SWC 虚拟项目
概要设计文档

版本号: 0.5
发布日期: 2020-09-21

版本历史

版本号	日期	制/修订人	内容描述
0.1	2020.9.15	汤健雄	建立文档
0.5	2020.9.15	汤健雄	1. 增加外部接口 2. 增加 debug 流程

版权所有 © 珠海全志科技股份有限公司。保留一切权利

目　录

1 概述 1
1.1 编写目的 . . . 1
1.2 适用范围 . . . 1
1.3 相关人员 . . . 1

2 项目介绍 2
2.1 项目背景 . . . 2
2.2 项目功能介绍 . . . 2
2.3 限制条件 . . . 2

3 软件总体结构设计 3
3.1 整体目录结构 . . . 3
3.2 整体结构框图 . . . 3
3.3 解耦设计 . . . 4
3.3.1 后台服务应用 A . . . 4
3.3.2 客户端 B . . . 5
3.3.3 客户端 C . . . 5
3.3.4 内核模块 K . . . 5
3.3.5 编解码模块 . . . 5
3.3.6 HASH 模块 . . . 5
3.3.7 数据封装模块 . . . 6
3.3.8 Log 日志模块 . . . 6

4 模块流程设计 7
4.1 A 与 B 通信过程 . . . 7
4.2 A 完成一次发送数据（向 B 或 C）的过程 . . . 9
4.3 A 完成一次接收数据（来自 B 或 C）的过程 . . . 10
4.4 K 完成一次转发数据的过程 . . . 11
4.5 A（B 或 C）与 K 具体通信过程（netlink 方式） . . . 11

5 接口设计 13
5.1 编解码模块接口 . . . 13
5.1.1 Encode . . . 13
5.1.2 Decode . . . 13
5.2 封装模块接口 . . . 14
5.2.1 Encapsulate . . . 14
5.2.2 Decapsulate . . . 14
5.3 HASH 计算模块接口 . . . 15
5.3.1 GetHash . . . 15

6 DEBUG 流程 17
6.1 函数测试 . . . 17
6.2 模块测试 . . . 17

ALLWINNER®

版权所有 © 珠海全志科技股份有限公司。保留一切权利

7 出错处理 **18**
7.1 函数返回值 18
7.2 数据有效性 18

8 总结 **19**

ALLWINNER®

插　图

3-1 整体结构框图 4
4-1 A 与 B 通信过程 8
4-2 A 完成一次发送数据过程 9
4-3 A 完成一次接受数据过程 10
4-4 K 完成一次转发数据过程 11
4-5 A 与 K 具体通信过程 12

版权所有 © 珠海全志科技股份有限公司。保留一切权利

1 概述

1.1 编写目的

此文档的编写目的是为了更加高效、快速并高质量的完成虚拟项目的开发任务，介绍 A100 平台上的 Demo 软件项目概要设计。

1.2 适用范围

适用范围为全志 A100 平台。

1.3 相关人员

全志 P616 项目 Demo 软件项目的设计者和开发者。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2 项目介绍

2.1 项目背景

M 公司是一家采用全志 SOC 集成方案的品牌大客户，7 月份跟全志合作立项，远程联合开发一个代号 P166 的重要项目，该项目基于全志 A100 平台，为了加快项目并行进度，P166 项目客户端项目经理 L，向全志 Aserver 平台提交了一项软件开发需求，要求全志方提供一套易用、稳定、可复用的软件 Demo，降低客户端前期开发工作量，加快二次开发整体进度。- 项目负责人：苏佳佳 - 参与人员：汤健雄、FAE 主管、项目经理

