[46];
if((Q0[47] ^ Q1[47]) != 0x80000000)
    continue;

/* B12 */
Q0[48] = RL(H(Q0[47], Q0[46], Q0[45]) + Q0[44]
    + X0[18] + 0xc4ac5665, 23) + Q0[47];
if((Q0[48] & 0x80000000) != (Q0[46] & 0x80000000))
    continue;
Q1[48] = RL(H(Q1[47], Q1[46], Q1[45]) + Q1[44]
    + X1[18] + 0xc4ac5665, 23) + Q1[47];
if((Q0[48] ^ Q1[48]) != 0x80000000)
    continue;

/* A13 */
Q0[49] = RL(I(Q0[48], Q0[47], Q0[46]) + Q0[45]
    + X0[16] + 0xf4292244, 6) + Q0[48];
if((Q0[49] & 0x80000000) != (Q0[47] & 0x80000000))
    continue;
Q1[49] = RL(I(Q1[48], Q1[47], Q1[46]) + Q1[45]
    + X1[16] + 0xf4292244, 6) + Q1[48];
if((Q0[49] ^ Q1[49]) != 0x80000000)
    continue;

/* D13 */
Q0[50] = RL(I(Q0[49], Q0[48], Q0[47]) + Q0[46]
    + X0[23] + 0x432aff97, 10) + Q0[49];
Q1[50] = RL(I(Q1[49], Q1[48], Q1[47]) + Q1[46]
    + X1[23] + 0x432aff97, 10) + Q1[49];
if((Q0[50] ^ Q1[50]) != 0x80000000)
    continue;

```

```

/* C13 */
Q0[51] = RL(I(Q0[50], Q0[49], Q0[48]) + Q0[47]
    + X0[30] + 0xab9423a7, 15) + Q0[50];
if((Q0[51] & 0x80000000) != (Q0[49] & 0x80000000))
    continue;
Q1[51] = RL(I(Q1[50], Q1[49], Q1[48]) + Q1[47]
    + X1[30] + 0xab9423a7, 15) + Q1[50];
if((Q0[51] ^ Q1[51]) != 0x80000000)
    continue;

/* B13 */
Q0[52] = RL(I(Q0[51], Q0[50], Q0[49]) + Q0[48]
    + X0[21] + 0xfc93a039, 21) + Q0[51];
if((Q0[52] & 0x80000000) != (Q0[50] & 0x80000000))
    continue;
Q1[52] = RL(I(Q1[51], Q1[50], Q1[49]) + Q1[48]
    + X1[21] + 0xfc93a039, 21) + Q1[51];
if((Q0[52] ^ Q1[52]) != 0x80000000)
    continue;

/* A14 */
Q0[53] = RL(I(Q0[52], Q0[51], Q0[50]) + Q0[49]
    + X0[28] + 0x655b59c3, 6) + Q0[52];
if((Q0[53] & 0x80000000) != (Q0[51] & 0x80000000))
    continue;
Q1[53] = RL(I(Q1[52], Q1[51], Q1[50]) + Q1[49]
    + X1[28] + 0x655b59c3, 6) + Q1[52];
if((Q0[53] ^ Q1[53]) != 0x80000000)
    continue;

/* D14 */
Q0[54] = RL(I(Q0[53], Q0[52], Q0[51]) + Q0[50]
    + X0[19] + 0x8f0ccc92, 10) + Q0[53];
if((Q0[54] & 0x80000000) != (Q0[52] & 0x80000000))
    continue;
Q1[54] = RL(I(Q1[53], Q1[52], Q1[51]) + Q1[50]
    + X1[19] + 0x8f0ccc92, 10) + Q1[53];
if((Q0[54] ^ Q1[54]) != 0x80000000)
    continue;

/* C14 */
Q0[55] = RL(I(Q0[54], Q0[53], Q0[52]) + Q0[51]
    + X0[26] + 0xffeff47d, 15) + Q0[54];
if((Q0[55] & 0x80000000) != (Q0[53] & 0x80000000))
    continue;
Q1[55] = RL(I(Q1[54], Q1[53], Q1[52]) + Q1[51]
    + X1[26] + 0xffeff47d, 15) + Q1[54];
if((Q0[55] ^ Q1[55]) != 0x80000000)
    continue;

/* B14 */
Q0[56] = RL(I(Q0[55], Q0[54], Q0[53]) + Q0[52]
    + X0[17] + 0x85845dd1, 21) + Q0[55];
if((Q0[56] & 0x80000000) != (Q0[54] & 0x80000000))
    continue;
Q1[56] = RL(I(Q1[55], Q1[54], Q1[53]) + Q1[52]
    + X1[17] + 0x85845dd1, 21) + Q1[55];
if((Q0[56] ^ Q1[56]) != 0x80000000)
    continue;

```

