

一种基于安卓系统的短消息加密方法

张晟骁,张 宏,李千目

(南京理工大学 计算机科学与工程学院,江苏 南京 210094)

摘 要:开源移动操作系统 Android,一经问世便因为其开放性以及灵活性,深受用户和厂商的青睐。但由于该系统的自由性,导致恶意软件泛滥。针对这样的情况,文中首先阐述了 Android 平台的一些设计思路并分析该平台中短信收发存在的包括信息泄露、信息拦截和信息伪造等潜在隐私问题,解析了现有的仅使用对称加密算法的短信加密防护手段的问题并研究了“一次一密”的密码学思想,设计了一个基于非对称加密体系的“一次一密”短信加密防护方法。在 Android 终端上实现后,测试了该方法与传统的加密方法的性能差异。得出的实验结果表明,在增加有限的时间开销的情况下,使用“一次一密”的短信加密手段能够有效防止密文与密钥的泄露并且能够在一定程度上抵御重放冲击。

关键词:Android;短信;加密;一次一密;密钥分发中心

中图分类号:TP309

文献标识码:A

文章编号:1673-629X(2015)09-0144-05

doi:10.3969/j.issn.1673-629X.2015.09.031

A SMS Encryption Method Based on Android

ZHANG Sheng-xiao, ZHANG Hong, LI Qian-mu

(School of Computer Science and Engineering, Nanjing University of Science and Technology,
Nanjing 210094, China)

Abstract: Since the open source mobile OS, Android, came out, which has been attracted by customers and manufactures due to its openness and flexibility. However, malicious programs have been widely spread. Due to this, describe the design idea of Android system and potential privacy issues such as message leak, message interception and faking, then analyze the existing SMS encryption system and OTP Cryptography and design a new SMS encryption system based on one-time padding and asymmetric encryption, finally implement the new system on Android platform and test the performance difference between this method and traditional encryption method. The experimental results show that with a limited time cost, the encryption system based on one-time padding is possible to prevent leakage of the cipher text and the key, and it can reduce the risk of replay attack.

Key words: Android; SMS; encryption; one-time pad; KDC

0 引 言

开源移动操作系统 Android,以其独特的开放性和灵活性获得了消费者和制造商的青睐。据统计,2014 年第二季度的全球智能手机市场份额中,Android 平台独占 84.7%,同比增长 33.3%,达到 2.55 亿台。在这样市场占有率的情况下,Android 平台的第三方应用数也达到了历史新高。截止 2014 年 6 月,Google Play 上的 Android 应用数已经超过 150 万个。

然而在这样的成绩背后不能掩盖 Android 平台的一些问题,尤其是因为开放性所导致安全风险越发凸显。360 公司于 2013 年度进行调查与分析,结果表

明:Android 平台 2013 年新增恶意应用 67.1 万个,较 2012 年提高了 4.4 倍,其中隐私窃取类在总数中提升至第三位。隐私窃取是指在用户不知情或未授权的情况下,应用程序自行获取敏感信息。在智能手机平台上,敏感信息主要有短信、通讯录、通信记录、聊天信息以及个人财务信息。由于用户群庞大,安全意识又良莠不齐,Android 平台的开放性反而会导致许多安全隐患。文中主要就是针对现有的 Android 平台短信加密防护系统进行分析与提高,从而更有效地保护用户数据,实现隐私保护。

收稿日期:2014-11-05

修回日期:2015-02-06

网络出版时间:2015-08-26

基金项目:国家自然科学基金资助项目(61272419);江苏省未来网络前瞻性研究项目(BY2013095-3-02)

作者简介:张晟骁(1990-),男,硕士研究生,研究方向为网络与信息安全;张 宏,硕士,教授,研究方向为信息安全;李千目,博士,教授,研究方向为云计算与信息安全。

网络出版地址:<http://www.cnki.net/kcms/detail/61.1450.TP.20150826.1603.062.html>

1 背景知识

1.1 Android 系统架构及组件

Android 系统采用了分层的设计思想,由低到高分别是:Linux 内核、C/C++库和 Android 运行环境、应用程序框架、应用程序^[1]。其中,Android 应用程序中使用了四大组件进行交互。