2.2 项目功能介绍

软件 Demo 包含后台服务应用 A、客户端应用 B、客户端应用 C 和内核模块 K 四个独立组件。K 作为 A 和 B、A 和 C 之间的通信中转站，B 和 C 之间不能通信。软件 Demo 主要有以下功能。
Demo 功能 1：A 和 B 发生一次通信，A 将数据包编码后发送给 K，K 受到数据包转发给 B，B 对数据包完成逆向解码还原，并将原始数据的 HASH 值字符串通过 K 返还给 A。A 受到 HASH 值字符串进行正确性校验，校验成功完成通信，校验失败后 Log 日志抛出异常码 ERN110。
Demo 功能 2：同理 A 和 C 发生通信过程如上，校验失败后 Log 日志抛出异常码 ERN120。

2.3 限制条件

- 规格软件开发：保证解耦设计，可被二次定制，具有一定的鲁棒性
  代码规范：代码风格符合 SWC 和 SW4 的代码规范要求，使用 git 进行统一的管理
  测试：各个模块支持多种方便、单独的调试手段，支持临时数据的调试，支持命令调试
  文档：符合软件设计文档规范，并需在内部评审通过

- 交付说明代码：提交至 git 仓库——SWC-Bootcamp
  文档：上传至 edoc，具体文档包括：虚拟项目任务计划书，软件概要设计文档，各个组件的测试列表、测试报告，各个模块代码的静态代码检查报告，组件之间的联调报告，代码的 ROM/RAM 分析报告，开发、调试过程的记录文档，总结文档。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3 软件总体结构设计

3.1 整体目录结构

```
├── sprit                          脚本文件
├── docs                           文档目录
├── out                            编译好的各模块文件
├── package                        外部依赖项
├── hdr                            头文件目录，便于各模块调用
| ├──codec.h                       编解码模块头文件
| ├──hash.h                        hash模块头文件
| ├──encapsulation.h               数据封装头文件
| ├──log.h                         日志记录头文件
├── src                            源码文件目录
| ├──kernel                        内核空间目录
| | ├── kernel_k.c                 内核模块K
| | ├── Makefile
| ├── user                         用户空间目录
| | ├── server_a.c                  后台服务A模块，与B、C通信
| | ├── client_b.c                  客户端应用B模块，主要向内核发送、接收信息
| | ├── client_c.c                  客户端应用C模块，主要向内核发送、接收信息
| | ├── codec.c                    编解码模块函数实现
| | ├── hash.c                     哈希算法模块函数实现
| | ├── encapsulation.c            数据封装函数实现
| | ├── log.c                      log日志实现
├── test                           测试目录
├── README.md                      介绍代码的用途、支持的软硬件平台，使用说明
├── tools                          编译工具
└── build.sh
```

3.2 整体结构框图

Demo 通信软件主要由四个子模块组成，分别是后台服务应用 A、客户端应用 B、客户端应用 C 和内核模块 K。除了四个子模块之外有四个公共模块：编解码模块、HASH 计算模块、封装模块和日志模块。

