基于 TrustZone 的指纹识别安全技术研究与实现

杨 霞[1,2] 刘志伟[1] 雷 航[1]

(电子科技大学信息与软件工程学院 成都 610054)[1] (保密通信重点实验室 成都 610041)[2]

摘 要 随着指纹识别技术在智能终端设备中的大量应用,指纹技术本身的安全问题也日益突出。为增强智能终端指纹识别的安全性,借助于 ARM TrustZone 安全扩展机制,提出了一种基于 TrustZone 的指纹识别安全保障技术和方法,其为指纹识别程序提供了可信执行环境,以保证其执行过程的安全性并防止恶意程序的攻击。同时,对指纹数据和指纹特征模板进行加密,并将密钥存储在受 TrustZone 保护的安全区域中以防止被窃取。此外,还实现了指纹数据的安全传输通道,以进一步确保敏感数据传输过程的安全性。最后,设计并实现了一个原型系统来验证所提技术和方法的有效性,实验结果证明所提出的技术和方法是可行的。

关键词 TrustZone,指纹识别,安全存储,指纹匹配,嵌入式系统

中图法分类号 TP309.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.7.026

Research and Implementation of Fingerprint Identification Security Technology Based on ARM TrustZone

YANG Xia[1,2] LIU Zhi-wei[1] LEI Hang[1]

(School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)[1]

(Science and Technology on Communication Security Laboratory, Chengdu 610041, China)[2]

Abstract The security of fingerprint technology itself is becoming increasingly prominent with its wide use in intelligent terminal device. With the security extension mechanism of ARM TrustZone, the technique and method of fingerprint identification security are put forward based on TrustZone to enhance the security of fingerprint identification for intelligent terminals. They provide trusted execution environment for the fingerprint identification program to ensure its safety in executing and prevent malicious code attacks. Meanwhile, the fingerprint data and fingerprint feature template are encrypted, the key is put into the secure area protected by TrustZone in order to prevent it from stealing. In addition, a secure channel for fingerprint data transmission is realized to further ensure the security of sensitive data transmission. At last, a prototype system is designed and implemented to verify the validity of the mentioned technique and method. The experimental results verify that the technique and method proposed in this paper are feasible.

Keywords TrustZone, Fingerprint identification, Secure storage, Fingerprint matching, Embedded system

当今社会,移动终端日益普及并朝着智能化的方向发展,在追求功能、便捷、时尚的同时,安全问题日渐突出。传统的密码方式不方便使用并且容易被恶意程序窃取,而随着指纹识别技术的成熟,其在智能终端设备中日益受到青睐。但是,目前对于指纹识别程序和用户指纹信息的保护缺乏有效的手段和方法。

为了保护指纹识别全过程的安全性,防止指纹信息被窃取。本文借助于 TrustZone[1] 空间隔离技术打造了一个可信的指纹处理执行环境和安全的存储空间。TrustZone 技术是从 CPU 内核设计的角度集成了安全控制机制,将安全保护措施集成到处理器内部,隔离所有 SoC 硬件和软件资源,使它们分别处于两个区域:处理关键资源的安全隔离区域和运行普通操作系统(如 Android、Linux 等)的普通区域。TrustZone 的硬件架构旨在提供安全框架,从而使设备能够抵御将遇到的众多潜在威胁。支持 TrustZone 的 AMBA3 AXI 总线构造中的硬件逻辑可确保普通区域组件无法访问安全区域的资源。将敏感资源放到安全隔离区域的设计以及在拥有安全扩展的处理器内核上运行安全可信的操作系统,可以有效地抵御众多潜在的攻击。

本文根据生理特性的指纹识别思想,结合嵌入式系统的安全扩展技术,设计并实现了一种基于 TrustZone 的指纹识别安全保障技术,提出了一种软硬件结合的指纹识别安全解决方案。本文通过借助 ARM 处理器的安全扩展技术,将系统划分为运行关键系统资源的安全环境和运行其他系统资源的非安全环境,将安全性要求较高的指纹识别程序运行于隔离区域所提供的可信执行环境中,保证了指纹识别的过程安

到稿日期:2015-05-28 返修日期:2015-08-17 本文受国家核高基重大专项(M1401060112ZX0103301),国家科技支撑计划(2012BAH44F00)资助。

杨 霞(1978—),女,博士,副教授,主要研究方向为嵌入式系统、可信计算、嵌入式操作系统安全理论和技术等,E-mail:xyang@uestc.edu.cn;刘志伟(1988—),男,硕士,主要研究方向为嵌入式操作系统安全理论和技术;雷 航(1960—),男,教授,博士生导师,主要研究方向为嵌入式系统。

全;并对指纹数据、指纹特征模板等敏感数据进行加密,然后将密钥存储在受 Trustzone 保护的安全存储空间中,保护了敏感信息的安全性。

1 相关研究

目前嵌入式平台上应用的指纹识别技术主要是运行在一个功能单一且封闭的系统之上,如门禁系统、指纹考勤系统。而对于智能终端的指纹识别而言,则主要是通过在智能终端的指纹图像传感器内置指纹识别芯片,来完成对指纹图像的采集、预处理、特征提取及匹配等全部操作。这些指纹操作全部运行在智能终端的 Android 普通环境下,而对于指纹这种敏感且唯一的生物特征信息而言,其若没有得到强有力的安全保护,随时都有可能被其他恶意的应用程序所截获。

