现代生活已经离不开的银行卡支付，背后的产品设计还是大有门道的。本文作者对银行卡支付的原理进行了分析梳理，与大家分享。

上次写了一篇『轻轻一扫，立刻扣款，付款码背后的原理你不想知道吗』，今天小黑哥再来跟大家聊聊支付。

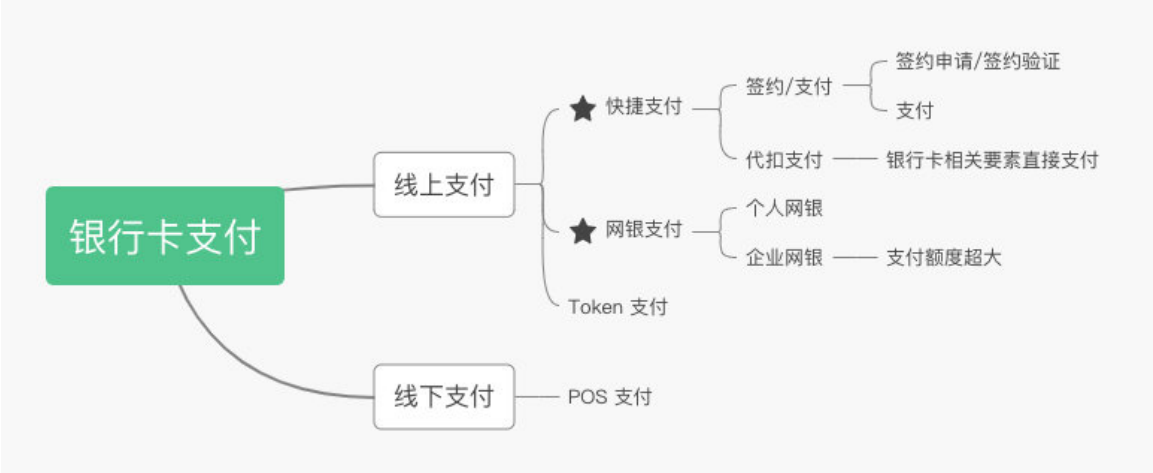
虽然现在我们主流的支付方式是使用支付宝/微信支付，但是当我们余额不足，或者选择从银行卡扣款时，将就会使用到银行卡支付。

所以今天我们就来来讲讲银行卡支付的相关原理，科普一下银行卡支付整个流程。

银行卡支付可以将其分为线上支付与线下支付。其中线下支付分类就比较简单，就是我们平常在商城购物时，POS 机刷卡支付。

而线上支付分类就比较多了，根据银行卡类别，可以分为信用卡支付与借记卡支付。按照支付行为，我们又可以将其分为快捷支付，网银支付，Token 支付。

今天我们主要来聊聊快捷支付与网银支付，这两种方式是目前比较流行的方式。其他几种方式，我们可以后面再来聊聊。



一、网银支付

首先我们来聊聊网银支付，这种方式在 10 年前，应该是最主流线上支付方式。

我们以电商购物为例，我们在网站上下单之后，选择银行卡支付通常会跳转到一个收银台页面。然后在收银台页面我们选择相关银行，点击到银行支付最后将会跳转到相应的银行页面。