整体的结构框图如下所示

版权所有 © 珠海全志科技股份有限公司。保留一切权利

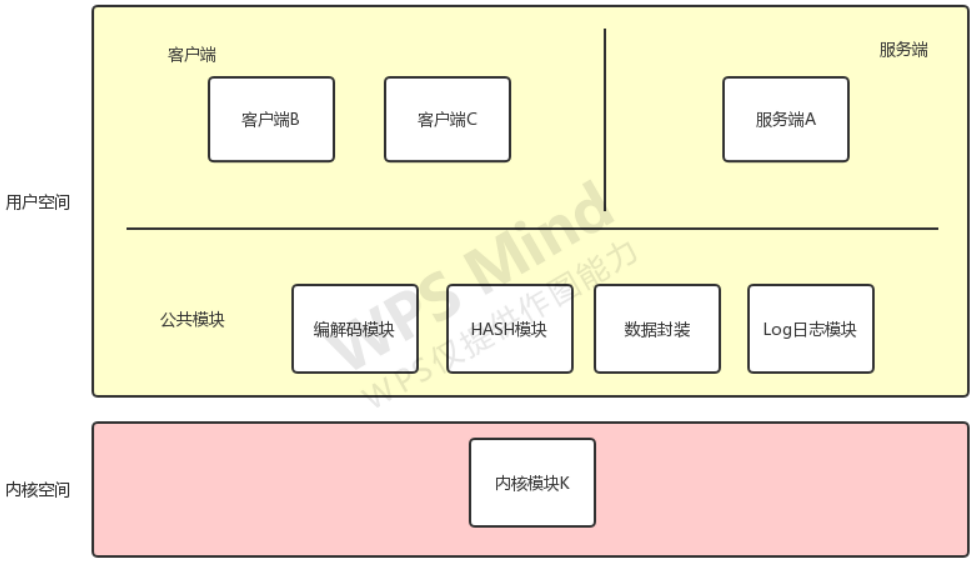

图 3-1: 整体结构框图

3.3 解耦设计

为了提高客户二次开发的效率，我们的 demo 软件包充分考虑了解耦设计，各个模块只需要调用公共模块的接口即可，若需要二次定制，只需修改公共接口的实现方式。

3.3.1 后台服务应用 A

后台服务应用 A，包括三个子功能，分别是发送数据、接收数据、HASH 校验模块。

发送数据:

对原始数据封装加密之后发送给 K，这一过程需要调用数据封装模块和编解码模块。

将原始数据的 HASH 值返还给 K，这一过程需要调用 HASH 计算模块和封装模块。

接收数据:

接收来自 K 的编码数据，解码还原。

-接收来自 K 的 HASH 值。

HASH 校验：根据 HASH 值完成 HASH 校验，如果成功则完成通信，失败则抛出异常码 ERN120 或 ERN110。此时需要调用 Log 日志模块完成记录。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3.3.2 客户端 B

后台服务应用 B，包括三个子功能，分别是发送数据、接收数据、HASH 校验模块。
发送数据：
对原始数据封装加密之后发送给 K，这一过程需要调用数据封装模块和编解码模块。
将原始数据的 HASH 值返还给 K，这一过程需要调用 HASH 计算模块和封装模块。
接收数据：
接收来自 K 的编码数据，解码还原。
接收来自 K 的 HASH 值。
HASH 校验：根据 HASH 值完成 HASH 校验，如果成功则完成通信，失败则抛出异常码 ERN110。此时需要调用 Log 日志模块完成记录。

3.3.3 客户端 C

后台服务应用 C，包括三个子功能，分别是发送数据、接收数据、HASH 校验模块。
发送数据：
对原始数据加密封装之后发送给 K，这一过程需要调用数据封装模块和编解码模块。
将原始数据的 HASH 值封装后返还给 K，这一过程需要调用 HASH 计算模块和封装模块。
接收数据：
接收来自 K 的编码数据，解码还原。
接收来自 K 的 HASH 值。
HASH 校验：根据 HASH 值完成 HASH 校验，如果成功则完成通信，失败则抛出异常码 ERN120。此时需要调用 Log 日志模块完成记录。

3.3.4 内核模块 K

内核模块 K，包括三个子模块分别是接收数据，转发数据和调试端口。
接收数据：接收来自 A（B 或 C）的数据，通过封装字段识别出数据的来源和去向，若是 BC 之间的通信，则停止转发并报错记录。这一过程需要调用 Log 日志模块。
转发数据：将受到的数据转发给 A（B 或 C）。

3.3.5 编解码模块

编解码模块需要具备编码功能和解码功能，能被 ABC 三个模块调用。

3.3.6 HASH 模块

HASH 模块需要具备计算 HASH 值的功能，能被 ABC 三个模块调用。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3.3.7 数据封装模块

对数据进行封装，添加发送者和接收者的信息，方便转发。对数据进行解封装，识别出发送和接收者的信息，还原原始数据。

3.3.8 Log 日志模块

记录 ABC 之间的通信信息。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4 模块流程设计

本项目的主要功能就是 AB 和 AC 之间的通信，本模块首先从宏观的角度出发，设计 A 与 B 发生通信过程时 A、B、C、K 各个模块的大致流程，然后从细节上设计整个通信过程中，各个子流程的流程图和各个模块的数据调用过程。

