ICS 35.030
L 80

GM

中华人民共和国密码行业标准

GM/T 0113—2021

在线快捷身份鉴别协议

Fast online identity authentication protocol

2021-10-18 发布　　2022-05-01 实施

国家密码管理局　　发 布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：中国科学院数据与通信保护研究教育中心、联想（北京）有限公司、中国科学院大学、国民认证科技（北京）有限公司、北京数字认证股份有限公司、浙江蚂蚁小微金融服务集团有限公司、北京天融信网络安全技术有限公司、中国科学院信息工程研究所、飞天诚信科技股份有限公司、成都卫士通信息产业股份有限公司、中国金融认证中心、吉大正元信息技术股份有限公司、中孚信息股份有限公司、杭州天谷信息科技有限公司、神州融安科技（北京）有限公司、北京握奇智能科技有限公司、郑州信大捷安信息技术股份有限公司、北京眼神智能科技有限公司。

本文件主要起草人：钱文飞、赵欣怡、贾世杰、刘丽敏、柴海新、荆继武、李俊、张永强、宋铮、景鸿理、王平建、牛莹姣、高彪、吕娜、陈天宇、张咪、朱鹏飞、罗俊、孙国栋、赵丽丽、苗功勋、程亮、李登峰、张渊、刘熙胖、杨春林。

在线快捷身份鉴别协议

1 范围

本文件规定了在线快捷身份鉴别协议，包括通用在线快捷身份鉴别协议、双因素在线快捷身份鉴别协议等内容。

本文件适用于在线快捷身份鉴别服务的开发、测试和评估。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 16262（所有部分） 抽象语法记法（ASN.1）

GB/T 16649.4 识别卡 集成电路卡 第4部分：用于交换的结构、安全和命令

GB/T 25069 信息安全技术 术语

GB/T 32905 信息安全技术 SM3密码杂凑算法

GB/T 32918（所有部分） 信息安全技术 SM2椭圆曲线公钥密码算法

GB/T 35276 信息安全技术 SM2密码算法使用规范

GB/T 36651 信息安全技术 基于可信环境的生物特征识别身份鉴别协议框架

GB/T 37092 信息安全技术 密码模块安全要求

GB/T 38636 信息安全技术 传输层密码协议（TLCP）

GM/Z 4001 密码术语

3 术语和定义

GB/T 25069、GB/T 36651、GM/Z 4001界定的以及下列术语和定义适用于本文件。

3.1

鉴别客户端 authentication client

在用户代理中处理双因素身份鉴别协议消息的软件组件，用于双因素身份鉴别协议中用户代理与鉴别器之间的通信。

3.2

鉴别客户端接口 authentication client interface

由鉴别客户端提供的协议接口，用于双因素身份鉴别协议中用户代理与鉴别客户端之间的通信。

3.3

生物特征识别密钥管理器标识符 biometric authentication protocol key manager identifier

分配给同一型号生物特征识别密钥管理器的唯一标识符，依赖方可通过该标识符唯一确定厂商公钥。

3.4

用户代理 user agent

安装在用户设备上的浏览器或者其他应用程序。

4 缩略语

下列缩略语适用于本文件。

APDU:应用协议数据单元(Application Protocol Data Unit)

AppID:应用程序标识符(Application Identifier)

ASN.1:抽象语法标记(Abstract Syntax Notation One)

BAP:生物特征识别身份鉴别协议(Biometric Authentication Protocol)

BAPV:生物特征识别身份鉴别协议版本(Biometric Authentication Protocol Version)

bkmID:生物特征识别密钥管理器标识符(BAP Key Manager Identifier)

KeyID:密钥标识符(Key Identifier)

KRD:密钥注册数据(Key Registration Data)

PII:个人识别信息(Personal Identifiable Information)

PIN:个人识别码(Personal Identification Number)

TAP:双因素在线快捷身份鉴别协议(Two-factor Authentication Protocol)

TEE:可信执行环境(Trusted Execution Environment)

TLCP:传输层密码协议(Transport Layer Cryptography Protocol)

TPM:可信平台模块(Trusted Platform Module)

VPN:虚拟专用网络(Virtual Private Network)

