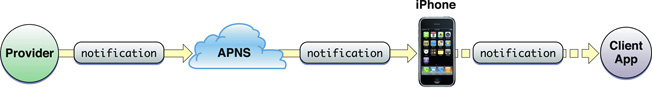
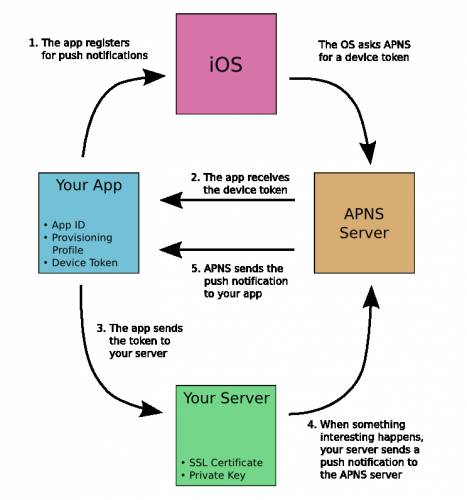
**APNS 是Apple Push Notification Service（Apple Push服务器）的缩写**



整个过程可以分为三个阶段

第 一阶段：push 消息服务器把要发送的消息、目的iPhone的标识打包，发给APNS。   
第二阶段：APNS在自身的已注册Push服务 的iPhone列表中，查找有相应标识的iPhone，并把消息发到iPhone。   
第三阶段：iPhone把发来的消息传递给相应的应用程序， 并且按照设定弹出Push通知。



 从上图我们可以看到。

   1、首先是应用程序注册消息推送。

   2、 IOS跟APNS Server要deviceToken。应用程序接受deviceToken。

   3、应用程序将deviceToken发送给PUSH服务端程序。

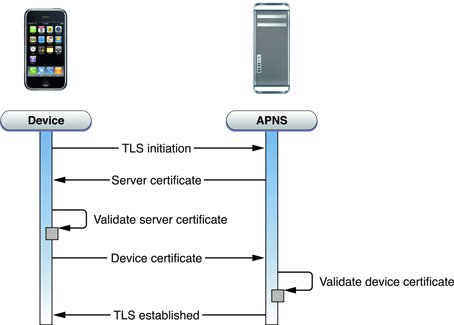
   4、 服务端程序向APNS服务发送消息。

   5、APNS服务将消息发送给iPhone应用程序。

Push认证

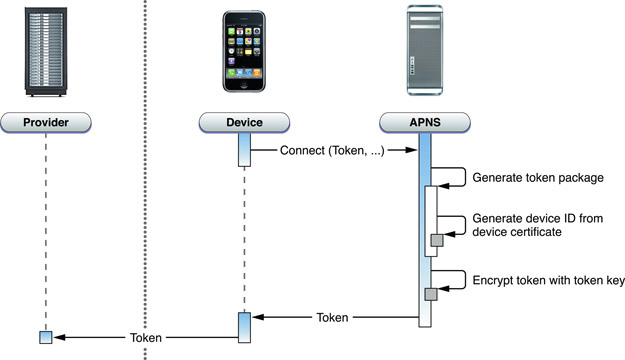
认证机制，实际上包含两层。一层是**物理连接上的认证**，另一层则才是涉及到**iPhone 设备令牌的认证。**

物理连接上的认证：SSL/TLS链接



iPhone在开启Push的时候，会连接 APNS建立一条TLS加密链接。每一台正常的iPhone都有一个独有的设备证书，而APNS也有一个服务器证书。两者建立的时候，会验证彼此的证书有效性。  
**TLS链接一旦建立，在没有数据的情况下，只需要每隔15分钟进行一次保活的握手，因此几乎不占流量。**而一旦因为意外原因导致链接中断，iPhone会不断重新尝试建立TLS链接，直到成功。

更高一层次：基于token(令牌)的认证  
在机制 简介里，我提到过**APNS判断Push推送消息该发给哪台iPhone的依据**是一个“目的iPhone的标识”，这个 标识就是**device token(设备令牌)**。  
**设备令牌是怎么生成的呢？是每次建立TLS 连接时，APNS通过前一层次(TLS层)里我们提到的每台正常的iPhone唯一的设备证书(unique device certificate)，并用令牌密钥(token key)加密生成的。**



参考地址：

<http://www.cocoachina.com/newbie/basic/2010/0401/900.html>

<https://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html>

<http://blog.serverdensity.com/how-to-build-an-apple-push-notification-provider-server-tutorial/>

<http://www.devdiv.com/iOS_iPhone-%E6%B5%85%E6%9E%90iOS%E4%B8%ADPush%E9%80%9A%E7%9F%A5%E7%9A%84%E4%BD%BF%E7%94%A8-thread-130543-1-1.html>

<http://www.devdiv.com/article-4042-1.html>

<http://hi.baidu.com/blackj_li/item/f6befa11655fe4e45f53b1ae>

**Feedback Service**

Apple provide a [feedback service](http://developer.apple.com/iphone/library/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/CommunicatingWIthAPS/CommunicatingWIthAPS.html#//apple_ref/doc/uid/TP40008194-CH101-SW3) which you are supposed to occasionally poll. This will provide a list of deviceTokens that were previously but are no longer valid, such as if the user has uninstalled your iPhone application. You can then remove the deviceToken from your database so you do not communicate with an invalid device.

