鸿蒙security_huks的文件层次分析

2021-9-30 赖述忠整理

1. frameworks
   1. crypto_lite
      1. cipher/src
         1. cipher_aes.c：主要定义了初始化AES算法数据和加密文本一系列函数(在frameworks/crypto_lite/js/cipher_module.cpp中调用)
         2. cipher_rsa.c：主要定义了RSA加解密的一系列函数(在frameworks/crypto_lite/js/cipher_module.cpp中调用)
      2. js/builtin/src
         1. cipher_module.cpp：定义了两个函数模块AES和RSA进行具体完整的AES和RSA过程执行

   2. huks_standard/main
      1. common/src
         1. hks_base_check.c：最终封装为HksCheckCihperData调用函数对各自算法的参数进行检查(在hks_check_paramset.c中被调用)
         2. hks_check_paramset.c：检查paramSet结构体中存储的各类信息，比如上面的AES RSA各种类型的key等
         3. hks_common_check.c：主要是HksCheckBlob函数检查传入的参数是否非空
         4. hks_crypto_adapter.c：进行加密材料参数的填充函数和对paramSet的构建
         5. hks_param.c：关于paramSet的建立、内容添加和检查、删除的各种功能函数(在各个模块的文件中都有引用，可以推测不止建立单一的paramSet)

      2. core/src
         1. hks_local_engine.c：local算法落地包括hash算法、Mac、取模、密钥生成、明文加密(AES RSA)、签名验证。我认为这里的local说明调用时机器应该出与服务器端，接收命令在本地进行服务再通过封装和网络传输将结果送回客户端

      3. crypto_engine
         1. mbedtls
         2. openssl

      4. os_dependency
         1. ipc/src
            1. hks_client_service_ipc.c：该文件面向进程client方，提供了用于向service方请求各类密钥服务的接口
            2. hks_client_service_passthrough.c：函数先添加进程名，然后调用上述接口，请求服务端
            3. hks_ipc_check.c：参数的检查
            4. hks_ipc_serialization.c：通过几个基础拷贝函数的封装，完成多种类型不同功能密钥的捆绑封装
            5. hks_ipc_slice.c：由于存在最大数据处理限制，数据切分(slice)用于密钥签名，验签，加密，解密以及MAC计算，每次按最大可处理数据量切分，分批操作
            6. hks_request.cpp：lient与service要通过代理才能实现通信，该文件用于客户端从SAMgr获取代理，从而获取服务端接口
         2. log：轻量和标准的日志输出，一般时四个类型：消息、错误、警告和BUG
            1. hks_log_lite.c
            2. hks_log.c
         3. posix
            1. hks_mem.c：四个关于内存分配的函数释放、分配和比较
         4. sysinfo——获取系统信息(包括进程名和硬件标识符)
            1. hks_get_process_info_passthrough.c：得到进程名“hks_client”

            2. hks_get_udid.c：获取udid--硬件标识符(在hks_rkc.c中RkcGetRmkRawKey使用)

   3. huks_lite/hw_keystore_sdk
      1. common
         1. hks_bn.c：基于国密算法的大数运算处理x = a^e mod n(在hks_service.c中用到)
         2. hks_common.c：封装了各种通用函数，比如缓冲区的初始化，随机数的生成，hash计算等
         3. hks_log_utils.c：日志的工具函数显示各种类型的日志
         4. hks_utility.c：封装了各种类型的转换函数(比如uint8--->time)

