消费机 PUSH 通讯协议

PUSH SDK

文档版本：V1.0 日期：2018年10月

push 协议版本：V1.0.0

ZKTeco

修改记录

日期	版本	描述	修改人	备注
2018/10/22	初版	1. 支持新架构设备CM102/CM108机型的push通信。	王温馨	

目录

1 摘要

本说明书是客户和开发者对原型设计说明，是开发者进行后续软件开发工作的依据。

本协议基于超文本传输协议（HTTP）的基础上定义的数据协议，建立在TCP/IP连接上，主要应用于中控新架构消费设备，定义了数据（用户信息、消费记录等）的传输格式、控制设备的命令格式。目前支持的服务器有ZKECO等。

2 特点

●新数据主动上传

●断点续传

●所有行为都由客户端发起，比如上传数据、服务器下发的命令等

2.1 编码

协议中传输的数据大部分都是ASCII字符，但是个别的字段也涉及到编码的问题，比如用户姓名， 所以对该类型数据做如下规定：

●为中文时，使用GB18030编码

●为其他语言时，使用UTF-8编码

目前涉及到该编码的数据如下：

●用户信息表的用户姓名

2.2 HTTP协议简介

Push协议是基于HTTP协议的基础上定义的数据协议，这里简单介绍下什么是HTTP协议，如果已经熟悉可跳过此部分。

HTTP协议是一种请求/响应型的协议。客户端给服务器发送请求的格式是一个请求方法（request method），URI，协议版本号，然后紧接着一个包含请求修饰符（modifiers），客户端信息，和可能的消息主体的类MIME（MIME-like）消息。服务器对请求端发送响应的格式是以一个状态行（status line），其后跟随一个包含服务器信息、实体元信息和可能的实体主体内容的类MIME（MIME-like）的消息。其中状态行（status line）包含消息的协议版本号和一个成功或错误码。如下例子

客户端请求：

GET http://113.108.97.187:8081/iclock/accounts/login/?next=/iclock/data/iclock/ HTTP/1.1
User-Agent: Fiddler
Host: 113.108.97.187:8081

服务器响应：

HTTP/1.1 200 OK
Server: nginx/0.8.12
Date: Fri, 10 Jul 2015 03:53:16 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Content-Language: en
Expires: Fri, 10 Jul 2015 03:53:16 GMT
Vary: Cookie, Accept-Language
Last-Modified: Fri, 10 Jul 2015 03:53:16 GMT
ETag: "c487be9e924810a8c2e293dd7f5b0ab4"
Pragma: no-cache
Cache-Control: no-store
Set-Cookie: csrftoken=60fb55cedf203c197765688ca2d7bf9e; Max-Age=31449600; Path=/
Set-Cookie: sessionid=06d37fdc8f36490c701af2253af79f4a; Path=/

HTTP通信通常发生在TCP/IP连接上。默认端口是TCP 80，不过其它端口也可以使用。但并不排除HTTP协议会在其它协议之上被实现。HTTP仅仅期望的是一个可靠的传输（译注：HTTP一般建立在传输层协议之上）；所以任何提供这种保证的协议都可以被使用。

3 定义

文档中引用定义使用格式为：${ServerIP}

●ServerIP：服务器IP地址

●ServerPort：服务器端口

●XXX：未知值

●Value1\Value2\Value3……\Valuen：值1\值2\值3……值n

●Required：必须存在

●Optional：可选

●SerialNumber：系列号(可以为字母、数字、字母+数字组合)

●NUL：null（\0）

●SP：空格

●LF：换行符（\n）

●HT：制表符（\t）

●DataRecord：数据记录

●CmdRecord：命令记录

●CmdID：命令编号

●CmdDesc：命令描述

●Pin：工号

●Workcode：workcode编码

●Reserved：预留字段

●OpType：操作类型

●OpWho：操作者

●OpTime：操作时间

●BinaryData：二进制数据流

●TableName：数据表名

●SystemCmd：系统命令

●Key：键

●Value：值

●FilePath：文件路径

●URL：资源位置

●DeviceName：设备名称

4 功能

客户端的角度来描述Push协议支持的功能

- 初始化信息交互
- 上传数据
- 获取命令
- 命令回复

5 流程

使用Push协议的客户端和服务器，必须由客户端先发起“初始化信息交互”请求成功之后，才能使用其他功能，比如上传数据、获取服务器命令、上传更新信息、回复服务器命令等，其中这些功能并没有先后顺序，取决于客户端应用程序的开发，如下图

6 初始化信息交互

客户端发起请求，将相应的配置信息发送给服务器，服务器接收到该请求，将相应的配置信息回复给客户端，只有当客户端获取到相应的配置信息，才能算交互成功；配置信息交互是按照规定好的格式进行的，具体如下

客户端请求消息

GET/iclock/cpos?SN=${SerialNumber}&options=all&fireVer=${XXX}&build=0001&ip=${ServerIP}&language=${XXX}&devicename=${DeviceName}&device_type=30&IsConsumeFunType=${XXX}&pushver=${XXX}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

……

注释：

HTTP请求方法使用：GET方法

URI使用：/iclock/cpos

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

options：${Required}表示获取服务器配置参数，目前值只有all

pushver：${Optional}表示设备当前最新的push协议版本，新开发的客户端必须支持且必须大于等于4.0.1 版本。

language：${Optional}表示客户端支持的语言。详细见附录2

IsConsumeFunType：设备支持的功能：

第0位，为1，表示支持消费补贴一体，为0，表示不支持

第1位，为1，表示支持双钱包功能，为0，表示不支持

……

Host头域：${Required}

其他头域：${Optional}

服务器正常响应

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

GET OPTION FROM:
${SerialNumber}${LF}OpStamp=${XXX}${LF}PosStamp=${XXX}${LF}FullStamp=${XXX}${LF}AllowStamp=${XXX}${LF}ZhiWeiLogStamp=${XXX}${LF}ZhiWeiLogStampId=${XXX}${LF}ZhiWeiBakLogStampId=${XXX}${LF}ErrorDelay=${XXX}${LF}ServerVer=${XXX}${LF}Delay=${XXX}${LF}TransTimes=${XXX}${LF}TransInterval=${XXX}${LF}Realtime=${XXX}${LF}Encrypt=${XXX}${LF}Timeout=${XXX}${LF}SyncTime=${XXX}${LF}TableNameStamp=YYYY-MM-DDThh:mm:ss${LF}MachineType=${XXX}${LF}UseSection=${XXX}${LF}BackSection=${XXX}${LF}CardPass=${XXX}${LF}UseStampId=${XXX}${LF}PosLogStampId=${XXX}${LF}FullLogStampId=${XXX}${LF}AllowLogStampId=${XXX}${LF}PosBakLogStampId=${XXX}${LF}FullBakLogStampId=${XXX}${LF}AllowBakLogStampId=${XXX}${LF}