如图 1 所示,四大基本组件分别为 Activity (活动)、Service (服务)、Broadcast Receiver (广播接收器)和 Content Provider (内容提供者)^[1-2]。

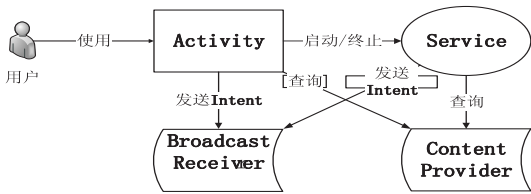


图 1 Android 四大组件交互

Activity 是 Android 最基本的组件,用于呈现用户界面和响应用户操作。一个 Activity 可以理解为一个界面,上面可以包含多个图形控件。一个应用可以包含多个 Activity,相互之间可以进行切换。Service 也是 Android 系统中的一个常用组件,本身没有图形界面,可以用于后台运行数据交互等操作,并与其他组件进行通信。Broadcast Receiver 是用于各个组件间通信的组件。系统的许多事件也是通过广播来发送,包括开机、接收短信、电池电量不足等。Content Provider 是用于应用程序对外共享数据时使用的。通过实现标准接口后,外部程序就可以访问本应用的数据。大部分应用程序都是通过这 4 个组件进行整合与通信实现。

1.2 Android 平台权限模型

Android 平台为了限制应用程序的行为,提出了应用程序权限机制。系统直接使用其 Linux 内核中的用户权限机制,将访问网络、发送短信等权限设计成为相应的用户 group。当一个 Android 应用运行时,系统会分配给应用一个 Linux 用户账户来运行应用,同时系统会读取应用的 AndroidManifest.xml 文件,根据 xml 文件中的权限将应用用户加入相应的 group,从而实现权限的获取。

Android 系统已经声明了约 100 个左右的内置权限,包括与短信和通话相关的一些系统权限。以短信通信相关权限为例,包括 SEND_SMS (发送短信)、WRITE_SMS (编写短信)、RECEIVE_SMS (接收短信)和 READ_SMS (读取短信)。任何应用在安装时都需要对所需要的权限进行展示,并且需要用户审核后才能执行安装。应用程序在安装以后无法修改其权限范围,如果强行越权执行的话,系统会强制关闭应用。

1.3 Android 平台的短信 API

实现 Android 平台短信收发功能首先需要的是前

文提到的相应权限。获取权限后,发送和接收短信主要是利用系统相关接口。在 Android 平台上发送短信主要是依赖应用程序框架中的 Android.telephony.SMSManager 类。SMSManager 类中包含的 sendTextMessage 方法,可向指定手机号发送文本短信。实现接收短信功能有两种方法。第一种,接收短信时,因为系统在硬件接收短信后会发布一条包含此条短信的所有信息的广播 (Broadcast),应用仅需实现一个接收此类广播的接收器,能在短信接收时处理短信内容。第二种,应用直接读取系统短信数据库,获得短信内容。

2 问题分析

2.1 Android 系统短信隐私

由于一般用户缺乏相应的安全意识,在软件安装时不会关注应用安装器提示的权限请求。恶意软件在传播过程中利用这一点,额外申请访问敏感信息和发送消息等权限。所以,一般的短信以及通话信息对于恶意软件来说都可以直接访问到。若配合其他权限,比如访问网络、发送短信时,用户的隐私数据将会直接泄露给第三方。

2.2 基于对称加密的解决方案

目前针对 Android 平台,已有能够实现短信加密的完整方案^[3]。如图 2 所示,通信双方使用对称加密算法,事先约定一个密钥。发送方发送时,调用应用程序框架,直接向接收方发送加密后的密文。接收方直接读取短信并解密获得明文。

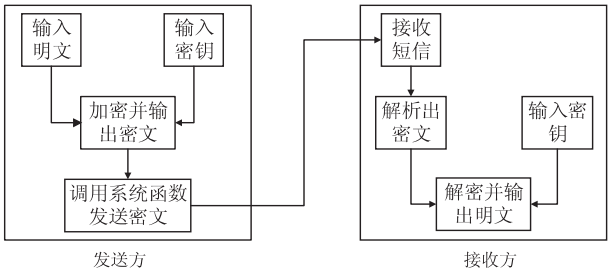


图 2 对称加密方案

现有方法,优点是需要的成本低、开发速度快,但是缺点也很明显:密文可能被截取,被其他应用读取,多次拦截后存在被破解的风险。同时,用户需要准备大量密钥来与不同用户通信,不能实现系统自动匹配。在分发密钥过程中,也存在密钥泄露的风险。针对这样的情况,需要设计一种密钥自动管理、明文自动保存并且密文需要有较高安全性的方案。

