**package** danji;

**public** **class** Matrix {

//形成密钥矩阵和最终安全密钥矩阵

//把alice密钥做列向量处理，分成array.length/row列，每个列向量有一百个数，矩阵长度为100\*array.length/row

**public** **static** **int**[][] intArray2Matrix(**int**[] array, **int** row){//enhance中Matrix.intArray2Matrix(this.aliceKeys, 100)，array为alice密钥，row为100

**int**[][] matrix = **new** **int**[row][array.length/row];//二维矩阵，行数为100，列数为alice密钥长度除以100

**for** (**int** i = 0; i < array.length/row; i++) {

**for** (**int** j = 0; j < row; j++) {

matrix[j][i] = array[i\*row + j];

}//内for循环，在第i列中，对每行的矩阵，写入 array的值，

}//外for循环，选择每一列循环，从第0列到第array.length/row列。

**return** matrix;

}

//保密增强计算，把托普利兹矩阵和密钥矩阵相乘

**public** **static** **int**[][] matrixMul(**int** a[][], **int** b[][]) {//Matrix.matrixMul(this.toeplitz, aliceKeysMatrix)，a的二维矩阵是托普利兹矩阵，b的二维矩阵是alicekey的矩阵

**if** (a[0].length != b.length)//如果a的列长不等于b的行长，即托普利兹矩阵的列不等于alice矩阵的行，无法进行矩阵乘法

**return** **null**;//当相等时，往下走

**int** y = a.length;//把托普利兹矩阵的行长记为y

**int** x = b[0].length;//把alice矩阵的列长记为x

**int** c[][] = **new** **int**[y][x];//构建二维矩阵c,长度为y\*x，即托普利兹矩阵的行长\*alice矩阵的列长

**for** (**int** i = 0; i < y; i++)//外for循环，选择托普利兹矩阵的行长，作为c矩阵的行数

**for** (**int** j = 0; j < x; j++)//内for循环，选择alice矩阵的列长，作为c矩阵的列数

**for** (**int** k = 0; k < b.length; k++)//矩阵乘法，再递归加法

c[i][j] ^= a[i][k] \* b[k][j];//因为是二进制数，用异或运算

//托普利兹矩阵的矩阵的第i行第k位和alice矩阵的第k行第j位相乘，后做异或处理，得到c矩阵中的第i行第j列的数

**return** c;

}

}