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Date头域：${Required}使用该头域来同步服务器时间，并且时间格式使用GMT格式，如Date: Fri, 03 Jul 2015 06:53:01 GMT

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述

服务器端配置信息：

第1行必须为该描述：GET OPTION FROM: ${SerialNumber}并且使用${LF}间隔配置信息，

其中${SerialNumber}为客户端发起请求的系列号，配置信息是使用键值对的形式（key=value）并且不同配置之间使用${LF}间隔

参数说明：

OpStamp：为设备最后上传人员数据的最新操作记录时间戳标记

PosLogStamp：为设备最后上传消费记录的记录时间戳标记

FullLogStamp：为设备最后上传充值记录的记录时间戳标记

ZhiWeiLogStamp：为设备最后上传支付宝微信充值记录的记录时间戳标记

AllowLogStamp：为设备最后上传补贴记录的记录时间戳标记

ErrorDelay：为联网失败后重新联接服务器的间隔时间（秒）

Delay：为正常联网时联接服务器的间隔时间（秒）

TransTimes：为定时检查并传送新数据时间（时:分，24小时格式），多个时间用分号分开，最多支持10个时间

TransInterval：为检查并传送新数据间隔时间（分钟）

TransFlag：为指示设备向服务器传送哪些数据的标识,最大字符长度为90(默认所有记录无选择，些字段保留后用)

Realtime：是否实时传送新记录。 为1表示有新数据就传送到服务器，为0表示按照 TransTimes 和 TransInterval：规定的时间传送

Encrypt：是否加密传送数据（加密传送使用中控专门的加密算法），请返回0

Timeout http：连接超时时间，(单位:秒),缺省60秒

SyncTime：同步时间间隔时间，（单位：秒）缺省0表示不自动同步时间

ServerVer：服务器版本号及时间（时间格式待定，旧版协议支持该参数）

TableNameStamp：自动上传数据时间日戳。TableName相应数据表名，与固件注册的数据表命名保持一致，Stamp为固定标志; 所有自动上传数据表的时间戳需返回给设备，采用如下形式，每个数据表的时间戳一行：

MachineType：机器类型 0消费机， 1充值机， 2补贴机

UseSection：使用主扇区

BackSection：使用备份扇区

CardPass：卡片密码

UseStampId：使用机器流水号上传记录

UseStampId：启动机器流水号标记上传消费记录(直接设置UseStampId=1)

PosLogStampId：为设备最后上传消费记录的记录戳标记

FullLogStampId：为设备最后上传充值记录的记录戳标记

ZhiWeiLogStampId：为设备最后上传支付宝微信充值记录的记录戳标记

AllowLogStampId：为设备最后上传补贴记录的记录戳标记

PosBakLogStampId：为设备最后上传消费备份记录的记录戳标记

FullBakLogStampId：为设备最后上传充值备份记录的记录戳标记

ZhiWeiBakLogStampId：为设备最后上传支付宝微信充值备份记录的记录戳标记

AllowBakLogStampId：为设备最后上传补贴备份记录的记录戳标记

示例

客户端请求：

GET
/iclock/cpos?SN=5158181900025&options=all&fireVer=Ver_2020.0.0.2-20181011&build=0001&ip=192.168.1.201&language=83&devicename=CM102&device_type=30&IsConsumeFunType=3&pushver=4.0.1 HTTP/1.1
Host: 192.168.1.201:8088
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*

服务器响应：
Date: Fri, 02 Nov 2018 09:05:36 GMT
Server: Apache/2.4.29 (Win64) mod_wsgi/4.5.24 Python/2.7
Content-Length: 515
Content-Language: zh-cn
Vary: Accept-Language,Cookie
Pragma: no-cache
Cache-Control: no-store
Connection: close
Content-Type: text/plain

GET OPTION FROM: 5158181900025
Stamp=0
OpStamp=0
ZhiWeiLogStamp=0
ZhiWeiLogStampId=0
ZhiWeiBakLogStampId=0
PosLogStamp=2018-11-02T10:16:58
FullLogStamp=0
AllowLogStamp=0
PosLogStampId=36
FullLogStampId=0
AllowLogStampId=0
PosBakLogStampId=2018-11-02T09:47:44
FullBakLogStampId=0
AllowBakLogStampId=0
UseStampId=1

TableNameStamp=0

PhotoStamp=0

ErrorDelay=60

Delay=30

TransTimes=00:00;14:05

TransInterval=1

TransFlag=1111101011

SyncTime=600

AllowOnline=0

MachineType=0

CustomSector=1

TimeZone=8

Realtime=1

Encrypt=0

7 交换公钥（支持通信加密的场合）

该功能设备推送设备公钥，接收服务器返回的服务器公钥。

客户端请求消息

POST /iclock/exchange?SN=$(SerialNumber)&type=publickey

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

PublicKey=${XXX}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/exchange

HTTP协议版本使用：1.1

Host头域：${Required}

其他头域：${Optional}

PublicKey：调用加密库返回的设备公钥。

服务器正常响应消息：

HTTP/1.1 200 OK

Server: ${XXX}

Set-Cookie: ${XXX}; Path=/; HttpOnly

Content-Type: application/push;charset=UTF-8

Content-Length: ${XXX}

Date: ${XXX}

PublicKey=${XXX}

注释：

PublicKey：服务器返回的服务器公钥。

8 交换因子（支持通信加密的场合）

该功能设备推送设备因子，接收服务器返回的服务器因子。

客户端请求消息

POST /iclock/exchange?SN=$(SerialNumber)&type=factors

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

Factors=${XXX}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/exchange

HTTP协议版本使用：1.1

Host头域：${Required}

其他头域：${Optional}

Factors：调用加密库返回的设备因子。

服务器正常响应消息：

HTTP/1.1 200 OK

Server: ${XXX}

Set-Cookie: ${XXX}; Path=/; HttpOnly
Content-Type: application/push;charset=UTF-8
Content-Length: ${XXX}
Date: ${XXX}