3 解决方案

3.1 一次一密

一次一密 (One-Time Padding) 是 1917 年由 Major Joseph Mauborgne 和 AT&T 公司的 Gilbert Vernam 发明

的。所谓一次一密是指在传输中加入了不确定因素,使每次传送信息都是用不同的密钥,使密文在短时间内不可能破解。

定义1:一次一密加密算法。 n 个字符长度的密文为 $M = (m_0, m_1, \dots, m_{n-1})$; 使用同样 n 个字符长度的随机密钥为 $K = (k_0, k_1, \dots, k_{n-1})$; 最后获得的密文为 $C = (c_0, c_1, \dots, c_{n-1})$ 。其中加密算法如下:

$$C_i = E_{k_i}(m_i) = (m_i + k_i) \bmod 26, 0 \leq i < n \quad (1)$$

算法中的密钥 K 需要使用随机序列,长度与原文一致,并且每段序列只能使用一次以保证加密强度。为了保存密钥,当时主要使用一次一密乱码本。此密码本是一个足够大的不重复真随机密钥字母集,这个密钥字母集被收入成册进行保管。每个密钥字母仅对一个明文字符使用一次,接收双方会一次销毁使用过的密钥。新的消息则用乱码本的新的密钥加密。如果信息偷取者没有整本密码本的话,该方法将无法破解。若使用暴力穷举密钥进行解密则会得到相应数量的伪明文,无法判断数据的真实性。在现阶段几乎是无法破解的加密方法。

但该方法有许多缺陷,比如密码本需要产生大量的随机数据,保存困难而且使用时顺序不能错误。所以仅仅是理论上完美的加密算法。现实中也仅在低速率信道上使用。

3.2 基于 KDC 思想的一次一密设计

现如今,基于权威认证中心的公钥交换体系是信息安全的核心与关键技术,旨在减少交换密钥时所面临的风险。同时,基于该技术的 Kerberos 协议,可以实现用户认证和通信管理等功能。Kerberos 协议的核心是认证服务器和密钥分发中心,通过这两个组件,用户与第三方服务器进行安全的认证与通信。将密钥分发中心(Key Distribution Center, KDC)的设计思路用到文中时,可以实现类似一次一密的功能,从而减少密文在短信渠道中被窃取解密的风险,实现身份认证和隐私保护的效果。

KDC 的主要功能为公钥的管理^[3],每个用户将公钥发送至本系统中的密钥管理服务器,服务器存储用户的公钥,并提供密钥验证和查询。用户间通讯之前,先通过非对称加密来进行用户鉴别,在正常通信时则使用一个随机密钥和非对称加密算法来实现类似一次一密的加密效果。使用该方案能够在保证系统性能的基础上,增加额外的用户验证和一定的随机性。

3.3 系统设计

将 KDC 和 Kerberos 协议的设计思想和安卓系统来整合,在短信流通过程中使用非对称加密来实现用户认证,使用对称加密来提升短信加密的效率。在现有的加密系统中加入第三方作为用户鉴别的手段^[4]。

这里使用一台带有短信猫的服务器作为第三方认证中心。手机与服务器通讯以及手机间互相验证时均使用 RSA 非对称加密算法。手机与手机间正式进行数据交互则使用一个随机生成的临时密钥进行通信,该密钥有生存周期,不能永久使用^[5]。

为了实现此平台,最简情况下需要3个实体,一台连接有短信猫的服务器和两台用于通信的 Android 平台手机。服务器主要功能是管理用户账户信息和公钥、处理客户端用户的注册和验证等操作。由于本系统使用时不依赖于网络,离线状态下服务器就能实现功能,可以减少通过网络传输带来的风险。客户端主要负责各用户的私钥保存、加密解密和保护短信的功能。在实际使用中主要实现两个功能:一个是用户的公钥注册;另一个是两个用户通讯前的用户验证和临时密钥协商。

