文章编号:1674-7070(2015)05-0422-05

陈遥[1] 夏亮亮[1] 谭辉[1]

移动互联网环境下终端与服务器安全交互的研究

摘要

移动互联网的飞跃式发展与智能终端成本的降低,惠及了众多的普通消费者,移动互联时代用户的隐私与信息安全变得更为重要.针对 WiFi 热点环境下用户隐私与数据容易泄露的环节,对比传统的常用加密算法与使用效果,提出了一种基于 OAuth 协议的令牌机制的安全交互方式,它可以在保证用户相关隐私与信息安全的同时,保证服务器的响应速度,且在用户的应用存在风险或异常时,可及时提醒并引导用户进行相关安全操作.压力测试结果表明该设计是有效可行的.

关键词

移动互联网;OAuth 协议;安全交互

中图分类号 TP311

文献标志码 A

0 引言

随着 3G/4G 通信技术的快速发展与智能终端的大量普及,移动互联网的成长得到了巨大的推动,以 iOS 与 Android 为代表的移动终端操作系统,为众多的应用提供了良好的“发育环境”.但移动互联网特殊的基因——无线接入方式,使众多应用在使用中存在巨大的安全风险,因为相对于传统的有线网络或者家庭、办公所使用的无线局域网络而言,因其接入方式更加快捷、灵活、开放和自由,以及公共无线热点随到、随连和随用的特性注定使其成为目前网络安全相对薄弱的一个环节.如果智能终端连接到不安全或者是恶意伪装的 WiFi 热点后,个人信息(如账号、密码和银行卡信息等)等敏感数据将面临泄露的风险,可能造成不可弥补的损失.

在实际应用中,应用系统与服务器的数据交换不可避免,要达到安全交互的目的,须对传输数据进行加密变形或根据应用的请求、意图更换新的终端与服务器交互形式.

移动终端常用的网络连接是 Http 协议的方式.以 Http 协议的交互方式为例,传统的 PC 端 Web 应用使用浏览器访问服务器时由 Session 存储并标识当前活动用户,从而在 Session 失效后或用户关闭浏览器时,服务器可以终止会话,释放资源;而移动智能终端设备在访问服务器时,由于并非使用浏览器的访问机制,为了标识自身会话信息,通常使用发送上次请求返回的 SessionID 的方式达到目的.但是,移动应用与传统 Web 应用差别很大且 SessionID 会有过期时间,当会话过期后,若要识别身份,则需要重新发送验证信息,而于某些移动应用来说,这样做会使操作变得极为繁琐与不便,所以也有很多开发者放弃使用会话的机制,转为使用发送特定识别信息的方式.例如最为简单直接的做法就是请求附带用户账号与密码,既保证了请求标识的唯一性,不用考虑会话过期所带来的多次额外操作,同时又能认证用户的有效身份.不过这种简单暴力的方式风险是最为突出的,频繁的请求每次都附带用户最为核心的信息,一旦请求数据在不安全的 WiFi 网络环境中传输时被捕获抓包,用户信息便直接暴露.大多数人都有这样一种习惯,为了便于记忆,各种网站或者应用的账号密码几乎保持一致,甚至各个银行卡与网银的密码也都一样.如果因一个应用的疏忽大意泄露用户敏感信息,不法分子得到这些信息去各个网站“撞库”,成功率是非常高的,由此带来的后果也是极为严重的.

收稿日期 2014-05-05

资助项目 南京信息工程大学滨江学院 2014 届优秀本科毕业论文支持计划(BYW002063)

作者简介

陈遥,女,副教授,主要研究方向为智能数据处理与数据挖掘.chenyao0077@163.com

1 南京信息工程大学 滨江学院,南京 210044

因此,为了保证用户信息的安全,必须对数据进行处理,不管使用什么方法,其终极目的是非明文或不传输用户关键信息.其中,非明文传输数据的典型做法就是加密,加密算法最具代表性的就是对称加密算法与非对称加密算法.WiFi 环境下容易在传输数据时发生隐私泄露,下文将对比常用的加密算法与使用效果,将其融入到移动应用中,以验证传统加密方式能否保障移动应用数据的传输安全.不传输用户关键信息的核心思想是找到一种可以与用户相关而安全,又可以得到保障的特殊标识.本文采用这种思想,将提出一种基于 OAuth 协议[1]令牌机制的安全交互方式的设计.