Factors=${XXX}

注释：

Factors：服务器返回的服务器因子。

9 获取时间请求

机器发送下列请求，从服务器获取当前服务器的时间:

GET /iclock/cpos?SN=xxxxxx&type=time

服务器返回数据：

Time=YYYY-MM-DDThh:mm:ss　hh:mm

（如：Time=2011-4-25T17:40:20+08:00）

10 上传数据

消费目前支持自动上传的数据包括：消费记录，充值记录，补贴记录，键值消费明细，商品消费明细等。

10.1上传方式

实时上传

间隔上传

定时上传

实时\间隔\定时三种上传方式，若支持实时，则间隔\定时方式不起作用。

实时上传，设备本身默认支持，服务器是可以控制的(详细见“[初始化信息交互]”的“Realtime”参数)

间隔上传，具体的间隔时间服务器是可以控制的(详细见“[初始化信息交互]”的“TransInterval”参数)

定时上传，具体的上传时间点服务器是可以控制的(详细见“[初始化信息交互]”的“TransTimes”参数)

10.2 上传消费记录

客户端请求消息

POST

/iclock/cpos?SN=${SerialNumber}&table=BUYLOG&Stamp=${timeStamp}&StampId=${RecNo}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

${DataRecord}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/cpos

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

table=BUYLOG：表示上传的数据为消费记录

Stamp：${timeStamp}表示消费记录上传到服务器的最新时间戳，格式为：YYYY-MM-DDThh:mm:ss　hh:mm （2018-12-20T17:20:13）

StampId：${RecNo}表示上传的消费记录的最大流水号

Host头域 ${Required}

Content-Length头域 ${Required}

请求实体：${DataRecord}，消费记录数据，数据格式如下

单钱包：

${SysID}${HT}${CARDNO}${HT}${PosTime}${HT}${PosMoney}${HT}${Balance}${HT}${CardRecID}${HT}${State}${HT}${MealType}${HT}${MealDate}&{HT}${RecNo}${HT}${OPID}

${SysID}：系统ID

${CardNo}：卡号

${PosTime}：消费时间

${PosMoney}：消费金额

${Balance}：余额

${CardRecID}：卡流水号

${State}：消费类型

${MealType}：餐别

${MealDate}：记餐日期

${RecNo}：机器流水号

${OPID}：操作员ID

双钱包：

${SysID}${HT}${CardNo}${HT}${PosTime}${HT}${PosMoney}${HT}${Balance}${HT}${CardRecID}${HT}${State}${HT}${MealType}${HT}${MealDate}&{HT}${RecNo}${HT}${OPID}${SubPosMoney}${SubBalance}${User_PIN}

${SysID}：系统ID

${CardNo}：卡号

${PosTime}：消费时间

${PosMoney}：消费金额

${Balance}：余额

${CardRecID}：卡流水号

${State}：消费类型

${MealType}：餐别

${MealDate}：记餐日期

${RecNo}：机器流水号

${OPID}：操作员ID

${SubPosMoney}：子钱包消费金额

${SubBalance}：子钱包余额

注：

${PosTime}：消费时间，类型为int，单位为秒（s）

多条记录之间使用${LF}连接

服务器正常响应消息

HTTP/1.1 200 OK

Content-Length: ${XXX}

……

OK

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述。

响应实体：当服务器接收数据正常并处理成功时回复OK，当出错时，回复错误描述即可。

10.3 上传商品消费明细记录

POST /iclock/cpos?SN=${SerialNumber}&table=storedetail
HTTP/1.1
Host: ${ServerIP}:${ServerPort}
Content-Length: ${XXX}
……

${DataRecord}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/cpos

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

table=storedetail：表示上传商品消费明细记录

Host头域：

Content-Length头域：

请求实体：${DataRecord}，商品消费明细数据，数据格式如下

${RecNo}${HT}${StoreNo}${HT}${Price}${HT}${RecSum}

注：

${RecNo}：对应消费记录流水号

${StoreNo}：商品编号

${Price}：商品价格

${RecSum}：本次消费流水

服务器正常响应消息

HTTP/1.1 200 OK

Content-Length: ${XXX}

……

OK

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述。

响应实体：当服务器接收数据正常并处理成功时回复OK，当出错时，回复错误描述即可。

10.4 上传键值消费明细记录

客户端请求消息

POST /iclock/cpos?SN=${SerialNumber}&table=keydetail

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

${DataRecord}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/cpos

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

table=keydetail：表示键值消费明细记录

Host头域：${Required}

Content-Length头域：${Required}

其他头域：${Optional}

请求实体：${DataRecord}，操作记录数据，数据格式如下

${RecNo}${HT}${KeyID}${HT}${Price}${HT}${RecSum}

${RecNo}：对应消费记录流水号

${KeyID}：键值编号

${Price}：商品价格

${RecSum}：本次消费流水

注：

多条记录之间使用${LF}连接

服务器正常响应消息

HTTP/1.1 200 OK

Content-Length: ${XXX}

……

OK

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述。

响应实体：当服务器接收数据正常并处理成功时回复OK，当出错时，回复错误描述即可。

10.5 上传充值记录

客户端请求消息

POST

/iclock/cpos?SN=${SerialNumber}&table=FULLLOG&Stamp=${TimeStamp}&StampId=${RecNo}

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

${DataRecord}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/cdata

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

table=FULLLOG：表示上传记录为充值记录

Stamp：${TimeStamp}表示充值记录上传到服务器的最新时间戳，格式为：YYYY-MM-DDThh:mm:ss　hh:mm （2018-12-20T17:20:13）

