结项报告

Student Full Name: 王小溪
Project ID: 22b970374

项目信息

- 项目名称：基于 secGear 实现部分国密算法
- 方案描述： 本项目采用国产机密计算框架—secGear，提供了可信执行环境，保障信息计算过程的安全性。secGear框架通过一组新的指令集扩展与访问控制机制，实现不同程序间的隔离运行，保障平台服务器的关键加解密代码和数据的机密性与完整性不受恶意软件的破坏。通过硬件上的隔离使得应用程序可以定义一个安全代码和数据区域，这一区域可以维护其机密性，即使攻击者能够物理上控制本项目平台以及产生对内存的直接攻击，也能够有效加以抵御。同时，相比于其他机密运算框架，极大缩短了开发移植适配时间，降低维护成本。

  为了保证代码的灵活性，产品性能更加优异，本项目计划不调用secGear框架提供的接口，而是通过C/C++自主编写一系列诸如异或、按位连接等基本运算操作，并对其进行封装。

- 时间规划：
  - 7.1-7.15：
    - 收集和学习国密算法相关资料
    - 收集和学习openEuler系统相关资料
    - 制定和修改项目开展方案
    - 环境安装及配置
      - openEuler系统安装及配置
      - secGear框架及相关环境安装及配置
  - 7.16-7.31：
    - 根据算法过程进行基础部分开发（secGear框架内的实用程序模块）
  - 8.1-8.15：
    - 实现SM2数字签名算法
    - 实现SM3密码杂凑算法（附加项，在SM2、SM9中使用）
    - 实现SM4分组密码算法
    - 实现SM9标识加密算法
    - 实现ZUC祖冲之密码算法
  - 8.16-9.10：
    - 测试及调试
    - 性能优化
    - 代码完善
  - 9.11-9.30：
    - 撰写报告
    - 整理、提交代码

项目总结

- 项目产出：

  1. 根据算法过程进行基础部分开发（secGear框架内的实用程序模块）：如期完成，包含：
  - `/include/common_utils.h`(通用)
  - `/include/ec.h`(用于SM2和SM9中椭圆曲线计算)
  2. 实现SM2数字签名算法：如期完成，包含：
  - `sm2/sm2.h`：SM2头文件
  - `sm2/sm2_sign.h`：SM2签名验签头文件
  - `sm2/sm2_sign_test.c`：SM2签名验签测试
  - `sm2/sm2_exchange.h`：SM2密钥交换头文件
  - `sm2/sm2_exchange_test.c`：SM2密钥交换测试
  - `sm2/sm2_crypt.h`：SM2加解密头文件
  - `sm2/sm2_crypt_test.c`：SM2加解密测试
  - `sm2/sm2_tests.h`：SM2测试头文件
  - `sm2/sm2alltest.c`：SM2完全测试
  3. 实现SM3密码杂凑算法：如期完成，包含：
  - `sm3/sm3.h`：SM3头文件
  - `sm3/sm3test.c`：SM3测试
  4. 实现SM4分组密码算法：如期完成，包含：
  - `sm4/sm4.h`：SM4头文件
  - `sm4/sm4test.c`：SM4测试
  5. 实现SM9标识加密算法：未能完成，完成部分包含：
  - `sm9/sm9.h`：SM9头文件（完成部分包含参数定义；辅助函数`H1`、`H2`、`KDF`；依赖`SM3`、`SM4`）
  6. 实现ZUC祖冲之密码算法：如期完成，包含：
  - `zuc/zuc.h`：ZUC头文件
  - `zuc/zuctest1.c`：ZUC第1部分（算法描述）测试
  - `zuc/zuctest2.c`：ZUC第2部分（机密性算法）测试
  - `zuc/zuctest3.c`：ZUC第3部分（完整性算法）测试

- 遇到的问题和解决方案：

  1. **问题：** 在C/C++中缺少对大数运算的相关库支持

     解决方案： 引用`openssl/bn.h`及其内置运算函数支持，并在编译选项中加入`-lcrypto`、在`$C_INCLUDE_PATH`/`$CPLUS_INCLUDE_PATH`包含`openssl`头文件所在目录、在`$LIBRARY_PATH`中包含`openssl/lib`所在目录，实现相关运算。

  2. **问题：** secGear框架安全侧编译问题

     解决方案： 咨询华为secGear开发团队，并且结合sgx和secGear开发实例，对CMAKELISTS进行修改

  3. **问题：** secGear框架运行时间缓慢（15-30s左右）

     解决方案： 首先，认为是创建enclave的时间过长导致，故选择两种方式来解决这个问题。一是一个执行完一个操作安全区不销毁，保留安全区的ID，二是在安全侧写一个不间断地程序，就是监听外部的请求，再选择操作，操作完不停止，继续监听下一个请求。但由于上述两种方案均存在一定问题。后经排查，发现是因为enclave设置的内存过大导致时间缓慢。调整后，时间恢复正常（1s内）

- 项目完成质量：
    - SM2数字签名算法：较高质量完成，并能通过标准文件中相关数据测试。
    - SM3密码杂凑算法：高质量完成，并能通过标准文件中相关数据测试。
    - SM4分组密码算法：高质量完成，并能通过标准文件中相关数据测试。
    - SM9标识加密算法：未能完成。
    - ZUC祖冲之密码算法：高质量完成，并能通过标准文件中相关数据测试。