此外由于短信本身的限制,密钥以及明文的长度和文本编码需要特别指定。考虑到手机本身的性能和文本切片时的系统消耗,当使用 RSA 非对称加密时,文中使用的密钥长度限定为 512 位,即 64 字节^[6]。针对短信文本的编码,所有的密文需要进行 Base64 编码转码。由于 RSA 加密算法的特性,将使用非对称加密时的明文长度限定为 53 字节,超出部分进行切片,不足时填充 0 比特。每条短信只存放一个切片的密文,不再额外填充。最后产生的密文经过一轮 Base64 编码后长度将为 88 个字节,不会超过短信长度上限。

比如,当需要传输用户的公钥时,因为公钥长度为 64 字节,所以需要公钥进行切片。一般文本是按照 53 个字节进行切分,由于公钥长度相对固定,所以将公钥切割成 32 个字节的两段。然后根据顺序,依次填充字符至 53 字节。最后对填充后的文本进行加密、Base64 转码和发送。解密时则根据接收短信的前后顺序进行。

进行通信时使用的对称加密算法可以自行选定,只需将密文控制在短信的长度之内。文中将使用 3DES 加密算法^[7]。

1) 用户公钥注册流程。

用户公钥注册主要是为了用户身份的鉴别和管理。每台手机在注册前,都需要按照标准生成一对 512 位的 RSA 密钥对。私钥自行保存,而公钥则通过用户注册流程传至服务器统一管理。

定义2:服务器公钥为 SPK,其私钥为 SSK。RSA 加密算法调用函数为 $\text{crypt} = \text{rsa}(\text{origin}, \text{key})$, 输入变量为明文 origin 与加密用的密钥 key , 输出为密文 crypt 。解密算法函数为 $\text{origin} = \text{deRsa}(\text{crypt}, \text{key})$, 输入密文 crypt 和密钥 key , 输出明文 origin 。

用户注册主要流程如图3所示。

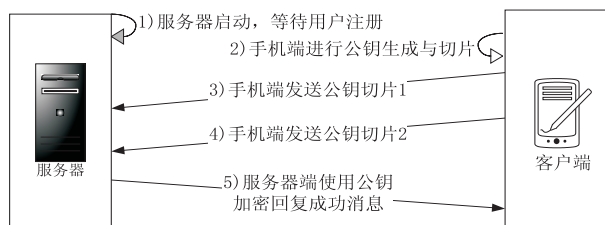


图3 用户公钥注册流程

(1)服务器处于开启状态,等待未登记公钥的用户进行注册。

(2)用户至系统管理员处登记需要使用服务的手机号,然后根据要求生成512位的密钥对,其中公钥为PK,私钥为SK。将PK分成 PK_1 和 PK_2 。对每一个分片使用服务器公钥进行加密,再进行Base64转码,其生成公式如下:

$$T_i = \text{Base64}(\text{rsa}(PK_i, SPK)), 0 \leq i < n \quad (2)$$

(3)客户端发送切片 T_1 至服务器。此时使用服务器的公钥进行加密。服务器进行解密后,判断用户状态,若不符合要求则通知用户联系管理员,并返回(1)。

(4)手机在有效时间内发送切片 T_2 至服务器。服务器若在有效时间内未收到 T_2 ,则回复用户注册等待超时,并返回(1)。

(5)服务器分别解析两条公钥短信后,使用服务器的私钥进行解密,并拼接成完整的公钥。若解密成功,则将此公钥存储至数据库并使用该公钥加密并向用户回复确认信息。若失败则直接通知用户再次尝试并返回(1)。其中用户公钥切片的解密公式见式(3)。服务器接收后只需拼接即可获得完整的公钥。

$$PK_i = \text{deBase64}(\text{deRsa}(T_i, SSK)), 0 \leq i < n \quad (3)$$

2) 用户通信。

用户之间进行通讯时,若一直使用RSA加密算法进行加密,则会因为其时间复杂度等原因将导致性能的下降。此时将通过引入一次一密的思想,在每次通信前双方进行身份验证并产生一个具有有效期的临时会话密钥。在使用期内,通信双方使用临时密钥进行非对称加密^[8-9]。