Host头域：${Required}

Content-Length头域：${Required}

其他头域：${Optional}

请求实体：${DataRecord}，充值记录数据，数据格式如下

${SysID}${HT}${CARDNO}${HT}${CardRecNo}${HT}${Suptime}${HT}${Money}${HT}${balance}${HT}${LogType}${HT}${OPID}${HT}${RecNo}

${SysID}：系统ID

${CardNo}：卡号

${CardRecNO}：卡流水号

${Suptime}：充值时间

${Money}：充值金额

${Balance}：余额

${LogType}：出纳类型(0充值记录 1退款　2优惠记录)

${RecNo}：机器流水号

${OPID}：操作员ID

多条记录之间使用${LF}连接

服务器正常响应消息

HTTP/1.1 200 OK

Content-Length: ${XXX}

……

OK

注释：

HTTP状态行：使用标准的HTTP协议定义。

HTTP响应头域：

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述。

响应实体：当服务器接收数据正常并处理成功时回复OK，当出错时，回复错误描述即可。

10.6 上传补贴记录

客户端请求消息

POST

/iclock/cpos?SN=${SerialNumber}&table=ALLOWLOG&Stamp={TimeStamp}&StampId=${RecNo}

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

${DataRecord}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/cpos

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

table=ALLOWLOG：表示上传记录为补贴记录

Stamp：${TimeStamp}表示上传补贴记录到服务器的最新时间戳

Host头域：${Required}

Content-Length头域：${Required}

其他头域：${Optional}

请求实体：${DataRecord}，补贴数据，数据格式如下

${SysID}${HT}${CARDNO}${HT}${CardRecID}${HT}${Batch}${HT}${AllowTime}${HT}${AllowMoney}${HT}${Balance}${HT}${state}${HT}${BaseBatch}${HT}${RecNo}

${SysID}：系统ID

${CARDNO}：卡号

${CardRecID}：卡流水号

${Batch}：补贴批号

${AllowTime}：补贴时间

${AllowMoney}：补贴金额

${Balance}：余额

${state}：补贴类型

${BaseBatch}：补贴基次

${RecNo}：机器流水号

多条记录之间使用${LF}连接。

服务器正常响应消息

HTTP/1.1 200 OK

Content-Length: ${XXX}

……

OK

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述。

响应实体：当服务器接收数据正常并处理成功时回复OK，当出错时，回复错误描述即可。

11 获取命令

如果服务器需要对设备进行操作，需先生成命令格式，等待设备发起请求时，再将命令发送到设备，对于命令的执行结果见[回复命令]

服务器下发命令并不会立即传送到设备，而是先把要向设备下发的命令缓存起来。根据配置，设备每隔一段时间（可配置）会向服务器发送如下请求，查询服务器是否有给自己的命令。

客户端请求消息

GET /iclock/posrequest?SN=${SerialNumber}&type=${machineType}&devno=${RecNo}
Host: ${ServerIP}:${ServerPort}
……

注释：

HTTP请求方法使用：GET方法

URI使用：/iclock/getrequest

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

type：${machineType}表示设备类别，type=pos为消费记录, type=full为充值记录, type=allow为补贴记录

devno：${RecNo}为当前最大记录流水号

Host头域：${Required}

其他头域：${Optional}

服务器正常响应消息

当无命令时，回复信息如下：

HTTP/1.1 200 OK
Date: ${XXX}
Content-Length: 2
……

OK

当有命令时，回复信息如下：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

${CmdRecord}

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Date头域：${Required}使用该头域来同步服务器时间，并且时间格式使用GMT格式，如Date: Fri, 03 Jul 2015 06:53:01 GMT

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述。

响应实体：${CmdRecord}，下发的命令记录，数据格式如下

C:${CmdID}:${CmdDesc}${SP}${XXX}

注：

${CmdID}：该命令编号是由服务器随机生成的，支持数字、字母，长度不超过16，客户端回复命令时需带上该命令编号，详细见下面“回复命令”功能。

${CmdDesc}：命令描述分为数据命令和控制命令，详见具体命令。

多条记录之间使用${LF}连接。

11.1 SHELL命令

执行操作系统命令，格式如下：

C:${CmdID}:SHELL${SP}${Shell_String}

命令返回：详见[回复命令]