4.1 A 与 B 通信过程

完成一次 A 与 B 通信的具体步骤如下：

- 后台服务端 A、内核模块 K、客户端 B 完成初始化，准备通信
- 后台服务端将合法数据编码封装后转发给 K，这一过程需要调用编解码模块和封装模块
- 内核模块 K 接受数据，解封装并判断发送者和接受者的合法性，这一过程需要调用封装模块
- 内核模块 K 转发数据包给 B
- 客户端 B 接收数据并进行解码还原，这一过程需要调用编解码模块
- 客户端 B 计算原始数据的 HASH 值字符串封装发送给 K，这一过程需要调用封装模块
- 内核模块 K 接受数据，解封装并判断发送者和接受者的合法性，这一过程需要调用封装模块
- 内核模块 K 转发 HASH 值字符串给 A
- A 接收 HASH 字符串并进行校验
- 完成通信或者抛出异常

版权所有 © 珠海全志科技股份有限公司。保留一切权利

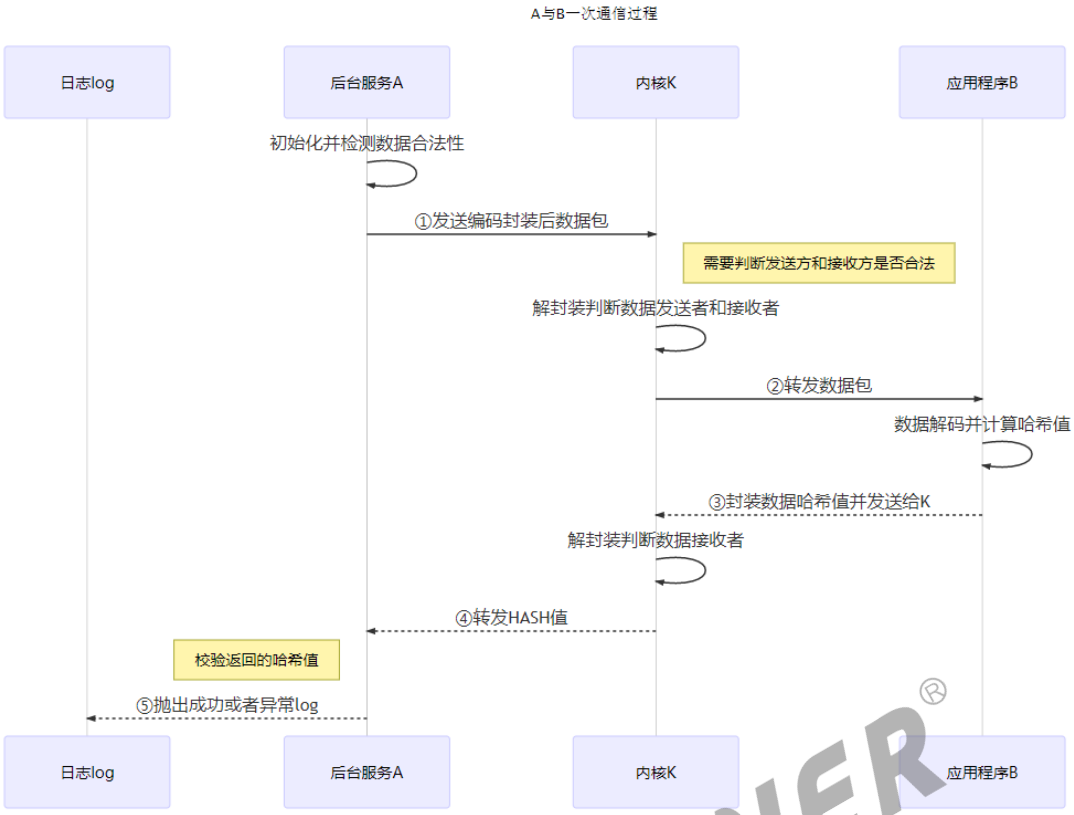