文献[2,3]通过采用移植 Linux 到嵌入式设备并外接指纹识别模块来构建一个嵌入式指纹识别系统,但是对敏感的指纹数据没有任何保护,仅仅是具有了指纹识别功能;文献[4]在指纹处理的过程中考虑到指纹数据的安全性,将指纹生物特征与加密技术相结合,在一定程度上保护了用户的指纹数据,但由于指纹识别程序与其他应用程序运行在同一个系统环境下,因此在实际的指纹采集、预处理和匹配的过程中,随时都存在被潜在的恶意程序所截获的风险。

ARM 于 2004 年推出了一种安全解决方案即 ARM TrustZone安全扩展。该安全架构通过以下方式来确保系统的安全性:

(1)安全区域和普通区域的隔离,可确保安全区域能够抵御来自普通区域的众多潜在攻击。

(2)虚拟化处理器核心,从而提供了两个虚拟核:一个用于运行安全环境,另一个用于运行普通环境,并增加了两个虚拟核上下文切换的监视器模式,普通处理器核心只能访问普通的系统资源,而安全处理器核心可以访问系统中所有的资源。

(3)严格控制普通或安全执行环境与监视器模式的切换,通过配置协处理器的安全配置寄存器(Secure Configuration Register,SCR)的 NS 位,来标识当前执行环境是否安全。而进入监视器模式的具体方法是可以通过执行一条专用指令 Secure Monitor Call(SMC),或者通过一些硬件异常机制,另外也可通过配置将发生 IRQ、FIQ 以及数据中止异常的情况跳转到监视器模式。

从目前的智能终端市场来看,基于 ARM TrustZone 技术的智能手机占据了主流的手机市场。例如:三星的 Galaxy 系列、iphone 的 Secure Enclave 以及高通的大部分处理器都已采用该项技术,足以证明其安全性已获得普遍认可。

针对文献[2,3]中存在的问题,为了在实现指纹识别功能的前提下同时保证用户指纹信息的安全性,提出一种基于 TrustZone 的指纹识别安全解决方案。通过借助于 ARM TrustZone 的安全扩展机制实现了安全区域和普通区域的隔离,其中将安全区域作为可信执行环境(Trusted Execution Environment,TEE),运行特定的安全操作系统(T-OS)[5],对敏感的指纹数据资源执行加密、安全存取以及特征匹配等操作,有效地解决了用户在使用智能终端操作敏感数据资源时,尤其像作为个人唯一生物特征的指纹信息被恶意应用程序所劫获的问题。

2 安全的指纹识别框架

本文在前期工作中借助于 TrustZone 的空间隔离机制和国际安全芯片组织 GlobalPlatform 所发布的可信执行环境(TEE)标准[6],提出了一个可信操作系统框架 Trust-E,并研究、实现了一个运行于 TEE 中的安全操作系统(T-OS)[7]。T-OS 包括安全内核、安全文件系统、安全通信机制、安全 GUI,以及 TUI 等模块。本文基于 Trust-E 安全框架和 T-OS 提出了一个安全的指纹识别框架,实现了指纹识别关键代码与普通代码的隔离运行,同时将敏感指纹数据与普通数据的存储空间隔离,有效地阻止了恶意程序对指纹信息的窃取,并保护了指纹处理过程的安全性。

2.1 指纹识别安全框架设计

基于 TrustZone 的智能终端指纹识别安全系统框架(见图 1)具体包含 6 大部分:运行于普通区域的指纹识别程序、运行于隔离区域的指纹识别安全服务、隔离区域上的加密模块、隔离区域上的安全存储模块、指纹图像数据的加密传输通道、负责两空间区域切换的监视器程序。

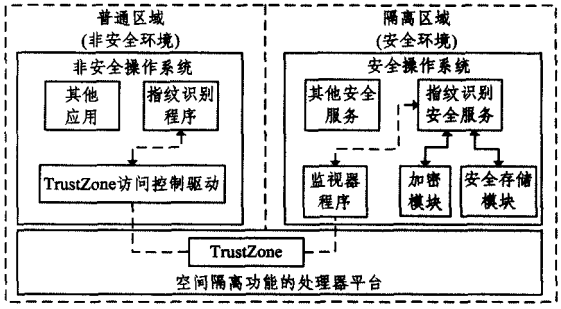

图 1 指纹识别安全系统框架

下面详细阐述每部分的功能和处理过程。

(1)指纹识别程序

指纹识别程序运行于普通环境,此普通环境是指智能终端的操作系统(如 Android 操作系统等)。由于该非安全环境可以安装来自安卓应用市场以及其他不受信任的应用程序,因此该系统的安全性并不能得到保证。运行在此环境上的指纹识别程序通过指纹图像采集器获得用户的指纹图像信息,并向安全操作系统发出请求以获取其内部存储的共享密钥,之后利用 AES-256 对称加密算法对其去头后的指纹图像数据进行加密处理,通过 Android 与 T-OS 之间的安全通信机制将密文传输到安全操作系统内部,再使用安全文件系统对其进行存储。

