基于可信硬件的智能手机短信加密方案*

马明阳[1]

(1. 上海交通大学软件学院，上海 200240)

摘　要: 手机短信已成为手机应用中双因子验证的常用方法，广泛用于网站登录、移动支付、银行转账等关键应用中。许多恶意软件利用手机系统漏洞来窃听短信，截取用户的验证码，从而对用户的财产安全产生了巨大的威胁。为防御这类恶意软件，许多基于对称密钥、公私钥体制等加密算法的手机短信加密软件已经被开始使用，给用户带来了较大便利。然而此类解决方案并不能保证在智能手机操作系统被攻破时短信明文数据与密钥数据的安全性。本文提出的 TrustSMS 系统通过利用 ARM TrustZone 技术，可以同时保证短信数据在传输过程中与在智能手机操作系统内的安全性。本文在三星 Exynos 4412 开发板上实现了 TrustSMS 的原型系统，实验数据表明 TrustSMS 系统对智能手机操作系统产生的影响极小，性能开销则低于 1%。

关键词: 短信加密；可信硬件；公钥密码体制；移动安全

中图分类号: TP309　　**文献标识码:** A　　**doi:**10.3969/j.issn.1006-2475.

SMS encryption scheme for smartphones based on trusted hardware

Ma Mingyang[1]

(1. School of Software, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: SMS has become a common used method in two-factor authentication, which is widely used for website login, mobile payment, bank transfer and other critical applications. However, many malicious applications take use of mobile operating system vulnerabilities to eavesdrop and intercept SMS for users' authentication code, which brings a great threat on the property of the user security. To defend such malicious applications, many SMS encryption applications based on symmetric key encryption algorithm or public/private key system have been started using, which bring great convenience to the users. However, such solutions cannot guarantee the confidentiality of the SMS plaintext or even the seeds when the mobile OS is compromised. This paper presents TrustSMS(Trusted Short Message Service), a secure SMS encryption scheme by using ARM TrustZone technology. TrustSMS can not only protect the confidentiality of the SMS against a malicious mobile OS, but also guarantee reliable end-to-end SMS transmission. A prototype of TrustSMS is developed on Samsung Exynos 4412. The experimental results show that TrustSMS has small impacts on the mobile OS and its performance overhead is less than 1%.

Key words: SMS encryption; ARM trustzone technology; NTRU public-key cryptosystem; mobile security

0 引 言

当近年来，随着智能手机设备的快速发展与普及，短信正在身份认证、日常交流等领域起到越来越重要的作用[1]。基于用户绑定性强、不需要额外设备、用户广泛拥有、校验成本极低等特点，短信被广泛的应用于双因子验证解决方案中[2]。然而，短信功能的使用却在当前面临着许多安全方面的威胁[3]，如窃听、拦截和篡改等攻击手段。由于短信数据在手机网络中是明文传输的，而 GSM 标准中用于手机到基站连接加密的 A5 算法却极易被攻破[4]，因此，对短信明文数据加密后再进行传输很有必要。有许多解决方案基于对称密钥或公私钥体制的加密算法对短信明文数据进行加密传输[5-8]，这些方案大多基于用户态应用软件实现，依靠加密算法保证短信明文数据的安全性。然而在另一方面，用户智能手机操作系统自身也面临着十分严峻的安全问题。

首先，当智能手机操作系统被攻破时，短信明文数据与密钥数据的安全性都无法得到保证。例如，攻击者可以通过截屏的方式获取包含短信明文数据的屏幕截图[9]。而如果用户用于收发短信的应用程序被篡改，被恶意注入的代码可以隐蔽地将短信明文数据发送给攻击者[10]。此外，一个恶意的操作系统甚至可以直接在内存或持久化存储设备中删除用于收发短信的应用程序文件或短信明文数据以及密钥数据，导致短信收发功能无法正常使用，用户短信数据永久性丢失等严重后果。

针对上述问题，本文提出了 TrustSMS (Trusted Short Message Service)系统：一个可信的智能手机短信加密解决方案，通过采用 NTRU 公钥密码体制实现了在智能手机平台上对短信数据的快速加解密，保证了短信数据在传输过程中的安全性；通过利用 ARM TrustZone 技术[11]实现了短信明