图 4-1：A 与 B 通信过程

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4.2 A 完成一次发送数据（向 B 或 C）的过程

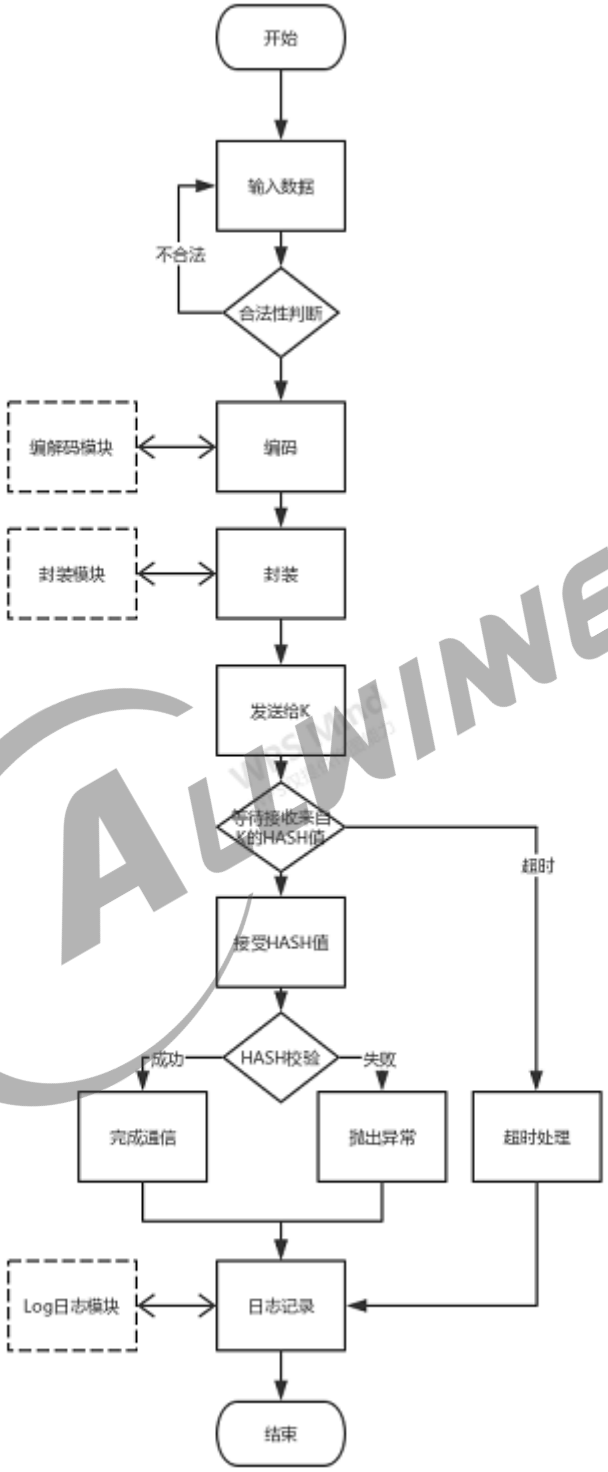

图 4-2: A 完成一次发送数据过程

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4.3 A 完成一次接收数据（来自 B 或 C）的过程

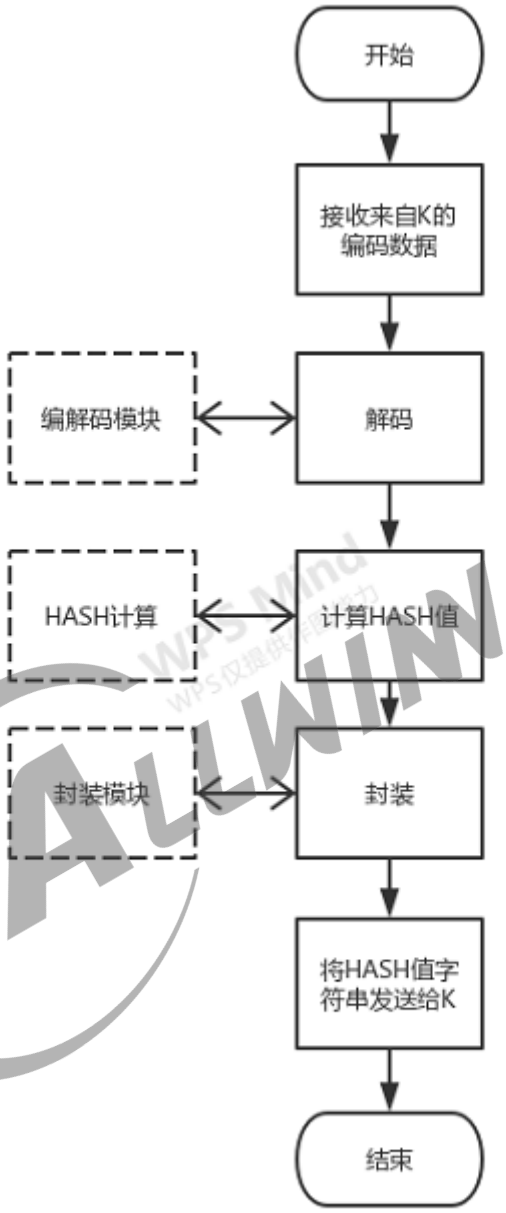

图 4-3: A 完成一次接受数据过程

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4.4 K 完成一次转发数据的过程

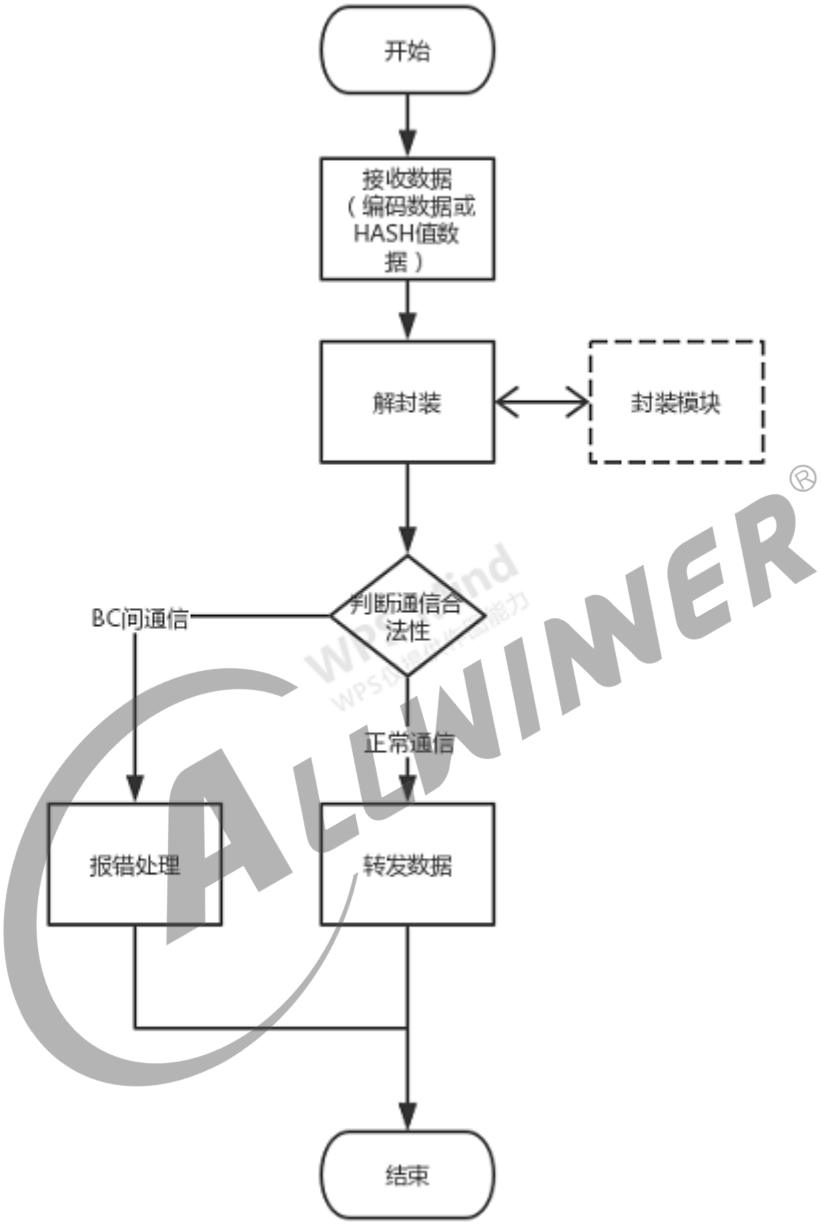