这个收银台页面可能是商户的页面，也可能是支付机构的页面，这个跟网银支付对接模式有关。

跳转到银行页面之后，我们首先需要下载按照银行安全控件，这样我们才能输入银行卡的相关信息。其次我们还需要使用银行给的安全设备，比如 USB 盾，令牌器，令牌码等。

在银行网站支付成功之后，就可以点击返回同步跳回到电商的网站，整个流程如下图所示：

第一步

收银台页面选择相关银行

第二步

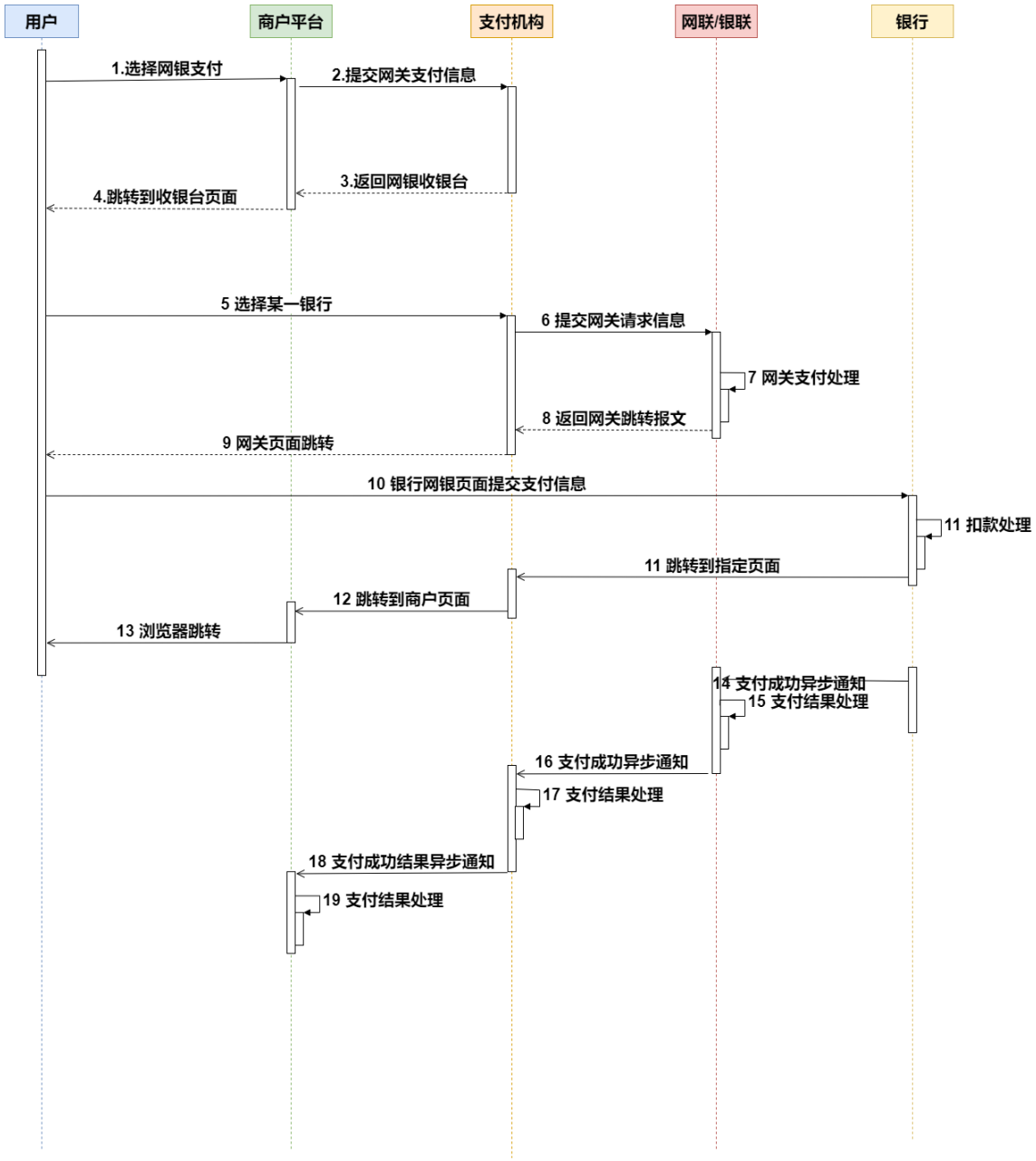
跳转到相关银行

第三步

支付成功，跳转到商户页面

网银支付流程

后台支付流程如下：



可以看到网银支付整个链路非常长，任何一步都可能发生失败，所以支付成功率不会很高。另外有部分银行网银页面只能在 IE 中打开，而且还有可能是很老版本的 IE。再加上网银支付为了保证安全性，还需要使用 U 盾，安装安全插件。