1 对称加密算法

对称加密算法[2]工作示意如图 1 所示,它效率高、速度快,但其加密和解密的密钥相同,密钥的泄露便意味着加密数据可被轻易查看,如安全性想达到预期效果,密钥的分发与管理系统则会变得异常庞大与复杂.例如,一个应用有 x 位用户,每个用户都使用对称加密方式安全传输数据,那么每人至少要交替使用 2 组密钥,服务器需要维护的密钥总数为 $x(x-1)$.而现阶段移动应用的特点是本地功能丰富强大,开发周期呈现出越来越短的趋势,且开发者在开发时间上的分配也多偏向于应用的功能与体验上,特别是分布式服务器系统,一个完整的密钥管理模块使用和维护成本较高,所以,这种算法不适用或者说只能作为移动应用安全加密的备选方案.

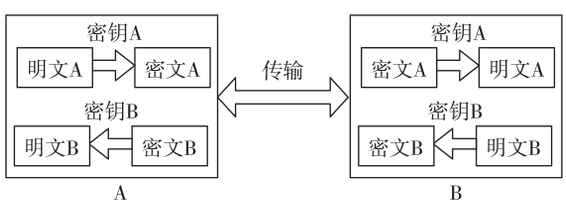

图 1 对称加密算法示意

Fig. 1 Diagram of symmetric encryption algorithm

2 非对称加密算法

非对称加密算法[2]工作示意如图 2 所示.相对于对称加密算法,它的安全性要高得多.它的另一个名字叫“公开密钥加密算法”.在移动应用中,终端与服务器各自持有对方的公钥,发送数据使用对方公钥加密,收到信息使用自己的私钥解密,服务器只需为每个用户生成一对公私钥即可.安全性和易用性虽然得以加强,但是在移动应用设计中使用非对称加密算法由于加密强度高且算法复杂,其效率远低于对称加密算法,甚至在服务器并发数达到一定程度后,极端状况下效率仅为对称加密算法 1/1 000.移动应用的特点是紧跟现代生活的快节奏操作,应用的平均响应时间如果高于用户的预期,就没有用户体验可言,应用的生存也就无从谈起.且由于近期曝出广为流行的网络加密软件 OpenSSL 存在“心脏出血”漏洞[3-5],即用欺骗性的 HeartBeat 数据包可诱使服务器返回小块内存数据,这些内存数据可能会包括用户信息、服务器关键信息,也使得很多使用密钥加密解密的数据面临威胁,因为一旦存有密钥的内存块泄露,服务器或用户的与之相关的各种信息就没有了安全保障.当然,在进行了相关漏洞的修复之后,非对称加密算法的安全性与以前无异,其最大瓶颈在于不能在移动应用频繁的请求中达到理想的响应时间.

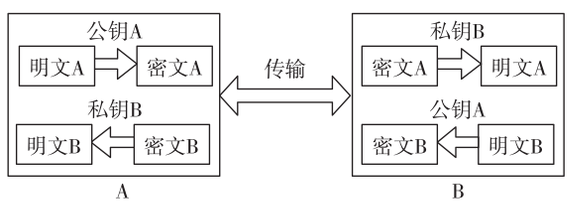

图 2 非对称加密算法示意

Fig. 2 Diagram of asymmetrical encryption algorithm

3 基于授权令牌的安全交互方式

传统的加密技术在移动应用中的安全性毋庸置疑,但并不适用于当前主流的移动应用开发节奏,而本文将采用的设计思想是,在不涉及用户敏感信息的同时去传输那些识别用户的标识(即不传输与用户直接相关的敏感数据),使得其即使被捕获抓取也毫无用处,从而保证交互时敏感数据的安全性.

3.1 OAuth 协议

OAuth 协议是一种授权用户资源的标准.OAuth 协议对于很多移动开发者来说并不陌生,特别是在开发类似新浪微博或腾讯 QQ 相关插件等应用中,必须遵循 OAuth 协议的流程步骤去获取用户的资源.因为 OAuth 授权是给第三方使用的,需要验证第三方厂商或应用的身份,然后厂商根据授权流程去逐步获取 Token,最终得到有效的可用的用户 Token.

OAuth 的最大特点是,在不接触用户账号密码的前提下可以安全地获取用户资源的授权,但是

OAuth 不能在应用系统中直接使用,因为 OAuth 协议提供的是一种安全标准,它并没有固定的实现,新浪、腾讯等互联网企业都有一套自己的 OAuth 实现,这样一套授权系统的主要作用也是授权给第三方应用或厂商.本文设计出一种基于授权令牌的从服务器到终端的交互方式,因不涉及第三方应用,如在云存储这个系统中不涉及到第三方,可以直接跳过授权流程,用户登录后就授权了,所以,其额外的类似于 OAuth 的授权过程便不用考虑,只需关注 OAuth 所使用的核心思想:Token(授权令牌)机制.在 OAuth 中,最终第三方应用拿到的唯一标识是用户的 Access Token,服务器通过对 Token 的处理最后定位到指定用户,确认令牌的有效性,继而进行下一步操作.

