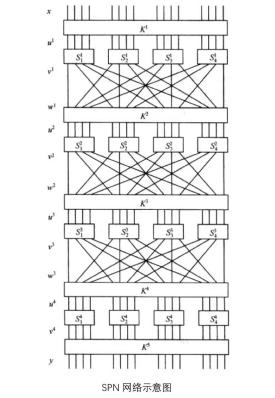
spn差分攻击实验报告

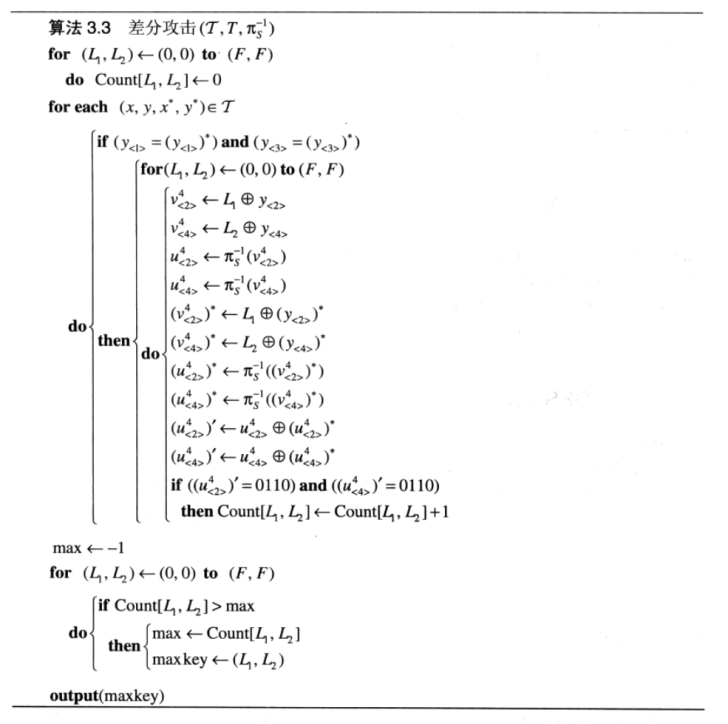
2112155 梁婧涵

1. 问题：



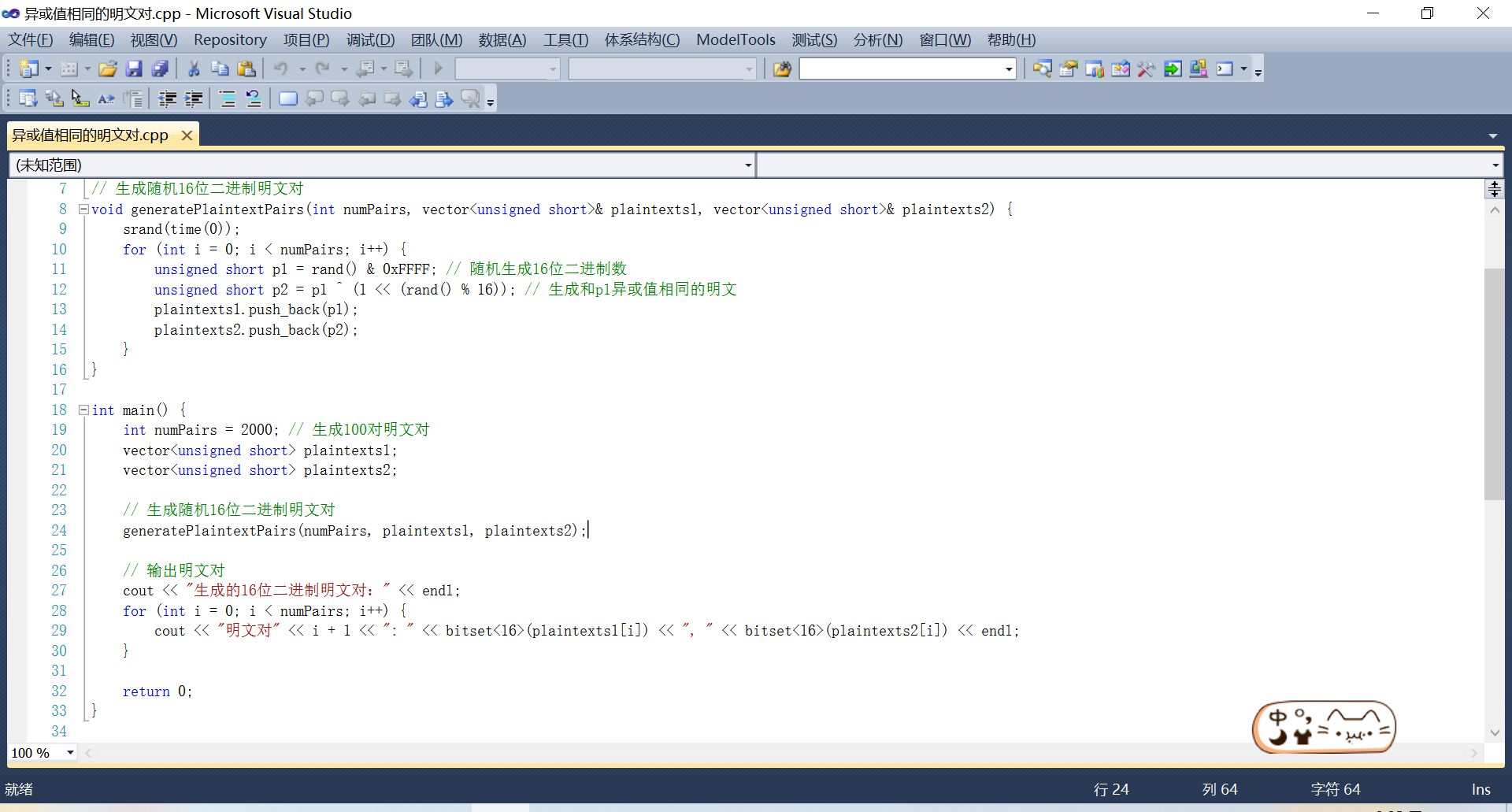
实现差分攻击算法，分析K^5轮密钥；

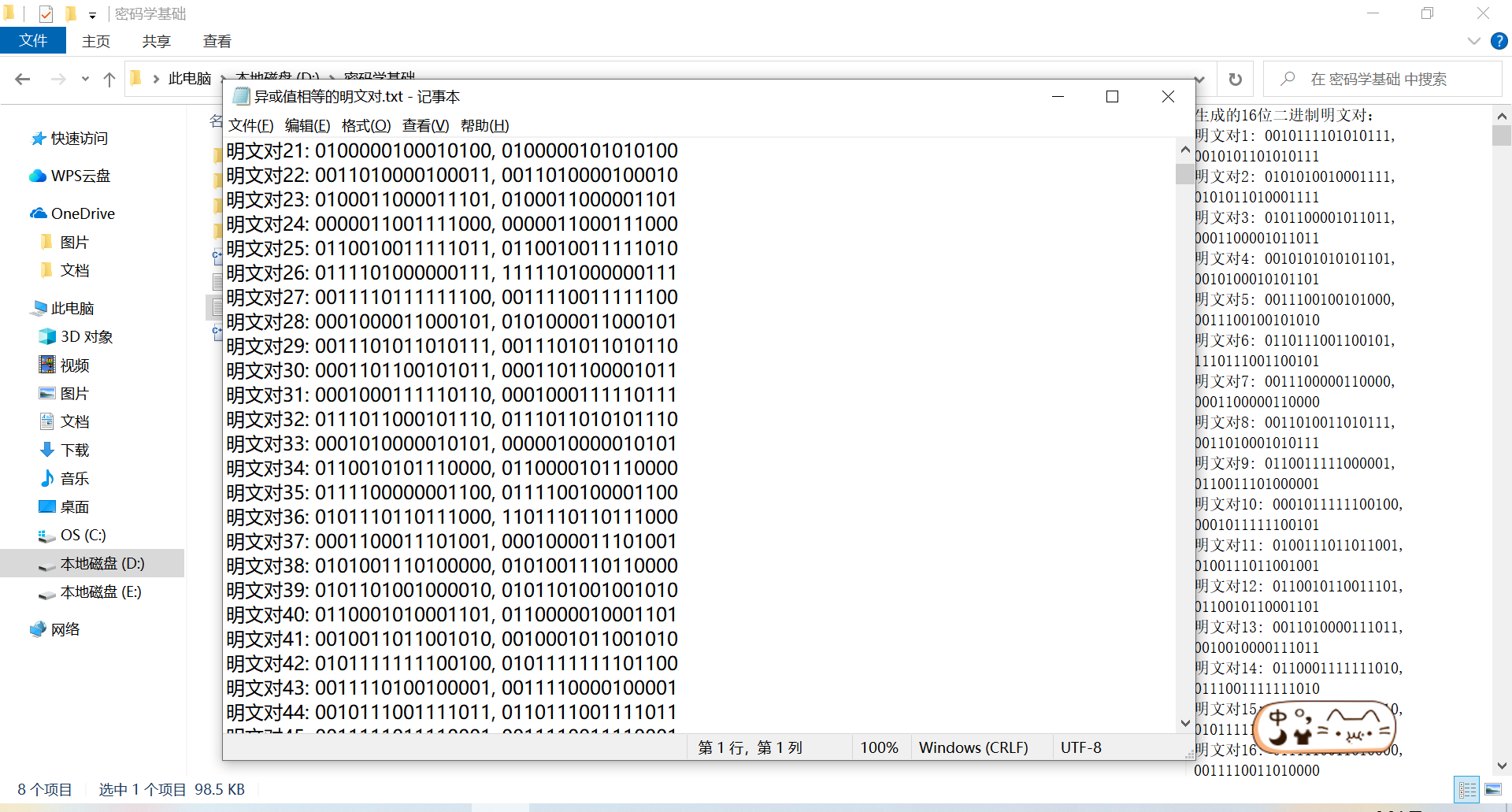
1. 实现方法



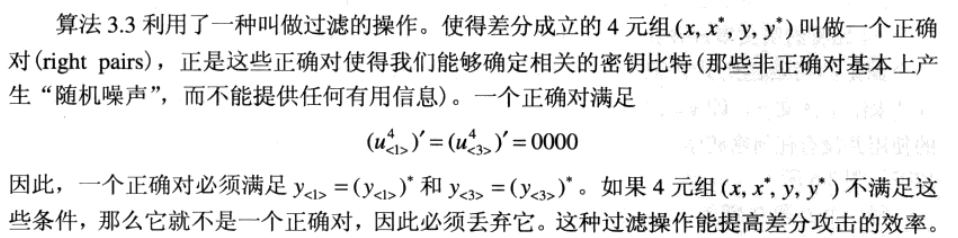