5 通用在线快捷身份鉴别协议

5.1 协议架构

5.1.1 概述

通用在线快捷身份鉴别协议借助智能设备,结合生物特征识别技术,提供了无口令、安全性能高、使用方便的身份鉴别方案,该协议在GB/T 36651规定的基于可信环境的生物特征识别身份鉴别协议框架基础上,针对协议架构、协议流程和要求等进行了细化,并在协议流程和要求中规定了协议的相关消息数据格式。其中,安全风险及措施建议见附录A,关于可信环境实现方式见附录B,关于协议接口见附录C。在实际部署过程中使用的商用密码产品,应当由具备资格的商用密码检测、认证机构检测认证合格后,方可部署到实际系统中。

在执行通用在线快捷身份鉴别协议之前,身份鉴别服务器通过用户代理调用生物特征识别密钥管理器接口的发现方法,检查用户设备是否支持通用在线快捷身份鉴别协议。本文件不规定身份鉴别服务器获取生物特征识别密钥管理器相关信息以及厂商公钥信息的方式,身份鉴别服务器可利用生物特征识别密钥管理器接口的发现方法中返回的结果,通过生物特征识别密钥管理器相关提供方的服务来获取这些信息。

通用在线快捷身份鉴别协议的交互图如图1所示。其中,图中实线表示消息流由图例中的左侧实体到右侧实体,虚线部分表示消息流由图例中的右侧实体到左侧实体或者是重定向消息流,双向箭头表示两个实体进行交互以完成某操作,跨越实体的虚线表示其他实体通过该实体进行通信,后续交互图中均以此说明为准。图中采用HTTP重定向方式对协议交互情况进行说明。

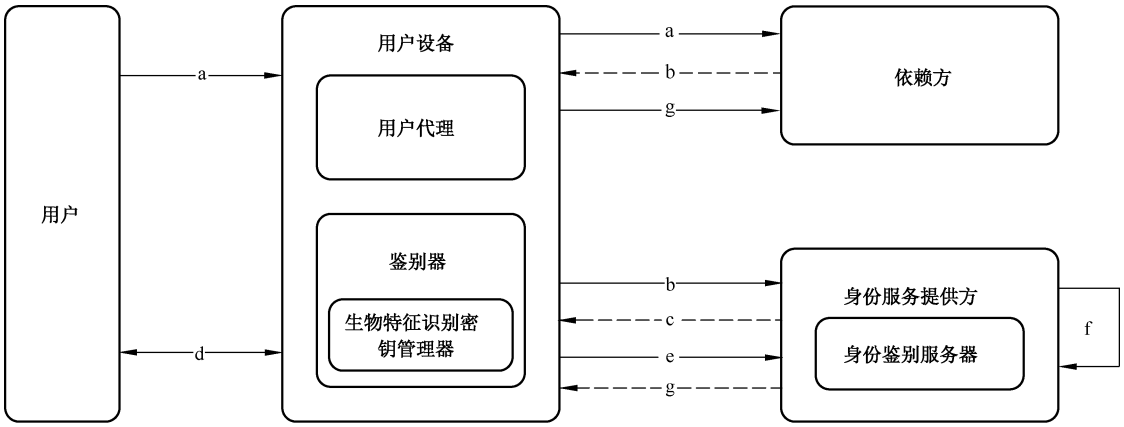

图 1　协议交互图

用户身份鉴别流程主要分为 8 个步骤：

a）　用户使用用户设备中的用户代理访问依赖方；

b）　依赖方将用户代理重定向至身份服务提供方；

c）　身份服务提供方向用户代理发送鉴别请求；

d）　用户使用用户设备的鉴别器进行身份验证；

e）　用户身份验证通过后，鉴别器将鉴别响应通过用户代理传递给身份服务提供方；

f）　身份服务提供方对鉴别响应进行验证；

g）　身份服务提供方通过用户代理将身份鉴别结果重定向至依赖方。

5.1.2　注册

在用户向身份鉴别服务器进行注册的流程中，用户使用生物特征识别密钥管理器创建一对新的鉴别公私钥并且将鉴别私钥保存在生物特征识别密钥管理器中，将鉴别公钥注册在身份鉴别服务器中。注册流程如图 2 所示，图中采用 HTTP 重定向方式对协议流程进行说明，详细流程和消息见 5.3.1。

