(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111563243 A
(43)申请公布日 2020.08.21

(21)申请号 202010357863.X

(22)申请日 2020.04.29

(71)申请人 中国人民解放军海军航空大学
地址 264001 山东省烟台市芝罘区二马路188号

(72)发明人 司维超 顾佼佼 宋超 张杰

(74)专利代理机构 西安研创天下知识产权代理事务所(普通合伙) 61239
代理人 郭璐

(51)Int.Cl.
G06F 21/32(2013.01)
G06F 21/44(2013.01)
G06F 21/46(2013.01)
H04L 9/32(2006.01)
H04L 29/06(2006.01)

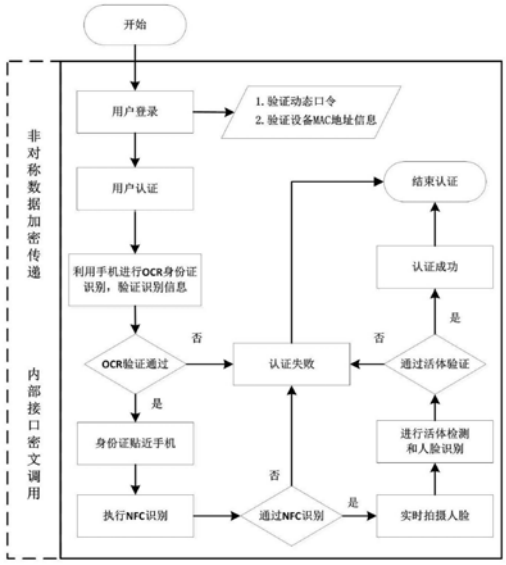
权利要求书2页 说明书11页 附图9页

(54)发明名称

一种基于微信小程序的可信身份认证平台

(57)摘要

本发明公开了一种基于微信小程序的可信身份认证平台,该平台中的数据通信安全机制是平台的基础模块,上层的认证均基于该模块,在用户登录认证、数据传输安全和内部接口密文调用等阶段分别应用了不同的数据安全机制,多维度保护了平台中的数据。采用了实名+实证+实人三重认证机制,其中实名认证初步确保了用户实名安全;实证认证对身份证进行识别,可以确保身份证的真实性;实人认证人脸识别和活体检测,以确保用户为真实活体本人。本发明基于微信小程序开发系统架构,充分利用OCR图像识别、NFC技术、人脸识别、活体检测等新一代身份识别技术,最终使用WXML等语言编程实现客户端和服务器端的设计,搭建手机端的可信任身份认证平台。



CN 111563243 A

1.一种基于微信小程序的可信身份认证平台,其特征在于,综合考虑数据通信安全,分别在用户登录认证、数据传输安全和内部接口密文调用等阶段应用了不同的数据安全机制;包括数据通信安全模块、客户端和服务器端,在保证平台使用安全的前提下,平台通过客户端和服务器端,实现三类不同的认证方式;

第一层次为实名认证,需要对二代身份证正面的人员信息拍照,通过OCR技术识别获取姓名、身份证号的卡面信息进行信息收集,并与后台数据库进行比对;

第二层次为实名+实证认证,在实名认证通过的基础上,利用具有NFC功能的智能设备对证件的合法性进行确认,并以此获取基础可信任根;以基础可信任根为索引,从公安部获取申请人的基础信息、人像比对源信息,集合生物行为识别及生物人脸特征比对的手段来确认申请人证件合法、基本信息有效及本人申请;

第三层次为实名+实证+实人认证,增加了人脸生物特征比对环节,利用活体检测技术,确定采集对象为生物人,以预置的人像/证件照片作为可信数据源进行对比,确定申请人信息的准确性。

2.根据权利要求1所述的基于微信小程序的可信身份认证平台,其特征在于,综合考虑数据通信安全,用户登录认证环节使用了动态口令卡或短信验证码的动态口令验证,以及登录设备MAC地址验证;数据传输环节对登录认证信息采用了MD5加密,对正常使用时的关键数据采用了RSA非对称加密。

3.根据权利要求1所述的基于微信小程序的可信身份认证平台,其特征在于,所述平台客户端的设计使用微信前端开发语言WXML和WXSS,以及脚本语言WXS,在WXML中将按钮绑定到相关函数上,拼接参数后以POST的方式发送JSON请求,通过URL地址发送给AI接口实现功能并回传参数,最后页面重定向到下一页。

4.根据权利要求1所述的基于微信小程序的可信身份认证平台,其特征在于,所述平台服务器端作为MVC设计中的M层,提供了读写数据库到API的封装;平台采用Node.js作为后端开发平台实现对MongoDB数据库的读取,与微信小程序端进行交互并返回JSON数据包的功能,其中使用Express框架搭建服务端;前后端使用HTTPS协议通信,传输JSON格式数据。

5.根据权利要求1所述的基于微信小程序的可信身份认证平台,其特征在于,平台第一层次实名认证中采用OCR身份证识别,绑定一个无参数的函数uploadIdImage函数到上传身份证照片按钮上,先调用微信API拉取相册或者摄像头,获取用户身份证照片后,将其用Base64编码,拼接百度身份证OCR参数后发送至百度AI接口;随后将识别的结果解析,与数据库中存储的内容进行比较,通过后进入下一步。

6.根据权利要求1所述的基于微信小程序的可信身份认证平台,其特征在于,平台第二层次实名+实证认证中,采用NFC近距离读卡,首先测试手机开启NFC功能的返回值并初始化NFC模块,然后完成wx.onHCEMessage的消息监听并发送消息,将shizhengCert函数绑定到读取NFC按钮上;通过南京亿数身份证读取API对身份证进行识别;在身份证芯片验证成功后,将修改flag值,转至人脸活体检测模块。

7.根据权利要求1所述的基于微信小程序的可信身份认证平台,其特征在于,平台第三层次实名+实证+实人认证中,采用人脸识别和人脸活体检测,将uploadRealImage函数绑定到实人认证按钮上;函数调用相机接口,拍照获取图片之后拼接参数,和服务端数据库中预置的身份鉴别图片一起经过Base64编码,发送给百度AI人脸识别API后端进行比对,返回对

比相似度比分，若比分大于等于80，则认证成功，为了检测用户为活体，还需要调用百度活体检测API进行活体检测，若用户按要求完成了预制的动作后，则确认用户为活体。

一种基于微信小程序的可信身份认证平台

技术领域

[0001] 本发明属于计算机技术领域，具体地说，涉及一种基于微信小程序的可信身份认证平台。

背景技术

[0002] 近年来，随着云计算、大数据等现代网络技术的迅速发展，身份认证在民用和军事领域的应用越来越广泛和深入，金融交易、社区管理、快递物流等行业的快速发展对在线身份认证技术的要求也越来越高。但鉴于网络环境错综复杂，网络在线身份认证在给人们带来便利的同时，也成为不法人员进行非法活动的滋生地，如何提高在线身份认证的安全等级，确保用户信息安全是急需进行研究的课题。为此，本文重点研究了在线身份认证安全隐患，针对安全通信潜在威胁设计了多重安全处理机制，一定程度上保证了用户信息传递安全；针对认证处理潜在威胁，以身份证为基础可信任根，根据认证业务需求，实现了“实名+实证+实人”三层安全可信的身份认证；最后实现了一个可信任的身份认证验证系统，用以验证各种安全机制，一定程度上满足了安全需求。

发明内容

[0003] 本发明的目的在于提供一种基于微信小程序的可信身份认证平台，综合考虑数据通信安全的基础上，实现了在用户登录认证、数据传输安全和内部接口密文调用等阶段分别应用了不同的数据安全机制，多维度保护了平台中的数据；在保证平台使用安全的前提下，设计并实现一个可信任身份认证平台。以身份证为基础可信任根，可根据认证业务需求，实现三类不同的认证方式，即“实名+实证+实人”认证。

