Report

在这个项目中任务是完成rfc6979中描述的确定性生成k的做法，其中在”确定性生成k策略.py“ 文件中，展示了该策略的正确性，可以看到，如果message和私钥不变，那么就不会改变，如果其中任何一个数改变，生成地伪随机数也会改变，如下图所示：

```
C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\SM2-RFC6979-deterministic-generate-k\确定性生成k策略.py"
message为： 3
私钥为： 00B9AB0B828FF68872F21A837FC303668428DEA11DCD1B24429D0C99E24EED83D4
生成的k为： 6562839476501080447343817551301358457417185688432752107582037361385442916374 9

C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\SM2-RFC6979-deterministic-generate-k\确定性生成k策略.py"
message为： 3
私钥为： 00B9AB0B828FF68872F21A837FC303668428DEA11DCD1B24429D0C99E24EED83D4
生成的k为： 6562839476501080447343817551301358457417185688432752107582037361385442916374 9

C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\SM2-RFC6979-deterministic-generate-k\确定性生成k策略.py"
message为： 4
私钥为： 00B9AB0B828FF68872F21A837FC303668428DEA11DCD1B24429D0C99E24EED83D4
生成的k为： 1104157897284012960066610198612173818299677156367623319178643522306454947351 10

C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\SM2-RFC6979-deterministic-generate-k\确定性生成k策略.py"
message为： 4
私钥为： 00B9AB0B828FF68872F21A837FC303668428DEA11DCD1B24429D0C99E24EED83D5
生成的k为： 8615412550835029636816170215833917271556363328633455814534171256476566900637 7
```

而在"确定性生成k在sm2中的运用.py"文件中，使用的是自己优化后的SM2类，并且将这个确定性生成k的策略封装进去，使其能被正确调用，最后在代码结尾尝试加密了字符串，发现可以验签通过，说明成功地封装进了SM2中

```
C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\SM2-RFC6979-deterministic
待签名消息为： Feng Xiangdi
私钥为： 00B9AB0B828FF68872F21A837FC303668428DEA11DCD1B24429D0C99E24EED83D5
生成的k为： 590016535239186178076493915230108615644392490538440849738689192005580160198 25
签名值的16进制表示为： 818d27473fd0e15c91698065daa98ec67058830f0c227e9eceda965f89c2d58b12d1f24e0d80356cc26fecc054aeea36575b123630ae0b6e357d2320a214c78b
验签成功！
```