If a notification for an application arrives when that application is not running, the device alerts the user that the application has data waiting for it.

如果一个应用通知到来但应用程序没有运行时，这个设备会提示这个应用有新的消息。

A notification is a short message consisting of two major pieces of data: the device token and the payload.

The device token is analogous to a phone number; it contains information that enables APNs to locate the device on which the client application is installed. APNs also uses it to authenticate the routing of a notification.

The payload is a JSON-defined property list that specifies how the user of an application on a device is to be alerted.

注意：

1,模拟器不能用push 通知，因为模拟器不能获得device token ，测试需要一个device。

2,发送的消息大小不能超过256 bytes。

3,apns不提供发送消息是否成功状态反馈。

Custom sounds must be under 30 seconds when played. If a custom sound is over that limit, the default system sound is played instead.

<http://blog.itotem.com.cn/?p=240>

用PHP为iOS做推送服务

首先，需要一个pem的证书，该证书需要与开发时签名用的一致。 具体生成pem证书方法如下：

*1. 登录到 iPhone Developer Connection Portal（*[*http://developer.apple.com/iphone/manage/overview/index.action*](http://developer.apple.com/iphone/manage/overview/index.action)*） 并点击 App IDs*

*2. 创建一个不使用通配符的 App ID 。通配符 ID 不能用于推送通知服务。例如，  com.itotem.iphone*

*3. 点击App ID旁的“Configure”，然后按下按钮生产 推送通知许可证。根据“向导” 的步骤生成一个签名并上传，最后下载生成的许可证。*

*4. 通过双击.cer文件将你的 aps\_developer\_identity.cer 引入Keychain中。*

5. 在Mac上启动 Keychain助手，然后在login keychain中选择 Certificates分类。你将看到一个可扩展选项“Apple Development Push Services”

6. 扩展此选项然后右击“Apple Development Push Services” > Export “Apple Development Push Services ID123”。保存为 apns-dev-cert.p12 文件。

7. 扩展“Apple Development Push Services” 对“Private Key”做同样操作，保存为 apns-dev-key.p12 文件。

8. 需要通过终端命令将这些文件转换为PEM格式：openssl pkcs12 -clcerts -nokeys -out apns-dev-cert.pem -in apns-dev-cert.p12

openssl pkcs12 -nocerts -out apns-dev-key.pem -in apns-dev-key.p12

9. 如果你想要移除密码，要么在导出/转换时不要设定或者执行：

openssl rsa -in apns-dev-key.pem -out apns-dev-key-noenc.pem

10. 最后，你需要将键和许可文件合成为apns-dev.pem文件，此文件在连接到APNS时需要使用：

cat apns-dev-cert.pem apns-dev-key-noenc.pem > apns-dev.pem

PHP代码如下：

*<?php*

*$deviceToken = $\_POST['token']; //取得设备的Token，获取方法便见下文*

*$body = array(“aps” => array(“alert” => ‘message’, “badge” => 2, “sound”=>’default’));  //推送方式，包括了提示内容，提示方式和提示声音。*

*$ctx = stream\_context\_create();*

*//如果在Windows的服务器上，寻找pem路径会有问题，路径修改成这样的方法：  
//$pem = dirname(\_\_FILE\_\_) . ‘/’ . ‘apns-dev.pem’;  
//linux 的服务器直接写pem的路径即可*

*stream\_context\_set\_option($ctx, “ssl”, “local\_cert”, “apns-dev.pem”);*

*//如果你的pem存有密码，需要加一个密码登陆语句*

*$pass = ”123123“；  
stream\_context\_set\_option($ctx, ‘ssl’, ‘passphrase’, $pass);*

*//此处有两个服务器需要选择，如果是开发测试用，选择第二名sandbox的服务器并使用Dev的pem证书，如果是正是发布，使用Product的pem并选用正式的服务器*

*$fp = stream\_socket\_client(“*[*ssl://gateway.push.apple.com:2195*](http://blog.itotem.com.cn/2195)*“, $err, $errstr, 60, STREAM\_CLIENT\_CONNECT, $ctx);*

*$fp = stream\_socket\_client(“ssl://gateway.sandbox.push.apple.com:2195″, $err, $errstr, 60, STREAM\_CLIENT\_CONNECT, $ctx);*

*if (!$fp) {*

*print “Failed to connect $err $errstrn”;*

*return;*

*}*

*print “Connection OK\n”;*

*$payload = json\_encode($body);*

*$msg = chr(0) . pack(“n”,32) . pack(“H\*”, str\_replace(‘ ‘, ”, $deviceToken)) . pack(“n”,strlen($payload)) . $payload;*

*print “sending message :” . $payload . “\n”;*

*fwrite($fp, $msg);*

*fclose($fp);*

*?>*

讲apns-dev.pem和php.php文件放到一个目录下。

进入终端，进入php.php目录下。键入命令php php.php