文数据与智能手机操作系统的隔离以及可信的显示、输入功能，保证了在手机操作系统不可信的前提下短信数据的安全性。

1 背 景

1.1 ARM TrustZone安全技术

ARM TrustZone 技术是在 ARM 平台上旨在建立硬件可信的技术，由 ARM 公司在 ARM v6 后期版本开始加入的安全拓展。与设计成固定功能设备并具有一个预定义的功能集的可信计算平

收稿日期：2015-12-14；

基金项目：国家自然科学基金 基于虚拟化与体系结构支持的移动平台系统安全研究（61303011）

作者简介：**马明阳**（1990-），男（回族），黑龙江省哈尔滨市阿城区人，上海交通大学软件学院硕士研究生，主要研究方向：移动安全

台模块(TPM，Trusted Platform Module)不同，TrustZone可以被认为是一种利用CPU作为TPM的更灵活的方法。在常规的正常世界之外，ARM 引入了一个特殊的CPU模式“安全世界”，从而建立了“安全的世界”的概念。此外，ARM TrustZone对状态的划分不仅限于CPU，还包括系统总线，外围设备和内存控制器。当安全世界模式被激活时，安全世界中运行的软件与在普通世界模式下运行的软件相比具有更高的访问权限。这样，一些系统功能，特别是安全功能和加密认证等，可以隐藏在正常世界之外以保证安全性。毫无疑问，这一概念远远比TPM芯片更灵活。

图1清晰地展现了TrustZone硬件体系结构，包括片上系统(SoC，System on Chip)与外围设备连接。SoC 包括一个核心处理器、直接内存访问(DMA，Direct Memory Access)，安全的RAM，安全的引导ROM，通用中断控制器(GIC，Generic Interrupt Controller)，TrustZone 地址空间控制器(TZASC, TrustZone Address Space Controller)、TrustZone 保护控制器(TZPC，TrustZone Protection Controller)，动态内存控制器(DMC，Dynamic Memory Controller)和DRAM，他们通过AXI总线相互通信。通过使用AXI-to-APB桥，SoC可以实现与外部设备的通信功能。AXI-to-APB可以获取当前访问外部设备事务的安全属性，当普通世界下的事务访问一个属性被设置为安全的外部设备时，AXI-to-APB会拒绝该访问。安全RAM与安全ROM利用软硬件机制进行隔离，它们用于存储安全世界中执行的操作系统。核心处理器工作在两种模式下：安全世界与普通世界。TZPC 用于设置外围设备的安全属性；TZASC负责控制对DRAM安全属性的划分，它可以将一部分DRAM设置为安全的，其余部分设为非安全的，如果处于非安全世界的处理器对安全的内存发起访问，该请求会被拒绝。普通世界下的DMA 对安全内存的访问也会被拒绝，这保证了安全内存不会被非安全世界下的软件或硬件访问。中断控制器GIC负责控制所有中断信息，它可以将某些中断设置为安全的，某些中断设置为普通的。

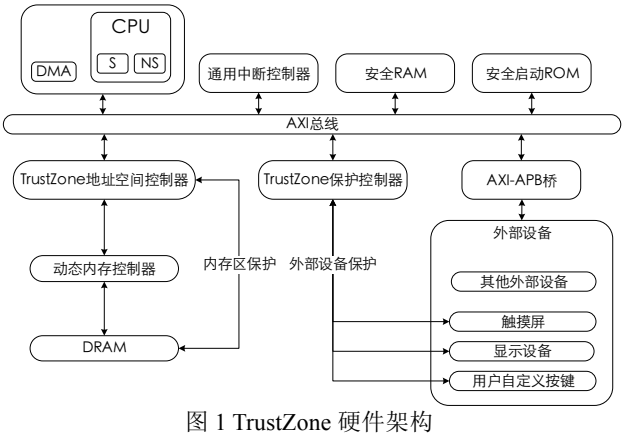

图1 TrustZone硬件架构

1.2 NTRU公钥密码体制

NTRU(Number Theory Research Unit)算法是由布朗大学的三位数学家：J.Hoffstein，J.Pipher和J.H. Silverman在1996年提出的一种全新的公钥密码体制。NTRU算法的安全性基于数论中从一个非常大的维数格中寻找一个最短向量的困难性。由于该算法只使用了简单的模乘法和模求逆运算，因而它具有产生密钥容易，加解密速度快，内存占用低等优点[12]。NTRU算法很好地解决了公钥密码体制加解密效率较低的难题，也十分适合应用于系统资源相对受限的智能手机平台。与RSA算法相比，NTRU的密钥生成速度快200倍，加密速度比RSA-2048快1000倍，解密速度快30倍[13]。

