(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111277418 B
(45)授权公告日 2023.05.12

(21)申请号 202010097133.0
(22)申请日 2020.02.17
(65)同一申请的已公布的文献号
申请公布号 CN 111277418 A
(43)申请公布日 2020.06.12
(73)专利权人 福建天晴在线互动科技有限公司
地址 350212 福建省福州市长乐市湖南镇大鹤村
(72)发明人 刘德建 叶伟 郑彬 李正 岳万恕 陈宏展
(74)专利代理机构 福州旭辰知识产权代理事务所(普通合伙) 35233
专利代理师 程勇
(51)Int.Cl.
H04L 9/32(2006.01)
H04L 9/40(2022.01)
(56)对比文件
CN 110071806 A,2019.07.30
CN 107135073 A,2017.09.05
CN 108183907 A,2018.06.19
CN 107705088 A,2018.02.16
US 2018309746 A1,2018.10.25
审查员 赵勇达

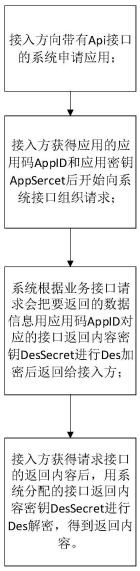
权利要求书2页 说明书5页 附图2页

(54)发明名称

一种实现Api接口安全性的方法

(57)摘要

本发明提供了一种实现Api接口安全性的方法,所述方法包括如下步骤:步骤1:接入方向带有Api接口的系统申请应用;步骤2:接入方获得应用的应用码AppID和应用密钥AppSercet后开始向系统接口组织请求;步骤3:系统根据业务接口请求会把要返回的数据信息用应用码AppID对应的接口返回内容密钥DesSecret进行Des加密后返回给接入方;步骤4:接入方获得请求接口的返回内容后,用系统分配的接口返回内容密钥DesSecret进行Des解密,得到返回内容。本发明有效的保证了接口的安全性。



CN 111277418 B

1.一种实现Api接口安全性的方法,其特征在于:所述方法包括如下步骤:

步骤1:接入方向带有Api接口的系统申请应用;

步骤2:接入方获得应用的应用码AppID和应用密钥AppSercet后开始向系统接口组织请求;该步骤2具体为:

步骤2.1:接入方要请求获取令牌接口来得到令牌Token;

所述步骤2.1进一步包括如下步骤:

步骤2.1.1:系统收到步骤2请求后,首先根据应用码AppID取得对应的应用密钥AppSercet,校验Sign签名是否合法有效;

步骤2.1.2:系统其次校验当前时间戳Ticket是否于服务器误差为一时间阈值范围内,超出范围的视为过期请求;

步骤2.1.3:最后系统校验请求IP是否在应用码AppID允许的IP白名单内;

步骤2.1.4:如果步骤2.1.1至步骤2.1.3都合法的情况下,系统组织令牌值返回给接入方;令牌Token是由应用码AppID、过期时间ExpireTime、随机字符串组成的Json字符串,再经过系统的DES加密得出的值,该值在返回给接入方之前会存入Redis缓存,用于校验合法性;

所述步骤2.1进一步具体为:接入方传输应用码AppID、当前时间戳Ticket、Sign请求获取令牌接口,所述Sign是由AppID+Ticket+AppSercet 3个参数拼接进行MD5加密后得到的值;所述令牌Token在一预设时间内有效,即在预设时间内无需重复请求令牌Token,令牌Token都视为有效;

步骤2.2:接入方获得令牌Token后有权向系统请求具体功能的业务接口,得到业务接口所需参数和签名规则组织出业务签名SdkSign;

步骤2.3:根据所述令牌Token和业务签名SdkSign,再加上业务接口所需参数,一并请求业务接口;

所述步骤2.3进一步包括如下步骤:

步骤2.3.1:系统收到所述业务接口请求后,首先解密令牌Token,并解析出包含的应用码AppID信息,如果解析失败,则视为非法请求;

步骤2.3.2:解析出应用码AppID后,根据应用码AppID得到Redis缓存key读取令牌Token,查看令牌Token是否存在,如果不存在,则视为非法请求;

步骤2.3.3:根据解析的应用码AppID获取AppSercet,根据签名规则,组织签名,比对SdkSign是否合法;