[0004] 其具体技术方案为：

[0005] 本发明设计的综合考虑数据通信安全的基于微信小程序的可信身份认证平台，包括数据通信安全模块、客户端和服务器端，实现三类不同的认证方式。

[0006] 第一层次为实名认证。需要对二代身份证正面的人员信息拍照，通过OCR技术识别获取姓名、身份证号等卡面信息进行信息收集，并与后台数据库进行比对。

[0007] 第二层次为实名+实证认证。在实名认证通过的基础上，利用具有NFC功能的智能设备对证件的合法性进行确认，并以此获取基础可信任根；以基础可信任根为索引，从公安部获取申请人的基础信息、人像比对源信息，集合生物行为识别及生物人脸特征比对等手段来确认申请人证件合法、基本信息有效及本人申请。

[0008] 第三层次为实名+实证+实人认证。增加了人脸生物特征比对环节，利用活体检测技术，确定采集对象为生物人，以预置的人像/证件照片作为可信数据源进行对比，可确定申请人信息的准确性。

[0009] 进一步，所述数据通信安全中，在用户登录认证环节使用了动态口令卡或短信验证码等动态口令验证，以及登录设备MAC地址验证等技术；数据传输环节对登录认证信息采用了MD5加密技术，对正常使用时的关键数据采用了RSA非对称加密技术。

[0010] 进一步，所述客户端的设计使用微信前端开发语言WXML和WXSS，以及脚本语言WXS，在WXML中将按钮绑定到相关函数上，拼接参数后以POST的方式发送JSON请求，通过URL地址发送给AI接口实现功能并回传参数，最后页面重定向到下一页。

[0011] 进一步，所述服务器端作为MVC设计中的M层，提供了读写数据库到API的封装；平台采用Node.js作为后端开发平台实现对MongoDB数据库的读取，与微信小程序端进行交互并返回JSON数据包的功能，其中使用Express框架搭建服务端；前后端使用HTTPS协议通信，传输JSON格式数据。

[0012] 进一步，第一层次实名认证中采用OCR身份证识别，绑定一个无参数的函数uploadIdImage函数到上传身份证照片按钮上，先调用微信API拉取相册或者摄像头，获取用户身份证照片后，将其用Base64编码，拼接百度身份证OCR参数后发送至百度AI接口；随后将识别的结果解析，与数据库中存储的内容进行比较，通过后进入下一步。

[0013] 进一步，第二层次实名+实证认证中，采用NFC近距离读卡。首先测试手机开启NFC功能的返回值并初始化NFC模块，然后完成wx.onHCEMessage的消息监听并发送消息，将shizhengCert函数绑定到读取NFC按钮上；通过南京亿数身份证读取API对身份证进行识别；在身份证芯片验证成功后，将修改flag值，转至人脸活体检测模块。

[0014] 进一步，第三层次实名+实证+实人认证中，采用人脸识别和人脸活体检测。将uploadRealImage函数绑定到实人认证按钮上；函数调用相机接口，拍照获取图片之后拼接参数，和服务端数据库中预置的身份鉴别图片一起经过Base64编码，发送给百度AI人脸识别API后端进行比对，返回对比相似度比分，若比分大于等于80，则认证成功。另外，为了检测用户为活体，还需要调用百度活体检测API进行活体检测，若用户按要求完成了预制的动作后，则可确认用户为活体。

[0015] 与现有技术相比，本发明的有益效果为：

[0016] 本发明基于微信小程序开发的系统架构，充分利用数据安全(MD5、RSA)、OCR识别、NFC技术、人脸识别、活体检测等新一代身份识别技术，最终使用WXML、Node.js等语言编程实现客户端和服务器端的设计，搭建了手机端的可信任身份认证平台。本平台优点体现如下：

[0017] 1、本发明充分考虑了数据通信安全，分用户登录认证、数据传输安全和内部接口密文调用等阶段分别应用了不同的数据安全机制，多维度保护了平台中的数据。其中用户登录认证保证了用户使用正确的口令和正确的设备登录平台。数据传输安全分类别采用不同的加密技术，保证了用户在与平台交互过程中的关键敏感数据的安全，实现了即使关键数据在传输时被截获，也无法被有效复现的目的。内部接口密文调用通过将平台提供给客户端调用的接口进行转义，利用不相关的“字母+数字”代号来表示，可以进一步避免被恶意用户非法调用或破解；另外，在内部接口中每次都会对调用的客户端进行身份信息验证，再次确保了调用安全。

[0018] 2、本发明采用了“实名+实证+实人”三重认证机制，确保了用户为真实本人且为活体。其中，实名认证通过OCR技术对用户身份证正面信息进行识别，并将识别后的信息与用户预先注册存储在平台的身份信息进行比对，初步确保了用户实名安全。实证认证通过NFC技术对身份证进行识别，获取基础可信任根，并基于它获取公安部身份证数据库真实身份信息，并进行再次比对，这样可以确保身份证的真实性。实人认证通过对用户的人脸进行拍

照,并与实证认证阶段获取的真实身份证人脸图像进行比对,以确保用户是本人;另外,为了进一步确保用户为真实活体本人,还增加了活体检测,并穿插抓拍人脸正面图片进行再比对,通过双重认证机制确保了用户的合法身份。

附图说明

[0019] 图1为登录认证时数据传输安全机制;

[0020] 图2为一次迭代;

[0021] 图3为正常使用时数据传输安全机制;

[0022] 图4为客户端调用接口安全验证示意图;

[0023] 图5为客户端调用实名认证接口示意图;

[0024] 图6为客户端调用实证认证接口示意图;

[0025] 图7为客户端调用实人认证接口示意图;

[0026] 图8为用户认证流程图;

[0027] 图9为平台功能模块图;

[0028] 图10为平台功能结构图;

[0029] 图11前端界面设计关系图;

[0030] 图12为服务器端项目架构;

[0031] 图13为登陆页面;

[0032] 图14为实名认证;

[0033] 图15为实证认证;

[0034] 图16为获取身份凭证。

[0035] 图17实人认证。

具体实施方式

[0036] 下面结合附图和具体实施过程对本发明的实施方案作进一步详细地说明。

[0037] 1、数据通信安全设计

[0038] 进行数据通信安全设计主要是防止非法用户。用户在使用前必须提前注册登记动态口令卡、手机号码、移动设备MAC信息。这样非法用户无法准确获取每次登录口令,且无法使用未注册设备进行系统登录,使得其无法非法使用系统。另外,正常用户在使用系统过程中,所传递的关键数据均进行非对称加密,所调用的服务接口均经过密文处理,使得非法用户即使截获信息,也无法复现数据,进一步阻挡了非法用户。

[0039] (1)用户登录认证

[0040] 用户登录认证作为可信身份认证平台的第一层安全防护,可以初步确保当前登录方式为准合法用户使用合法设备登录。

[0041] 1)动态口令验证

[0042] 用户在使用可信身份认证平台时,无需再使用传统的“用户名+密码”,而是改用动态口令方式。此处动态口令可以采用两种方式:一是动态口令卡;二是短信验证码。

[0043] 方式一:动态口令卡。

[0044] 此种方式适用于用户登录设备不具备短信通信功能的情况。用户在可信身份认证

平台注册时,附带下发电子口令卡,该卡记录了一个50×50的二维矩阵,矩阵每一单元格为4位“数字+字母”组合。样例如下:

[0045] 表1电子动态口令卡

[0046]

	A1	A2	……	A50
B1	aE3h	4BkN	……	5R6V
B2	H9d0	qT7m	……	2GJ9
……	……	……	……	……
B50	F0T3	2UpL	……	W5Q8

[0047] 用户在登录时,系统会随机给用户下发一个标签组合,例如“A2B2”。当用户获取这个随机标签后,可以查找电子口令卡,进而获知当前动态密码为“qT7m”。服务端同样维护了与该用户相同的电子口令卡,用于提取相同的动态口令以验证客户端用户口令的正确性。

[0048] 方式二:短信验证码。

[0049] 此种方式适用于用户登录设备具备短信通信功能的情况。用户在可信身份认证平台注册时,提供手机号码,用于接收服务端发送的随机验证码。用户在每次登录时,选择短信验证登录方式,则系统会向用户注册手机号码发送随机验证码。用户可借此验证码登录系统。