11.2 CLEAR命令

清楚数据，命令格式如下：

格式1：清除消费记录

C:${CmdID}:CLEAR${SP}POSLOG

格式2：清除充值记录

C:${CmdID}:CLEAR${SP}FULLVALUE

格式3：清除补贴记录

C:${CmdID}:CLEAR${SP}SIDYLOG

格式4：清楚全部数据：

C:${CmdID}:CLEAR${SP}DATA

格式5：先备份消费记录，然后按日期删除日期以前的消费记录

C:${CmdID}:CLEAR${SP}ONRESPOSLOG${SP}DATE=${PosTime}

说明：

DATE：${PosTime}表示消费时间，格式为YYMMDD。例如20181201。

目前该命令只支持清楚，暂时不支持备份

。

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=CLEAR

11.3 INFO命令

发送机器的信息到服务器上，具体命令格式如下：

C:${CmdID}:INFO

说明：

返回服务器参数包括：

⑴黑名单数量：BlackUserCount

⑵白名单数量：WhiteUserCount

⑶指纹数量：FPCount

⑷内存大小：FlashSize

⑸剩余内存大小：FreeFlashSize

⑹Push版本号：PUSHVersion

⑺消费记录数量：PosLog

⑻充值记录数量：FullLog

⑼补贴记录数量：AllowLog

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=INFO

11.4 设置设备选项

命令格式如下

C:${CmdID}:SET${SP}OPTION${SP}ITEM=${Value}

说明：

ITEM 为选项的内容，VALUE为选项的值。例如：

SET OPTION IPAddress=192.168.1.225

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=SET OPTION

11.5 REBOOT命令

设备重启，命令格式如下

C:${CmdID}:REBOOT

注意：

若服务器一次返回设备多条命令，该命令必须是的最后一条，否则其后的其他命令将会被忽略。

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=REBOOT

11.6 数据命令

数据命令包括对数据表的增、删、改、查，具体命令格式如下：

11.6.1 UPDATE命令

命令格式如下

C:${CmdID}:UPDATE${SP}${TableName}${SP}${Value}

支持数据表以及下发格式如下表：

表名	数据格式
USERINFO	UPDATE USERINFO SysID=XXX\tUSerID=XXX\tPIN=XXX\tCardNo=XXX\tName=XXX\tPassword=XXX\tSegNo=XXX\tUserType=XXX\tPrivage=XXX\n
PRESSKEY	UPDATE PRESSKEY KeyID=XXX\tPrice=XXX\n
STOREINFO	UPDATE STOREINFO StoreNo=XXX\tName=XXX\tBc=XXX\tPrice=XXX\tagio=XXX\n
MEALTYPE	UPDATE MEALTYPE Mlid=XXX\tName=XXX\tStart=XXX\tEnd=XXX\n
TIMESEG	UPDATE TIMESEG SegID=XXX\tTsID=XXX\tStart=XXX\tEnd=XXX\n
FIXED	UPDATE FIXD TsID=XXX\tStart=XXX\tEnd=XXX\tPrice=XXX\n
CARDTYPE	UPDATE CARDTYPE SortID=XXX\tName=XXX\trebate=XXX\tTimemaxmoney=XXX\tDaymaxmoney=XXX\tMealmaxmoney=XXX\tDaymaxtimes=XXX\tMealmaxtimes=XXX\tLowlimit=XXX\tMaxlimit=XXX\tMealType=XXX\tEnable=XXX\tlimit=XXX\tBatchNo=XXX\tUseFinger=XXX\n
SUBSIDYLOG	UPDATE SUBSIDYLOG SysID=XXX\tCardNo=XXX\tBatch=XXX\tMoney=XXX\tallowDate=XXX\tdowndate=XXX\n
SUBSIDYLOG（双钱包）	UPDATE SUBSIDYLOG SysID=XXX\tCardNo=XXX\tBatch=XXX\tMoney=XXX\tallowDate=XXX\tdowndate=XXX\tStartDate\tClearFlag\tEndFlag\n

多条记录之间使用${LF}连接。

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=UPDATE

11.6.2 DELETE命令

命令格式如下

C:${CmdID}:DELETE${SP}${TableName}${SP}${Cond}

说明：

${TableName}：表示要删除表的表名

${Cond}：表示删除数据条件

多条记录之间使用${LF}连接。

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=DELETE

11.6.3 QUERY命令

命令格式如下

C:${CmdID}:QUERY${SP}${TableName}${SP}${Cond}

说明：

${TableName}：要查询数据表的表名

${Cond}：查询条件。如果未指定查询条件则删除所有数据

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=QUERY

11.7 CHECK命令

检查并传送新数据。

命令格式如下

格式1：要求设备立即从服务器更新参数设置。

C:${CmdID}:CHECK${SP}OPTION

说明：

设备收取到该命令会发送初始化连接请求到服务器，重新获取服务器参数。详见[初始化信息交互]

格式2：要求设备立即检查是否有新的消费数据，并立即把新数据传送到服务器上。

C:${CmdID}:CHECK${SP}POSLOG

格式3：要求设备立即检查是否有新的充值数据，并立即把新数据传送到服务器上。

C:${CmdID}:CHECK${SP}FULLLOG

格式4：要求设备立即检查是否有新的补贴数据，并立即把新数据传送到服务器上。

C:${CmdID}:CHECK${SP}ALLOWLOG

格式5：要求设备立即从服务器更新参数设置，并立即把所有新数据传送到服务器上。

C:${CmdID}:CHECK${SP}ALL

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=CHECK

11.8 SYNC命令

按照指定记录流水号上传记录。

命令格式如下：

SYNC${SP}POSLOG${SP}START={RecNO}${SP}END${RecNO}

格式2：按机器流水号同步充值记录

SYNC${SP}FULLLOG${SP}START={RecNO}${SP}END${RecNO}

格式3：按机器流水号同步补贴记录

SYNC${SP}ALLOWLOG${SP}START={RecNO}${SP}END${RecNO}

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=SYNC

11.9 COLL命令

按日期上传记录命令

命令格式如下：

格式1：按日期上传机器内的消费记录

COLL${SP}POSLOG${SP}START={DATE}${SP}END${DATE}${SP}CARDNO={No}

格式2：按日期上传机器内的充值记录

COLL${SP}FULLLOG${SP}START={DATE}${SP}END${DATE}${SP}CARDNO={No}

格式3：按日期上传机器内的补贴记录

COLL${SP}ALLOWLOG${SP}START={DATE}${SP}END${DATE}${SP}CARDNO={No}

说明：

CARDNO=0，上传此时间段内的所有记录。

CARDNO非零，上传此时间段内此卡号的所有记录。

DATE格式：Y-m-d H:M:S。例如：2018-12-24 12:33:01

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=COLL

11.10 GetFile命令

获取设备文件。

命令格式如下

C:${CmdID}:GetFile${SP}${FilePath}

说明：

设备发送系统文件到服务器

${FilePath}为系统文件路径

客户端发送：

POST /iclock/posdevicecmd?SN=${SerialNumber}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

${DataRecord}

注释：

HTTP请求方法使用：POST方法

URI使用：/iclock/cpos

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

Host头域 ${Required}

Content-Length头域 ${Required}

请求实体：${DataRecord}，设备返回数据，数据格式如下：

ID=${CmdID}${LF}SN=${SerialNumber}${LF}FILENAME=${fileName}${LF}CMD=GetFile${LF}Return=${Value}${LF}Content=${FileValue}

说明：

ID：命令id

SN：${SerialNumber}设备序列号

FILENAME：${fileName}文件名称

CMD：命令名

Return：${Value}命令返回值

Content：${FileValue}为文件内容，可以使多行文本，也可以是二进制内容

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=GetFile。

11.11 PutFile命令

发送文件到设备。要求设备下载服务器上的文件，并保存到FilePath指定的文件中（如果是tgz文件，下载后将自动解压到FilePath指定目录，未指定目录则解压到/mnt/mtdblock目录，其他格式文件需指定文件保存路径及文件名）。该文件必须由服务器以HTTP方式提供，并给出获取该文件的URL。

