用户内容数据结构:

1.

不需要分片的情况下，**明文是String**，构建成BigInteger作为算法输入，密文是长度为2的BigInteger[]

解密后是和明文相同的BigInteger，可以构建成明文String

输入：String 加密输入：String 编码后加密输入：BigInteger

密文：BigInteger[]

解密：BigInteger 解码：String 输出：String

2.

输入长度过长时，**明文是String**，若直接构建成BigInteger大小将超过N(明文空间Zn)，需要先分片成String[]，来确保String[]中的每个元素构建成的BigInteger都位于明文空间内。分片时为了防止解码错误，在每个元素前加一个英文字母或数字，解密成String[]后再依次去掉。实现时采用了z。

将String分片成String[]，再将String[]构建成BigInteger[]，因为是对数组加密，所以对BigInteger[]的加密方式可以有多种实现，默认采用对每个BigInteger[]以相同的公钥h和公共参数加密，相当于用户提供了多组输入依次加密。用String[]构建出BigInteger[]，加密生成BigInteger[][]

输入：String 加密输入：String[] 编码后加密输入：BigInteger[]

密文BigInteger[][]

解密：BigInteger[] 解码：String[] 合并：String 输出：String

3.

输入长度更长，明文超过了String的表示能力(length()返回int值,String作为不可变类型内部采用final char[]存储数据，多个char组成String，char[]的每个元素存储相同索引的String的那个字符,数组的length也返回int值，可知String和数组最大长度都是32位有符号整型，21亿字节，2.1G字节)，而数据超过2.1GBytes常有，此时需要用户分批输入，无法做到和之前一样一次将所有内容输入。每次输入一个长度在2G字节左右的String，分批输入

直到输入完成，**用户输入的明文为String[]**,用2.中相同的方法依次对String[]中每个元素分片，得到String[][]，采用2.中的方法通过String[][]构建出BigInteger[][]，加密得到BigInteger[][][]，解密得到BigInteger[][]，重构建还原出明文String[][]，将分片合并得到原始输入String[]，再依次将String[]的每个元素输出到文件，合并。

这种方式让用户每段输入作为字符串数组的一个元素，每个元素为2G字节，最大可以2G个元素，即21亿组，即可以接受21e\*21e

=4.41\*10的18次方字节，4.41EB（PBx1024），世界上还不存在这么大的内存，因此可以满足任何情况下用户的输入

输入：String[] 加密输入:String[][] 编码后加密输入：BigInteger[][]

密文：BigInteger[][][]

解密: BigInteger[][] 解码：String[][] 合并：String[]

输出到文件中：String