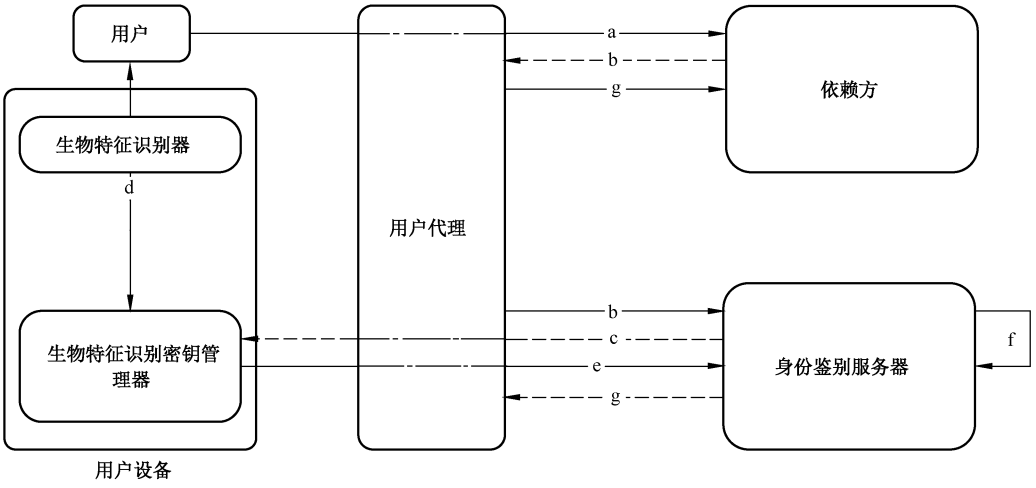

图 2　注册/鉴别流程

a) 用户使用用户设备中的用户代理访问依赖方。

b) 当用户需要进行生物特征识别身份鉴别注册时，依赖方将用户定向到身份鉴别服务器。可使用 HTTP 重定向方式将用户设备重定向到身份鉴别服务器，或者使用消息转发方式。

c) 身份鉴别服务器向用户设备中的生物特征识别密钥管理器发送注册请求消息；用户设备的生物特征识别密钥管理器在收到身份鉴别服务器的注册请求消息时，验证身份鉴别服务器的真实性，验证通过则提示用户选择本地可用的生物特征识别器，否则拒绝该消息。本文件中涉及生物特征识别密钥管理器验证身份鉴别服务器真实性的方法，宜采用证书方式或者其他可验证身份鉴别服务器真实性的方法。

d) 用户选择本地合适的生物特征识别器，使用生物特征识别信息解锁生物特征识别密钥管理器，完成用户生物特征识别验证。如果用户之前未将生物特征识别信息登记到该生物特征识别器，则进行登记；如果用户已进行登记，则使用已登记的生物特征识别信息完成解锁过程。用户生物特征识别验证成功后，生物特征识别密钥管理器创建一对与生物特征识别密钥管理器、身份鉴别服务器相关联的唯一的鉴别公私钥，鉴别私钥保存在本地的生物特征识别密钥管理器，并且不允许从生物特征识别密钥管理器导出。如果生物特征识别密钥管理器没有能力保存鉴别私钥，则该生物特征识别密钥管理器将鉴别私钥进行加密，然后将加密后的鉴别私钥保存在用户设备中，用于加密用户私钥的密钥则保存在生物特征识别密钥管理器中并且不允许从生物特征识别密钥管理器导出；本协议中有关密钥等关键安全参数的全生命周期管理或安全管理的要求应符合 GB/T 37092 的规定。

e) 生物特征识别密钥管理器生成密钥注册数据，然后生成注册响应消息，将注册响应消息发送到身份鉴别服务器。其中，密钥注册数据中包含上一步生成的鉴别公钥，注册响应消息中包含密钥注册数据，以及使用厂商私钥对密钥注册数据进行签名的签名值。厂商公私钥为用户设备制造厂商在生物特征识别密钥管理器中预先植入的用于证明生物特征识别密钥管理器身份的密钥对。

f) 身份鉴别服务器通过 5.2.7 的生物特征识别密钥管理器信息获取厂商公钥，利用厂商公钥验证注册响应消息中的签名，签名正确则提取出鉴别公钥并保存该鉴别公钥，同时应保存该鉴别公钥与用户之间的对应关系。

g) 身份鉴别服务器将注册结果返回给依赖方。

5.1.3 鉴别

在鉴别流程中，用户通过生物特征识别密钥管理器使用鉴别私钥对服务器挑战签名，向身份鉴别服务器证明其拥有该私钥，完成身份鉴别过程。鉴别流程如图 2 所示，图中采用 HTTP 重定向方式对协议流程进行说明，协议详细流程和消息见 5.3.2。

a) 用户使用用户设备中的用户代理访问依赖方。

b) 当用户需要进行生物特征识别身份鉴别时，依赖方将用户定向到身份鉴别服务器，可使用 HTTP 重定向方式将用户设备重定向到身份鉴别服务器，或者使用消息转发方式。