步骤2.3.4:校验请求IP是否在应用码AppID允许的IP白名单内;

步骤2.3.5:判断系统当前小时内是否有超量请求,即系统会运行一个计划任务,每个5分钟会统计一次每个应用接口系统当前小时的请求量,如果大于最大请求频率阀值,则存储进Redis缓存,代表已超量;

步骤2.3.6:如果步骤2.3.1至步骤2.3.5都通过的情况下,代表请求合法,进入步骤3;否则,为非法请求,结束流程;

步骤3:系统根据业务接口请求会把要返回的数据信息用应用码AppID对应的接口返回内容密钥DesSecret进行Des加密后返回给接入方;

步骤4:接入方获得请求接口的返回内容后,用系统分配的接口返回内容密钥

DesSecret进行Des解密,得到返回内容。

2.根据权利要求1所述的一种实现Api接口安全性的方法,其特征在于:所述签名规则组织出业务签名SdkSign具体为:根据统一签名规则,业务接口所需参数根据参数名ASCII从小到大排序,将业务接口所需参数进行组合一起得到text1,text1转化为小写获得text2,text2+AppSecret进行MD5加密后获得32位的MD5值,这个就是SdkSign。

一种实现Api接口安全性的方法

技术领域

[0001] 本发明涉及网络通讯技术领域,特别是一种实现Api接口安全性的方法。

背景技术

[0002] 现如今各种Api接口层出不穷,例如公司内部不同小组间的接口调用、对面用户的功能性接口、第三方服务接口等等。一个好的Api有很多维度来衡量,其中安全性和简易性是一个Api接口最基本也是最重要的两个维度,但是往往这2个维度会自相矛盾,设计的越复杂安全性相对会更高,但是对于接入方来说就是一件费劲的事件,太难接入不易看懂,就会产生很多无畏的沟通成本。设计的太简易也不行,接口有被篡改调用的风险,由其涉及Api充值、奖励发放、敏感数据等,安全与否直接影响到个人或企业的财产。所以设计一个好的Api接口,即简单又安全尤为重要。

发明内容

[0003] 为克服上述问题,本发明的目的是提供一种实现Api接口安全性的方法,能进有效的保证了接口的安全性。

[0004] 本发明采用以下方案实现:一种实现Api接口安全性的方法,所述方法包括如下步骤:

[0005] 步骤1:接入方向带有Api接口的系统申请应用;

[0006] 步骤2:接入方获得应用的应用码AppID和应用密钥AppSercet后开始向系统接口组织请求;该步骤2具体为:

[0007] 步骤2.1:接入方要请求获取令牌接口来得到令牌Token;

[0008] 步骤2.2:接入方获得令牌Token后有权向系统请求具体功能的业务接口,得到业务接口所需参数和签名规则组织出业务签名SdkSign;

[0009] 步骤2.3:根据所述令牌Token和业务签名SdkSign,再加上业务接口所需参数,一并请求业务接口;

[0010] 步骤3:系统根据业务接口请求会把要返回的数据信息用应用码AppID对应的接口返回内容密钥DesSecret进行Des加密后返回给接入方;

[0011] 步骤4:接入方获得请求接口的返回内容后,用系统分配的接口返回内容密钥DesSecret进行Des解密,得到返回内容。

[0012] 进一步的,所述步骤2.1进一步具体为:接入方传输应用码AppID、当前时间戳Ticket、Sign请求获取令牌接口,所述Sign是由AppID+Ticket+AppSercet 3个参数拼接进行MD5加密后得到的值;所述令牌Token在一预设时间内有效,即在预设时间内无需重复请求令牌Token,令牌Token都视为有效。

[0013] 进一步的,所述步骤2.1进一步包括如下步骤:

[0014] 步骤2.1.1:系统收到步骤2请求后,首先根据应用码AppID取得对应的应用密钥AppSercet,校验Sign签名是否合法有效;

[0015] 步骤2.1.2:系统其次校验Ticket是否于服务器误差为一时间阈值范围内,超出范围的视为过期请求;

[0016] 步骤2.1.3:最后系统校验请求IP是否在应用码AppID允许的IP白名单内;