1.3 威胁模型

本文假设运行在普通世界中的手机操作系统极易受到恶意代码的攻击，甚至会被用于直接攻击用户短信数据。

首先，恶意代码可能会试图通过直接读取内存或显示设备数据的方式获取短信数据；第二，它可能通过搜索存储设备的方式来试图获取用于加密短信的密钥数据；第三，它可能通过篡改TrustSMS的代码或控制流窃取短信数据明文。

直接针对硬件的攻击则不在本文的讨论范围内，本文假设攻击者可以物理接触到移动设备，同时运行在安全世界中的代码都是可信的。

1.4 相关工作

目前的手机平台端到端短信加密解决方案大多假设用户持有的智能手机可信，依靠所选取的加密算法的安全性保证短信明文数据的安全性[8-11]。此类解决方案可以较为有效的抵御在短信数据传输过程中针对GSM等传输协议的缺陷所进行的短信包窃听、拦截、修改等攻击手段，却极易受到来自智能手机内部的恶意程序或恶意操作系统的攻击。

本文旨在利用支持ARM TrustZone技术的智能手机设备上的可信硬件，将短信加密系统TrustSMS与易受攻击的普通世界操作系统隔离，从而为用户提供可信的短信数据加解密功能与短信明文数据的显示、输入功能。

2 TrustSMS框架设计

2.1 概 述

本文利用ARM TrustZone技术设计并实现了可信智能手机短信加密系统TrustSMS，其基础框架如图2所示。一个手机Android操作系统运行在普通世界中，TrustSMS进程运行在安全世界的T6操作系统[14]中，包含三个主要组件：基于NTRU算法的短信数据加解密程序，可信显示控制器和可信触摸屏驱动。运行在普通世界Android操作系统内的代理程序负责接收用户输入。当用户长按某个短信密文文本框时，代理程序主动切换至安全世界，执行TrustSMS 程序进行解密操作并将短信明文数据显示给用户。当用户选择发送短信功能时，代理程序先切换至安全世界，等待用户输入短信明文，加密后将短信密文数据传回普通世界的代理程序并继续短信发送流程。

由于 TrustSMS 与普通世界中的操作系统共享相同的显示设备，因此需要一个安全显示控制器来保证短信明文数据的显示安全，一块安全显示缓冲区将被保留以用于存储需要显示的短信明文数据。同时为了支持新建短信的功能，TrustSMS 需要提供可信的触摸屏驱动以便用户进行输入。

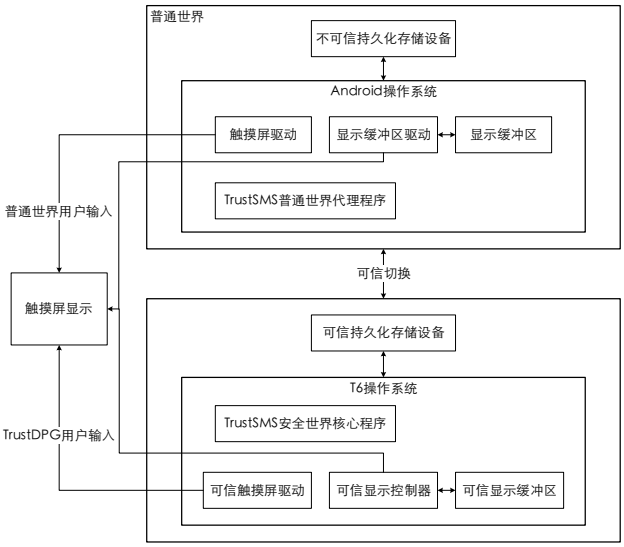

图2 TrustSMS基础框架

2.2 短信数据加解密过程

TrustSMS 中的短信数据加解密功能基于 NTRU 公钥密码体制实现。建立 NTRU 密码体制需要三个整数参数(N, p, q)和四个次数为 N-1 的多项式集合 Lf, Lg, L ψ, Lm。其中 p 和 q 互素，且 q 远大于 p。