(2)指纹识别安全服务

指纹识别安全服务运行于隔离区域,此处的隔离区域是指运行在受 Trustzone 保护的安全操作系统(T-OS)之上,该系统主要完成安全设备管理、任务管理、安全隔离的内存管理,还提供了安全的文件存储机制和通信机制;并根据非安全环境的服务请求标识判断请求的安全性,创建相应的安全服务进程。运行在此环境上的指纹识别安全服务程序用于完成指纹数据的预处理、特征提取、特征模板的加密存取、特征匹配等指纹识别的关键操作,从而确保指纹识别过程安全。

(3)加密模块

设隔离区域上的加密模块,在指纹注册阶段主要为隔离区域提供对指纹特征模板的加密处理,最终将特征模板密文存储

在安全文件系统中，生成一个安全的指纹特征模板库；在指纹匹配阶段为已加密的指纹特征模板提供模板解密功能，将其与采集到的指纹图像特征样板比较，获得指纹匹配的结果。

(4)安全存储模块

隔离区域上的安全存储模块通过挂载 T-OS 的安全文件系统，主要提供 3 个功能：1)为普通区域存储加密指纹数据的密钥，指纹识别程序使用该密钥对指纹数据进行加密；2)为隔离区域存储加密指纹特征的密钥，用于指纹特征的加密存储；3)为隔离区域提供对指纹特征密文的安全存储。

(5)加密传输通道

图 1 虚线部分即为指纹图像数据的加密传输通道，它是普通区域中的指纹识别程序与隔离区域的指纹识别安全服务之间建立的安全传输桥梁，包括加密指纹的密钥和指纹信息密文的传输。

(6)监视器程序

监视器是负责两空间区域切换的监视器程序，该程序运行在监视器环境下，负责安全操作系统与非安全操作系统的运行环境的切换，存储和恢复切换时系统环境的运行状态信息。

2.2 指纹识别安全访问控制协议设计

为确保整个指纹识别系统的安全性，本文设计了一种基于 TrustZone 的指纹识别安全访问控制协议(见图 2)，具体包括 4 个模块：1)Android 端的指纹识别应用程序(FP_App)，其定义了针对安全操作系统(T-OS)的唯一目的 ID 标识(dest_uuid)、获取密钥的命令标识(FP_ENCRYPT)、执行指纹匹配的命令标识(FP_MATCH)；2)TrustZone 驱动模块，其会随机生成唯一用户 ID(client_uuid)和服务会话 ID(session_id)；3)监视器模块，由 monitor_smc_handle 函数负责 Android 与 T-OS 的上下文保存和切换操作；4)安全操作系统(T-OS)，其提供安全指纹识别服务(FP_service)、用于指纹数据加密的密钥(Pkey)以及用于指纹特征加密的公私钥对(Ku，Kr)。