[0017] 步骤2.1.4:如果步骤2.1.1至步骤2.1.3都合法的情况下,系统组织令牌值返回给接入方;令牌Token是由应用码AppID、过期时间ExpireTime、随机字符串组成的Json字符串,再经过系统的DES加密得出的值,该值在返回给接入方之前会存入Redis缓存,用于校验合法性。

[0018] 进一步的,所述签名规则组织出业务签名SdkSign具体为:根据统一签名规则,业务接口所需参数根据参数名ASCII从小到大排序,将业务接口所需参数进行组合一起得到text1,text1转化为小写获得text2,text2+AppSecret进行MD5加密后获得32位的MD5值,这个就是SdkSign。

[0019] 进一步的,所述步骤2.3进一步包括如下步骤:

[0020] 步骤2.3.1:系统收到所述业务接口请求后,首先解密令牌Token,并解析出包含的应用码AppID信息,如果解析失败,则视为非法请求;

[0021] 步骤2.3.2:解析出应用码AppID后,根据应用码AppID得到Redis缓存key读取令牌Token,查看令牌Token是否存在,如果不存在,则视为非法请求;

[0022] 步骤2.3.3:根据解析的应用码AppID获取AppSercet,根据签名规则,组织签名,比对SdkSign是否合法;

[0023] 步骤2.3.4:校验请求IP是否在应用码AppID允许的IP白名单内;

[0024] 步骤2.3.5:判断系统当前小时内是否有超量请求,即系统会运行一个计划任务,每个5分钟会统计一次每个应用接口系统当前小时的请求量,如果大于最大请求频率阀值,则存储进Redis缓存,代表已超量;

[0025] 步骤2.3.6:如果步骤2.3.1至步骤2.3.5都通过的情况下,代表请求合法,进入步骤3;否则,为非法请求,结束流程。

[0026] 本发明的有益效果在于:1、接入简易方便:提供统一签名规则,多语言通用的加密规则;2、接口安全性:运用账号令牌、业务参数签名、IP限制、接口权限、接口请求限量和返回信息加密等来保护接口安全。

附图说明

[0027] 图1是本发明的方法流程示意图。

[0028] 图2是本发明一实施例的方法具体流程框图。

具体实施方式

[0029] 下面结合附图对本发明做进一步说明。

[0030] 请参阅图1所示,本发明的一种实现Api接口安全性的方法,所述方法包括如下步骤:

[0031] 步骤1:接入方向带有Api接口的系统申请应用;

[0032] 步骤2:接入方获得应用的应用码AppID和应用密钥AppSercet后开始向系统接口组织请求;该步骤2具体为:

[0033] 步骤2.1:接入方要请求获取令牌接口来得到令牌Token;所述步骤2.1进一步具体为:接入方传输应用码AppID、当前时间戳Ticket、Sign请求获取令牌接口,所述Sign是由AppID+Ticket+AppSercet 3个参数拼接进行MD5加密后得到的值;所述令牌Token在一预设时间内有效,即在预设时间内无需重复请求令牌Token,令牌Token都视为有效。

[0034] 步骤2.2:接入方获得令牌Token后有权向系统请求具体功能的业务接口,得到业务接口所需参数和签名规则组织出业务签名SdkSign;所述签名规则组织出业务签名SdkSign具体为:根据统一签名规则,业务接口所需参数根据参数名ASCII从小到大排序,将业务接口所需参数进行组合一起得到text1,text1转化为小写获得text2,text2+AppSecret进行MD5加密后获得32位的MD5值,这个就是SdkSign。

[0035] 步骤2.3:根据所述令牌Token和业务签名SdkSign,再加上业务接口所需参数,一并请求业务接口;

[0036] 步骤3:系统根据业务接口请求会把要返回的数据信息用应用码AppID对应的接口返回内容密钥DesSecret进行Des加密后返回给接入方;

[0037] 步骤4:接入方获得请求接口的返回内容后,用系统分配的接口返回内容密钥DesSecret进行Des解密,得到返回内容。

[0038] 其中,所述步骤2.1进一步包括如下步骤:

[0039] 步骤2.1.1:系统收到步骤2请求后,首先根据应用码AppID取得对应的应用密钥AppSercet,校验Sign签名是否合法有效;