```

/* A15 */
Q0[57] = RL(I(Q0[56], Q0[55], Q0[54]) + Q0[53]
            + X0[24] + 0x6fa87e4f, 6) + Q0[56];
if((Q0[57] & 0x80000000) != (Q0[55] & 0x80000000))
    continue;
Q1[57] = RL(I(Q1[56], Q1[55], Q1[54]) + Q1[53]
            + X1[24] + 0x6fa87e4f, 6) + Q1[56];
if((Q0[57] ^ Q1[57]) != 0x80000000)
    continue;

/* D15 */
Q0[58] = RL(I(Q0[57], Q0[56], Q0[55]) + Q0[54]
            + X0[31] + 0xfe2ce6e0, 10) + Q0[57];
if((Q0[58] & 0x80000000) != (Q0[56] & 0x80000000))
    continue;
Q1[58] = RL(I(Q1[57], Q1[56], Q1[55]) + Q1[54]
            + X1[31] + 0xfe2ce6e0, 10) + Q1[57];
if((Q0[58] ^ Q1[58]) != 0x80000000)
    continue;

/* C15 */
Q0[59] = RL(I(Q0[58], Q0[57], Q0[56]) + Q0[55]
            + X0[22] + 0xa3014314, 15) + Q0[58];
if((Q0[59] & 0x80000000) != (Q0[57] & 0x80000000))
    continue;
Q1[59] = RL(I(Q1[58], Q1[57], Q1[56]) + Q1[55]
            + X1[22] + 0xa3014314, 15) + Q1[58];
if((Q0[59] ^ Q1[59]) != 0x80000000)
    continue;

/* B15 */
Q0[60] = RL(I(Q0[59], Q0[58], Q0[57]) + Q0[56]
            + X0[29] + 0x4e0811a1, 21) + Q0[59];
Q1[60] = RL(I(Q1[59], Q1[58], Q1[57]) + Q1[56]
            + X1[29] + 0x4e0811a1, 21) + Q1[59];
if((Q0[60] ^ Q1[60]) != 0x80000000)
    continue;

/* A16 */
Q0[61] = RL(I(Q0[60], Q0[59], Q0[58]) + Q0[57]
            + X0[20] + 0xf7537e82, 6) + Q0[60];
Q1[61] = RL(I(Q1[60], Q1[59], Q1[58]) + Q1[57]
            + X1[20] + 0xf7537e82, 6) + Q1[60];
if((Q0[61] ^ Q1[61]) != 0x80000000)
    continue;
if((A0 + Q0[61]) != (A1 + Q1[61]))
    continue;

/* D16 */
Q0[62] = RL(I(Q0[61], Q0[60], Q0[59]) + Q0[58]
            + X0[27] + 0xbd3af235, 10) + Q0[61];
Q1[62] = RL(I(Q1[61], Q1[60], Q1[59]) + Q1[58]
            + X1[27] + 0xbd3af235, 10) + Q1[61];
if((D0 + Q0[62]) != (D1 + Q1[62]))
    continue;

/* C16 */
Q0[63] = RL(I(Q0[62], Q0[61], Q0[60]) + Q0[59]
            + X0[18] + 0x2ad7d2bb, 15) + Q0[62];
Q1[63] = RL(I(Q1[62], Q1[61], Q1[60]) + Q1[59]

```

```

        + X1[18] + 0x2ad7d2bb, 15) + Q1[62];
    if((C0 + Q0[63]) != (C1 + Q1[63]))
        continue;

    /* B16 */
    Q0[64] = RL(I(Q0[63], Q0[62], Q0[61]) + Q0[60]
        + X0[25] + 0xeb86d391, 21) + Q0[63];
    Q1[64] = RL(I(Q1[63], Q1[62], Q1[61]) + Q1[60]
        + X1[25] + 0xeb86d391, 21) + Q1[63];
    if((B0 + Q0[64]) != (B1 + Q1[64]))
        continue;
    break;
}
if(i >= LOOP_22)
    goto block2_again;
return;
}

int main(int argc, char *argv[])
{
    size_t i;

    if(argc == 5)
    {
        IV[0] = strtoul(argv[1], NULL, 0);
        IV[1] = strtoul(argv[2], NULL, 0);
        IV[2] = strtoul(argv[3], NULL, 0);
        IV[3] = strtoul(argv[4], NULL, 0);
    }

    srand(time(NULL) ^ (getpid() << 16));
    block1();
    printf("block #1 done\n");
    block2();
    printf("block #2 done\n");

    printf("unsigned int m0[32] = {\n");
    for(i = 0; i < 32; i++)
    {
        printf("0x%08x, ", X0[ i]);
        if((i & 3) == 3)
            printf("\n");
    }
    printf("};\n\n");
    printf("unsigned int m1[32] = {\n");
    for(i = 0; i < 32; i++)
    {
        printf("0x%08x, ", X1[ i]);
        if((i & 3) == 3)
            printf("\n");
    }
    printf("};\n\n");
    return 0;
}

```