2.2.1 密钥生成

先随机选取两个多项式f, g∈L_g，且f关于模p和模q的逆FpFq必须存在，即：

$$F_p * f \equiv 1(mod\ p) \tag{1}$$

$$F_q * f \equiv 1(mod\ q) \tag{2}$$

然后计算：

$$h \equiv F_q * g(mod\ q) \tag{3}$$

由此得到多项式 h 为公钥，多项式 f 为私钥，但在实际中也需要存储 F_p 作为私钥的一部分，记为(f, F_p)。客户端将私钥保存在可信持久化存储设备中，将公钥上传至 TrustSMS 服务器供其他通信方发送短信时用于加密。

2.2.2 加密过程

首先，发送方选择出需要加密的短信明文数据 m，然后随机选择一个多项式 ψ ∈L ψ，从 TrustSMS 服务器获取接收方的公钥 h 后计算：

$$e \equiv p\psi * h + m(mod\ q) \tag{4}$$

e 即为加密后的短信密文数据。

2.2.3 解密过程

接收方收到加密后的消息m后，用自己的私钥(f, F_p)进行解密。首先计算：

$$a \equiv f * e(mod\ q) \tag{5}$$

这里选择的a的系数在-q/2到q/2之间，然后计算：

$$F_p * a(mod\ p) \tag{6}$$

计算结果即为短信数据明文。

2.3 可信显示

TrustSMS 需要一个可信的图形化用户界面以向用户显示解密后的短信明文数据，可信显示控制器负责将显示数据从安全显示缓冲区复制到显示设备上。由于智能手机通常只有一块显卡和一个显示屏，这些硬件设备由安全世界和普通世界共享，因此为了防止短信明文数据被泄露，需要将显示数据缓冲区预先保留一部分作为安全显示缓冲区，并标记为安全状态。系统切换至安全世界后，TrustSMS 首先保存显卡和显示设备的状态，然后重新设置安全显示缓冲区数据为需要显示的短信明文数据，显示完成后清除当前数据并恢复普通世界操作系统的显示状态，最后切换回普通世界。

2.4 可信启动

TrustSMS 在系统启动时被加载至安全世界并一直持续到系统重启或关机，系统启动流程如图 3 所示。手机开机后首先执行只读存储器(ROM)中的代码，将可信引导装载程序从可信持久化存储设备中加载至安全内存，之后可信引导装载程序取得系统控制权并开始初始化安全世界。完成后将 T6 操作系统的镜像文件从可信持久化存储设备中加载至安全内存，再将不可信的引导装载程序和 Android 操作系统镜像文件加载至普通世界的不可信内存。最后，可信引导装载程序将系统 CPU 状态从安全状态切换至非安全状态，并执行不可信的引导装载程序。不可信引导装载程序对普通世界进行初始化并开始启动普通世界的操作系统。

当用户在普通世界启动 TrustSMS 程序后，长按已收到的短信文本框或点击新建短信将触发可信切换。此时普通世界的操作系统将被挂起，系统切换至安全世界，根据切换请求解密选中的短信或开始等待用户输入新建短信内容。切换请求基于普通世界 TrustSMS 代理程序主动调用 smc(Secure Monitor Call)命令切换至安全世界实现。

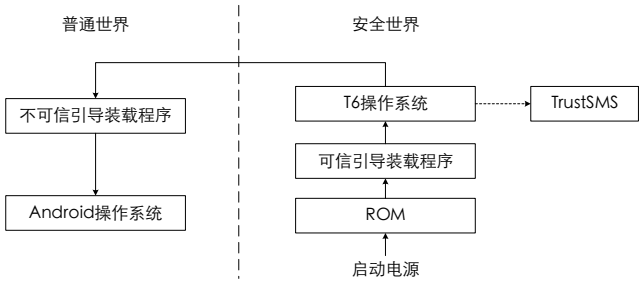

图 3 TustSMS 可信启动顺序

2.5 可信触摸屏

除了显示解密后的短信明文数据外，TrustSMS 还支持对新建短信数据的加密。当用户需要向其联系人发送短信时，首先需要在相关的认证系统注册并获取共享密钥，然后将密钥数据输入到智能手机