图 4-4: K 完成一次转发数据过程

4.5 A（B 或 C）与 K 具体通信过程（netlink 方式）

内核模块 K 和服务应用 A 通信的过程如下：

版权所有 © 珠海全志科技股份有限公司。保留一切权利

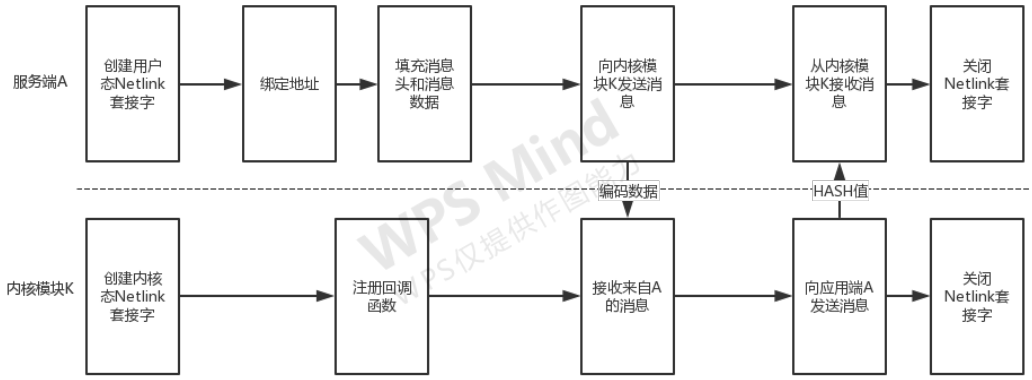

图 4-5: A 与 K 具体通信过程

后台服务 A 首先在用户态使用 socket() 函数创建套接字，然后通过 bind 函数绑定地址，然后按照相关的规则封装消息，然后通过 sendmsg() 将封装好的编码信息发送给内核模块 K，使用 recvmsg 接收来自内核模块 K 的 HASH 值消息，结束通信后通过 close 函数关闭套接字，释放资源。

内核模块 K 在加载时创建一个内核态的 Netlink 套接字，指明接收数据的处理函数 input()。当收到后台服务应用 A 的消息时，内核模块 K 调用 input 函数进行处理，需要发送消息时，调用 netlink_unicast() 函数进行单播或 netlink_broadcast() 函数进行组播，向用户进程发送信息。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

5 接口设计

5.1 编解码模块接口

5.1.1 Encode

- 函数原型：

```
int Encode(unsigned char *original_data_str,
          int len,
          unsigned char *encode_data_str);
```

- 作用：对原始数据进行编码加密
- 参数：
  - 参数 1：原始数据
  - 参数 2：原始数据长度
  - 编码后数据
- 返回：
  - 1：编码成功
  - -1：编码失败，原始数据类型不合法
  - -2：编码失败，原始数据长度不合法