3.2 Token 生成策略

在应用中,借助令牌授权的思想,设计一套新的服务机制,首要的工作是设计出一种 Token 的生成策略.

Token 的设计原则是,Token 必须具有唯一性且不可复制.因为"唯一性"显而易见是保证能准确地定位到指定用户,这里的"不可复制性"是指生成 Token 的算法有能力在大量的暴力计算之下抗住足够的压力.

从 Token 的设计原则出发,将采用在数据库表中经常设计为主键使用的 UUID(Universally Unique Identifier,通用唯一识别码)来作为 Token 令牌的生成策略.因为 UUID 是由 32 位十六进制数字组成,它可以保证对在同一时空中的所有机器都是唯一的,并按照 OSF(Open Software Foundation,开放软件基金会)制定的相关标准进行计算,使用硬件的网卡地址、ns 级的时间及指定的芯片 ID 码等各种可能的数字,可以最大程度地保证每一个节点所生成的标识都不会产生重复.根据 UUID 设计的初衷,UUID 不仅需要确保彼此之间是不重复的,而且至少也是与公元 3400 年之前其他任何生成的 UUID 有非常大的区别.所以,使用 UUID 可以作为生成 Token 的首选策略.

3.3 基于授权令牌的安全机制

通过使用授权的 Token,应用便不必每次请求都试图去加密用户的关键信息,现在可以直接使用 Token 向服务器发出请求,Token 的使用及更新流程如图 3 和图 4 所示.其中,图 3 为终端的操作流程,图 4 为服务器处理流程.验证的核心是当一组 Token 验证通过,且该组 Token 的使用次数正常,则判定为合法.验证次数是为防止 Token 被拦截使用,也作为更新 Token 的判断依据.

用户每次登录时,首先默认初始化一个 Token 令牌组,例如,本实例中采用由 2 个令牌(即 TokenA 与 TokenB)组成一个令牌组.服务器在初始化用户 Token 时,默认随机分配一个使用本 Token 的最大次数 p,但 p 不能超过指定阈值(如 50 次,设置阈值的目的是防止随机出特别巨大的次数而增大安全风险),这样用户登录(这里使用非对称或其他加密算法是可行的)后便获取自己的一个 Token 组存储在本地,但应用并不知道这个 Token 组的使用上限 p.应用每次发送请求且使用 Token 时,在本地记录 TokenA 的使用次数 n(用十六进制表示),并与 TokenA 相加,得到新的 TokenA,将它与未做处理的 TokenB 发送到服务器.服务器收到这个 Token 组,再根据 TokenB 查找到服务器对应的 TokenA,同时更新本组 Token 的已用次数 m,如果这时服务器存储的 TokenA+m 与终端请求的 TokenA 值相等,则用户认证结束,若 Token 有效且用户有效,则操作指令生效.同时,为了防止同一组 Token 使用次数太多将被多次监听而产生的风险,在判定当前操作有效的前提下,当 $m \geqslant p$ 时,后台异步启动 Token 更新的操作,重新分配终端的 Token 并重置 n,更新服务器的对应 Token 与 p,同时重置 m.

图 4 中 m、n 的自增完全可以做成动态的,具体自增值可以基于本组 TokenB 第 x 位的值.设计目标是通过无规律的动态 Token 变化增强安全性.

在此过程中用户在终端是无需进行多余操作的,无需更改密码便可在使用中不断保护应用安全.若必须发送相关敏感信息如真实姓名等,则可适量使用非对称加密算法进行处理再连同 Token 发送至服务器;对于非敏感信息如应用本身的指令、请求等,则直接发送相关信息和 Token,这样既平衡了服务器的响应速度又达到了安全传输的目的.如果 Token 被挟持或被网络抓包,因识别标识不涉及用户任何敏感信息,大大降低了用户相关账户的风险,而且因为每次请求的 Token 都会根据次数的不同发生变化,那么被截获的当前一组 Token 是已失效的 Token.如果这组 Token 被伪装成用户访问的形式再次向服务器请求,那么,服务器验证 Token 有效性时便会察觉数据的异常,将拒绝请求并触发用户安全模块,使用电邮或手机短信的方式对相关用户进行安