[0040] 步骤2.1.2:系统其次校验Ticket是否于服务器误差为一时间阈值范围内,超出范围的视为过期请求;

[0041] 步骤2.1.3:最后系统校验请求IP是否在应用码AppID允许的IP白名单内;

[0042] 步骤2.1.4:如果步骤2.1.1至步骤2.1.3都合法的情况下,系统组织令牌值返回给接入方;令牌Token是由应用码AppID、过期时间ExpireTime、随机字符串组成的Json字符串,再经过系统的DES加密得出的值,该值在返回给接入方之前会存入Redis缓存,用于校验合法性。

[0043] 进一步的,所述步骤2.3进一步包括如下步骤:

[0044] 步骤2.3.1:系统收到所述业务接口请求后,首先解密令牌Token,并解析出包含的应用码AppID信息,如果解析失败,则视为非法请求;

[0045] 步骤2.3.2:解析出应用码AppID后,根据应用码AppID得到Redis缓存key读取令牌Token,查看令牌Token是否存在,如果不存在,则视为非法请求;

[0046] 步骤2.3.3:根据解析的应用码AppID获取AppSercet,根据签名规则,组织签名,比对SdkSign是否合法;

[0047] 步骤2.3.4:校验请求IP是否在应用码AppID允许的IP白名单内;

[0048] 步骤2.3.5:判断系统当前小时内是否有超量请求,即系统会运行一个计划任务,每个5分钟会统计一次每个应用接口系统当前小时的请求量,如果大于最大请求频率阀值,则存储进Redis缓存,代表已超量;

[0049] 步骤2.3.6:如果步骤2.3.1至步骤2.3.5都通过的情况下,代表请求合法,进入步骤3;否则,为非法请求,结束流程。

[0050] 下面结合一具体实施例对本发明作进一步说明:

[0051] 如图2所示,本发明一种实现Api接口安全性的方法包括以下步骤:

[0052] 步骤1:接入方向带有Api接口的系统申请应用;

[0053] 步骤1.1:系统管理员分配好应用后,向接入方提供属于该应用的AppID(应用码)、AppSercet(应用密钥)和DesSercet(接口返回内容密钥),并且设置接口请求频率,接口请求频单位:次/小时;

[0054] 步骤2:接入方获得AppID和AppSercet后可以开始向系统接口组织请求;举例AppID是uLJqf20200113,AppSercet是TLnkJRtFTVjdTnYBKvTw,为了用于后面的步骤举例使用。

[0055] 步骤2.1:首先接入方要请求“获取令牌”接口来得到令牌Token。接入方传输AppID、Ticket、Sign请求“获取令牌”接口,Ticket为当前时间戳,Sign是由AppID+Ticket+AppSercet 3个参数拼接MD5后的值。令牌Token有2个小时的有效期,2个小时内无需重复请求令牌Token,令牌Token都视为有效。

[0056] 步骤2.1.1:系统收到2.1的请求后,首先根据AppID取得对应的AppSercet,校验Sign签名是否合法有效;

[0057] 步骤2.1.2:系统其次校验Ticket是否于服务器误差为5分钟范围内,超出范围的视为过期请求。

[0058] 步骤2.1.3:最后系统校验请求IP是否在AppID允许的IP白名单内。

[0059] 步骤2.1.4:如果2.1.1至2.1.3都合法的情况下,系统组织令牌值返回给接入方。令牌Token是由AppID、ExpireTime过期时间、随机字符串组成的Json字符串,再经过系统的DES加密得出的值。该值在返回给接入方之前会存入Redis缓存,用于校验合法性。举例Redis缓存Key为“ApiToken_uLJqf20200113”。

[0060] 步骤2.2:接入方获得令牌Token后将有权向系统请求具体功能的业务接口,根据实际业务接口所需参数和签名规则组织出SdkSign业务签名一起请求。举例请求的是一个下单业务接口,接口所需参数有:Type充值类型、Amount充值金额、Body充值标题。根据统一签名规则,参数根据参数名ASCII从小到大排序,即Amount+Body+Type得到text1,text1转化为小写获得text2,text2+AppSecret MD5加密后获得32位的MD5值,这个就是SdkSign。