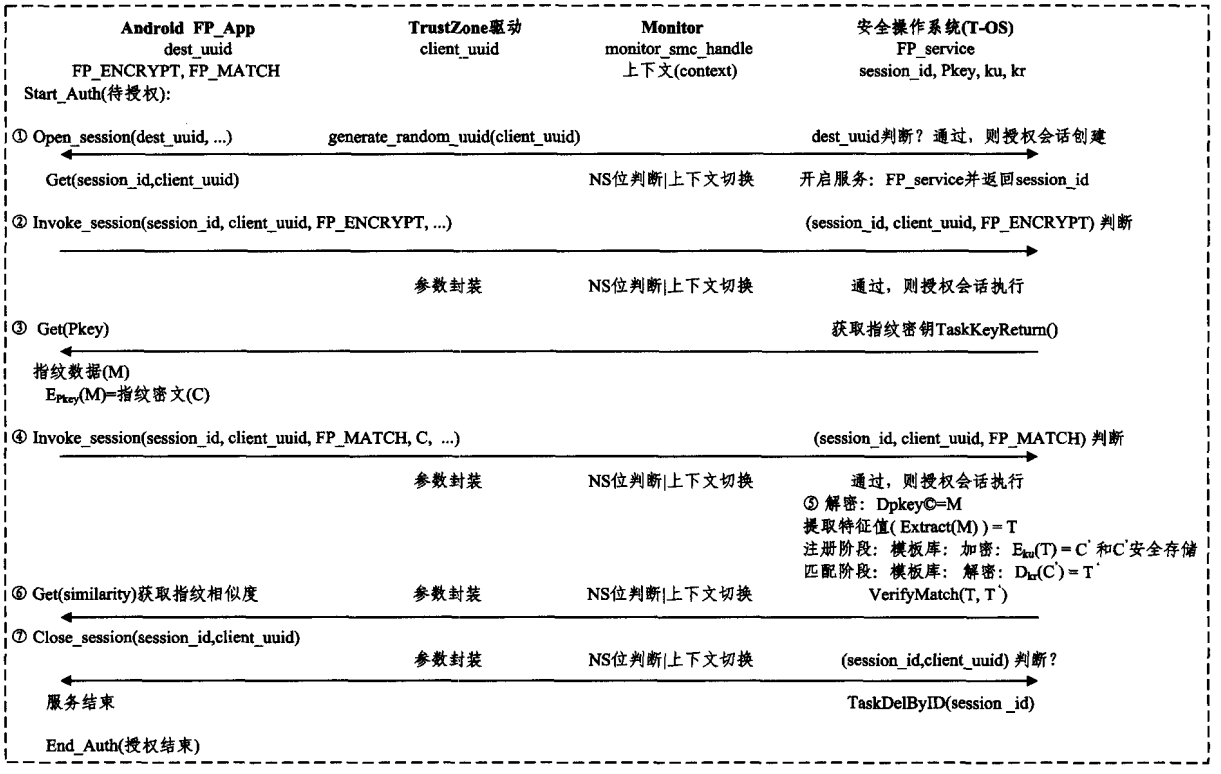

图 2 指纹识别安全访问控制协议

Android 的应用程序如何获得 T-OS 的指纹识别服务程序的服务授权，是该安全访问控制协议的关键所在。其中授权总共涉及以下 3 个环节。

(1)Android 与 T-OS 间的安全服务会话的授权创建

首先，Android 环境下的 FP_App 应用程序通过声明与 T-OS 的指纹识别服务相同的目的 ID(dest_uuid)，来指明自己想要获取的服务类型；其次，调用打开会话函数(Open_session)尝试与 T-OS 建立安全服务会话，在整个会话的创建过程中，TrustZone 驱动会产生用于标识该 FP_APP 的唯一客户 ID(client_uuid)，而 Monitor 模块的 monitor_smc_handle 函数会判断当前的安全配置寄存器(SCR)中的 NS 标识位的值(0 代表安全环境，1 代表非安全环境)以及保存和恢复上下文(context)。最终，T-OS 会根据用户程序指定的 dest_uuid 与系统内部预定的 ID 比较，若相同，则授权 FP_App 应用程序创建该会话并返回成功创建的安全会话 ID(session_id)。

(2)指纹密钥会话的授权执行

FP_App 应用程序根据获得的 session_id 和 client_uuid 以及指纹密钥的命令标识 FP_ENCRYPT 等参数信息，调用执行会话函数(Invoke_session)尝试获得与 T-OS 的安全服务程序(FP_service)执行安全会话的授权。其执行过程会通过执行 TrustZone 驱动模块的 CallTrustZone 函数陷入到监视器环境并由 Monitor 模块切换到安全环境，T-OS 通过对传入参数的判断，最终授权该安全会话执行并根据命令标识 FP_ENCRYPT 调用指纹识别安全服务(FP_service)的获取指纹密钥函数(TaskKeyReturn)，将最终预存储在安全隔离区域上的对称密钥(Pkey)返回给 FP_App 应用程序。

(3)指纹匹配会话的授权执行

FP_App 应用程序用获得的密钥对称加密采集到的指纹数据(M)得到密文(C)，并使用之前生成的 session_id、client_uuid和执行指纹匹配的命令标识 FP_MATCH 等参数尝

试获得 T-OS 的 FP_service 执行安全会话的授权。同样会依次调用 TrustZone 驱动模块和 Monitor 模块，完成安全环境的切换，由 T-OS 对参数 session_id 和 client_id 执行判断，最终授权该安全会话执行并根据命令标识 FP_MATCH 调用 FP_service 的相应处理函数。首先，解密指纹密文得到指纹数据(M)；其次，依次对指纹数据进行特征值提取、模板制作、指纹匹配等操作。其中，为确保模板的安全，对其进行安全加密，并将得到的模板密文(C′)存储到隔离的安全区域中。最终将匹配的相似度结果返回给用户程序 FP_App。而安全会话执行完成后，通过调用 Close_session 函数关闭 FP_App 与 T-OS 的 FP_service 间的安全会话通道，释放已申请的内存空间，同时也避免会话通道被其他程序恶意使用。

3 指纹图像数据的安全传输技术

在非安全环境下，通过指纹图像采集器获取指纹图像数据到传输到安全环境之前，该阶段的指纹数据是极容易被恶意程序劫获的，因此通过 TrustZone 驱动模块在指纹识别程序和指纹识别安全服务之间建立起一个安全的指纹数据传输通道，以保障指纹数据传输的安全性。

3.1 安全通道框架设计

图 3 展示了指纹数据安全传输通道的框架。首先，在非安全环境下，在采集到指纹图像数据后，通过安全传输通道从安全环境中获取密钥，然后采用对称加密算法对指纹图像数据进行加密，接着由监控器将系统的执行环境切换到受 TrustZone 保护的安全环境中；在安全环境下，调用加解密模块对安全通道传输来的指纹数据的密文执行解密操作，进而将获得的指纹数据交由指纹识别安全服务进行指纹识别的具体操作。

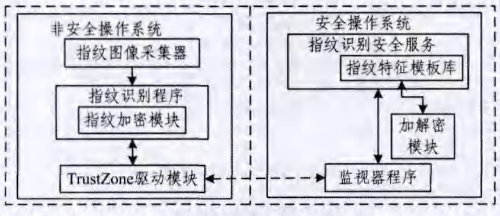

图 3 指纹数据的安全传输框架

安全通道的关键功能由 TrustZone 驱动完成，此模块实现了 Android 与 T-OS 之间的安全通道的会话创建、会话执行和会话关闭功能，并通过 CallTrustZone 函数实现由非安全环境陷入到监视器环境的关键操作。

3.2 指纹信息安全通道的传输

指纹信息安全通道的传输主要包括两部分操作：从安全操作系统中获取加密指纹数据的密钥；向安全操作系统传输指纹数据的密文。由于两个操作的过程类似，此处以第一个操作为例对安全通道传输过程进行详细阐述，其具体步骤如下。

