|  |
| --- |
|  |
|  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 文档信息 | | | | | | | |
| **文档更新历史** | | | | | | | |
| 版本号 | 日期 | 作者 | | 更新描述 | | | |
| V1.0 | 2014-11-17 | Chen Jun | | 初版 | | | |
| V1.1 | 2014-11-18 | Chen Jun | | 增加一次性用户注册及登录处理 | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
|  |  |  | |  | | | |
| **文档审核历史** | | | | | | | |
| 版本号 | 评估日期 | 评估人 | 签核日期 | | 签核人 | 签核人角色 | 备注 |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |
|  |  |  |  | |  |  |  |

文档目录

[概述 4](#_Toc404025387)

[定义 4](#_Toc404025388)

[设计概要 4](#_Toc404025389)

[方案选择 4](#_Toc404025390)

[0.公钥存放、取得方式 4](#_Toc404025391)

[1.用户注册方式 4](#_Toc404025392)

[2.用户登录方式 5](#_Toc404025393)

[前提与限制 6](#_Toc404025394)

[代替方案 6](#_Toc404025395)

[扩展 6](#_Toc404025396)

[备注 6](#_Toc404025397)

# 概述

本文档用于说明用户注册Ba账号及登录Ba服务器的相关处理。

本文档中出现的产品、商标、专利等相关内容之权利均归属各权利所有人所有。若无本文档权利人之明确书面许可，本文档的任何部分不可被复制、存储或引进检索系统，或者以任何形式、任何方式（电子、机械、复印、录音或其他）或为任何目的进行传播。

# 定义

1. 上行数据：从客户端传向服务端的数据；
2. 下行数据：从服务端传向客户端的数据；
3. 一次性用户：Ba绑定时间为24小时的用户。该类用户使用手机号的方式进行注册和登录。

# 设计概要

用户在注册及登录过程伴随着大量敏感数据的交互行为，一旦发生信息泄露将造成极大的损失。因此，在处理注册、登录及其他涉及用户信息的操作时，必须遵行交互最小化原则与数据加密原则。

本系统采用非对称RSA进行加密，私钥永远只保存在服务器端，并且不提供任何对外的获取、修改接口；公钥在一定条件下向外部调用公开。用户信息，除去用户名（登录名）、公钥以及下文中特别声明以外，均应该加密交互。

上行数据的加密采用公钥加密、私钥解密后再以私钥加密的方式存在于数据库中。

下行数据是直接将私钥加密的数据用于交互。

# 方案选择

## 0.公钥存放、取得方式

客户端应负责用户及其公钥的保存，一旦发现某一用户的公钥信息在当前的本地存储中不存在时，应当调用服务器的取得公钥服务获取该用户的公钥并进行保存。

## 1.用户注册方式

用户注册分为两个步骤。

**STEP1** ：用户通过各客户端进行注册，此时只需要填写用户名即可。服务端在检查用户名无重复后，向客户端返回以下三类信息，该三类信息不进行加密处理：

用户名

用于JS的公钥

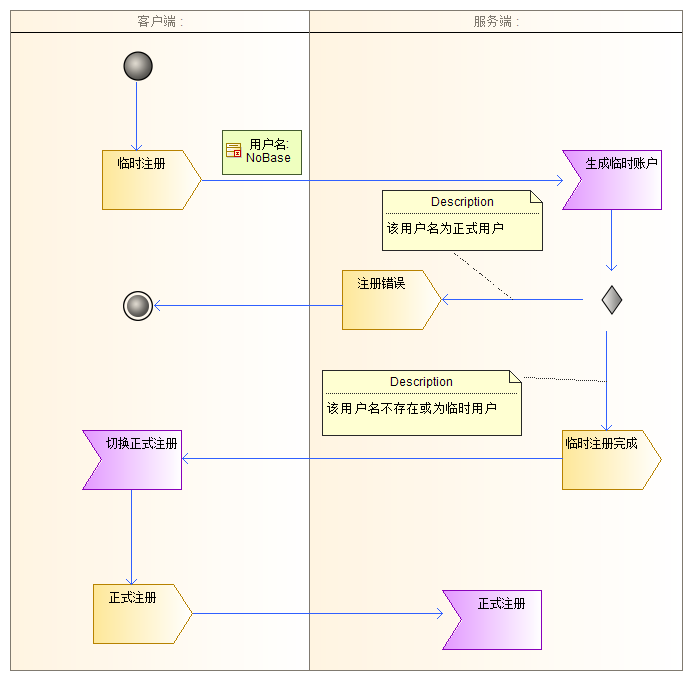
临时注册号

客户端应当记录这些信息，并将用户导向正式注册页面进行正式注册。

**STEP2** ：当用户填写完毕注册信息并通过客户端校验后，客户端将使用该用户的公钥对信息进行加密，并将加密后的数据提交服务器进行正式注册。

**STEP2’**：如果出现异常情况导致用户无法连续完成STEP1与STEP2，而只是STEP1完成时，应当允许用户仍然使用临时注册的页面输入STEP1中完成的用户名。服务端将判断该用户是否可以继续注册，或是已为注册用户而因用户存在导致注册失败。