1. 代码实现

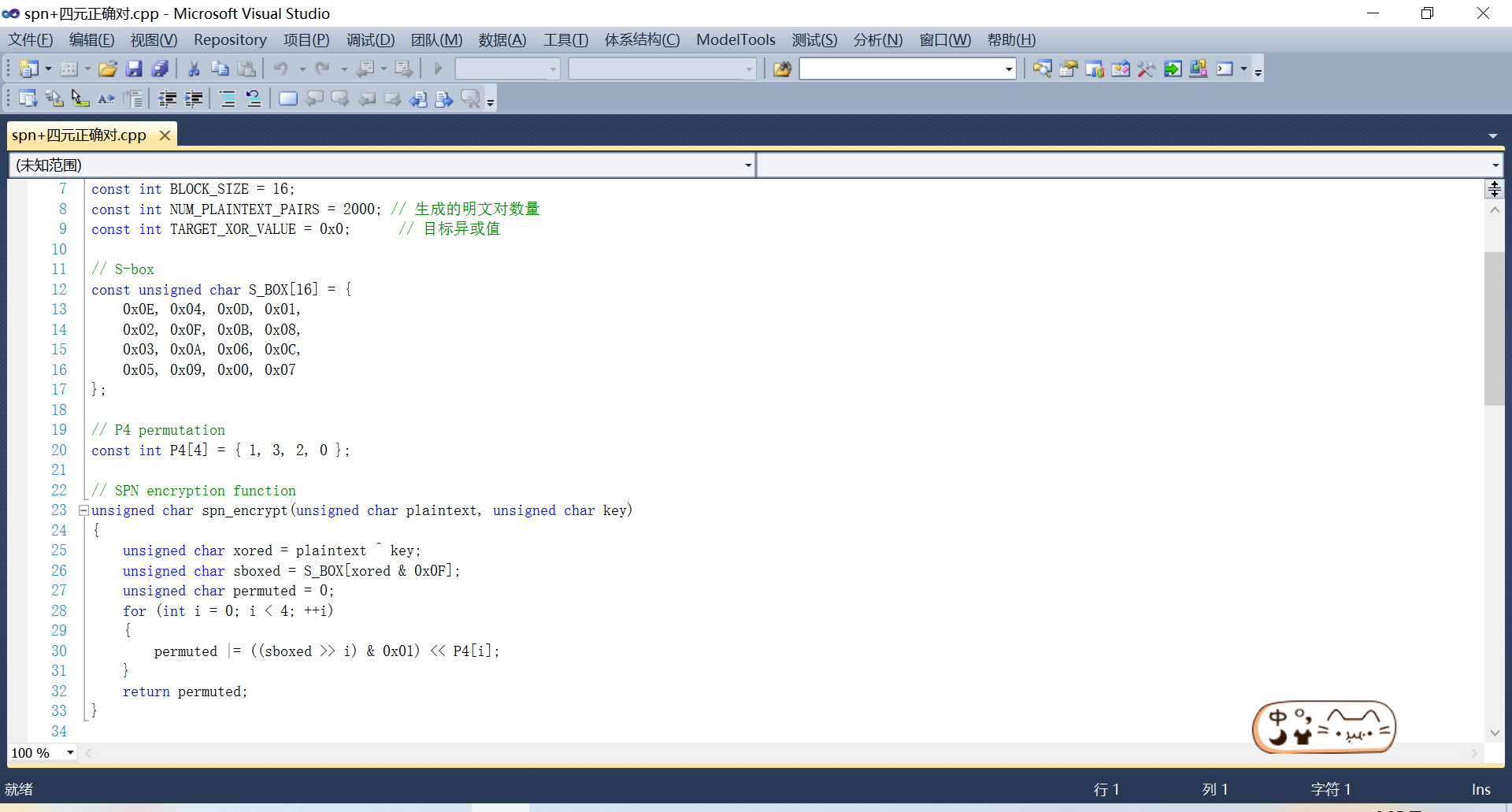
1.随机生成16位二进制数并得到2000对异或值相同的明文对：



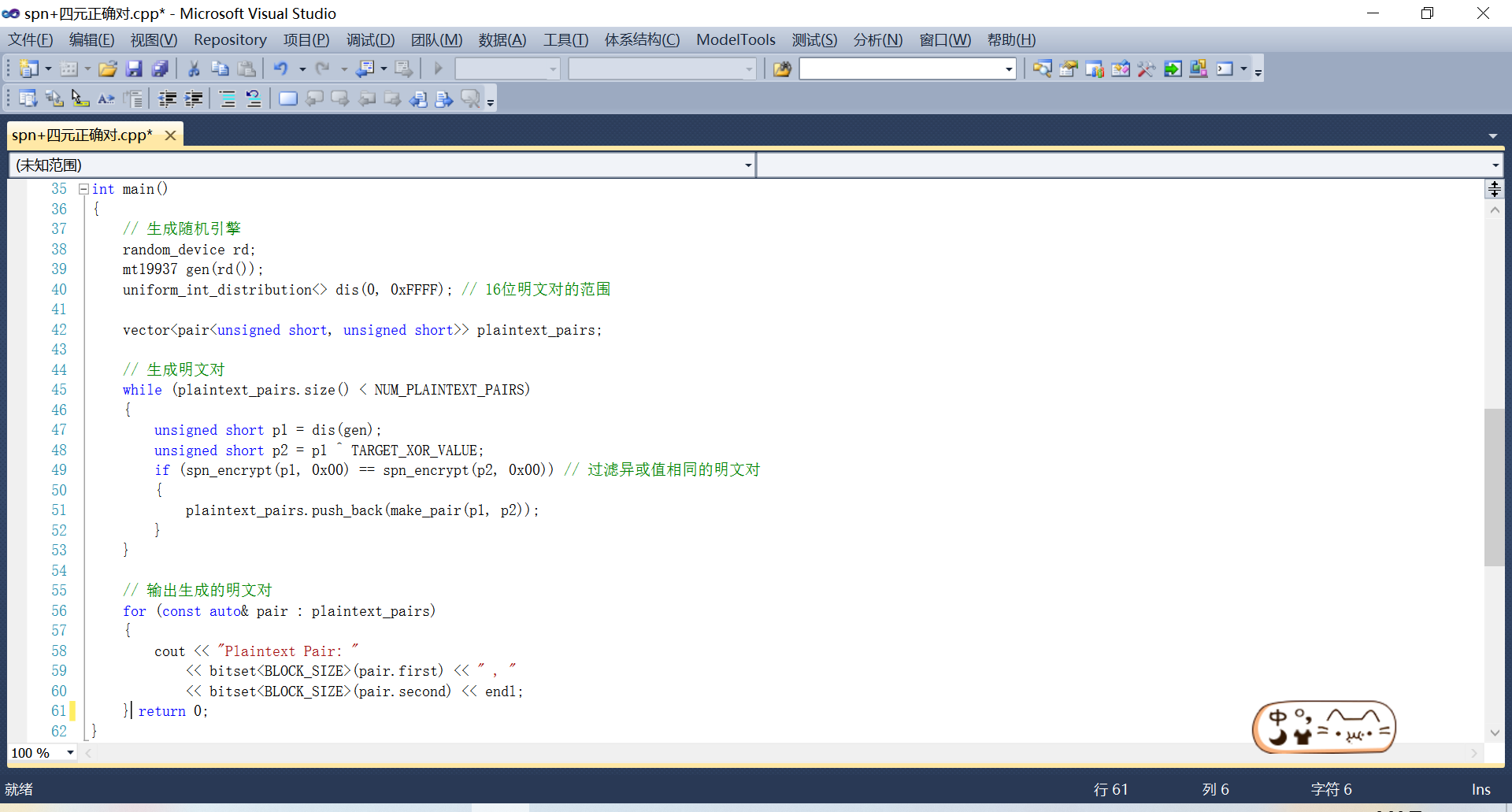


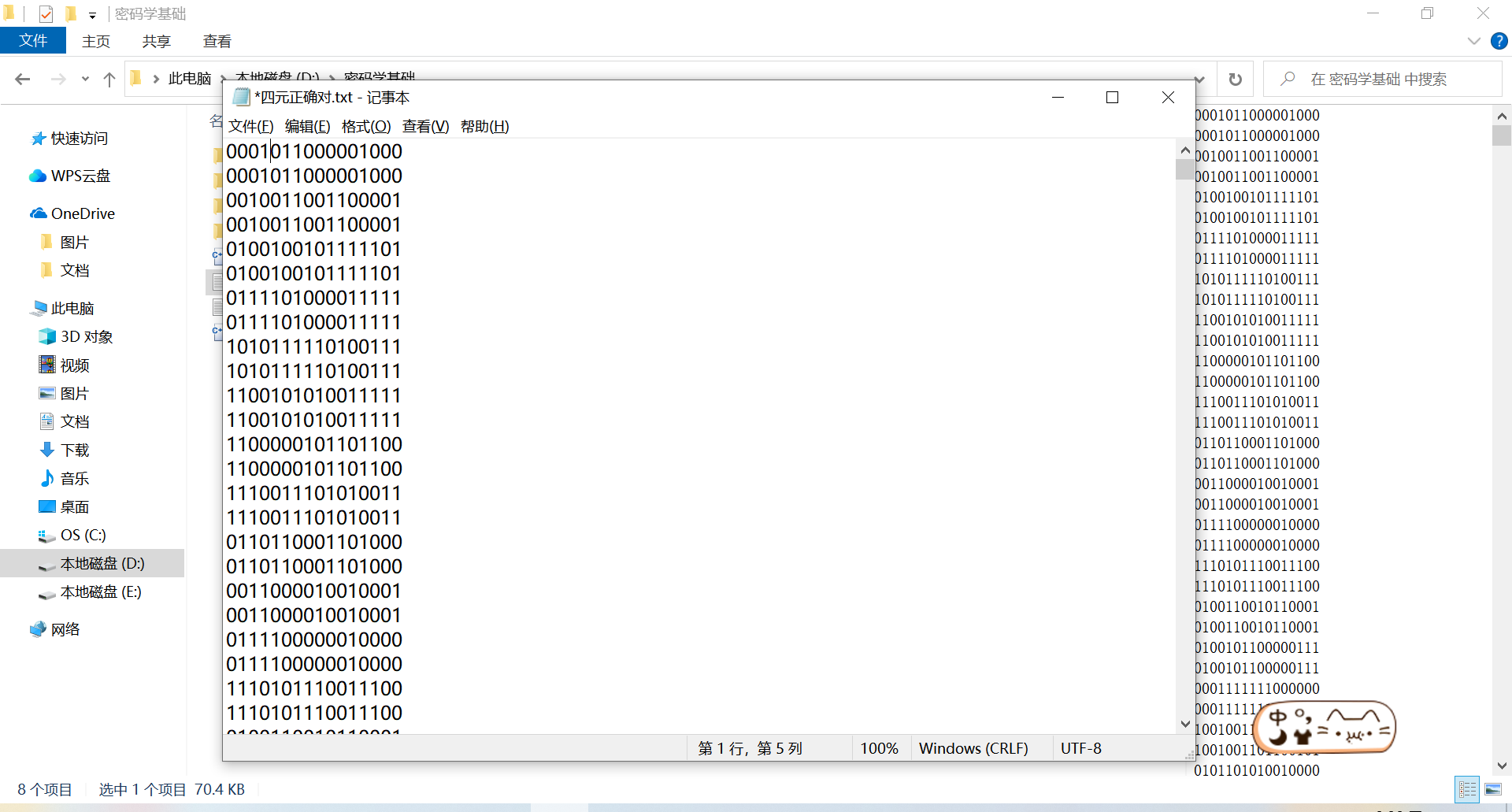
1. 过滤得到四元正确对(x,y,x\*,y\*)





生成随机引擎，输出明文对，过滤异或值相同的明文对，输出生成的明文对：





3.差分攻击

（1）十进制转四位二进制