(1)采集用户指纹图像

运行在普通区域上的指纹识别程序通过用户点击注册或匹配按钮，触发指纹图像采集器采集用户指纹；之后通过读取采集到的指纹图像文件来获取它的文件类型、文件大小以及实际数据大小等文件头信息，以此来判断指纹图像是否采集成功。对成功采集到的指纹图像去掉文件的头部信息，以减小指纹数据加密的代价。

(2)获取安全操作系统的密钥

用于加密用户指纹数据的密钥已经预先存储在安全操作系统的文件系统上，因此需要从安全操作系统内部获取密钥。

该部分密钥的获取涉及到两部分的执行授权(如图 2 的流程①②所示)：Android 与 T-OS 间的安全服务会话创建的授权、指纹密钥会话执行的授权。具体对应指纹识别安全访问控制协议设计部分的(1)和(2)。其中该安全会话在创建和执行的过程中，在通过 TrustZone 驱动模块调用 CallTrustZone 函数陷入到监视器环境时，监视器程序通过读取安全配置寄存器(SCR)的 NS 位来判断得到有来自非安全环境的服务请求，保存当前 Android 的上下文并恢复即将切换到的安全环境的上下文。

(3)加密指纹数据

利用之前从安全操作系统内部获取的密钥，调用 AES-256 对称加密算法对实际的指纹数据进行加密处理，最终得到用户指纹数据的密文。

指纹信息传输的安全通道既能保证获取安全操作系统的密钥过程的安全性，又能保证指纹数据在向安全操作系统传输的过程中不会被潜在的恶意程序截获，从而保证了指纹数据传输过程的安全性。

4 指纹识别安全服务的实现

指纹识别安全服务作为指纹识别过程中最关键的应用程序，运行于安全操作系统 T-OS 之上，其执行环境为 TrustZone 提供的安全隔离区域。

下面分别对指纹识别安全服务框架、指纹预处理、指纹特征值提取、安全存储、匹配等内容进行详细阐述。

4.1 基于 T-OS 的指纹识别安全服务框架

指纹识别安全服务程序的安全服务处理流程如图 4 中的灰色隔离区域所示，该服务首先对非安全环境传输来的指纹信息的密文采用对称加密算法解密，以获取实际的指纹数据；之后对指纹数据依次执行预处理、特征提取、特征模板库的加密存储、特征匹配等一系列的指纹操作。

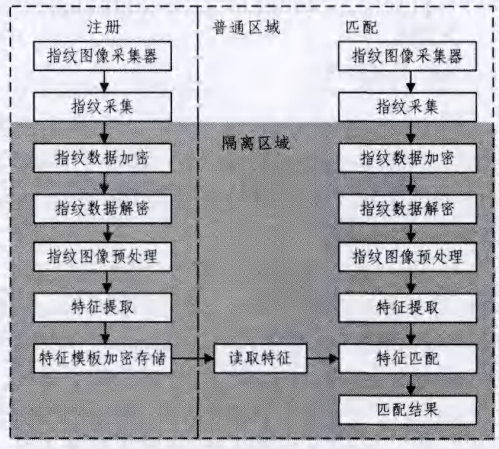

图 4 安全指纹识别系统流程

为保障指纹处理过程的安全性，首先，运行于普通执行环境的指纹识别程序建立一个安全会话通道，之后通过发送 FP_MATCH 指纹匹配命令标识和指纹数据密文等参数到安全操作系统，来完成指纹密文解密、指纹的预处理、特征提取和匹配操作。指纹识别安全服务框架将指纹识别的关键处理

过程与非安全环境的应用程序隔离开来，使其不受非安全环境的干扰，保障了指纹识别的安全性。

4.2 指纹数据的预处理流程

隔离区域上指纹识别安全服务，首先利用解密获得的指纹信息进行指纹数据的预处理操作。指纹预处理的主要原因是通过指纹图像采集器获得的指纹图像并不是完全的清晰、完整，指纹传感器、手指的干湿度以及手指的按压力度等外部因素导致采集到的指纹图像并不是很理想。此时需要对采集到的指纹数据进行预处理操作，去除夹杂的噪点及背景，并对二值化的指纹数据采用纹线细化操作，从而最终获得清晰的指纹纹路。

图 5 为指纹图像的预处理的全过程。

原始数据→计算方向场→前景分割→定向滤波→二值化→去除噪声→清除背景→细化

图 5 预处理全过程

指纹图像的预处理过程较复杂，下面以定向滤波环节为例具体描述。由于外部因素会影响采集到的图像的质量，为了弥补图像质量的不足，借助于求得的图像场的相关信息，采用 Gabor 小波模型[8]在指纹的纹线方向及垂直方向上进行图像增强处理。

采用了高斯函数[9]与三角函数的 Gabor 小波图像增强数学建模，其二维数学表达式如下：

$$H(x,y)=\frac{1}{2\pi\lambda\sigma^2}\exp\{-\frac{(\frac{x'}{\lambda})^2+(y')^2}{2\sigma^2}\}\exp(2\pi\, if\, x') \quad (1)$$