[0050] 2)登录设备MAC地址验证

[0051] 通过动态口令认证确保了当前用户为准合法用户,但无法保证该准合法用户所使用的登录设备的合法性。为此,本发明增加了对登录设备MAC地址的验证。

[0052] 用户在登录时,客户端自动获取当前登录设备的MAC地址信息,会同动态口令一起发送给服务端进行验证。服务端验证收到的MAC地址信息与用户注册时提供的MAC地址信息是否一致,以确定是否是合法登录设备。

[0053] (2)数据传输安全

[0054] 无论是用户登录信息还是正常认证过程信息,都需要通过无线网络经Internet进行传输,当数据不经过任何处理,则是以明文形式进行传输,这样就存在数据传输泄密安全隐患。为了提高可信身份认证平台的安全等级,则需要对传输数据进行加密。所传输的数据主要分为两类:一是登录认证信息;二是正常使用过程中的关键信息。下面分别针对这两类信息进行加密处理。

[0055] 种类一:登录认证信息

[0056] 本发明为了防止登录认证信息在网络传输时被截获破解,平台考虑增加MD5(Message-digest Algorithm 5)对登录认证信息进行加密。MD5作为目前常用的一种Hash加密算法,专门针对理论上任意长的输入产生固定128位长输出。在用户A和B通信的过程中,为了保证认证信息在传送过程中没有被修改,A把自己的信息通过散列函数生成哈希值,附在消息尾部,B收到消息后,重新计算哈希值,如果哈希值比对一致,则表示在传送过程中信息未被篡改,反之则表明信息被篡改。流程如图1所示。

[0057] 算法加密处理过程具体包含了如下几个步骤:

[0058] Ⅰ.消息填充

[0059] 首先填充消息,使其长度为(n×512-64)位,其中n为大于0的整数。注意,即使消息本身长度为448=(1×512-64)位,满足要求,还是需要继续填充512位为960=(2×512-64)

位。填充的内容由一个1和后续的0组成。然后，在填充内容的后面再附上64位，这64位存放的内容是填充之前消息的长度，一旦消息长度大于264，则取其对264取模的结果进行填充。

[0060] 填充过后，消息的长度为512×L，则可将消息分为每512位长一组的Y0，Y1，……，YL-1，再把每一个分组Y分为16个32比特长的字单位，这样消息为N＝L×16个字，因此消息还可以按字表示为M[0，……，N-1]。

[0061] Ⅱ.缓冲区初始化

[0062] Hash函数的中间结果和最终结果保存在128位的缓冲区当中，缓冲区用32位的寄存器表示。可用4个32比特长的字表示：A、B、C、D。初始值以十六进制表示为A＝01234567，B＝89ABCDEF，C＝FEDCBA98，D＝76543210。

[0063] Ⅲ.HMD5运算

[0064] 以分组为单位对信息进行处理，每一分组Yq(q＝0，……，L-1)都经过压缩函数HMD5处理。HMD5是算法的核心，一次算法包括有4轮处理过程。HMD5的4轮处理过程结构一致，但所用的逻辑函数不同，可分别表示为F、G、H、I。只不过每一轮用到的逻辑函数各不相同，同时每一步的输入函数也有所不同。

[0065] 表2 4轮的逻辑函数

[0066]

轮	基本函数g	逻辑函数g(b,c,d)
1	F(b,c,d)	(b∧c)∨(b-∧d)
2	G(b,c,d)	(b∧d)∨(c∧d-)
3	H(b,c,d)	b⊕c⊕d
4	I(b,c,d)	c⊕d∨d-

[0067] 每轮的输入为当前处理的512位分组Yq和缓冲区当前的A、B、C、D值，输出继续存储缓冲区中生成新的A、B、C、D。每轮要完成16步重复迭代运算，四轮共计64步完成。64步途中每一步都按照图2示流程进行迭代，其中X[k]是当前分组Yq的第k个32位字，T[i]作为消除输入数据规律性的数据。

[0068] 每一轮循环左移的情况不同，第一轮分别左移7、12、17、22位，第二轮分别左移5、9、14、22位，第三轮分别左移4、11、16、23位，第四轮分别左移6、10、15、21位。

[0069] 最终第四轮的输出与第一轮的输入进行模232相加结果为输出，也就是下一个分组进行运算的缓冲区初始值。

[0070] Ⅳ.输出

[0071] 全部L个512位分组按照上述描述处理完，最后一个分组的128位输出即为所有迭代得出的哈希值。

[0072] 种类二：正常使用时关键数据传输信息

[0073] 在用户登录成功后进行关键数据传输时，也需要进行加密处理。此处主要是解决用户登录后正常进行数据操作的传递安全问题。此时数据分为两大类：一是字符信息数据，如姓名、身份证号、身份证基础可信任根等；二是图片数据，如人脸图片等。为了缩短处理时间，提高加密效率，此处仅对于字符敏感信息数据使用非对称密码加密技术(RSA)。令客户端持有公钥，而后台数据处理端持有私钥，在保证私钥安全的基础上，完成加密传输数据。利用RSA加密实现平台的数据传递安全原理如图3所示。

[0074] 使用RSA算法，把选取的两大素数之积n和公钥e存储于手机端数据库SQLite中，另

外还要把整数n和私钥d存储于平台数据库中。在数据传输的过程中，客户端向服务端发送的内容包括用户真正需要传输的正常关键信息数据(如身份证号等)，以及已被MD5算法加密处理过的“动态口令+设备MAC”客户身份信息(身份认证在用户登录中已验证完备，此处进行验证的是用户每一步操作的合法性)。全部内容利用RSA算法加密后，再发送至服务端。

[0075] 每一次数据传输的目的都是调用在服务端的某个或某些具体的接口。以用户进行实证认证为例，在服务端接收到经RSA公钥加密过的客户身份信息和身份证基础可信任根等数据后，服务端的处理过程如下：

[0076] i解密。首先对加密内容进行RSA私钥解密得到身份信息和正常数据。

[0077] ⅱ验真。其次需要再次根据服务端存储的动态口令(MD5加密值)进行内容比较验证身份信息。

[0078] ⅲ服务。最后验证身份信息无误后，处理正常的数据请求，实现后续服务。

[0079] 每一次数据传输过程都需要调用某个或某些接口服务，每个服务都必须经过解密验真无误的步骤才处理真正的服务。因为对所有服务而言解密和验真都属于必须经过的步骤，为了简化代码的书写，在服务端中实现将解密函数和比较动态口令(MD5加密值)函数作为公共函数供所有的服务函数调用。

[0080] (3)内部接口密文调用

[0081] 用户客户端若要使用可信身份认证平台提供的服务，必须通过网络调用平台提供的接口。一般服务端为了方便用户调用，会将功能接口以显示的具有一定含义的命名方式暴露给用户，这种方式对于确需公开给大众用户自主调用时可以避免混淆，达到“见名知意”的目的，如用户开发地图应用时所调用的百度地图接口等。但是对于本可信身份认证平台来说，将服务端接口以这种方式提供给客户端调用，必然会造成一定的安全隐患。为此，本发明通过将具有含义的明文接口及其参数转换为“字母+数字”组合代号方式，可以一定程度上防止接口被非法人员截获后通过接口名字和参数获知接口功能，进而进行恶意调用。经过转换后接口即使被非法获取也无法“见名知意”，增加了安全性。如下表3所示。

[0082] 表3接口转义示例

[0083]

原始接口示例	转义后接口示例
Login(Sting name, String password)	L(String L1, String L2)

[0084] 另外，为了进一步增加接口的安全性，平台在每个接口中还会验证接口调用设备的MAC地址，这样可以防止非法用户通过非法设备调用接口。流程如图4所示。

[0085] 2、实名、实证和实人认证设计

[0086] 通过以上验证后，可以初步确保“准合法用户”使用“合法设备”登录可信身份认证平台，之所以是“准合法用户”是因为可能会存在非法用户盗取合法用户设备进行登录的情况。为此，还需要对“准合法用户”的身份进行深层次的认证。本发明采用“实名+实证+实人”认证设计，主要是认证合法用户。首先需要认证用户身份证信息，通过OCR识别用户身份证表面文字信息，确保用户信息与注册的身份证信息相一致。其次，为了防止身份证为非法仿制证件，还必须通过手机NFC功能对用户提供的身份证进行读卡验证操作，确保身份证卡片的真实性。最后，为了确保用户为真实本人且为活体，必须通过活体检测和人脸识别功能对用户进行验证。