如果URL以"http://"开头，设备将把URL看成是完整的URL地址，否则，设备将把本服务器的/iclock/地址附加到指定的URL上。例如：

PutFile file/fw/X938/main.tgz main.tgz或PutFile file/fw/X938/main.tgz

将要求设备下载 http://server/iclock/file/fw/X938/main.tgz并解压缩main.tgz到/mnt/mtdblock文件夹中。 PutFile file/fw/X938/main.tgz /mnt/将要求设备下载 http://server/iclock/file/fw/X938/main.tgz并解压缩main.tgz到/mnt/文件夹中。PutFile file/fw/X938/ssruser.dat /mnt/mtdblock/ssruser.dat

将要求设备下载 http://server/iclock/file/fw/X938/ssruser.dat并保持为/mnt/mtdblock/ssruser.dat文件。

命令格式如下

C:${CmdID}:PutFile${SP}${URL}${SP}${FilePath}

说明：

${URL}：服务器上需要下载的文件地址

${FilePath}：文件存入客户端的目标路径

命令执行结果如何回复见[回复命令]功能，Return返回值见（附录1），返回内容格式如下：

ID=${XXX}&Return=${XXX}&CMD=PutFile

12 挂失、解挂、改密码

12.1挂失

客户端请求格式：

Get /iclock/posrequest?SN=${SerialNumber}&RepType=2${XXX}&Card=${XXX}&Pwd=${XXX}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

……

成功服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

OK

失败服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

NO

12.2解挂

客户端请求格式：

Get /iclock/posrequest?SN=${SerialNumber}&RepType=3${XXX}&Card=${XXX}&Pwd=${XXX}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

……

成功服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

OK

失败服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

NO

12.3请求修改密码

客户端请求修改密码格式：是否可以改密码， 服务器须验证卡号是否合法，旧密码是否正确

Get

/iclock/posrequest?SN=${SerialNumber}&RepType=4${XXX}&Card=${XXX}&OldPwd=${XXX}&NewPwd=${XXX}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

……

成功服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

OK

失败服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

NO

客户端改密完成格式：可以改密码后，终端修改卡片密码，修改完成后通知服务器可以将数据库中密码修改。

Get

/iclock/posrequest?SN=${SerialNumber}&RepType=5${XXX}&Card=${XXX}&OldPwd=${XXX}&NewPwd=${XXX}

HTTP/1.1

Host: ${ServerIP}:${ServerPort}

……

成功服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

OK

失败服务器响应：

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: ${XXX}

……

NO

13 命令回复

客户端在[获取到服务器下发的命令]后，需要对相应的命令进行回复

客户端请求消息

POST /iclock/posdevicecmd?SN=${SerialNumber}

Host: ${ServerIP}:${ServerPort}

Content-Length: ${XXX}

……

${CmdRecord}

注释：

HTTP请求方法使用：GET方法

URI使用：/iclock/devicecmd

HTTP协议版本使用：1.1

客户端配置信息：

SN：${Required}表示客户端的序列号

Host头域：${Required}

Content-Length头域：${Required}

其他头域：${Optional}

响应实体：${CmdRecord}，回复的命令记录，回复的内容都会包含ID\Return\CMD信息，含义如下

ID：服务器下发命令的命令编号

Return：客户端执行命令之后的返回结果

CMD：服务器下发命令的命令描述

少部分回复会包含其他信息，具体回复内容格式请看各个命令的说明

多条命令回复记录之间使用${LF}连接

服务器正常响应消息

HTTP/1.1 200 OK

Date: ${XXX}

Content-Length: 2

……

OK

注释：

HTTP状态行：使用标准的HTTP协议定义

HTTP响应头域：

Date头域：${Required}使用该头域来同步服务器时间，并且时间格式使用GMT格式，如Date: Fri, 03 Jul 2015 06:53:01 GMT

Content-Length头域：根据HTTP 1.1协议，一般使用该头域指定响应实体的数据长度，如果是在不确定响应实体的大小时，也支持Transfer-Encoding: chunked，Content-Length及Transfer-Encoding头域均是HTTP协议的标准定义，这里就不在详述

示例

客户端请求：

```
POST /iclock/posdevicecmd?SN=0316144680030 HTTP/1.1
Host: 58.250.50.81:8011
User-Agent: iClock Proxy/1.09
Connection: close
Accept: */*
Content-Length: 143

ID=info8487&Return=0&CMD=DATA
ID=info8488&Return=0&CMD=DATA
ID=info8489&Return=0&CMD=DATA
ID=info7464&Return=0&CMD=DATA
ID=fp7464&Return=0&CMD=DATA
```

服务器响应：

```
HTTP/1.1 200 OK
Server: nginx/1.6.0
Date: Tue, 30 Jun 2015 01:24:48 GMT
Content-Type: text/plain
Content-Length: 2
Connection: close
Pragma: no-cache
Cache-Control: no-store
.....
OK
```

14 附录

14.1 附录1

通用错误码	描述
0	成功
-1	参数错误
-2	传输用户照片数据与给定的Size不匹配
-3	读写错误
-1001	容量限制
-1002	设备不支持
-1003	命令执行超时
-1004	数据与设备配置不一致
-1005	设备忙
-1006	数据太长
-1007	内存错误
-1008	获取服务器数据失败

14.2 附录2

语言编号	意义
83	简体中文
69	英文
97	西班牙语
70	法语
66	阿拉伯语
80	葡萄牙语
82	俄语
71	德语
65	波斯语
76	泰语
73	印尼语
74	日本语
75	韩语
86	越南语
116	土耳其语
72	希伯来语
90	捷克语
68	荷兰语
105	意大利语
89	斯洛伐克语
103	希腊语
112	波兰语
84	繁体