中。基于对普通世界中操作系统的不信任，TrustSMS 提供了在安全世界中的输入支持以便于用户输入要发送的短信明文数据。

3 TrustSMS 框架实现

3.1 概 述

本文在一款三星 Exynos 4412 开发板上实现了 TrustSMS 的原型系统，该开发板配备了 ARM Cortex-A9 处理器，CPU 频率为 1.4GHz，运行在普通世界的操作系统是 Android IceCream Sandwich(Android 4.0 版本)，Linux 内核版本为 3.0.2。运行在安全世界的安全操作系统为 T6[14]。TrustSMS 作为 T6 的一个安全应用程序运行。

3.2 可信持久化存储

提供可信的持久化存储功能是 TrustZone 技术的一项关键特性，然而 TrustZone 将具体的实现机制交由厂商决定，因此不同厂商或许会对存储资源的分配与隔离存在不同的设计方案。为了防止某些厂商不提供专用的可信持久化存储设备，我们可以将已有的持久化存储设备转为可信设备。

智能手机通常都装有多种持久化存储设备。例如，智能手机一般将一块 NAND 闪存卡作为主存储设备，将 MicroSD 卡设为扩展存储设备。因此，我们可以将其中一块存储设备设为只允许在安全世界访问，将另一块存储设备提供给普通世界使用。由于在 Exynos 4412 开发板上没有 NAND 闪存卡，但拥有一个 MicroSD 卡接口和若干 USB 接口，因此 TrustSMS 的原型系统将 MicroSD 卡分配给安全世界使用，而将一块 8G 存储空间的 U 盘分配给普通世界使用。在 MicroSD 卡中存储了可信引导装载程序，TrustSMS 可执行程序，不可信的引导装载程序和普通世界操作系统的镜像。在不可信的 U 盘中则只保存了普通世界操作系统的文件系统。由于不可信的引导装载程序和普通世界操作系统的镜像均保存在可信 MicroSD 卡中，系统可以保证即使普通世界操作系统受到了攻击或破坏，它的静态内核镜像仍然是安全的，并且可以在系统重启之后恢复。我们修改了 Android 操作系统的 init.rc 文件以使普通世界操作系统启动时将 U 盘加载为文件系统。

3.3 内存隔离

Exynos 4412 开发板提供了内存标记机制以将安全内存区与非安全内存区隔离，通过多主存多内存接口(M4IF，Multi Master Multi Memory Interface)控制。Exynos 4412 开发板包含两块 512MB 大小的 RAM 空间，M4IF 最高可以设置 256MB 大小的连续安全内存。在 TrustSMS 的原型系统中将 RAM 最高位的 16MB 内存保留为安全内存。显示缓冲区包含 800*480 个像素点，每个像素点由 2 字节 TGB565 值控制。系统在启动时即将 NTRU 密钥数据读入到安全内存中，可以大大减少短信密文数据解密是的时间消耗。

3.4 可信启动

TrustSMS 系统的详细启动过程如图 4 所示。当手机开机时系统首先执行 ROM 中的代码，将可信启动装载程序从 MicroSD 卡中加载至安全内存，可信启动装载程序负责将 TrustSMS 从 MicroSD 卡中加载至安全内存，并将不可信启动装载程序和普通世界操作系统的镜像加载至非安全内存。最后，可信启动装载程序将系统切换至普通世界并执行不可信启动装载程序，初始化普通世界的环境设置，启动普通世界操作系统，随后将 U 盘加载为普通世界操作系统的文件系统。启动时系统会检查可信启动装载程序和 TrustSMS 镜像的完整性，以确保对篡改 MioSD 卡上静态存储数据行为的检测。

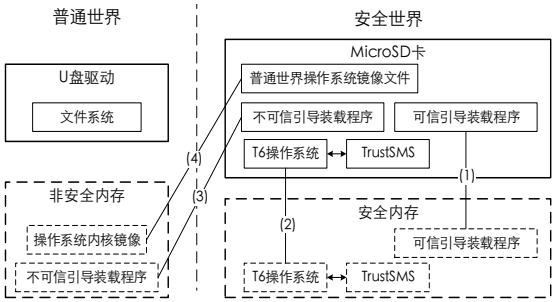

图4 TrustSMS可信引导装载流程

3.5 短信密文加密

TrustSMS 原型系统实现了基于 NTRU 公钥密码体制的短信数据加解密算法，并将 NTRU 开源项目[15]中的数据加解密相关代码移植到了安全世界中。

NTRU 加密算法的函数为 ntru_crypto_ntru_encrypt(), 其声明如下所示：

```
extern uint32_t ntru_crypto_ntru_encrypt(
                uint16_t        pubkey_blob_len,
                uint8_t const   *pubkey_blob,
                uint16_t        pt_len,
                uint8_t const   *pt,
                uint16_t        *ct_len,
                uint8_t         *ct);
```