[0087] (1)实名认证

[0088] 实名认证主要是验证用户身份证正面信息与其在可信身份认证平台预留的身份信息是否一致。过程为:“准合法用户”通过移动客户终端(如具有NFC和拍照功能的手机等)调用平台实名认证接口,对身份证正面进行拍照,并上传至平台服务端;然后平台服务端采用OCR技术对其中的关键信息(如姓名、出生年月、性别、身份证号等)进行识别;最后比对客户端提供的信息与客户在平台预留的信息一致性,并返回结果。流程如图5所示。

[0089] (2)实证认证

[0090] 通过实名认证后,可以初步确认用户的身份关键信息的合法性,但是也存在一定漏洞。比如非法用户提供的身份证是仿制证件或提供的身份证照片为其它途径获取等,此时仅通过实名认证已经无法确认“准合法用户”的身份,为此必须再增加实证认证。

[0091] 实证认证主要是验证“准合法用户”提供的身份证是否是真实证件,可以排除假冒仿制证件或仅提供身份证照片所带来的安全隐患。过程为:“准合法用户”通过具有NFC功能的移动客户端调用平台实证认证接口,此时要确保客户端的NFC功能是开启状态;将身份证实体贴近客户端,客户端会与身份证之间简历NFC连接,并获取身份证中芯片所存储的基础可信任根等信息;客户端将识别出的基础可信任根等信息上传至服务端;服务端以基础可信任根为索引,从公安部网站获取申请人的基础信息、人像比对源等信息,并跟客户端在实名阶段所提供的身份证关键信息进行再次比对,从而确保了“准合法用户”提供的身份证的真实性。流程如图6所示。

[0092] (3)实人认证

[0093] 通过实名认证和实证认证后,可以初步确保“准合法用户”提供了真实的身份证件来进行身份认证,但仍然存在一定漏洞。比如非法用户盗用了合法用户的手机和身份证时,若仅基于以上认证方式,则无法准确识别出其中的非法性,为此必须再增加实人认证。

[0094] 实人认证主要是验证“准合法用户”确系本人,且为活体,可以排除非法用户仅通过证件冒名认证的安全隐患。过程为:“准合法用户”通过移动客户端的摄像头对自己头像进行拍照,将照片上传至服务端;服务端通过人脸识别技术将客户端上传的人脸图片与通过实证认证阶段获取的身份证上的头像进行比对,确认当前操作验证过程的是“本人”。但此时还无法完全确保是真实的“本人”,因为客户端提供的用于进行人脸识别的人脸图片可能是预先拍摄的,为此还需要进行活体检测。系统预设眨眼、摇头等多种动作,并自定义生效动作及校验顺序。每次进行活体检测时,“准合法用户”按照系统提示,做出相应的动作,系统会随机抓取多图进行活体判断。这样即可保证当前用户为活体,又防止非法用户提前录制。为了进一步确保当前活体是合法用户本人,在进行活体检测过程中会对用户正面图片进行抓拍,并与实证认证阶段获取的身份证上的头像进行再次比对。这样通过双重验证机制,最终确定“准合法用户”为“合法用户”本人,且为活体。流程如图7所示。

[0095] 通过“实名+实证+实人”认证后,基本可以确定“准合法用户”为“合法用户”,进而可以进行后续操作。

[0096] 3可信任身份认证平台总体方案

[0097] 本发明基于微信小程序的系统架构,综合考虑数据通信安全,设计并实现了一个可信任身份认证平台。该平台根据认证业务需求,实现了“实名+实证+实人”三重认证机制。

[0098] 3.1平台认证流程

[0099] 平台认证流程如图8所示。

[0100] (1)平台分别在用户登录认证、数据传输安全和内部接口密文调用等阶段应用了不同的数据安全机制,多维度确保数据安全。

[0101] (2)通过“实名+实证+实人”三重认证机制,最终确保用户的合法性。

[0102] 第一层次为“实名”认证。需要对二代身份证正面的人员信息拍照,通过OCR技术识别获取姓名、身份证号等卡面信息进行信息收集,并与后台数据库进行比对。

[0103] 第二层次为“实名+实证”认证。在实名认证通过的基础上,利用具有NFC功能的智能设备对证件的合法性进行确认,并以此获取基础可信任根。以基础可信任根为索引,从公安部身份证数据库获取申请人的基础信息、人像比对源信息,通过比对再次确认申请人证件合法、基本信息有效及本人申请。

[0104] 第三层次为“实名+实证+实人”认证。因系统无法确认现场采集对象是生物人像还是照片人人像,因此“实名+实证”仍然存在假冒的漏洞。而“实名+实证+实人”认证较“实名+实证”认证有了很大的进步,增加了人脸生物特征比对环节,利用活体检测技术,确定采集对象为生物人,以预置的人像/证件照片作为可信数据源进行对比,可确定申请人的合法性。

[0105] 基于平台的认证流程,设计平台功能模块如图9所示,功能结构如图10所示。

[0106] 3.2平台数据库设计

[0107] (1)服务端MongoDB数据库设计

[0108] 由于身份认证涉及的数据大多是个人身份信息,具有数据体量大、结构简单、频率快等特点,对读写数据的效率要求非常高,因此使用MongoDB作为服务端后台数据库。另外出于安全性的考虑,数据库中的个人身份信息并没有采用用户输入方式验证,而是在部署应用时由DBA从可信数据源进行导入。数据字典如下表4所示:

[0109] 表4 MongoDB数据库数据字典

[0110]

Key名	示例数据	数据类型
_id	5cd40d2782f6e32a541bdd4c	ObjectId
Phone	13811111111	String
Password	test1	String
Name	1qazxsw2	String
Idnum	张XX	String
Sex	男	String
Nat	汉	String
Addr	XX省XX市XX街道XXX	String
Birth	19970120	String
Idpic	/9j/4AAQSkZJRgABAQEAYA…..	String

[0111] (2)客户端数据库设计

[0112] 在认证过程中,客户端需要加密存储动态口令卡、RSA非对称加密公钥等信息。本发明采用SQLite数据库用于客户端数据的存储。

[0113] 表5动态口令卡

[0114]

Key名	示例数据	数据类型

_id	9yk40d2782f6e32a541bkh7t	ObjectId
RowId	A1	String
ColId	B1	String
Value	aE3h	String

[0115] 表6 RSA公钥

[0116]

Key名	示例数据	数据类型
_id	4hn00d2782f6e32a541b97ug	ObjectId
name	RSAPublicKey	String
Value	5m9m14XH3oqLJ8bNGw9e4rGpXpck...	String

[0117] 注意:以上数据库内容需要设置访问权限,只有注册客户端才保存该数据库,且数据库中的内容,尤其是动态口令要加密存储。

[0118] 3.3客户端设计

[0119] 平台基于微信小程序,对各个认证功能模块的调用实现可信的身份认证,力求便捷、高效、简约,在这里使用微信前端开发语言WXML和WXSS,以及脚本语言WXS。其中WXML是微信基于HTML设计的一套标签语言,实现了基本的组件、事件系统和数据双向绑定,并采用Mustache模板引擎;WXSS是兼容CSS的微信小程序样式表规范,在CSS的基础上扩展了许多有用的功能;WXS则是基于JS语法的微信脚本语言,内部封装了微信的基本API并可以兼容大多数JS脚本。根据功能需求,该框架所要设计实现的特点如下:

[0120] 1.系统界面布局大方简约,融合安全元素;

[0121] 2.实现“实名+实证+实人”三层认证过程,操作流程简单方便;

[0122] 3.系统可扩展性较强,便于日后根据需求变更修改系统;