这个过程说实话还是很复杂，还记得当年使用某行网银充值购买黄钻的时候，搞了一下午都没成功的，各种证书安装失败啥的。第一次在线充值，就这么失败告终。

感谢某行为我省下 10 元零花钱~

二、快捷支付

快捷支付，指的用户提供卡信息给电商等商户，商户会在后台将卡信息传递给支付机构，然后进行协议绑定。一旦绑定成功，下次支付，无需再让用户提供卡号等信息。

还是以电商购物支付为例，首次支付，需要经历绑卡过程。

第一步　　第二步　　第三步

输入银行卡信息,获取验证码　　回填验证码，确认支付　　支付成功


09:41
快捷支付管理
支付宝
财付通
拉卡拉
京东支付
百度钱包
易宝支付
快钱支付
通联支付
宝付
中金
银联电子
财付通
百度钱包

第一步

历次支付，直接获取验证码
这里也可以是免密支付，跳过第二步

第二步

回填验证码，确认支付

第三步

支付成功

快捷支付接口一般可以归为两类：

签约/支付 代扣支付

1. 签约/支付

签约/支付需要分为两个步骤：

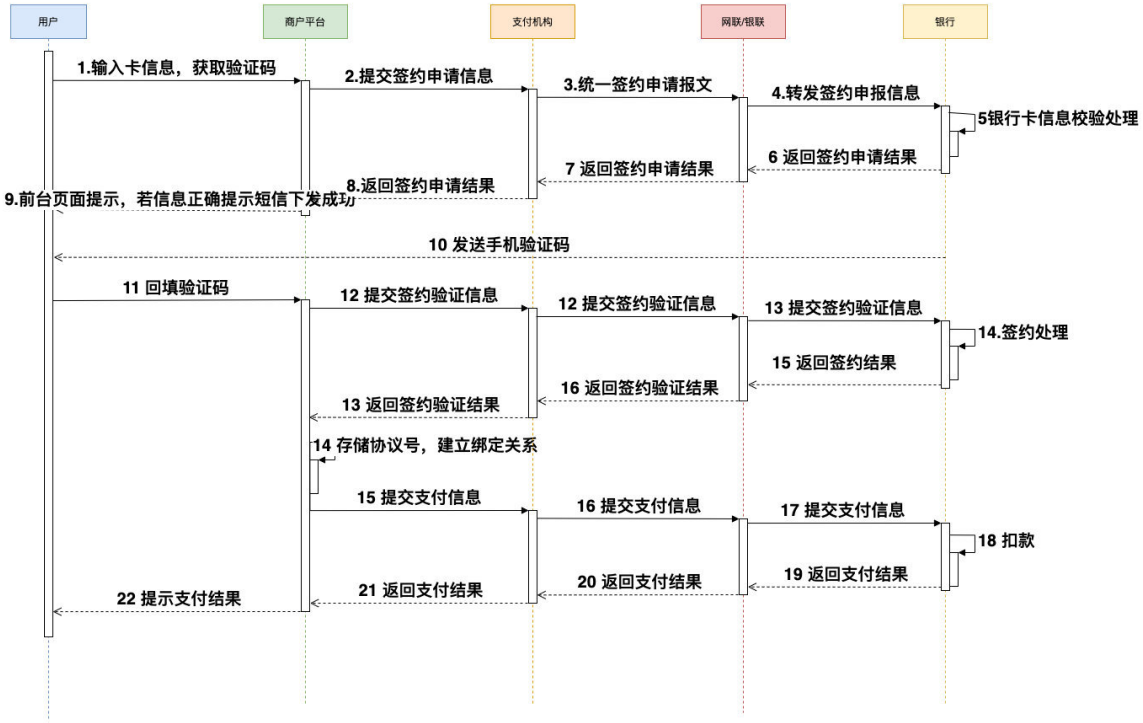
签约申请/签约验证 支付

签约过程需要传入银行卡信息，银行卡号，户名，身份证号，手机号，信用卡的话可能还需要传入 cvv2 以及有效期。这个过程主要是为了鉴权，校验银行卡信息的正确性。

一旦支付机构/银行端信息校验成功，将会下发短信。用户回填短信，就代表同意开通快捷支付，建立绑定关系。绑定成功之后，支付机构将会返回给商户协议号。

支付过程，商户就可以拿着协议号进行扣款。

整个后台流程如下所示：



2. 代扣支付

代扣支付的过程相比签约/支付就比较简单，每次直接上送银行卡信息，就可以直接扣款。代扣支付原则上可以做到整个过程无密支付，即不需输入验证码，完成扣款。

流程较为简单，详情可以参考快捷支付支付过程。

相比于签约/支付过程，代扣支付看起来更快捷，但是这种方式安全风险就会比签约支付大，可能就会出现盗刷现象。原本代扣接口本应适用于水电煤等扣费场景，但是发展过程一度被用于金融支付等场景。

现在这类接口正在慢慢下线，正在被新的商业委托接口（类似于签约/支付）所代替。

虽然快捷支付支付体验好，整个流程无需跳转到银行页面，支付过程不会被打断，支付成功率高。

但是易用跟安全性，永远都是矛盾。由于这个过程用户向商户提供银行卡相关信息，这些数据如果一旦被窃取，资金就可能会被盗取。另外，快捷支付，手机验证码可能是最后一道防线，手机如果丢失，那么银行卡资金也可能被盗取。