[0061] 步骤2.3:根据2.1获得的令牌Token,2.2获得的SdkSign,再加上业务接口所需参数,一并请求业务接口。

[0062] 步骤2.3.1:系统收到2.3的请求后,首先解密令牌Token,并解析出包含的AppID信息。如果解析失败,则视为非法请求。

[0063] 步骤2.3.2:解析出AppID后,根据AppID得到Redis缓存key“ApiToken_uLJqf20200113”读取令牌Token,查看令牌Token是否存在。如果不存在,则视为非法请求。

[0064] 步骤2.3.3:根据解析的AppID获取AppSercet,根据签名规则,组织签名,比对SdkSign是否合法。

[0065] 步骤2.3.4:校验请求IP是否在AppID允许的IP白名单内,存在,则合法请求,不存在,则非法请求。

[0066] 步骤2.3.5:判断系统本小时内是否有超量请求,AppID加上当前时间的小时yyyyMMddHH来作为缓存key,例如uLJqf20200113_2019011310,读取Redis缓存,查看是1月13号10小时这个时段否超量。系统会运行一个计划任务,每个5分钟会统计一次每个应用接

口本小时(即10小时这个时段)的请求量,如果大于阀值(设定的最大请求频率)则存储进Redis,代表已超量。这个步骤不做同步实时的校验,因为适当的超量是允许的,通过异步的校验方式也能提高系统的性能。

[0067] 步骤3:如果步骤2.3.5通过,则代表请求合法。系统会把要返回的数据信息用AppID对应的DesSecret进行Des加密后返回给接入方。

[0068] 步骤4:接入方获得请求接口的返回内容后,用系统分配的DesSecret进行Des解密,得到返回内容。

[0069] 以上所述仅为本发明的较佳实施例,凡依本发明申请专利范围所做的均等变化与修饰,皆应属本发明的涵盖范围。

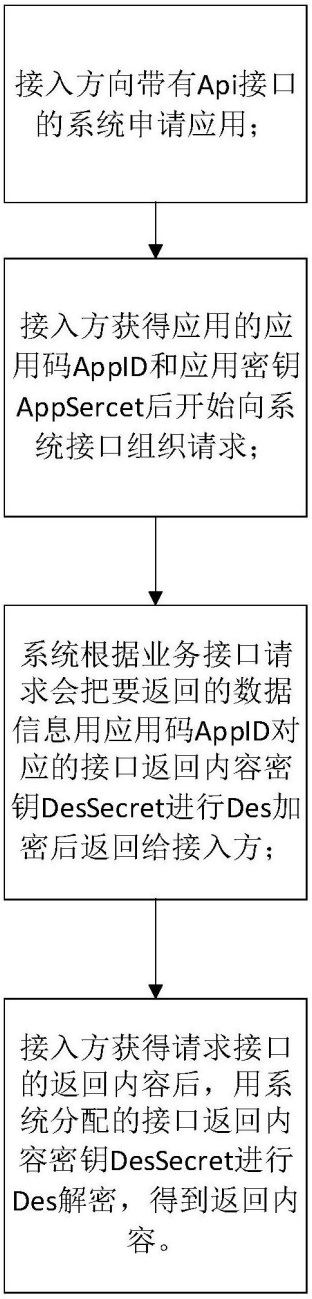

接入方向带有Api接口的系统申请应用；
接入方获得应用的应用码AppID和应用密钥AppSercet后开始向系统接口组织请求；
系统根据业务接口请求会把要返回的数据信息用应用码AppID对应的接口返回内容密钥DesSecret进行Des加密后返回给接入方；
接入方获得请求接口的返回内容后，用系统分配的接口返回内容密钥DesSecret进行Des解密，得到返回内容。