c) 身份鉴别服务器向用户设备中的生物特征识别密钥管理器发送鉴别请求消息；用户设备的生物特征识别密钥管理器在收到身份鉴别服务器的鉴别请求消息时，验证身份鉴别服务器的真实性，验证通过则提示用户选择可用的生物特征识别器，否则拒绝该消息。本文件中涉及生物特征识别密钥管理器验证身份鉴别服务器真实性的方法，宜采用证书方式或者其他可验证身份鉴别服务器真实性的方法。

d) 用户选择本地合适的生物特征识别器，使用生物特征识别信息解锁存储在生物特征识别密钥管理器中的用户鉴别私钥，完成用户生物特征识别验证。用户生物特征识别验证成功后，生物

特征识别密钥管理器选择相应的鉴别私钥对待签名数据签名。

e) 生物特征识别密钥管理器将包含签名结果的鉴别响应消息发送到身份鉴别服务器。

f) 身份鉴别服务器使用相应的鉴别公钥对签名进行验证，验证成功后，用户身份鉴别成功，否则用户身份鉴别失败。

g) 身份鉴别服务器将鉴别结果返回给依赖方。

5.1.4 注销

当用户删除在依赖方注册的账户，或丢失用户设备，或依赖方在需要删除用户绑定的公私钥等情形下，身份鉴别服务器应发起注销流程。其中，身份鉴别服务器在执行注销流程前应鉴别用户的身份，应使用 5.1.3 中规定的鉴别流程，若无法使用上述鉴别流程，则使用其他身份鉴别方式，例如线下身份鉴别方式。具体注销流程如图 3 所示，图中采用 HTTP 重定向方式对协议流程进行说明，图中为完整的注销流程，若丢失用户设备，则不需要向生物特征识别密钥管理器发送注销请求消息。

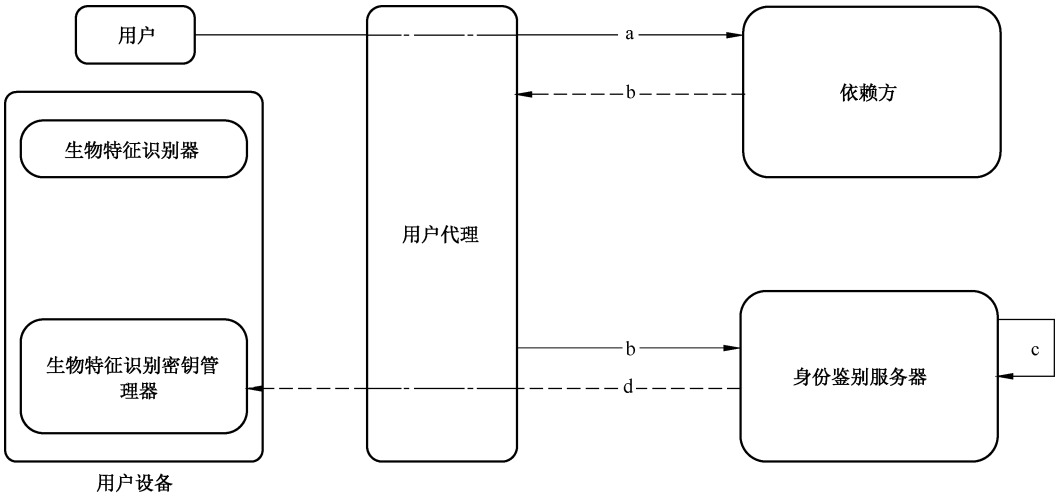

图 3 注销流程

a) 用户使用用户设备中的用户代理向依赖方发起注销流程。

b) 依赖方将用户定向到身份鉴别服务器，可使用 HTTP 重定向方式将用户设备重定向到身份鉴别服务器，或者使用消息转发方式。首先进行用户的身份鉴别，鉴别通过后执行后续注销流程，否则拒绝执行注销流程。

c) 身份鉴别服务器删除相应的鉴别公钥，并向生物特征识别密钥管理器发送注销请求消息。

d) 用户设备的生物特征识别密钥管理器在收到身份鉴别服务器的注销请求消息时，验证身份鉴别服务器的真实性，验证通过则删除相应的鉴别私钥，否则拒绝该消息。

5.2 协议消息相关数据结构

5.2.1 概述

通用在线快捷身份鉴别协议所有数据格式采用 GB/T 16262 描述。

5.2.2 版本

```
Version::=SEQUENCE{
    major       INTEGER,
```

```
    minor      INTEGER
}
```

major：主要版本。

minor：次要版本。

本文件的协议主要版本为1，次要版本为0。

5.2.3 操作类型

```
Operation::=PrintableString
```

本文件有以下三种操作类型：

Reg：注册，Operation值为“Reg”。

Auth：鉴别，Operation值为“Auth”。

Dereg：注销，Operation值为“Dereg”。

5.2.4 扩展

```
Extension::=SEQUENCE{
    id              PrintableString,
    data            PrintableString,
    fail_if_unknown    BOOLEAN
}
```