三、银行支付相关问题

总得来说，对接银行卡支付渠道，整个过程不是很难的，无非就是按照接口文档，拼接参数，然后做一些相应的调试。但是这个过程有些点需要特别注意。

1. 加签/验签

银行卡支付一般通过互联网传输，这个过程为了防止报文被串改，通常会采用 RSA2 ，国密等加密算法加密报文，得到签名串，然后一起上送给支付机构。

支付机构方会进行相应的验签，验签失败，就会驳回支付请求，这样可以有效保证支付请求是从合法商户发起。所以对于商户来说，一定要保存好相应公私钥，不要随意泄漏。

另外，对于支付请求的响应信息/网银结果异步通知，支付机构端也会进行加签。商户端一定要进行验签，只有验签通过才能进行下一步。

ps:发送请求由于不加签，交易无法进行，所以这一步肯定会做的。 但是返回信息你不进行验签，也能处理报文，这个可能就会被忽略。 我第一次对接相关支付渠道的时候，嫌麻烦，就没进行验签。现在想想，真的是心大。。。

2. 终态判定

对于快捷支付这类同步接口，对于支付接口请求响应消息，我们需要判定请求是否成功，需要根据接口返回的响应码。有些接口也可能返回响应码与支付状态，那么我们就需要根据两者结合起来一起判断。

这个过程，不是说除了成功的响应码之外，其他都算失败。我们需要根据相关的接口文档进行相应的分类，有些如余额不足，卡要素不正确等错误码，当然可以明确归类为失败。

但是比如一些处理中，或者系统异常等返回码，这种无法明确到底是成功还是失败的，我们不能置为失败，需要结合支付查询或者异步通知结果，然后在做处理。

对于网银支付这类同步接口，这类只能等待渠道端的异步通知。一般来说，渠道端只会通知的成功的支付订单。

这个具体根据渠道端接口文档。

一般来说渠道异步通知接口，若没有给渠道端异步通知返回成功响应，该通知将会重复通知，直到到达一定次数或者得到成功的相应。

所以接受到异步通知之后，一定要内部逻辑处理成功之后，才能返回成功响应码给渠道端。这样即使内部逻辑处理错误，还能再次通过异步通知处理内部逻辑。

另外还需要注意内部处理逻辑的幂等性。

3. 请求参数相关

（1）支付金额

请求过程一定要注意接口文档中支付金额的单位，是分，还是元。如果不注意单位，很有可能造成少收，多收的情况。

对于成功响应的信息，我们还需要注意校验上送金额与扣款金额（如果有返回的话）一致性。如果不一致，**一定不要将订单更新为成功，**及时人工介入查单。

最后支付渠道上线之后，还需要做一些真实扣款，比如小额 0.1,渠道最大额度测试。扣款成功之后，还要及时查看银行卡真实扣款金额是否与上送金额一致，原因见下文。

（2）请求流水号(订单号)

除了支付金额，我们还需要注意请求流水号/订单号唯一性，需要使用唯一 id 当做请求流水号，切勿使用时间戳等方式。

对于重复流水号，如果未成功，是允许重复支付的。如果成功，不允许再次支付的。但是也不乏有些机构接口没做好这部分校验。

举一个自己趟过的坑，一个几万的教训。之前对对接过某银行的系统，测试的时候为了方便，直接采用时间戳当流水号。

上线时未及时发现这个问题，某天恰好同一秒产生两笔流水号一样的单子，上送给银行。然后对方返回两笔都收款成功，但是第二天对账时发现仅收到一笔单子的资金。所幸最后通过人工追回这笔资金，不然当时卖了我，也赔不起啊。。。