[0123] 4.系统运行稳定,鲁棒性强;

[0124] 5.基于微信小程序实现,不依赖其他终端设备,便于部署使用;

[0125] 6.可以集成在其他系统中作为子系统提供身份认证的功能。

[0126] 界面设计分为三级界面,具体关系如图11所示。

[0127] 3.4服务器端设计

[0128] 服务器端作为MVC设计中的M层,提供了读写数据库到API的封装。平台采用Node.js作为后端开发平台,实现对MongoDB数据库的读取、与微信小程序端进行交互并返回JSON数据包的功能,其中使用Express框架搭建服务端。前后端使用HTTPS协议通信,传输JSON格式数据。该后端设计的优点有:

[0129] 1.Node.js代码原生异步处理,在可能具有阻塞的场景中(如DB读写,网络、磁盘请求)具有响应速度快的优势;

[0130] 2.Express.js框架以路由栈为线索构建代码,封装了其他服务器端,便于快速实现后端API搭建;

[0131] 3.MongoDB数据库原生提供异步连接支持,每次读写都是异步请求,避免在后端使用线程池,可以节省计算资源提高处理效率。

[0132] 服务器端架构如图12所示。app.js为框架入口,其中注册了Express.js框架所需的一系列中间件,进行了基础路由配置等。bin/文件夹存放框架启动配置,www文件中配置了服务器启动的端口、IP等信息。cert/文件夹存放SSL证书,用以启动HTTPS服务器。

package.json文件为项目依赖树。public/文件夹为静态资源目录。routes/文件夹为路由目录。

[0133] 4身份识别方案

[0134] 4.1OCR身份证识别

[0135] OCR身份证识别技术基于深度学习算法,融合多种图像处理技术,可以精准识别公民身份证上的图像文字信息并返回和后台数据库进行比对,经过对比选用百度OCR证件识别API实现,具体设计如下:

[0136] 绑定一个无参数的函数uploadIdImage函数到“上传身份证照片”按钮上,先调用微信API拉取相册或者摄像头,获取用户身份证照片后,将其用Base64编码,拼接百度身份证OCR参数后发送至百度AI接口。随后将识别的结果解析,与数据库中存储的内容进行比较,通过后进入后续认证环节。

[0137] 4.2NFC近距离读卡

[0138] 因为我国公民的身份证由公安使用国密算法加密,没有密钥无法对身份证卡面信息进行解密,可以采用南京亿数信息科技有限公司的身份证网络读取SDK用于对身份证芯片信息进行读取。另外,本发明基于微信小程序的框架开发,而小程序正好能够提供基于NCE模式的NFC开发接口,通过获取证件的NFC信号确认证件的真实性。以下是结合小程序NFC接口开发文档进行的开发步骤:

[0139] 首先测试手机开启NFC功能的返回值并初始化NFC模块,然后完成wx.onHCEMessage的消息监听并发送消息,将shizhengCert函数绑定到读取NFC按钮上。在身份证芯片验证成功后,将修改flag值,转至后续认证环节。

[0140] 4.3人脸活体检测

[0141] 目前人脸活体检测可以接入的平台有Face++、阿里云、腾讯云、百度云等,综合考虑接口稳定性、价格技术等因素,最重要的是百度人脸识别API可以实现手机端调用相机实时拍摄人脸的功能,可以判断重复拍摄照片、视频中的人脸所产生的摩尔纹。另外百度人脸识别AI实现,具有稳定可靠、识别精度高,并且采用接口+SDK双重融合检测,可抵御照片、视频等攻击。接入的步骤和OCR身份证识别技术类似。具体设计如下:

[0142] 将uploadRealImage函数绑定到实人认证按钮上。函数调用相机接口进行人脸拍摄,并将图片传至服务端。服务端之后拼接参数,将客户端上传图片和数据库中预置的身份鉴别图片一起经过Base64编码,发送给百度AI人脸识别API后端进行比对,返回对比相似度比分,若比分大于等于80,则认证成功。另外,为了检测用户为活体,还需要调用百度活体检测API进行活体检测,若用户按要求完成了预制的动作后,则可确认用户为活体。

[0143] 5功能测试

[0144] 使用微信开发者工具进行平台认证过程的测试,内容包括用户登录、实名认证、实证认证、实人认证、获取身份凭证。测试过程如下:

[0145] (1)开启后台腾讯云服务器;

[0146] (2)打开微信开发者工具,导入项目代码,输入AppID;

[0147] (3)Node手动部署,建立与服务器的连接;

[0148] (4)设置相关参数进行测试。

[0149] 登录界面如图13所示。实名认证界面如图14所示。实证认证界面如图15、图16所

示。实人认证界面如图17所示。

[0150] 以上所述，仅为本发明较佳的具体实施方式，本发明的保护范围不限于此，任何熟悉本技术领域的技术人员在本发明披露的技术范围内，可显而易见地得到的技术方案的简单变化或等效替换均落入本发明的保护范围内。

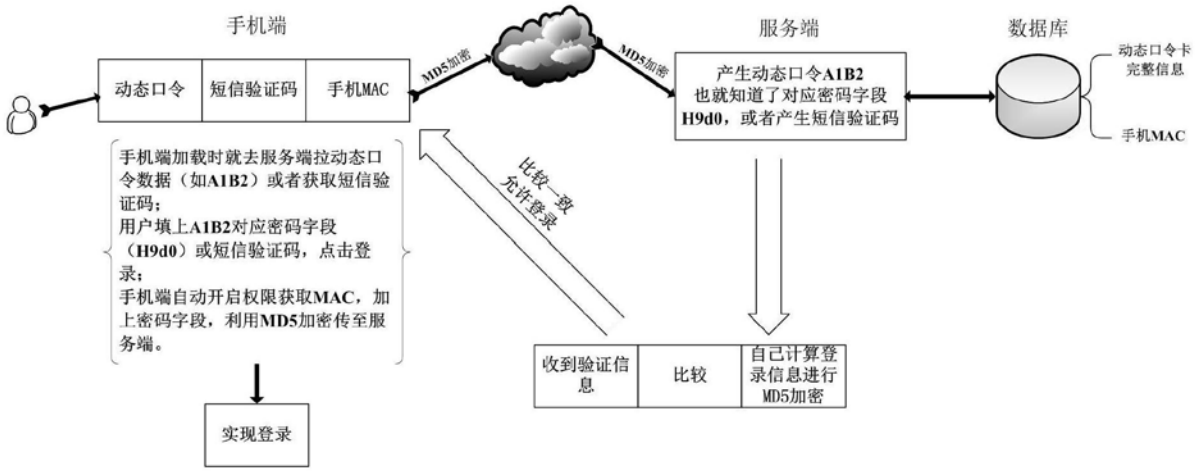

手机端
服务端
数据库
动态口令
短信验证码
手机MAC
MD5加密
MD5加密
产生动态口令A1B2
也就知道了对应密码字段
H9d0，或者产生短信验证码
动态口令卡
完整信息
手机MAC
手机端加载时就去服务端拉动态口令数据（如A1B2）或者获取短信验证码；
用户填上A1B2对应密码字段（H9d0）或短信验证码，点击登录；
手机端自动开启权限获取MAC，加上密码字段，利用MD5加密传至服务端。
比较一致
允许登录
收到验证信息
比较
自己计算登录信息进行MD5加密
实现登录