      2. soft_service
         1. hks_file.c：文件相关操作为主，主要包括文件数据的读写，文件数据的初始化以及文件安全性管理等功能，文件是密钥存储的集成管理系统，因此对文件相关的操作便尤为重要
         2. hks_rkc.c：主要是有关根密钥组件相关操作函数的结合，其中包括将密钥信息填入缓冲区中以及从缓冲区中提取密钥信息，读取、检查、初始化、恢复、销毁密钥库文件信息，进行特定密钥的获取以及信息的加密解密操作等
         3. hks_service.c：以密钥服务为核心内容，其中包括了各种密钥生成与加密方案，密钥算法与协议的匹配，常用hash、大数指数模运算方法，密钥初始化、传输、更新、修改、删除等管理功能以及注册文件回调相关服务，是整个软服务的核心部分
         4. hks_storage.c：主要涉及内存中密钥缓冲区的读写，计算内存中存储密钥的相关信息（数量、插槽信息……）、以及密钥相关空间的变动等
      3. hks_access.c：猜测：应该时客户端接入服务器端获取服务的access过程函数
      4. hks_client.c：客户端在access后建立与服务端的连接从而获得一系列可支持的服务(通过hks_access.c中的通道进行服务的请求和结果返回)

2. interfaces/innerkits
   1. huks_lite
   2. huks_standard/src
      1. hks_api.c：用于调用各种加密算法、验证等服务的API

3. services/huks_standard
   1. huks_engine
      1. core/src
         1. hks_auth.c：对于用户进行authParam和requestParam的检查
         2. hks_core_service.c：核心服务的构造，比如数据的加解密，各种密钥的封装，签名的验证，消息摘要的生成等
         3. hks_keyblob_lite.c：各种类型的keyblob的构造加密和解密，包括密钥和paramSet等
         4. hks_keyblob.c：标准版的各种keyblob的构造加密和解密，功能与lite相同，但是实现略有不同
         5. hks_rkc_rw.c：提取(extract)日期，masterKey等信息
         6. hks_rkc.c：rootKey component
         7. hks_upgrade_key_info.c： KeyInfo结构体的初始化、更新和删除
      2. os_dependency

   2. huks_service
      1. core/src
         1. hks_client_check.c：客户端的各种常规参数的检查比如proessName，密钥参数等
         2. hks_client_service_adapter.c：非lite版本使用的加密框架为openSSL，实现客户端和服务端数据格式的转换--X509格式跟其他格式的适配
         3. hks_client_service_adapter_lite.c：lite版本使用mbedtls与标准版功能相似
         4. hks_client_service.c：为进程间客户端(请求方)提供各类服务的具体实现，涉及密钥存储管理实现、各类加密算法实现等相关函数，hks_ipc_service.c中经过准备相关函数参数后，直接调用hks_client_service.c其中的函数即实现对应服务请求
         5. hks_double_list.c：双向列表的初始化、插入、移除
         6. hks_operation.c：operation操作链的各种操作函数的封装
         7. hks_storage_adapter.c：存储类型的适配，将密钥的信息blob拆封再封装
         8. hks_storage_lite.c：轻量版的密钥文件读写封装

9. hks_storage.c：密钥文件的读写，路径的查找填充，密钥文件名的统计，各类型文件存储和销毁
10. hks_upgrade_storage_data.c：围绕着本地密钥存储文件的更新与删除进行功能编写。通过进程名来对本地文件进行访问和更新，通过偏移量和内存拷贝实现文件的更新

2. os_dependency
    1. idl
        1. ipc
            1. hks_ipc_serialization.c：通过几个基础的拷贝函数的封装得到的类型和参数不同的copy函数，构造各种信息体验证获取的hks函数
            2. hks_ipc_service.c：每个进程通过uid唯一标识，该文件为各进程提供了相关的密钥服务与证书服务接口（内部具体实现在于core文件夹），保证其数据安全
            3. hks_response.cpp：调用reply中的写函数将response的内容写入缓冲区
        2. passthrough
            1. hks_passthrough_access.c：封装一层access——表示通道入口——实则调用hkscore的核心函数

    2. posix
        三种形式的文件读写和删除——一种是鸿蒙自己封装的utils工具文件读写，一种是基于C库的fread等指针文件读写，还有一种是基于Linux系统的接口调用——通过文件标识符来进行文件的唯一识别，读写和删除，这里分为了轻量版和标准版
        1. hks_file_operator_lite.c
        2. hks_file_operator.c

    3. sa
        1. huks_sa.cpp：定义了消息类型和操作处理以及服务的登记初始化开始等预备功能函数