14.3 附录3

协议版本规则

- **已发布的协议版本：**

```
2.2.14

2.3.0

加密协议版本：2.4.0及以上
```

- **设备端：**

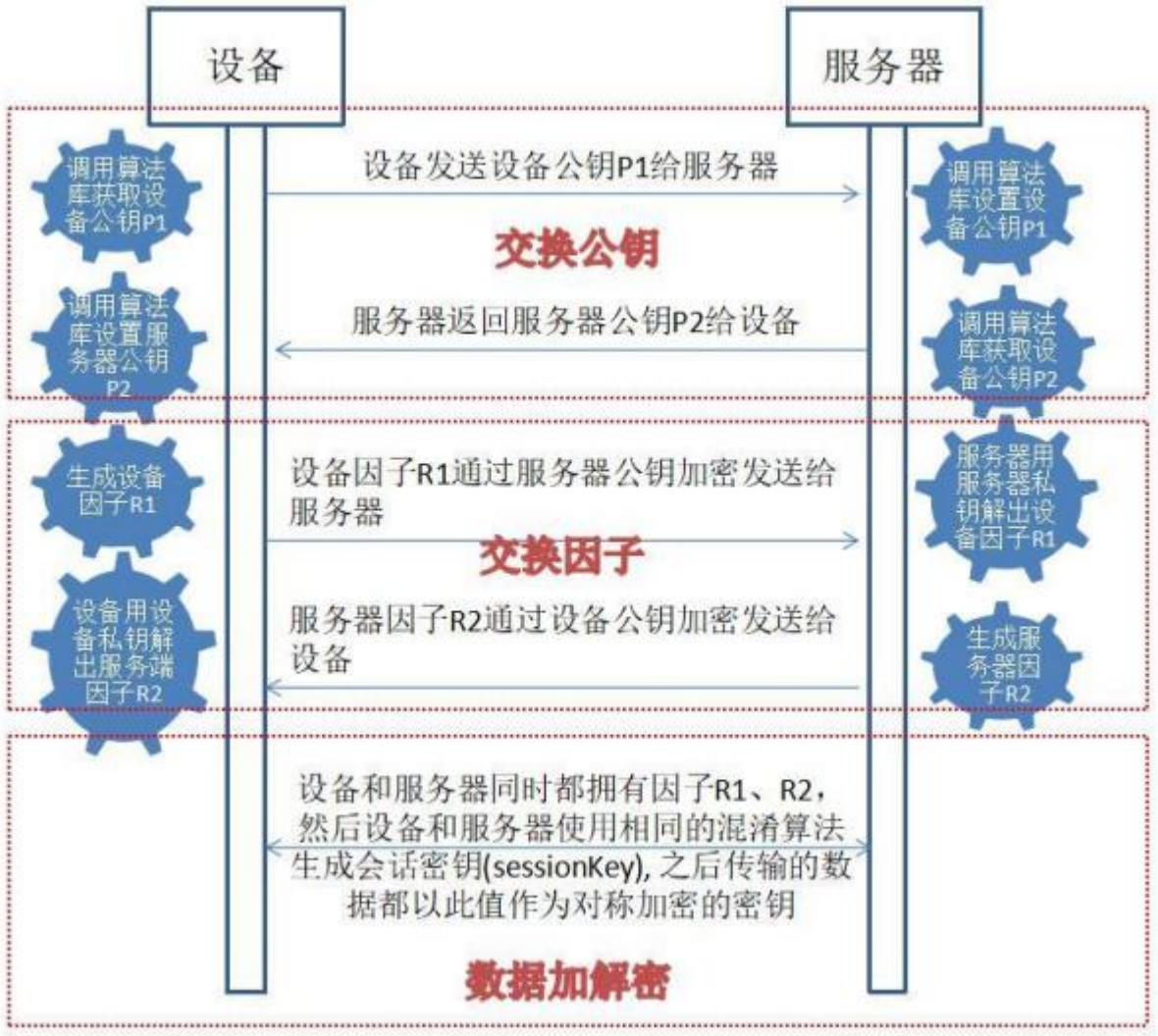
```
设备将当前push使用的协议版本通过下面协议推送给服务端

GET
/iclock/cpos?SN=${SerialNumber}&options=all&pushver=${XXX}&language=${XXX}&pushcommkey=${XXX}

服务端针对这个请求返回服务端使用哪个发布协议版本开发，将协议版本返回给设备。

PushProtVer=xxx，没返回这个参数的话，设备默认服务器使用的协议版本为2.2.14

设备根据当前push使用的协议版本与服务端返回的协议版本比较，使用较低的那个版本交互。
```

- **服务端：**

```
服务端根据下面的请求获得设备端push使用的协议版本，如果没有pushver字段，那么默认设备使用的是2.2.14协议版本。

GET
/iclock/cpos?SN=${SerialNumber}&options=all&pushver=${XXX}&language=${XXX}&pushcommkey=${XXX}

服务端需要返回软件使用哪个发布协议版本：

PushProtVer=xxx

服务端根据软件使用的协议版本与设备端上传的协议版本比较，使用较低的那个版本交互。
```