图1

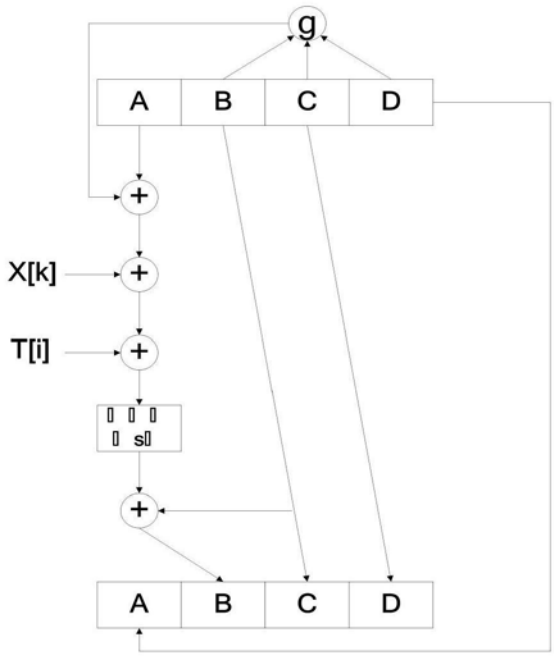

g
A
B
C
D
X[k]
T[i]
s
A
B
C
D

图2

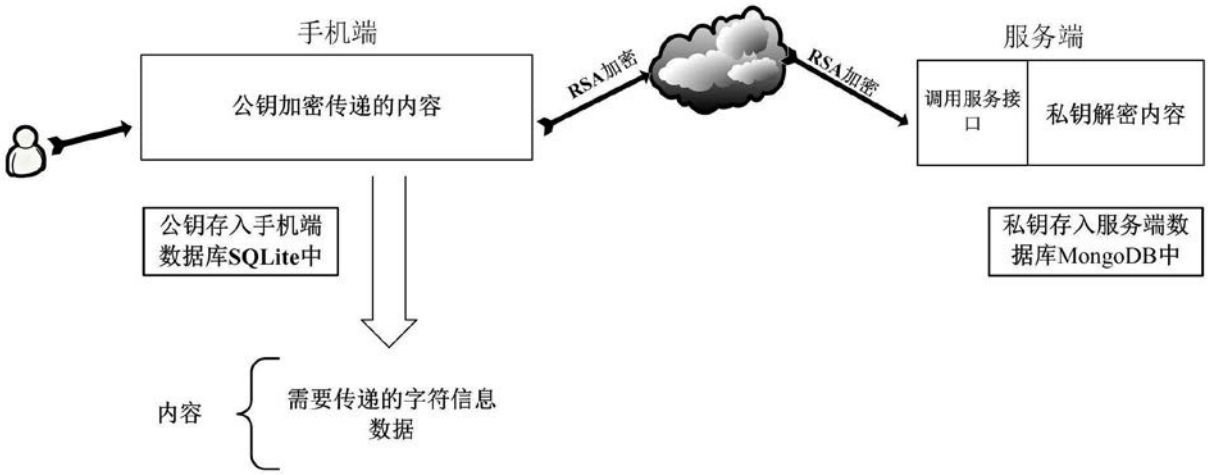

手机端
公钥加密传递的内容
RSA加密
RSA加密
服务端
调用服务接口
私钥解密内容
公钥存入手机端数据库SQLite中
私钥存入服务端数据库MongoDB中
内容
需要传递的字符信息数据

图3

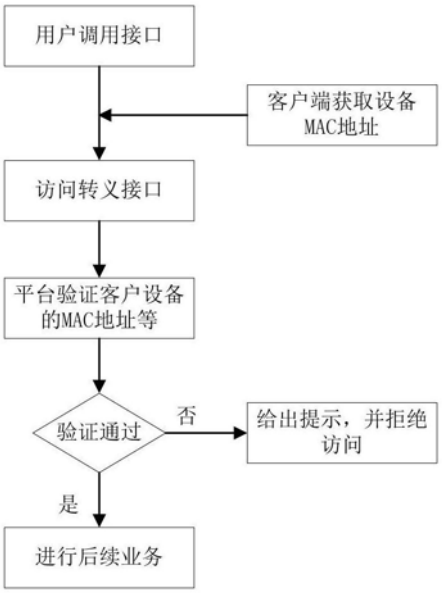

用户调用接口
客户端获取设备MAC地址
访问转义接口
平台验证客户设备的MAC地址等
验证通过
否
给出提示，并拒绝访问
是
进行后续业务

图4

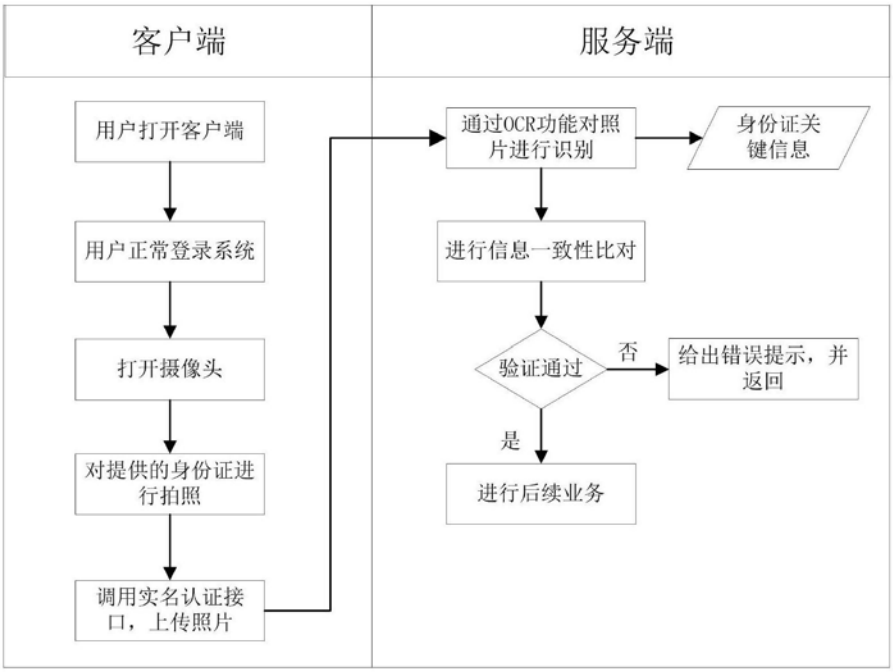

客户端
服务端
用户打开客户端
用户正常登录系统
打开摄像头
对提供的身份证进行拍照
调用实名认证接口，上传照片
通过OCR功能对照片进行识别
身份证关键信息
进行信息一致性比对
验证通过
否
给出错误提示，并返回
是
进行后续业务

图5

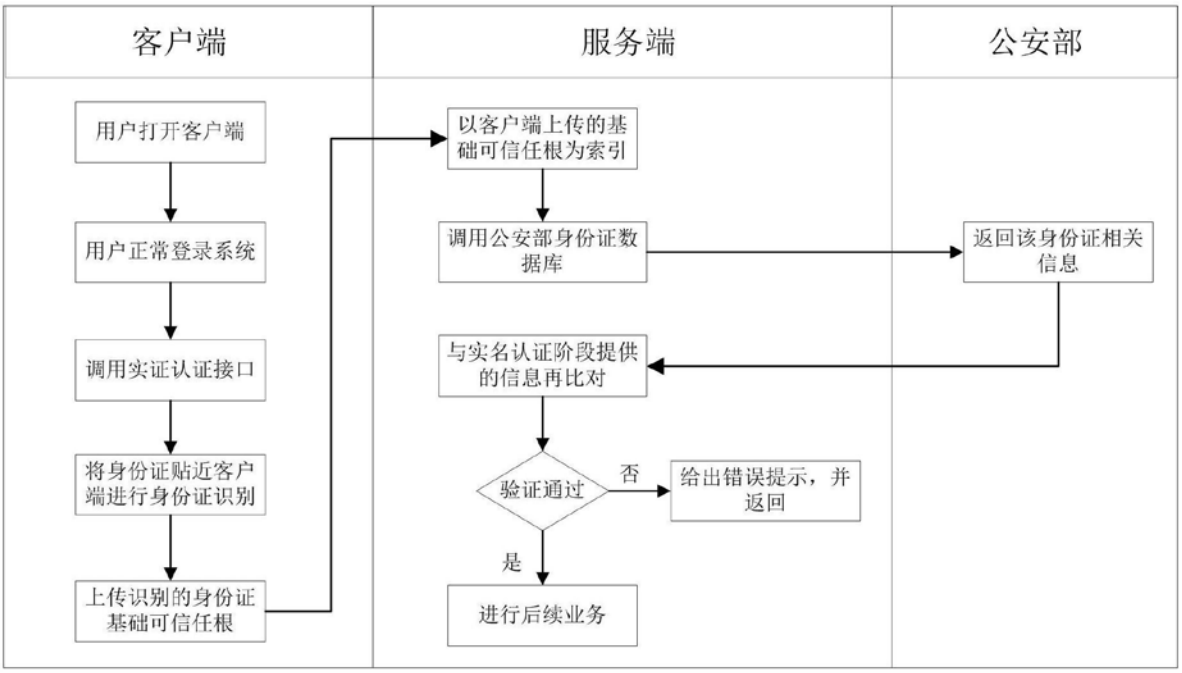

客户端
服务端
公安部
用户打开客户端
用户正常登录系统
调用实证认证接口
将身份证贴近客户端进行身份证识别
上传识别的身份证基础可信任根
以客户端上传的基础可信任根为索引
调用公安部身份证数据库
返回该身份证相关信息
与实名认证阶段提供的信息再比对
验证通过
否
给出错误提示，并返回
是
进行后续业务