图1

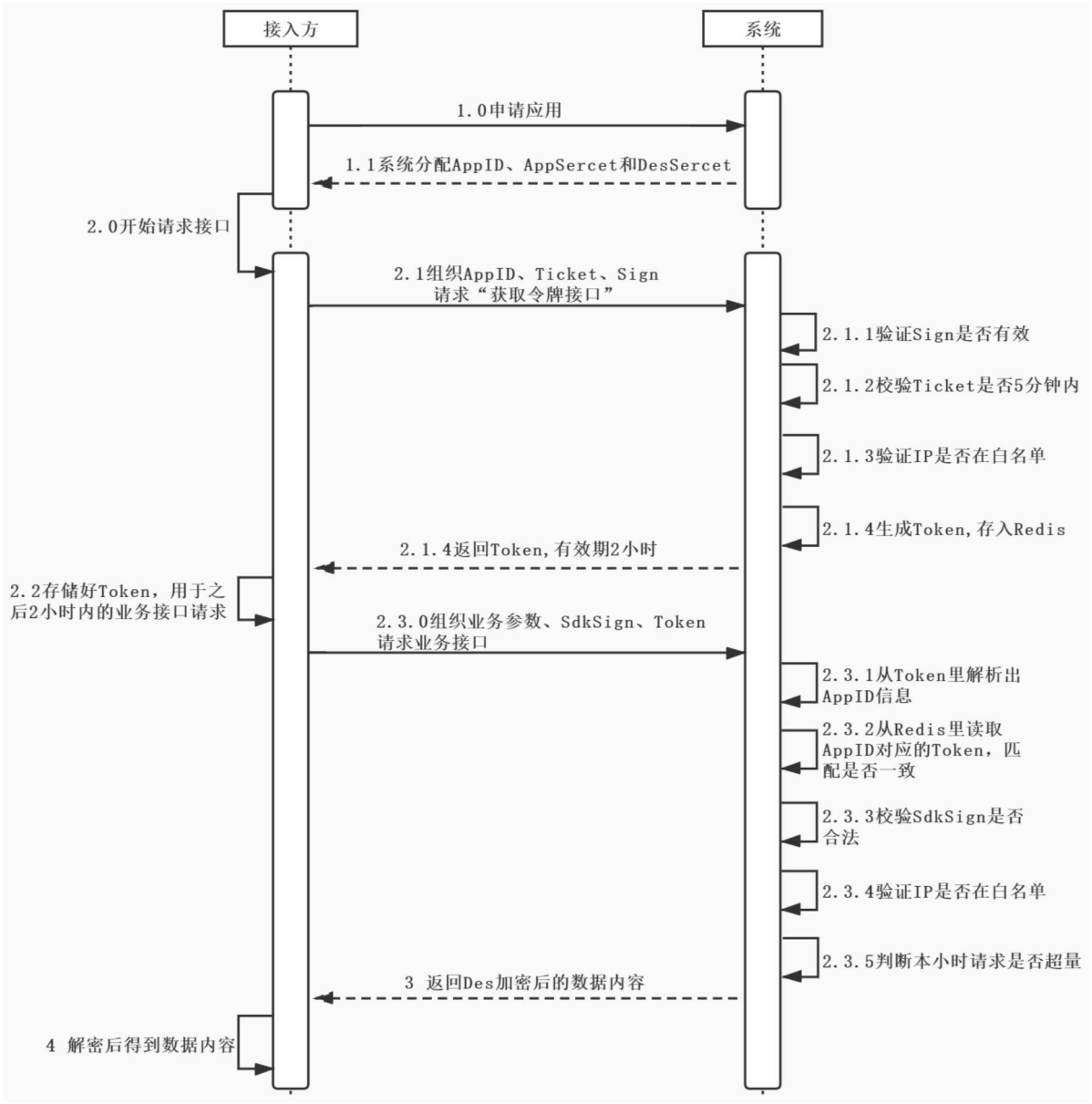

接入方
系统
1.0申请应用
1.1系统分配AppID、AppSercet和DesSercet
2.0开始请求接口
2.1组织AppID、Ticket、Sign
请求“获取令牌接口”
2.1.1验证Sign是否有效
2.1.2校验Ticket是否5分钟内
2.1.3验证IP是否在白名单
2.1.4生成Token,存入Redis
2.1.4返回Token,有效期2小时
2.2存储好Token，用于之后2小时内的业务接口请求
2.3.0组织业务参数、SdkSign、Token
请求业务接口
2.3.1从Token里解析出AppID信息
2.3.2从Redis里读取AppID对应的Token，匹配是否一致
2.3.3校验SdkSign是否合法
2.3.4验证IP是否在白名单
2.3.5判断本小时请求是否超量
3 返回Des加密后的数据内容
4 解密后得到数据内容

图2