14.4 附录4

数据加密秘钥交换方案



- 算法：加密算法库将统一进行封装，设备使用的算法库为静态库。
- 方案：

  a) 设备和服务器重连的时候初始化非对称加密的公私钥。

  b) 设备和服务器交换公钥：

  ■设备发送设备公钥 P1 给服务器。

  ■服务器返回服务器公钥 P2 给设备。

  ■完成公钥交换。设备和服务器同时都拥有公钥 P1、P2。

  c) 设备和服务器交换因子：

  ■设备生成因子 R1，并通过服务器公钥加密发送给服务端。

  ■服务器用服务器私钥解出设备因子 R1。

  ■服务器生成因子 R2，并通过设备公钥加密发送给设备。

  ■设备用设备私钥解出服务端因子 R2。

  ■完成因子交换。设备和服务器同时都拥有因子 R1、R2。

  d) 设备和服务器同时都拥有因子 R1、R2，然后设备和服务器使用相同的混淆算法生成会话密钥(sessionKey)，之后传输的数据都以此值作为对称加密的密钥。

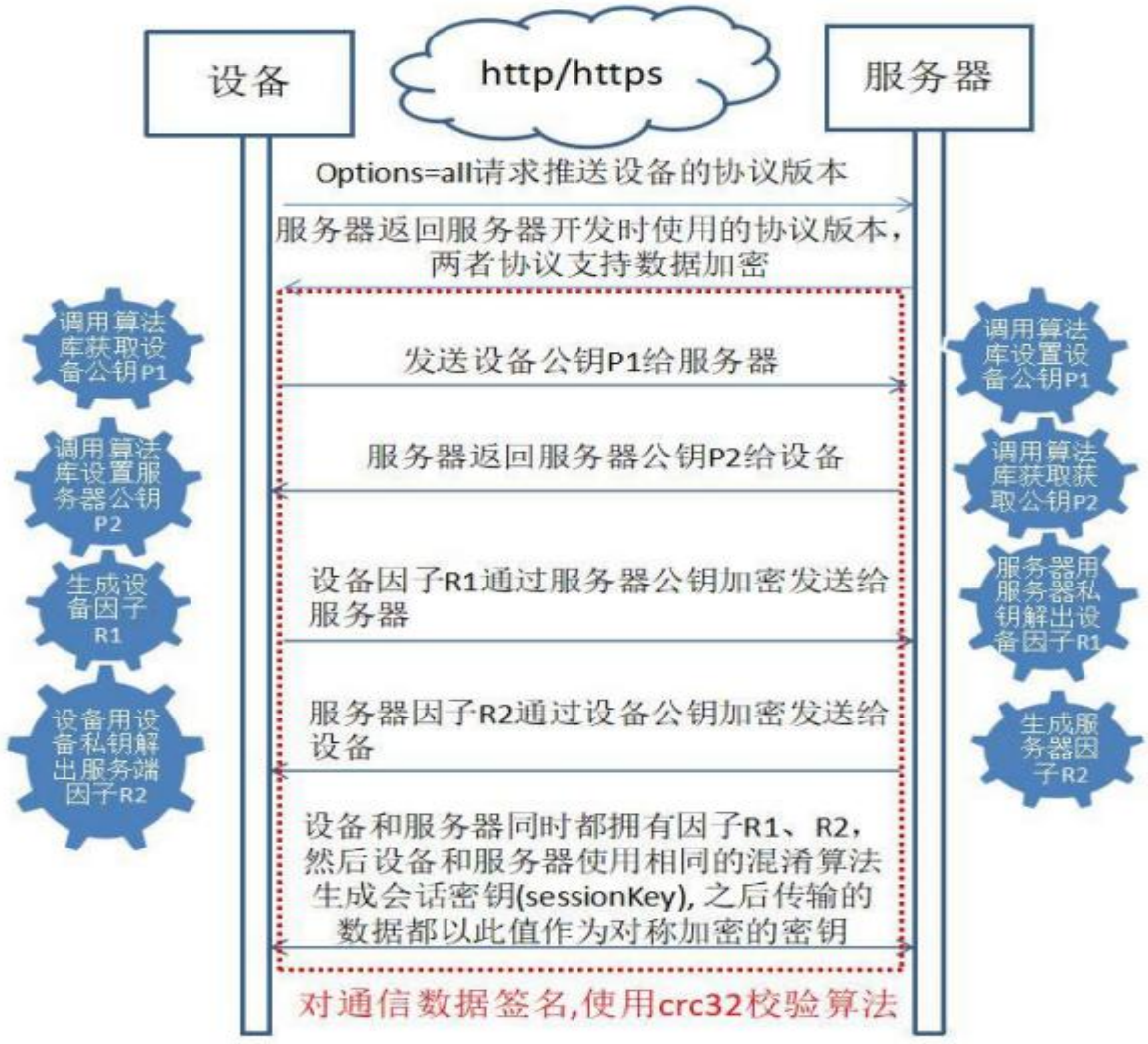
兼容方案

根据设备和服务器使用的协议版本实现兼容，情况如下：

- 情况一


设备
http/https
服务器
Options=all请求推送设备的推送协议版本或者没推送协议版本
服务器返回后者未返回服务器开发时使用的协议版本，两者协议没有都支持数据加密
对通信数据明码传输

- 情况二


设备
http/https
服务器
Options=all请求推送设备的协议版本
服务器返回服务器开发时使用的协议版本，两者协议支持数据加密
调用算法库获取设备公钥P1
发送设备公钥P1给服务器
调用算法库设置设备公钥P1
调用算法库设置服务器公钥P2
服务器返回服务器公钥P2给设备
调用算法库获取获取公钥P2
生成设备因子R1
设备因子R1通过服务器公钥加密发送给服务器
服务器用服务器私钥解出设备因子R1
设备用设备私钥解出服务端因子R2
服务器因子R2通过设备公钥加密发送给设备
生成服务器因子R2
设备和服务器同时都拥有因子R1、R2，然后设备和服务器使用相同的混淆算法生成会话密钥(sessionKey)，之后传输的数据都以此值作为对称加密的密钥
对通信数据签名,使用crc32校验算法

注释：

a) 设备根据设置的服务器地址来判断使用https还是http传输。

b) 设备现有的第一个请求协议头中增加pushver字段为设备当前通信协议版本号，软件返回的数据内容中增加PushProtVer表示软件是基于哪个协议版本开发的。设备和服务器两个协议版本比较取最低的，按最低协议版本进行通信。

情况一： 当服务器和设备的协议版本不都支持，则使用对数据通信进行明码传输。

情况二： 设定某个协议版本是支持数据加密的，当服务器和设备的协议版本都支持，则使用数据加密方案。

交互顺序如下：

■新增协议对设备和服务器的公钥 P1、P2 进行交换。

■新增协议对设备和服务器的因子 R1、R2 进行交换。

■对通信数据签名进行 crc32 校验，设备和服务器同时都拥有因子 R1、R2，然后设备和服务器使用相同的混淆算法生成会话密钥(sessionKey)，之后传输的数据都以此值作为对称加密的密钥。

全国免费技术咨询热线：4006-900-999
广东省东莞市塘厦平山188工业大道26号中控智慧产业园
广东省深圳市龙岗区坂田五和大道北中控智慧大厦
厦门市集美区软件园三期B02栋20层
www.zkteco.com