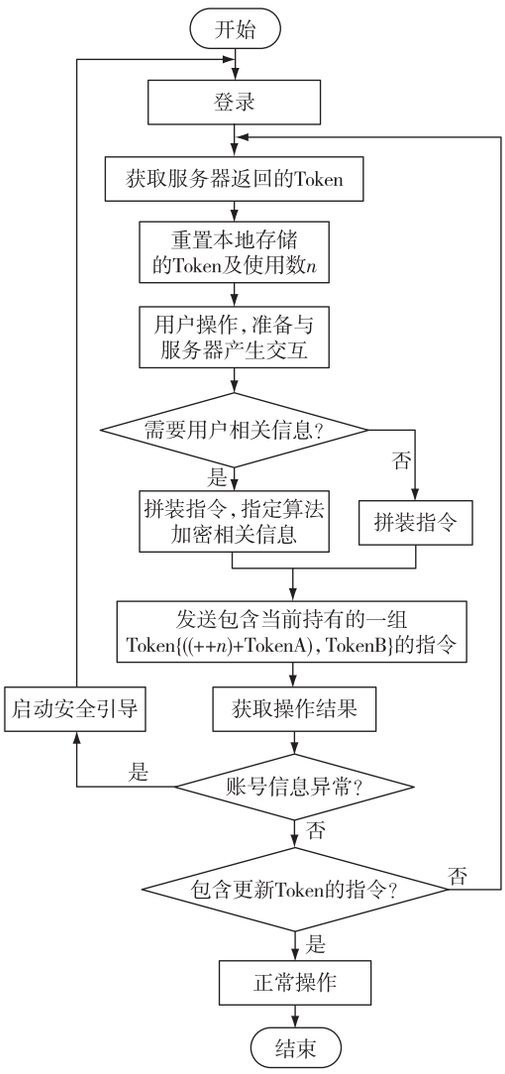

图 3　智能终端 Token 使用及更新流程

Fig. 3　Flow chat of using and updating Token at the intelligent terminal

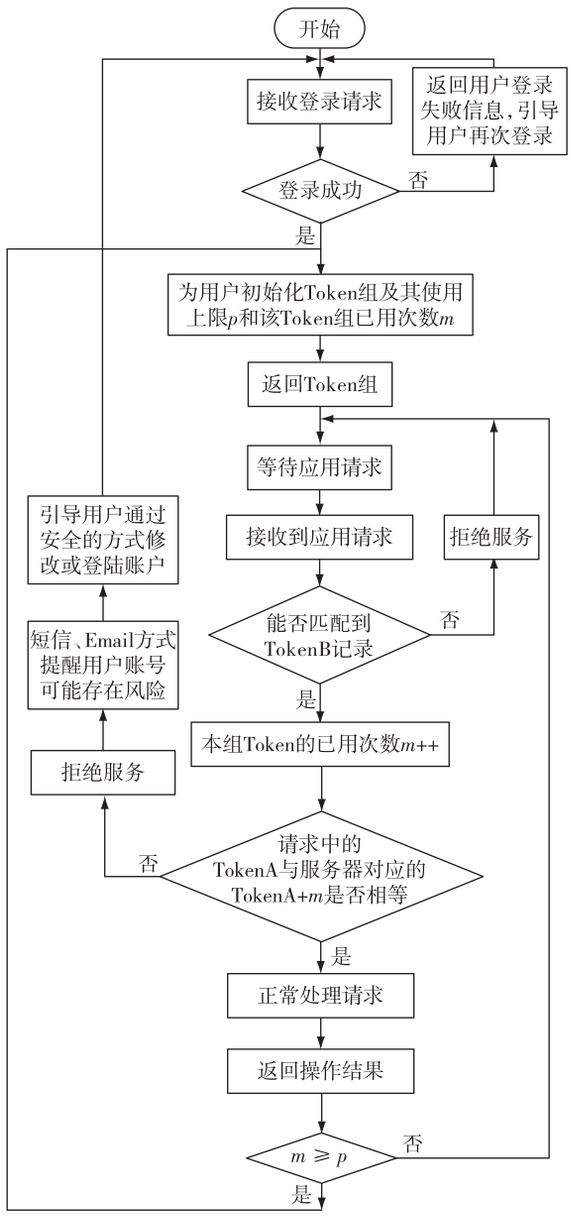

图 4　服务器验证 Token 及更新流程

Fig. 4　Flow chat of verifying and updating Token in the server

全引导.由此可见,由于 UUID 特殊的生成机制,并结合使用一组 Token 的组合验证方式,使用暴力破解的尝试性请求基本无成功可能.