NTRU 加密函数的各个参数意义如表 1 所示。

表1 NTRU加密算法参数

参数	参数含义说明
pubkey_blob_len	密钥长度
pubkey_blob	密钥数据
pt_len	明文数据长度
pt	明文数据
ct_len	加密后得到的密文数据长度
ct	加密后得到的密文数据

NTRU 解密算法函数为 ntru_crypto_ntru_decrypt()，其声明如下所示：

```
extern uint32_t ntru_crypto_ntru_decrypt (
                uint16_t        privkey_blob_len,
                uint8_t const   *privkey_blob,
                uint16_t        ct_len,
                uint8_t const   *ct,
                uint16_t        *pt_len,
                uint8_t         *pt);
```

NTRU 解密函数的各个参数意义如表 2 所示。

表2 NTRU解密算法参数

参数	参数含义说明
privkey_blob_len	密钥长度
privkey_blob	密钥数据
ct_len	密文数据长度
ct	密文数据
pt_len	解密后得到的明文数据长度
pt	解密后得到的明文数据

3.6 可信触摸屏驱动

由于新建短信时用户需要在安全世界中输入准备发送的短信数据明文，安全世界必须包含一个可信的触摸屏驱动以确保用户输入不会被普通世界操作系统截获或屏蔽。在 TrustSMS 原型系统中，一个四线电阻式触摸屏连接至电源管理集成芯片(PMIC，Power Management Integrated Circuit)，PMIC 监控触摸屏的电压值并将模拟信号转成代表触摸点 X-Y 坐标的数字信号，每次触摸的坐标值存储在 PMIC 的模拟信号-数字信号转换器(ADC，Analogto-Digital Converter)寄存器中，值为零时表示当前没有触摸事件。

当用户点击触摸屏时会触发一个 PMIC 中断，中断处理程序调用触摸屏驱动读取 ADC 寄存器以获取此次触摸的位置信息，触摸屏驱动根据点击位置和当前显示信息执行相应的响应操作。由于触摸屏被普通世界和安全世界共享，因此其拥有两个中断处理程序分别在两种模式下调用。当系统切换至安全世界后，触摸屏驱动被设置为安全状态，此后的所有触摸事件将由安全模式下的中断处理程序响应。当系统由安全世界切换至普通世界时，触摸屏中断将被重新设置为非安全中断。

3.7 可信显示控制器

由于普通世界和安全世界共享 LCD 显示屏，而普通世界操作系统极易受到攻击与篡改，为了防止其通过直接读取显示缓冲区数据的方式获得短信数据明文和密钥数据，安全世界必须拥有独立的显示缓冲区驱动。

TrustSMS 原型系统实现了可信显示控制器，直接在安全世界内设置开发板的图像处理单元(IPU，Image Processing Unit)数据以显示短信明文数据。Exynos 4412 开发板的显示系统主要由 IPU 和 LCD 显示器两部分组成，IPU 负责将数据从显示缓冲区发送至显示设备，LCD 则负责显示相应数据，其使用之前需要被初始化。

在普通世界操作系统启动时，IPU 被设置为非安全状态，可以将非安全显示缓冲区的数据传送至 LCD 显示器。当系统切换至安全世界时，可信显示控制器首先保存 IPU 的当前状态，然后由可信触摸屏驱动对 IPU 进行重新设置，并将其设置为安全状态，开始从预先保留的安全显示缓冲区读取数据。系统切换回普通世界前可信显示控制器会首先清除安全世界的显示数据，随后恢复之前保存的 IPU 状态以保证普通世界操作系统正常运行。

TrustSMS 保留了 RAM 的 16MB 最高位内存作为安全内存，其中 15MB 用作安全显示缓冲区。显示缓冲区包含 800*480 个像素点，每个像素点由 2 字节的 RGB565 值控制。