坐标轴的旋转表达式：

$$x'=x\cos\theta+y\sin\theta \quad (2)$$

$$y'=-x\sin\theta+y\sin\theta \quad (3)$$

其中，θ 为旋转因子，即纹路与 X 轴正向的夹角；λ 为坐标轴比例因子，即图像平面的长短轴比例；f 为频率值；σ 为标准差。

4.3 指纹数据的特征提取流程

经过指纹数据的预处理操作后，成功获得了脉络清晰的指纹骨架拓扑结构，借此再通过指纹特征提取模块获取指纹特征，包括指纹的端点、交叉点以及中心点。图 6 为指纹特征提取流程。

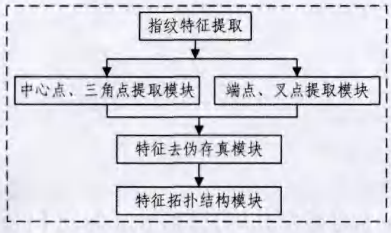

图 6 指纹特征提取全过程

首先根据特征点的设定规则按图像的宽度与高度进行遍历搜索，摒弃边缘点、方向场周围变化剧烈的虚假点；其次统计获得的端点及交叉点的数目；最终将端点及交叉点装配到拓扑结构中，形成指纹特征模板。在经过指纹特征提取操作后，将获得指纹的特征信息。图 7 为根据预处理的细化指纹数据(图 7(a))提取到的指纹特征标记(图 7(b))。

(a) (b)

图 7 指纹特征点标记

指纹特征点标记包括：方框即中心点；单线圈即端点；双线圈即叉点。

4.4 指纹特征模板库的安全存储

在用户指纹的注册阶段，需要将用户的指纹特征作为比对模板存储到安全操作系统的安全文件系统中，用于指纹匹配阶段的特征匹配操作。而指纹特征模板库的制作前提是成功提取到指纹特征，之后会对该指纹特征执行加密操作，保证指纹特征以密文的形式安全地存储在安全操作系统中。

通过调用安全操作系统的加解密模块，使用 RSA-2048 非对称算法对指纹特征模板执行加密操作，由于指纹注册阶段会累计提取用户的指纹特征多次，因此会执行多次该加密操作，从而构成指纹特征模板库文件。利用安全操作系统 T-OS 提供的 yaffs 文件系统，挂载自定义的盘符，创建模板库文件并依次写入固定大小的数据，便成功在该安全文件系统中搭建起指纹特征模板库。

指纹特征的安全存储为指纹特征信息提供了双重的保护，即指纹特征的密文形式存储和受 TrustZone 保护的安全隔离区域。该存储方式有效地阻隔了非安全环境的应用程序劫获指纹特征数据的可能，而且指纹特征密文存储的形式也避免了安全操作系统内部的其他安全服务获得指纹特征数据，从而为指纹特征提供了最大限度的安全保护。

4.5 指纹数据的特征匹配流程

指纹特征的匹配操作主要涉及到两部分：解密读取指纹注册阶段创建的指纹特征模板库、匹配阶段的指纹特征与成功读取的指纹特征的匹配操作。

4.5.1 指纹特征模板库的解密读取

通过挂载指定的盘符，打开并循环读取指纹特征模板库文件中固定大小的内容，确保每次读取一个指纹特征密文，之后通过调用安全操作系统的加解密模块，使用 RSA-2048 非对称算法解密指纹特征密文而获取一个完整可用的指纹特征模板数据，以供指纹特征匹配操作使用。

4.5.2 指纹特征与指纹特征模板库的匹配

将采集到并且成功提取获得的指纹特征数据作为特征样板，将从指纹模板库解密获得的指纹特征作为特征模板，因此指纹特征的匹配就是特征样板与特征模板的匹配。

首先根据特征样板和特征模板的中心点来确定两指纹特征比对的坐标系，之后由匹配程序循环读入模板特征点和样板特征点，通过自定义的特征点的类型、静态角度和半径距离等阈值条件，找到符合的特征点并统计匹配的得分以及匹配

的特征点数量。为了提高匹配的效率，设定匹配特征点的最小阈值，当匹配的特征点数目达到该值时，便会根据统计的得分动态地计算当前指纹匹配的相似度并与设定的匹配成功的相似度阈值比较，若小于该阈值则继续循环匹配，若大于阈值则会停止匹配，将匹配成功的相似度结果写入到 Android 的指纹识别程序分配的内存空间，最终由安全监视器程序切换回 Android 环境，至此完成了指纹识别的所有操作。

5 实验

5.1 实验平台介绍

本文实验的硬件平台为 Cortex-A9(含 TrustZone 隔离区域)的 CES4412 开发平台。软件平台为运行于普通区域的非安全的操作系统(Android 4.0 版本)和运行于隔离区域的安全操作系统(T-OS)。

5.2 实验可行性验证

5.2.1 指纹处理的安全性功能验证

图 8 为 Android 指纹识别程序的指纹采集界面。此过程运行于非安全的 Android 操作系统中。界面中包括指纹注册、匹配、相似度选项，其中相似度的结果由 T-OS 安全服务处理返回得到。在用户指纹匹配之前，需要首先进行用户的指纹注册，其具体的执行流程如图 2 所示。本图仅完成 Android 下指纹识别程序的用户的指纹信息采集、获取指纹密钥会话的授权执行和指纹加密操作，如图 2 所展示的流程①、②、③。在实验的原型系统中，本部分执行流程得到了验证。如图 9 所示，Android 指纹识别程序成功获取到 T-OS 的安全授权并将指纹加密成功。其它流程会在 T-OS 内部调用指纹识别服务程序(FP_service)完成。

