GPRS OTA方案征求意见稿

根据我目前的理解，GPRS OTA的产品特点如下（不正确的地方请指出）：

1. 价格敏感，可能会更换更低价格的MCU
2. 设备功能单一，扩展IO的可能性不大
3. GPRS通信，带宽需要资费
4. 功耗敏感
5. GPRS通信由模块完成

首先介绍两种bootload的策略，及主要特点

Standalone bootload

Bootload自带无线通信模块，bootload中自己完成下载、校验和更新三步。

Bootload功能多，体积大

Application bootload

Bootload不带无线通信模块，Application中完成下载，bootload中完成校验和更新。

Bootload功能少，体积小

Standalone bootload需要有无线通信模块，如果需要还需要协议栈。

Application bootload需要完整下载整个固件文件，对存储器有较高要求，尤其是flash。

综合目前的产品特点，因为GPRS模块的协议栈不需要单片机实现，无线通信模块软件代价较小，而flash资源可能会很紧张，所以采用Standalone bootload

典型的Application bootload的方案如下：



大致流程如下：

1. 升级人员通过http界面选择更新文件，点击升级
2. 通过TCP向设备发送进入bootload界面
3. 设备的app改写进入Bootload标志并复位，进入bootload
4. 设备的bootload等待权限认证
5. 主机根据更新文件中信息完成权限认证
6. 设备反馈通过权限认证，并等待下载任务
7. 主机根据更新文件启动下载任务
8. 设备反馈进入下载状态，并等待数据内容
9. 主机根据更新文件逐条发送数据
10. 设备收到数据后立刻烧写
11. 主机发送完最后一条数据后发送下载完成命令
12. 设备收到下载完成命令后校验整个烧写文件，反馈校验状态
13. 主机收到校验正确后发送复位命令
14. 设备收到复位命令后直接复位

此流程未包括bootload的OTA，如要支持bootload的OTA，需要：

1. 额外的RAM，用于缓存bootload文件
2. 更新文件中对下载任务分类，设备在收到下载bootload的任务时将收到的数据包缓存到RAM
3. 校验完成后要启动“烧写bootload并复位”功能，通过主机下发

需要资源：

1. flash资源=APP + bootload
2. 少量EEPROM用于存储标志，或者NVRAM寄存器也可，或者占用1~2page的flash
3. RAM资源 = max(APP使用量，栈大小+bootload变量+bootload文件大小)

OTA文件格式（暂未描述）

考虑使用标准格式（未找到，帮忙找找），否则自定义。

OTA协议（暂未描述）

考虑使用标准协议（未找到，帮忙找找），考虑使用全ASCII码的协议，带来少量流量的额外开销。

采用TCP协议，设备做server，主机client，固定端口接收client的命令并服务。部分命令需要在APP下处理（如进入bootload），因此APP下也要支持这个server

权限算法

考虑使用伪随机数算法，主机请求一个seed，再发送一个key，设备判断是否正确

压缩算法

RAM资源紧张，不适合使用压缩算法，为了方便使用AT指令（ASCII码），考虑使用BASE64编码

加密算法（暂未描述）

GPRS模块是否支持标准的加密算法？