4　压力测试

本文设计使用 Token 令牌授权机制并应用在云存储终端,对于移动应用来说,其服务器对于请求的响应速度是非常重要的参数之一,因而在新的安全交互机制下,测试服务器是否能够达到理想的响应速度.

本部分主要测试服务器在解析多个包含完整的有效的 Token 组请求时,其并发性能是否可以达到理想状态,响应速度是否可以保持在可控范围内(< 30 ms).因此,通过 Java 线程与网络模拟访问的方式进行了一系列压力测试,具体结果如表 1 所示.其中,QPS 是每秒查询率,2XX 延时是指成功请求延时,服务器 idel 是指服务器空载率.

表 1　Token 请求方式时服务器压力测试结果

Table 1　Stress test result in the sever when sending request by Token

并发数	QPS	请求数	2XX 延时/ms	Error 数量	服务器 idel/%
30	302	43 296	<1	8	25
40	451	62 190	<1	11	19
50	533	70 936	3	7	15
100	529	72 963	40	14	13

从模拟数据测试结果看,单个服务器并发数小于50时每秒查询率在500以内,服务器可以保证良好的性能与响应速度,Token机制的授权验证未成为服务器的瓶颈,可以认为服务器在Token机制下能够正常工作.

因受限于设备与测试环境,服务器与模拟请求之间没有带宽瓶颈,且模拟请求使用的都是简单命令,真实请求数目与并发性能可能会略低于此测试数据.

5 小结

本文针对WiFi环境下容易产生传输数据泄露的环节[6-8],以Http交互方式为例,对比了常用的加密算法与使用效果,从OAuth协议中得到启发,重新设计了一种安全交互方式,在保证用户相关信息安全的同时,并未加重服务器的负担,同时在用户的应用存在风险或异常时,可及时提醒并引导用户进行相关的安全操作.本文方案在理想状态下测试是可行的,但是由于真实网络环境千变万化,网络不通或者网络质量不佳的情况时有发生,如果中间有正常请求未送达,由于终端与服务器的数据不一致性,下次请求便有可能触发安全模块保护,造成"狼来了"的假象,所以,下一步工作是继续完善这套新的安全交互方式.

参考文献

References

[1] IETF.The OAuth 2.0 authorization framework[EB/OL].[2014-04-30].http://tools.ietf.org/html/rfc6749

[2] Stinson D R.密码学原理与实践[M].2版.冯登国,译.北京:电子工业出版社,2005
Stinson D R.Cryptography theory and practice[M].2nd Ed.Translated by FENG Dengguo.Beijing: Publishing House of Electronics Industry,2005

[3] 安慧科技.OpenSSL心脏出血漏洞全回顾[EB/OL].[2014-04-25].http://www.freebuf.com/articles/network/32171.html
Anhui Sci & Tech.Review on OpenSSL[EB/OL].[2014-04-25].http://www.freebuf.com/articles/network/32171.html

[4] Lodderstedt T, McGloin M, Hunt P.OAuth 2.0 threat model and security considerations[EB/OL].[2014-03-30].http://tools.ietf.org/html/rfc6819

[5] IETF.The OAuth 2.0 authorization framework: Bearer token usage[EB/OL].[2014-04-30].http://self-issued.info/docs/draft-ietf-oauth-v2-bearer.html

[6] Shrawankar M,Shrivastava A K.Comparative study of security mechanisms in multi-cloud environment[J].International Journal of Computer Applications,2013,77(6):9-13

[7] Wang C,Chow S S M,Wang Q,et al.Privacy-preserving public auditing for secure cloud storage[J].IEEE Transactions on Computers,2013,62(2):362-375

[8] Bessani A, Correia M, Quaresma B, et al.DepSky: Dependable and secure storage in a cloud-of-clouds[C]// Proc of the 6th Conf on Computer System.New York: ACM,2011:31-46

Secure communication between terminal and server in the mobile internet environment

CHEN Yao[1] XIA Liangliang[1] TAN Hui[1]

1 Binjiang College,Nanjing University of Information Science & Technology,Nanjing 210044

Abstract The rapid development of mobile Internet and the continuing decline of intelligent terminal cost attract more and more users,thus the user's privacy and information security becomes significant in the mobile Internet era. In this paper,we analyze the weak link in user's information security in WiFi hotspot environment,then compare the application effect of two conventional encryption algorithms,thereafter propose a secure communication based on token mechanism in OAuth protocol,which can ensure the security of user privacy,keep the server at high response speed,even can timely remind and guide the user in case of information risk or abnormality in the user's application. Finally,this design is applied in the system,and the stress test results show that the proposed design is effective.

Key words mobile internet;OAuth protocol;secure communication