定义3:通信发起方为A,其私钥为SKa,公钥为PKa;响应方为B,其私钥为SKb,公钥为PKb;会话密钥为SessionK^[10]。

临时会话密钥的创建流程如图4所示^[11]。

(1)A生成一串25字节长度随机值RandomA,并于此随机值后方填充0至53字节。使用SKa进行加密并发送给用户B。

(2)用户B接收到短信后,首先向服务器查询A的公钥。若不能查询到PKa,则返回(1);若接收到

PKa,则用PKa解密短信,提取RandomA。

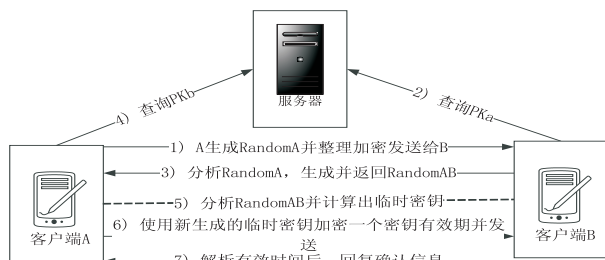


图4 临时会话创建流程

(3)用户B也生成一段25字节长的随机值RandomB。将RandomB接在RandomA后,组成50字节长的RandomAB。在RandomAB后填充0比特至53字节并使用B的私钥SKb加密发送给A。

(4)用户A接收到B的回复后,向服务器查询B的公钥PKb。若无法查询,则返回(1);若能接收到PKb,则解密短信提取RandomAB。比较前25字节的RandomA,若不一致则返回(1)。

(5)用户A与B此时都能获取到RandomAB,根据临时会话使用的算法通过RandomAB生成密钥SessionK。文中使用3DES算法进行加密,则需要3条64位的密钥进行3次运算。RandomAB一共50字节,共400位。提取前、正中以及最后3条,各64位作为3DES算法的 K_1, K_2, K_3 。

(6)用户A使用3DES算法,加密一个密钥有效期,再次使用私钥SKa加密密文,发送给B。

(7)用户B接收短信后,解析有效时间,并使用3DES加密算法加密确认信息,并在私钥SKb加密,发送给A。A解析确认信息后,双方在有效期内将直接使用3DES加密算法与B进行通信。

3.4 与Android系统进行整合

本系统发送短信时直接调用系统API发送密文。接收短信时则通过广播接收器实现:应用将会记录还在有效期内的联系人,接收短信时,拦截这些用户的密文,通过解密存储在私有数据库中。整个系统使用过程中不需要用户进行干预,加密的流程均由系统托管^[12]。

4 测试结果及分析

系统将在Google提供的SDK上进行开发,开发时启用3部Android虚拟机,其中一部模拟服务器操作,另外两部作为通讯双方。服务器端使用Android的Service组件和SQLite数据库,在后台处理短信内容、存取公钥和查询认证等工作。通讯的双方则使用Activity组件进行交互并实现短信的发送。客户端界面如图5所示,与传统的未加密短信发送客户端软件基本相似,包括接收方号码、短信记录、输入框以及发

送按钮。



图 5 系统界面

用户进行发送接收时,后台 Service 将会完成用户认证、加解密等操作。客户端接收到短信后会将内容直接显示在界面上,过程无需用户干预。

测试实验机型为华为 C8813 手机,预装版本为 4.1 的 Android 操作系统,搭载了高通骁龙 MSM8625 双核处理器以及 512 MB 的内存,硬件规格属于中档智能手机。测试时,模拟两个用户在一个密钥有效期内的通信流程,并且逐渐增大短信的发送量。同时对比直接使用固定密钥在发送相同短信量的时间。测试数据使用 Android SDK 的 TraceView 工具。此工具能够统计一个应用中各个组件的调用时间。由于工具的限制,测试时忽略短信在系统中传播的时间,仅计算加密解密和收发的时间。

测试结果如图 6 所示。虽然测试过程中系统忽略了短信收发的时间,但文中方法使用的验证流程相比原有方法增加的时间开销主要是商定临时密钥时的 8 次短信交互,这个损耗的时间相对而言是固定的。根据不同的短信发送量,整个流程中 CPU 增加的运算量约为 3%~8%,比例与发送量成反比。所以,为了平衡效率与临时密钥的安全性,用户可以适当调整密钥的有效时间。