4 实验与性能评估

4.1 实验环境

TrustSMS 原型系统实现于一款三星 Exynos 4412 开发板，该开发板配备了 ARM Cortex-A9 处理器，CPU 频率为 1.4GHz，普通世界运行的操作系统是 Android IceCream Sandwich(Android 4.0 版本)，Linux 内核版本为 3.0.2。运行在安全世界的安全操作系统为 T6[14]，TrustSMS 作为 T6 的一个安全应用程序运行。

4.2 性能评估

本文首先测试了 TrustSMS 原型系统中短信数据的解密与显示功能性能开销。从用户执行普通世界 TrustSMS 代理程序并长按某条短信密文数据选择解密触发可信切换后，到短信明文数据由 LCD 显示给用户，系统执行主要分为四个步骤，各步骤的具体执行操作和所用时间如表 3 所示，实验选取的短信长度平均为三十个汉字，所用时间为三十次短信密文解密显示过程耗时的平均值。

表3 加密短信解密各步骤所用时间

执行步骤	执行操作	所用时间(毫秒)
1	可信切换	0.0013
2	短信密文解密	16.57
3	可信显示背景绘制	35.61
4	短信明文绘制	5.74

从普通世界切换到安全世界需要 1858 个 CPU 周期，共 1.3 微秒；短信密文解密函数执行平均需要 16.57 毫秒；系统切换至安全世界后，为了便于用户交互，系统会重新设置显示缓冲区首先绘制纯色背景，此过程平均需要 35.61 毫秒完成，这也是整个切换过程中最大的性能开销部分。最后系统根据解密后的短信数据明文将对应的像素点信息写入可信显示缓冲区以完成短信数据明文的显示，此步骤需 5.74 毫秒。

综上，从用户选中短信密文选择解密到短信数据明文在屏幕上显示出来平均需要 57.92 毫秒，而 Android 操作系统的一次触摸屏点击事件响应时间通常需要 60 毫秒以上，用户在使用 TrustSMS 系统时几乎不会感受到延迟。

对于发送短信时的加密过程，用户完成短信明文数据输入后系统会首先请求接收方最新的 NTRU 加密公钥数据，这一网络通信过程的性能开销受网络环境影响较大，在 wifi 网络环境下三十次请求的平均用时为 217.91 毫秒，在 4G 网络环境下的平均用时为 273.22 毫秒。短信明文数据的平均长度为三十个汉字，三十次加密平均用时为 9.12 毫秒。因此用户完成短信明文数据输入后 TrustSMS 系统带来的额外开销不超过 300 毫秒，对用户体验影响较小。

同时，智能手机作为与用户频繁交互的设备，对普通世界操作系统实时性要求很高，因此 TrustSMS 需要保证较低的性能开销以减小对用户体验的影响。本文采用 Android 操作系统测评工具 Antutu[16]对启动时加载了 TrustSMS 的系统与原生的 Android 操作系统性能进行了对比。图 5 为测试结果，得分越高表示性能越好。从图中可知，加载了 TrustSMS 的系统具有较好的总体性能，其性能开销低于 1%。这是由于 TrustSMS 几乎不对普通世界的 Android 操作系统做任何修改，只会对内存访问、外部存储设备访问性能产生极低的影响。

