第二次大作业

本次大作业要求编程实现如下分组密码和哈希函数：

1. 编程实现AES-128算法的加解密运算。进一步实现AES-128的CTR工作模式的加解密。
2. 编程实现SHA-256、SHA3-256哈希算法。

要求：

1. 正确实现算法。可以参考网站<https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/example-values>中提供的测试向量进行验证和Debug。
2. 分别使用8Kbits和8Mbits两种数据流进行多次加密（哈希）运算估计算法效率，算法实现效率不小于100Mbps。
3. 在“./data”文件夹中有若干输入文件，利用你实现的密码算法对数据进行处理：
   1. 使用AES-128-CTR以text\_1.txt～test\_5.txt中数据为明文分别进行加密，结果输出至文件text\_aes\_128\_ctr\_1.txt～text\_aes\_128\_ctr\_5.txt。
   2. 使用AES-128-CTR以ciphertext\_1.txt～ciphertext\_5.txt中数据为密文分别进行解密，将解密结果输出至文件ciphertext\_aes\_128\_ctr\_1.txt～ciphertext\_aes\_128\_ctr\_5.txt。
   3. 使用SHA-256对text\_1.txt～text\_5.txt中数据流进行压缩，输出摘要结果至文件text\_sha\_256\_1.txt～text\_sha\_256\_5.txt。
   4. 使用SHA3-256对text\_1.txt～text\_5.txt中数据流进行压缩，输出摘要结果至文件text\_sha3\_256\_1.txt～text\_sha3\_256\_5.txt。

说明：

* 输入输出数据均采用txt文件，内容为十六进制表示的字节流，各字节之间空格分隔。输出文件请统一保存至文件夹“./output\_学号”下。
* a)、b)中使用的密钥均为01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10，初始计数器均为f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff（十六进制字节序列表示）。
* 为方便大家验证，我们提供了三个算法对示例文件text\_0.txt的处理结果，见“./output\_example”文件夹。

1. 相关实现代码、实验报告、数据打包提交。报告中包含算法实现逻辑、优化细节、算法实现效率的测试结果截图。

注：建议使用C或C++语言，第三次大作业中会用到本次实现的密码算法。由于算法接口为字节序列，需要进行字节序列与算法内部运算状态的转换，注意实现时的字节序问题，本次作业字节序规定同参考网站中测试向量文件相同。记一个字节序列为，其中每个为一个字节，最低位记为，最高位记为。三个算法约定如下：

* AES-128：对应课件中算法状态的。
* SHA-256：采用大端序。字节序列对应比特序列为：

消息填充时优先填充字节的高位。对字节序列表示的消息，解析为32位消息字序列，记消息字最低位为，最高位为，对应关系为

* SHA3-256：采用小端序。字节序列对应比特序列为：

消息填充时优先填充字节的低位。参考网站测试向量文件中也提供了lanes的64位整数表示，设状态的字节序列解析为64位lanes的序列，记的最低位为，最高位为，对应关系为