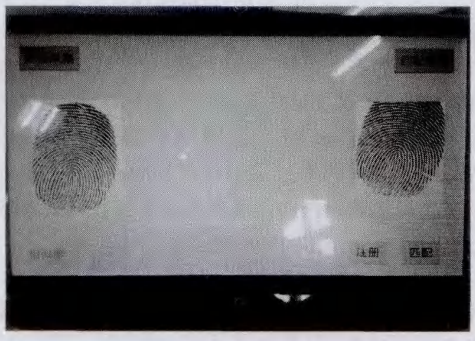

图 8 同一用户指纹采集截图

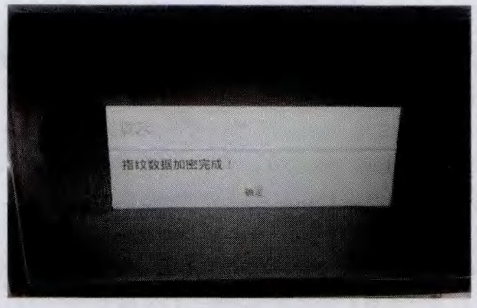

图 9 同一用户指纹数据加密成功截图

5.2.2 基于 T-OS 的指纹识别安全服务功能验证

图 10 为运行于安全操作系统(T-OS)中的指纹识别服务对同一用户的 GUI 指纹匹配结果展示截图，图中结果分别表示指纹匹配的相似度和匹配结果。由于得到的相似度大于相似度阈值，从而得出指纹匹配成功的结论。

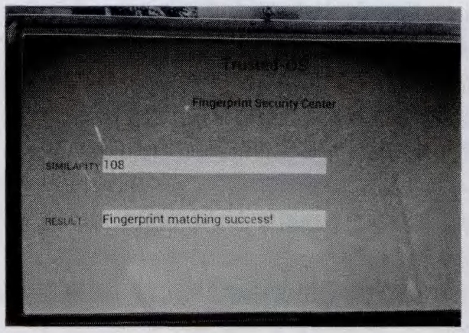

图 10 同一用户指纹匹配结果截图

如图 2 所示，T-OS 通过流程④的安全通道得到指纹密文 C，经过流程⑤的解密处理获得指纹数据 M，将 M 提取到的特征值与指纹模板库进行循环比对，调用图 2 中的指纹匹配函数(Verify_Match)。区别于注册阶段，匹配阶段在流程⑤中，需要解密读取指纹模板库。通过在原型系统中对本部分的指纹匹配的多次执行结果可以看出，T-OS 的指纹识别安全服务的功能是正常的，并且其安全性受 TrustZone 的隔离保护。

5.2.3 指纹敏感信息的安全存储功能验证

在本实验部分，为验证存储的安全性，通过采用多个 Android 应用程序来不断地尝试获取 T-OS 的安全会话授权。由累计的多次实验结果可知，除指纹识别程序外的其他 Android 程序都是无法获得会话授权的，从而印证其在无法获得 sessid_id 和 client_uuid 之前，Android 应用程序是无法获取指纹服务的，并且在没有获得正确的指纹服务命令标识的前提下，指纹识别服务也不会执行。

5.2.4 指纹算法功能验证

本实验共进行了 20 次不同用户的指纹匹配算法验证，表 1 展示了前 6 次指纹识别操作的识别时间和匹配的相似度结果。其中样板 1 同模板 1、模板 5、模板 6 为同一人，匹配时间项的起止时间为成功采集到用户指纹开始到指纹匹配获得相似度的值为止。相似度值为采集到的指纹样板与预存储的指纹模板的匹配相似度结果，其匹配阈值设定为 60，若大于该阈值则说明指纹匹配成功，否则匹配失败。由于本身的指纹素材有限，限制了实验的次数。不过从整个匹配的结果可看出，指纹算法匹配结果与预期相同。

表 1 累计多次的指纹匹配结果

实验	指纹样本	指纹模板	匹配时间(ms)	相似度
1	1	1	1214	115
2	1	2	1193	0
3	1	3	1192	0
4	1	4	1198	0
5	1	5	1123	103
6	1	6	1182	118

5.3 安全性分析

本文所采用的 ARM TrustZone 安全扩展机制和密码学体制能够有效抵御众多潜在攻击，其中包括：对指纹数据攻击、对指纹信息传输过程攻击、对指纹模板库攻击、对指纹识别过程攻击。

(下转第 176 页)

(CSMR). 2012:223-232

[13] Couto C, Pires P, Valente M T, et al. Predicting software defects with causality tests[J]. Journal of Systems and Software, 2014, 93(6):24-41

[14] D'Ambros M, Lanza M, Robbes R. An extensive comparison of bug prediction approaches[C]// 7th IEEE Working Conference on Mining Software Repositories (MSR), 2010. 2010:31-41

[15] Granger C. Investigating causal relations by econometric models and cross-spectral methods[J]. Econometrica, 1969, 37(3):424-438

[16] Chow G C. Econometrics[M]. New York: McGraw Hill, 1983

[17] Zhang M Z, He C Z, Gu X, et al. D-GMDH: A novel inductive modelling approach in the forecasting of the industrial economy [J]. Economic Modelling, 2013, 30(2):514-520