图6

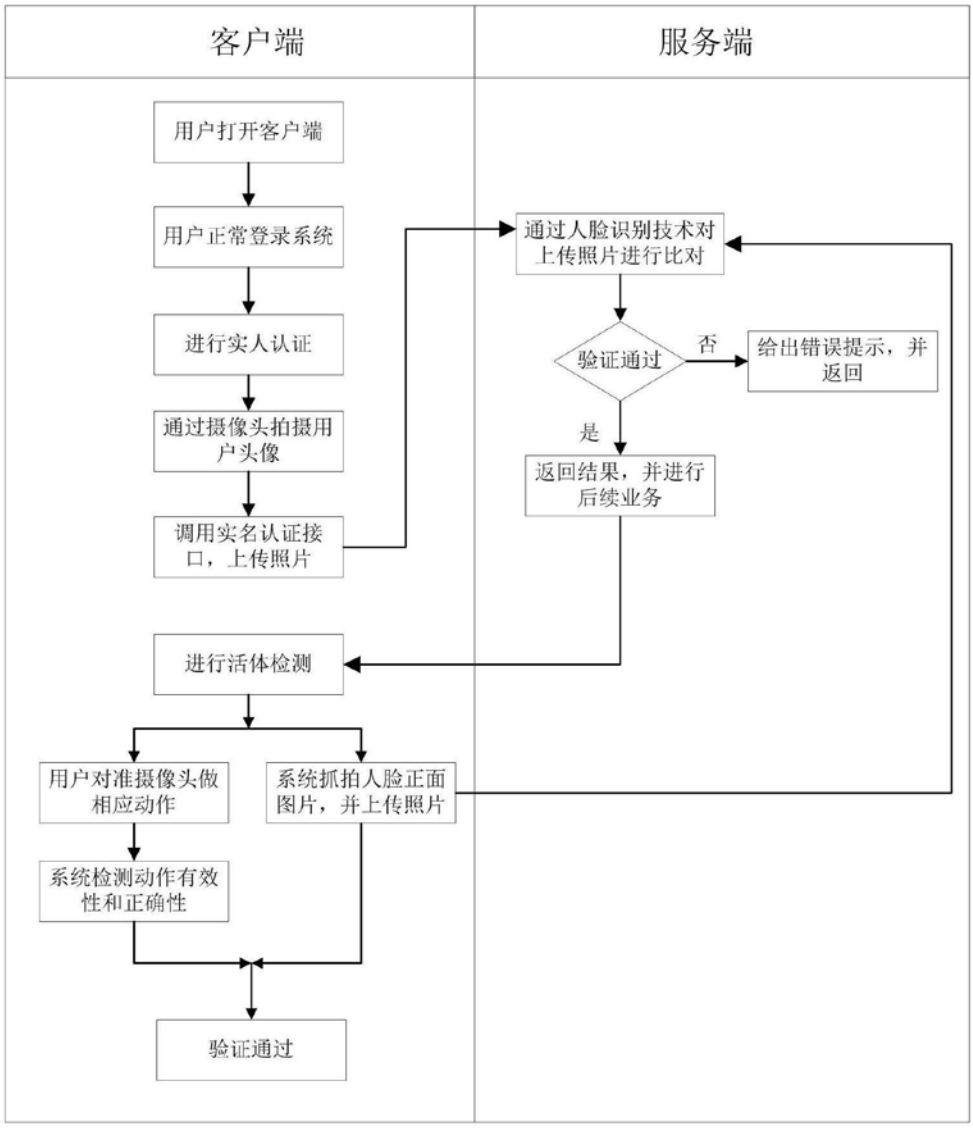

客户端
服务端
用户打开客户端
用户正常登录系统
进行实人认证
通过摄像头拍摄用户头像
调用实名认证接口，上传照片
通过人脸识别技术对上传照片进行比对
验证通过
否
给出错误提示，并返回
是
返回结果，并进行后续业务
进行活体检测
用户对准摄像头做相应动作
系统检测动作有效性和正确性
系统抓拍人脸正面图片，并上传照片
验证通过

图7

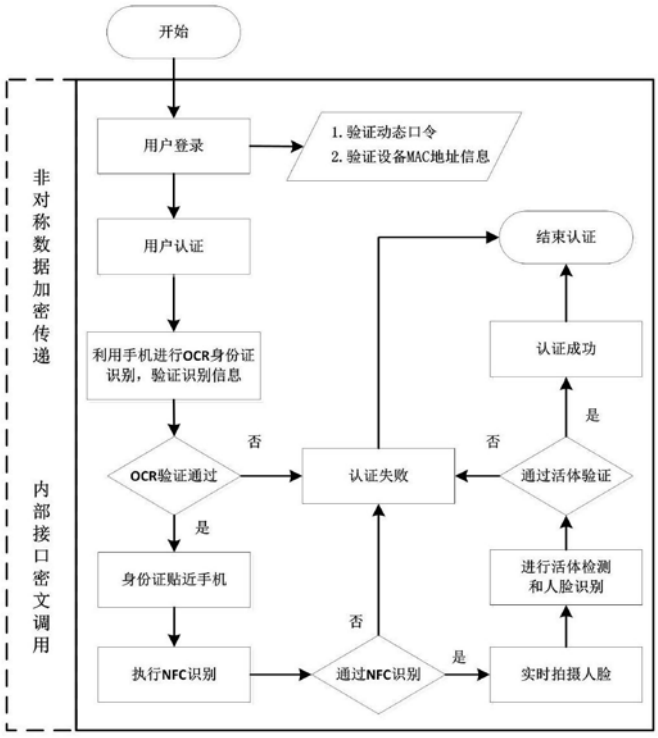

开始
非对称数据加密传递
内部接口密文调用
用户登录
1. 验证动态口令
2. 验证设备MAC地址信息
用户认证
结束认证
利用手机进行OCR身份证识别，验证识别信息
认证成功
否
是
否
否
OCR验证通过
认证失败
通过活体验证
是
身份证贴近手机
进行活体检测和人脸识别
否
执行NFC识别
通过NFC识别
是
实时拍摄人脸