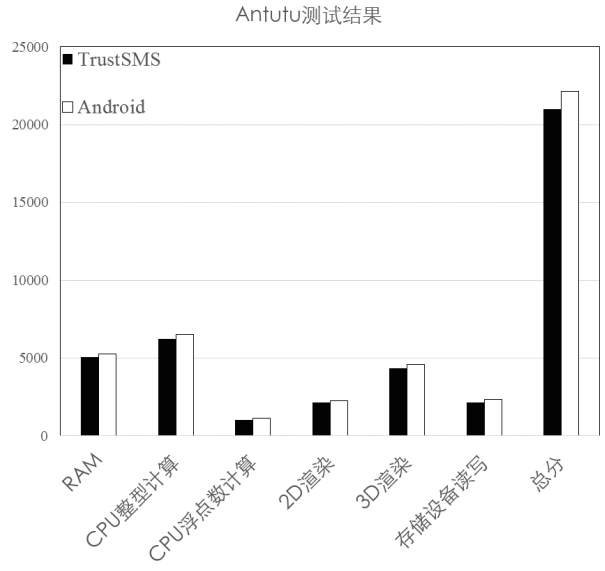

图 5 Antutu 测试结果

5 结 论

本文设计并实现了智能手机平台可信短信加密解决方案 TrustSMS，通过利用 ARM TrustZone 技术，TrustSMS 保证了短信数据在端到端传输中的安全性，同时可以抵御智能手机平台内普通世界操作系统针对短信明文数据与密钥数据的多种类型的攻击，保证了短信数据的加解密功能与显示功能正常、可信。TrustSMS 解决方案不需要对普通世界操作系统做任何修改，针对原型系统的性能测试显示其带来的性能开销低于 1%。

与传统基于对称密钥加密的解决方案相比，TrustSMS 系统由于密钥交换过程的存在，会额外产生 200 毫秒左右的延迟。一个可能的解决方案是参考动态口令生成算法设计一种轻量级的具有足够安全性的短信加密方案，以减少对通信和计算资源的消耗。同时本文的未来工作还将着眼于实现安全世界中不同安全进程之间的隔离以防止某些存在恶意的安全进程对 TrustSMS 进程进行攻击，一个可能的解决方案是将不同的安全进程运行在不同的 Linux 容器中[17]。

参考文献：

[1] Amani N. Using SMS as a business communication tools for SMES[J]. 6th Regional innovation system and innovation clusters in Africa Tanzania, 2009.

[2] Symantec.Whitepaper:Two-factor Authentication:A TCO Viewpoiont[OL]. https://www4.symantec.com/mktginfo/whitepaper/user_authentication/whitepaper-twofactor-authentication.pdf.

[3] BABU K S, SALEEMS A F. SMS Encryption for Mobile Communication[J]. 2013.

[4] Mueller R, Schrittwieser S, Fruehwirt P, et al. Security and privacy of smartphone messaging applications[J]. International Journal of Pervasive Computing and Communications, 2015, 11(2).

[5] Rayarikar R, Upadhyay S, Pimpale P. SMS encryption using AES Algorithm on Android[J]. International Journal of Computer Applications, 2012, 50(19): 12-17.

[6] Agoyi M, Seral D. SMS security: an asymmetric encryption approach[C]//Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on. IEEE, 2010: 448-452.

[7] Qi N, Pan J, Ding Q. The implementation of FPGA-based RSA public-key algorithm and its application in mobile-phone SMS encryption system[C]//Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on. IEEE, 2011: 700-703.

[8] Mahmoud T M, Abdel-Latef B A, Ahmed A A, et al. Hybrid compression encryption technique for securing SMS[J]. International Journal of Computer Science and Security (IJCSS), 2010, 3(6): 473.

[9] Lin C C, Li H, Zhou X, et al. Screenmilker: How to milk your android screen for secrets[C]//21st Annual Network and Distributed System Security Symposium (NDSS), San Diego, California, USA. 2014..

[10] Arzt S, Rasthofer S, Bodden E. Instrumenting android and java applications as easy as abc[C]//Runtime Verification. Springer Berlin Heidelberg, 2013: 364-381.

[11] Alves T, Felton D. TrustZone: Integrated hardware and software security[J]. ARM white paper, 2004, 3(4): 18-24.

[12] Shen X, Du Z, Chen R. Research on NTRU algorithm for mobile java security[C]//Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009. SCALCOM-EMBEDDEDCOM'09. International Conference on. IEEE, 2009: 366-369.

[13] Kaur I E A, Singh N. SMS Encryption using NTRU Algorithm[J]. 2013.

[14] T6, a secure os and tee for mobile devices [OL]. http://trustkernel.org/, 2015.

[15] Whyte W, Etzel M, Jenney P. Open Source NTRU Public Key Cryptography Algorithm and Reference Code, 2013[J]. Available under the Gnu Public License (GPL) at https://github.com/NTRUOpenSourceProject/ntru-crypto.→ Cited on, 71.

[16] Antutu Benchmark [OL]. http://www.antutu.com/en/Ranking.shtml.

[17] Booting the Android LXC container [OL]. https://wiki.ubuntu.com/Touch/ContainerArchitecture.