活动图如下（省略正式注册完成后的活动）：



## 2.用户登录方式

用户登录时，由客户端向服务器提交用户名、密码信息。依照《ba-TD-131104-01[异常处理及异常体系]》的规定，当登录成功时，服务器将返回HTTP Code204作为成功标志；而不成功时，服务器将返回HTTP Code417，并在HTTP Content中列明错误代码。

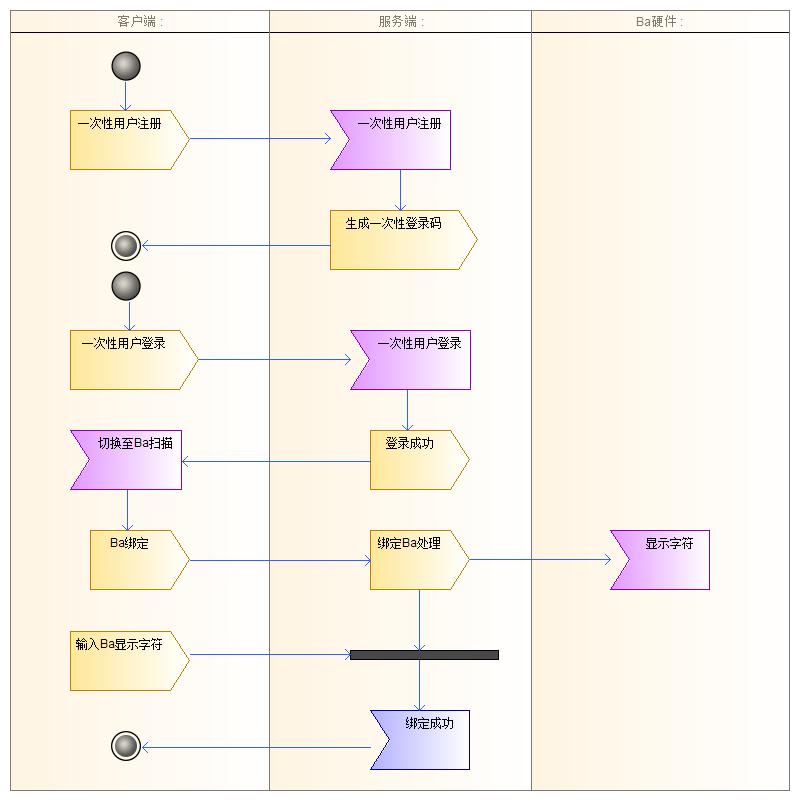
## 3.一次性用户注册及登录方式

一次性用户通过手机号提交注册申请。服务器端在收到申请后，与正式用户注册相同，会以手机号码的SHA256散列为用户名保存新的记录（用户类型为一次性用户），同时生成一次性登录密码发送给该手机号，该登录密码生成24小时内有效。用户通过客户端，使用手机号和该一次性密码进行登录。

用户登录时，服务端首先会检查该用户的密码是否已经过期，如过期则返回登录错误异常，要求用户重新注册。

绑定设备时，用户需要通过客户端扫描Ba硬件上的标识（二维码、条码等），并将用户名/手机号、硬件标识信息传给服务端。服务端会生成2字节的随机字符通过指令使其显示在Ba上，用户需要输入这两个字符以完成绑定。绑定时间默认为24小时。

活动图如下所示：



# 前提与限制

客户端与服务端可能使用不同的代码进行开发，因此在RSA填充算法可能存在不一致的情况。需要根据各种客户端开发语言对RSA的支持情况进行调查分析，找出交集。

JavaScript可以使用以下的类库：

<http://www-cs-students.stanford.edu/~tjw/jsbn>

作为参考，加密耗时的列表如下

http://www-cs-students.stanford.edu/~tjw/jsbn/speed.html

# 代替方案

暂无

# 扩展

为进一步提高安全性，应该使用https代替http情况。

# 备注

无