void fuzhi(char\* w, int\* r)

{

int len = strlen(w);

for (int i = 0; i < len; i++)

{

if ( w[i] == '1' || w[i] == '0' )

{

r[i] = w[i] - '0';

}

}

}

void siweierjinzhi(int decimal, int\* binary, int j)

{

for (int i = j; i >= j - 3; i--)

{

binary[i] = decimal % 2;

decimal /= 2;

}

}

（2）S盒的逆

//S盒的逆矩阵

void niS1(int\* S, int\* S1)

{

for (int i = 0; i < 16; i++)

{

S1[S[i]] = i;

}

}

1. SPN差分攻击，测试候选子密钥(La,Lb)，异或值正确，则结果+1，找到结果值最大的(La,Lb)可能的子密钥。

部分代码块：

for (int j = 0; j < 16; j++)

{

siweierjinzhi(j, Lb, 3);

v[4] = (La[0] + y[4]) % 2;

v[5] = (La[1] + y[5]) % 2;

v[6] = (La[2] + y[6]) % 2;

v[7] = (La[3] + y[7]) % 2;

v[12] = (Lb[0] + y[12]) % 2;

v[13] = (Lb[1] + y[13]) % 2;

v[14] = (Lb[2] + y[14]) % 2;

v[15] = (Lb[3] + y[15]) % 2;

v2 = 0;

if (v[4] == 1)

v2 +=8;

if (v[5] == 1)

v2 +=4;

if (v[6] == 1)

v2 +=2;

if (v[7] == 1)

v2 +=1;

u2 = s1[v2];

siweierjinzhi(u2, u, 7);

v4 = 0;

if (v[12] == 1)

v4+=8;

if (v[13] == 1)

v4+=4;

if (v[14] == 1)

v4+=2;

if (v[15] == 1)

v4+=1;

u4 = s1[v4];

siweierjinzhi(u4, u, 15);

1. 实验结果

最少利用7组正确对可攻击成功；