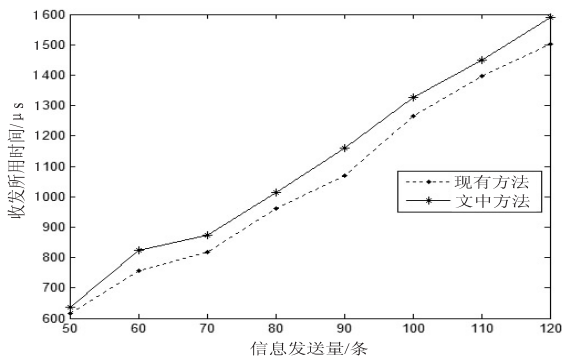


图 6 实验结果

对于整个系统来说,在增加有限的性能和时间损失后,能够大幅增加系统的安全性与保密性,可以在很大程度上解决用户管理密钥与安全性的矛盾^[13-14]。

5 结束语

通过对前文的分析与实验结果,使用文中的一次一密的短消息加密防护方法,能在简化用户操作的同时加强现有短消息加密系统的安全性。使用非对称加密算法进行用户认证加强了通讯的安全性,而继续使用对称加密算法可以在提高安全性的基础上保证通讯效率。然而进行多次加解密还是会存在时间开销,需要设计更加完善的认证流程和随机会话密钥生成算法。所以,该密钥生成流程还需进一步研究和提高。

参考文献:

- [1] 杨丰盛. Android 应用开发揭秘[M]. 北京:机械工业出版社,2010.
- [2] Google. Android SDK[EB/OL]. [2012-06-21]. <http://developer.android.com/sdk/index.html>.
- [3] 朱哲明,赵泽茂,吕金鹏. 基于 Java 语言实现手机短信加密[J]. 保密科学技术,2012(4):52-56.
- [4] 汤阳,张宏,李千目. 一种能量均衡的异构传感网密钥管理协议[J]. 南京理工大学学报:自然科学版,2011,35(6):738-743.
- [5] 徐建,荆文娟,严悍,等. 一种软件体系结构风险评估方法[J]. 南京理工大学学报:自然科学版,2010,34(5):680-685.
- [6] Hasan S, Al-Bakri M L, Kiah M, et al. Securing peer-to-peer mobile communications using public key cryptography: new security strategy[J]. International Journal of the Physical Sciences, 2011, 6(4):930-938.
- [7] Buckingham S. Short message peer-to-peer protocol specification[EB/OL]. 2000. <http://www.smsforum.net>.
- [8] 徐向文. 蓝牙技术中的一种基于 DES 加密的安全策略[J]. 通信技术,2008,41(11):150-152.
- [9] 李玉荣. DES 加密算法及其在 Java 中的实现[J]. 软件导刊,2009,8(4):55-57.
- [10] Das M L, Saxena A, Gulati V P. A dynamic ID-based remote user authentication scheme[J]. IEEE Transactions on Consumer Electronics, 2004, 50(2):629-631.
- [11] Hassinen M, Markovski S. Secure SMS messaging using Quasi-group encryption and Java SMS API[C]//SPLST03. Finland: [s. n.], 2003.
- [12] Rivest R L, Shamir A, Adleman L. A Method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2):120-126.
- [13] Agoyi M, Seral D. SMS security: an asymmetric encryption approach[C]//Proc of 2010 sixth international conference on wireless and mobile communications. [s. l.]: [s. n.], 2010: 448-452.
- [14] 李昭,王跃武,雷灵光,等. 基于动态密钥的 Android 短信加密方案[J]. 中国科学院研究生院学报,2013,30(2):272-277.

一种基于安卓系统的短消息加密方法

作者：[张晟骁](#)，[张宏](#)，[李千目](#)，[ZHANG Sheng-xiao](#)，[ZHANG Hong](#)，[LI Qian-mu](#)
作者单位：[南京理工大学 计算机科学与工程学院,江苏 南京,210094](#)
刊名：[计算机技术与发展](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2015(9)

引用本文格式：[张晟骁](#).[张宏](#).[李千目](#).[ZHANG Sheng-xiao](#).[ZHANG Hong](#).[LI Qian-mu](#) [一种基于安卓系统的短消息加密方法](#)[期刊论文]-[计算机技术与发展](#) 2015(9)