5.1.2 Decode

- 函数原型：

```
int Decode(unsigned char *encode_data_str,
           int len,
           unsigned char *decode_data_str);
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 作用：检索指定属性的值
- 参数：
  - 参数 1：编码数据
  - 参数 2：编码数据长度
  - 参数 3：解码后数据
- 返回：
  - 1：编码成功
  - -1：编码失败，原始数据类型不合法
  - -2：编码失败，原始数据长度不合法

5.2 封装模块接口

5.2.1 Encapsulate

- 函数原型：

```
int Encapsulate(unsigned char *original_data_str,
                int len,
                unsigned char *encapsulate_data_str);
```

- 作用：对数据进行打包，加入发送者和接收者的信息
- 参数：
  - 参数 1：待打包数据
  - 参数 2：带打包数据长度
  - 参数 3：接收打包完成数据
- 返回：
  - 1：封装成功
  - -1：封装失败，数据类型不符
  - -2：封装失败，数据长度不符

5.2.2 Decapsulate

- 函数原型：

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
int Decapsulate(unsigned char *encapsulate_data_str,
                  int len,
                  unsigned char *original_data_str,
                  int sender,
                  int receiver);
```

- 作用：检索指定属性的值
- 参数：
  - 参数 1：封装数据
  - 参数 2：封装数据长度
  - 参数 3：原始数据
  - 参数 4：发送者 id
  - 参数 5：接收者 id
- 返回：
  - 1：解封成功
  - -1：封装失败，数据类型不符
  - -2：封装失败，数据长度不符
  - -3：封装失败，其他

5.3 HASH 计算模块接口

5.3.1 GetHash

- 函数原型：

```
int GetHash(unsigned char *original_data_str,
              int len,
              unsigned char *hash_str)
```

- 作用：计算原始字符串的 HASH 值
- 参数：
  - 参数 1：原始数据
  - 参数 2：原始数据长度
  - 参数 3：原始数据 HASH 值字符串
- 返回：

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 1：计算成功
- -1：计算失败，数据类型不符
- -2：计算失败，数据长度不符
- -3：计算失败，其他

版权所有 © 珠海全志科技股份有限公司。保留一切权利

6 DEBUG 流程

6.1 函数测试

完成函数测试有如下方法：

- 关键位置打印
- 函数返回值错误码测试
- 不同函数之间互相测试

6.2 模块测试

内核模块测试
- 设备节点创建正常 - 调试节点创建正常服务端 A 测试
- 数据发送正常 - 数据接收正常 - HASH 值校验正常客户端 B 和 C 测试
- 数据发送正常 - 数据接收正常 - HASH 值校验正常公共模块
- 编解码模块编码功能正常 - 编解码模块解码功能正常 - 数据封装模块数据打包拆包正常 - HASH 计算模块计算功能正常 - log 日志模块添加日志功能正常不同模块之间的互相交叉测试

版权所有 © 珠海全志科技股份有限公司。保留一切权利

7 出错处理

7.1 函数返回值

对每个带返回值的函数进行函数返回值的有效性校验，确保代码有效运行。

7.2 数据有效性

每个模块接受的数据都需要进行有效性检验，如不符合标准则会将其丢弃。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

8 总结

这篇概要设计旨在完成客户的最基本的软件需求，还有很多深层次的功能需求并未完成，后续有待改进，改进点如下：

- 为了完成基本通信功能，通信方式上采用了主流的 Netlink，并未考虑其他种类的方式
- 在数据的传输过程中，本次设计暂时只考虑了单客户端通信过程，并未考虑并发控制
- 数据的 HASH 计算方法，暂时考虑 Linux 自带的 md5
- 打包方式，自定义最简单的添加表头的方法，后续有待改进

版权所有 © 珠海全志科技股份有限公司。保留一切权利

著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、、全志科技、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。