虽然这个例子银行端肯定也是存在问题的，未做防重处理，但是只要我们做好唯一流水号的逻辑，也能避免该问题。

真实惨痛例子

上面注意的问题聊了这么多，其实想引起对接渠道技术同学注意。不要片面认为支付机构或银行等系统很稳，不会有问题。

程序毕竟是人写的，一次升级改动，就有可能引起血崩。

所以不要过分相信对方系统的稳定性，我们能做的就是做好我们自己系统的稳定性，加入各种参数校验，尽量降低风险的发生。

给大家举几个惨痛的例子：

曾经对接过某银行，小额测试，完全没问题。但是我们在测试限额的时候，比如说限额 1000 元，我们测试 1000.01 的时候，讲道理这笔支付应该会失败。

但是这笔扣款成功了，并且查看银行扣款记录，仅仅只扣了 0.01 。看到这个，你是否有很多问号？？？这 TM 竟然发生限额溢出。。。

哎，这种问题，只能紧急下线该渠道，然后等待银行端修复。

最后再举几个来自网上的例子，关于支付的漏洞。

原文地址:http://drops.wooyun.org/papers/345

0x00 背景介绍

随着网民越来越习惯于网上购物，出现了越来越多的电商网站，在线交易平台等。

其中肯定要涉及在线支付的流程，而这里面也有很多逻辑。

由于这里涉及到金钱，如果设计不当，很有可能造成0元购买商品等很严重的漏洞。

0x01 检测方法与案例

根据乌云上的案例，支付漏洞一般可以分为五类，如果发现其他的类型，欢迎补充：

1、支付过程中可直接修改数据包中的支付金额

这种漏洞应该是支付漏洞中最常见的。

开发人员往往会为了方便，直接在支付的关键步骤数据包中直接传递需要支付的金额。

而这种金额后端没有做校验，传递过程中也没有做签名，导致可以随意篡改金额提交。

只需要抓包看到有金额的参数修改成任意即可。

我们来看一看乌云上的几个案例：

WooYun: 必胜客宅急送支付表单伪造金额

WooYun: 肯德基宅急送支付表单伪造金额

WooYun: 新浪微号存在支付绕过漏洞

WooYun: 淘宝网某处存在严重支付漏洞

WooYun: 佳域手机官方商城支付漏洞 （这个亮点是真的到货了……）

WooYun: 91分站存在支付绕过

WooYun: 江西移动1元钱买手机漏洞

WooYun: 爱拍主站存在严重漏洞

WooYun: 再爆苏宁某站点重大漏洞

WooYun: 苏宁某站点存在严重漏洞

WooYun: TP-Link官方商城支付漏洞

WooYun: 鲜果网支付漏洞

WooYun: 京东商城购买商品时，可以修改商品金额，并且支付成功

WooYun: 京东团购订单金额可在客户端修改并提交网银支付

WooYun: 网通营业厅客户信息泄露、充值支付价格修改漏洞

2、没有对购买数量进行负数限制

这种案例也比较常见，产生的原因是开发人员没有对购买的数量参数进行严格的限制。

这种同样是数量的参数没有做签名，导致可随意修改，经典的修改方式就是改成负数。

当购买的数量是一个负数时，总额的算法仍然是"购买数量x单价=总价"。

所以这样就会导致有一个负数的需支付金额。

若支付成功，则可能导致购买到了一个负数数量的产品，也有可能返还相应的积分/金币到你的账户上。

WooYun: 百脑汇商城支付漏洞

WooYun: m1905电影网存在严重支付漏洞

WooYun: 国美网上商城支付漏洞1元订购Iphone 4S！

WooYun: 又拍网旗下某站存在严重支付漏洞

WooYun: 新蛋中国支付漏洞

WooYun: 拉卡拉商店0元购支付问题

WooYun: 中粮52buy商城的支付漏洞

WooYun: 115网盘存在支付绕过

最后一个漏洞与其他不同的是把数量改成一个超大的数，而不是负数。

结果导致支付的金额可能超过一定数值而归0。

3、请求重放

购买成功后，重放其中请求，竟然可以使购买商品一直增加~

阿里云主机多次下订单，会出现0元订单情况，不知道程序员后端是如何写的……

WooYun: 豆丁网购买豆元后可以将豆元倍增

WooYun: 阿里云0元订单，服务器随便买

4、其他参数干扰

此案例金钱已经做了签名认证，修改后不通过。

但是仍然有一个参数会对最后的金额产生影响而没有一起做签名导致问题产生。

WooYun: 新东方逻辑支付漏洞

0x02 修复方案

来源：https://wooyun.js.org/drops/在线支付逻辑漏洞总结.html

总结

今天我们主要聊了下银行卡支线上支付的两种主流模式，快捷支付与网银支付。

快捷支付目前是现在最主流银行卡支付方式，因为使用体验最好，支付流程不易被打断。但是该模式相对来说安全性较低。不过现在支付机构端与银行端会有相应的风控手段，大家不用过分担心。

另外一点快捷支付，一般额度较小，通常最高额度可能只有几万。

所以对于支付金额较大的场景，只能采用网银支付这种方案。

最后聊了下银行卡支付对接过程中一些问题，有些例子，可以集成到测试案例中。每当对接一个渠道时，就可以按照案例执行。

最后

支付系列的文章，小黑哥已经更新几篇，历史文章可以查看下面相关阅读。

后续，小黑哥还会更新几篇，聊聊支付宝/微信支付相关支付方式，聊聊支付过程中重复扣款等等。

如果各位同学还想了解其他支付相关的话题，可以在评论区留言。

参考文档：

支付系统设计：银行卡支付（三）

#相关阅读#

“轻轻一扫，立刻扣款”，解读付款码背后的原理

作者：楼下小黑哥；微信公号@程序通事，支付行业，后端技术