图8

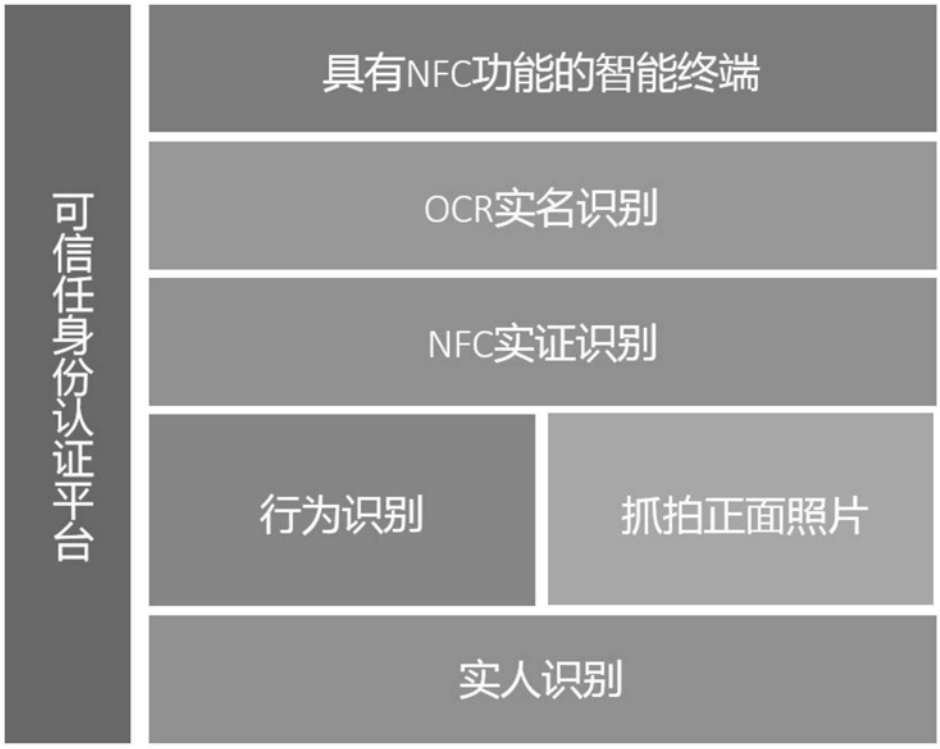

可信任身份认证平台
具有NFC功能的智能终端
OCR实名识别
NFC实证识别
行为识别
抓拍正面照片
实人识别

图9

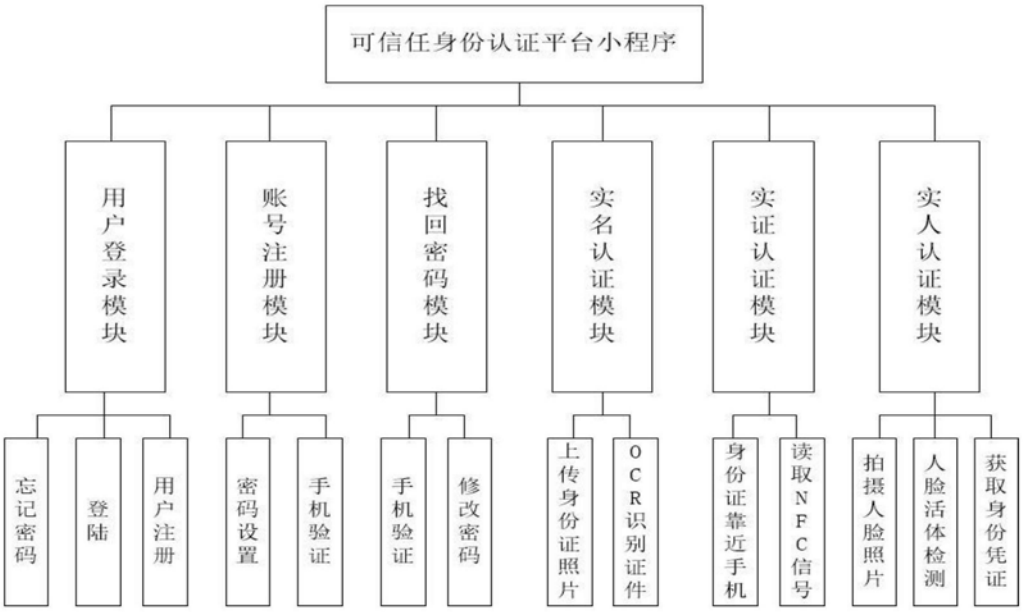

可信任身份认证平台小程序
用户登录模块
账号注册模块
找回密码模块
实名认证模块
实证认证模块
实人认证模块
忘记密码
登陆
用户注册
密码设置
手机验证
手机验证
修改密码
上传身份证照片
OCR识别证件
身份证靠近手机
读取NFC信号
拍摄人脸照片
人脸活体检测
获取身份凭证

图10

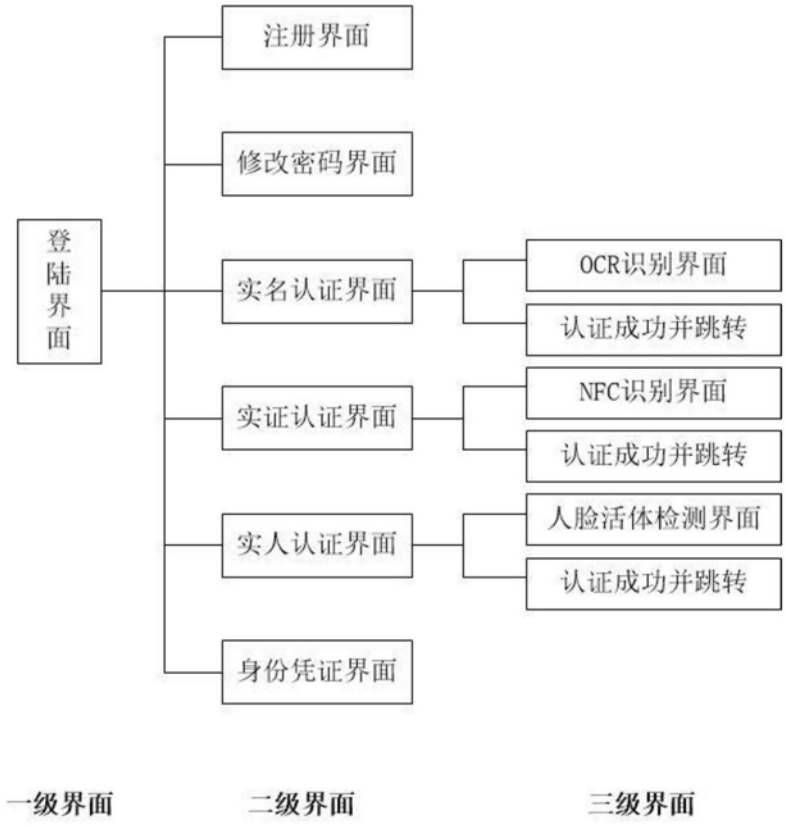

登陆界面
注册界面
修改密码界面
实名认证界面
实证认证界面
实人认证界面
身份凭证界面
OCR识别界面
认证成功并跳转
NFC识别界面
认证成功并跳转
人脸活体检测界面
认证成功并跳转
一级界面
二级界面
三级界面

图11


.
|-- app.js
|-- bin
| `-- www
|-- cert
| |-- cert.crt
| |-- csr.pem
| `-- private.pem
|-- package.json
|-- public
| |-- images
| |-- javascripts
| `-- stylesheets
| `-- style.css
|-- routes
| |-- api.js
| |-- index.js
| `-- users.js

图12


10:07
WeChat
性能数据
基础
CPU: 0%
内存: 146m
启动和切换
页面切换耗时: 456 ms
启动耗时: 1295 ms
渲染
其它
数据缓存: 89 B
用户登录
请输入用户名
请输入密码
登录
忘记密码
账号注册
地址:山东省烟台市芝罘区二马路188号 邮编:264001
vConsole

图13


可信任身份认证平台
初级实名认证
上传身份证照片
WeChat
20:54
验证成功! 进入下一步

图14


可信任身份认证平台
中级实证认证
请将身份证靠近设备读取
NFC
WeChat
20:54
验证成功! 进入下一步

图15


certification_platform - 微信开发者工具 RC v1.02.1905081
文件 编辑 工具 界面 设置 微信开发者工具
模拟器
编辑器
调试器
云开发
100%
WiFi
模拟操作
WeChat
可信任身份认证平台
获取身份凭证
身份凭证:
bab90ae18b3b6cb1901be52e50ccc83
5ddf04df25ab659b9d28b91aec269f17f
时间戳: 1558616154120

图16


certification_platform - 微信开发者工具 RC v1.02.1905081
文件 编辑 工具 界面 设置 微信开发者工具
小程序模式
模拟器
编辑器
调试器
云开发
Phone 7
100%
WiFi
模拟操作
WeChat
可信任身份认证平台
高级实人认证
进行真人检测
通过检测，得分为: 90.44335938
恭喜通过认证，点击获取验证凭证

图17