[18] Tamura H, Kondo T. Heuristics Free Group Method of Data Handling Algorithm of Generating Optimal Partial Polynomials with Application to Air Pollution Prediction[J]. International Journal of Systems Science, 1980, 11(9):1095-1011

[19] Liu W, Tian S B. An Improved GMSM Method and its Application[J]. ACTA Automatic Sinica, 1993, 19(4):468-471

[20] Ramakanta M, Ravi V. Software Reliability Prediction Using Group Method of Data Handling[C]// 12th International Conference of Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. 2009:13-19

[21] Couto C, Piresa P. Predicting software defects with causality tests[J]. Empirical Software Engineering, 2014, 19:154-181

(上接第 152 页)

(1)指纹采集阶段的安全性分析

对指纹数据的攻击,是指用户指纹数据被采集并传输到安全操作系统内部之前,被潜在的恶意程序截获。为了避免此种攻击,本文采用通用的加密算法对指纹数据进行加密,以确保本阶段指纹数据的安全,而用于指纹数据加密的密钥则保存在受 TrustZone 保护的硬件隔离区域,该密钥的获取则是通过指纹安全通道进行传输。

(2)指纹信息传输过程的安全性分析

针对指纹信息传输过程的攻击,主要发生在把指纹数据从 Android 环境传输到 T-OS 系统阶段。由于 TrustZone 驱动模块会在内核层申请内存空间,拷贝用户层内存空间的指纹数据,并且整个阶段指纹以密文形式存在,从而保证了本过程的安全性。

(3)指纹模板库的安全安全性分析

指纹模板库作为指纹匹配操作的模板资源,是整个指纹识别过程的基准和依据。指纹模板库由于存储在安全隔离区域中,因此可以阻隔来自非安全环境的安全威胁。此外,为防止 T-OS 内的其他安全服务获取该指纹模板库,通过使用非对称加密算法并结合存储在隔离区域内的密钥,完成对指纹模板库的加密保护。

(4)指纹识别处理程序的安全性分析

针对指纹识别过程的攻击,主要发生在指纹预处理、特征值提取和匹配等阶段,保证该流程的安全可靠是指纹识别服务的关键。本文通过借助于 ARM TrustZone 技术所搭建的指纹识别安全框架,确保了整个识别过程运行在安全环境内部,从而有效地避免了恶意程序对该过程的攻击。

结束语 目前,将指纹识别技术大量地应用于便携设备上,不管是手机厂商,还是互联网企业、移动运营商、银联等,都将会极大地推进移动支付业的发展,而作为安全支付关键技术的指纹识别技术的重要性是毋庸置疑的。保障用户指纹的机密性和指纹处理的安全性是本文研究的目标,本文借助于 TrustZone 的硬件隔离保护机制以及安全加密和安全存储等安全技术实现了对敏感资源操作的安全性保护。其中包括以下 3 个方面的安全:用户指纹数据的安全加密避免了 Android 恶意程序截获用户指纹信息的风险;Android 与 T-OS 间的安全通道确保了指纹密文数据的安全传输;T-OS 内部的指纹预处理、特征值提取与安全存储、指纹匹配操作确保了指纹识别服务过程的安全。通过保障每个环节的安全,保证了整个指纹识别过程的安全。而随着指纹识别技术发展的不断深入,智能终端支付的便捷性和安全性有着一定的研究意义和应用价值。

参 考 文 献

[1] ARM. Building a Secure System using TrustZone Technology [M]. 2009

[2] Wang Dong, Fan Jian-ying. Application Study of Fingerprint Identification Based on Linux Embedded System[D]. Harbin: Harbin Engineering University, 2009(in Chinese)
王东. 嵌入式 Linux 系统在指纹识别中的应用研究[D]. 哈尔滨:哈尔滨理工大学, 2009

[3] Luo Fan, Pei Yi-jian. Research and design of embedded fingerprint recognition system based on ARM+Linux[D]. Kunming: Yunnan University, 2014(in Chinese)
罗凡,裴以建. 基于 ARM+Linux 的嵌入式指纹识别系统研究与设计[D]. 昆明:云南大学, 2014

[4] Shen Yong, Zhu Wen-jing. Design and Implementation of Security-enhanced Scheme for Embedded Database[J]. Modern Electronics Technique, 2010(14):21-24(in Chinese)
沈勇,朱文静. 一种嵌入式数据库安全增强方案的设计与实现[J]. 现代电子技术, 2010(14):21-24

[5] Luo Jing, Yang Xia, et al. Design and Implementation of Security OS based on the TrustZone[C]// ICEMI. IEEE Press, 2013: 1027-1032

[6] Global Platform Device Technology. TEE Internal API Specification[EB/OL]. [2011-01]. http://www.globalplatform.org/specificationsdevice.asp

[7] Yang Xia, Luo Jing, et al. Trust-E: A Trusted Embedded Operating System Based on the ARM Trustzone[C]// UIC-ATC-ScalCom. 2014

[8] Wang Jin-xiang. Gabor Filter based Fingerprint Image Enhancement[C]// International Society for Optics and Photonics. 2013

[9] Wang W, Li J, Huang F, et al. Design and implementation of Log-Gabor filter in fingerprint image enhancement[J]. Pattern Recognition Letters, 2008, 29(3):301-308