该结构描述的是在各种操作中使用的通用扩展。

id：扩展的标识符。

data：该值可包含任意字符串，身份鉴别服务器和生物特征识别密钥管理器应对该值的语义协商一致。该值可为空。

fail_if_unknown：当fail_if_unknown是false时表示未知扩展应被忽略，当fail_if_unknown是true时表示未知扩展应导致错误。

5.2.5 操作头

```
OperationHeader::=SEQUENCE{
    bapv            Version,
    op              Operation,
    appID           PrintableString,
    serverData      PrintableString OPTIONAL,
    exts            SEQUENCE OF Extension OPTIONAL
}
```

bapv：在线快捷身份鉴别协议版本，抽象语法标记(ASN.1)描述见5.2.2。

op：值应为“Reg”，“Auth”或者“Dereg”，在线快捷身份鉴别协议操作的类型，ASN.1描述见5.2.3。

appID：依赖方应用程序标识符。

serverData：依赖方创建的会话标识符。

exts：扩展列表，ASN.1描述见5.2.4。

5.2.6 最终挑战参数

```
FinalChallengeParams::=SEQUENCE{
    appID           PrintableString,
```

```
    challenge          PrintableString
}
```

appID：该参数值设置为操作头的 appID 参数值。

challenge：该参数值设置为注册请求或鉴别请求中的服务器挑战参数值，服务器挑战是服务器提供的随机参数值。

5.2.7 生物特征识别密钥管理器信息

身份鉴别服务器可获取生物特征识别密钥管理器相关信息，例如生物特征识别密钥管理器厂商在生成厂商密钥时所使用的相关算法和信息，以及生物特征识别密钥管理器生成鉴别公私钥时所使用的相关算法和信息。身份鉴别服务器可使用生物特征识别密钥管理器相关提供方的服务来获取这些信息。

5.2.8 断言描述信息

```
AssertionInfo::=SEQUENCE{
    version                    Version,
    mode                       AuthenticationMode,
    signatureAlgAndEncoding    INTEGER,
    publicKeyAlgAndEncoding    INTEGER OPTIONAL
}
```

version：断言的版本，ASN.1 描述见 5.2.2。

mode：断言模式。

AuthenticationMode：用户验证方式，ASN.1 描述如下：

```
AuthenticationMode::=INTEGER{explicitly(0x01)}
```

explicitly：应进行显式的用户验证，例如指纹输入、虹膜扫描等。

signatureAlgAndEncoding：该值代表签名算法相关信息，身份鉴别服务器和生物特征识别密钥管理器应对该值的语义协商一致，即能够通过该值在身份鉴别过程中使用一致的签名算法和相关参数。在通用在线快捷身份鉴别协议中，宜采用 SM2 算法，SM2 算法的实现应符合 GB/T 32918 的规定，该值为 0x0007。

publicKeyAlgAndEncoding：该值代表公钥算法相关信息，身份鉴别服务器和生物特征识别密钥管理器应对该值的语义协商一致，即能够通过该值在身份鉴别过程中使用一致的公钥算法和相关参数。在通用在线快捷身份鉴别协议中，宜采用 SM2 算法，SM2 算法的实现应符合 GB/T 32918 的规定，该值为 0x0007。

5.2.9 生物特征识别密钥管理器计数器值

```
Counters::=SEQUENCE{
    SignCounter        INTEGER,
    RegCounter         INTEGER
}
```

SignCounter：签名计数器的值。

RegCounter：注册计数器的值。

5.2.10 发现数据

```
DiscoveryData::=SEQUENCE{
```

```
    supportedBAPVersions      SEQUENCE OF Version,
    clientVendor              PrintableString,
    clientVersion             Version,
    availableBAPKeyManagers SEQUENCE OF PrintableString
}
```

描述：

supportedBAPVersions：支持的在线快捷身份鉴别协议版本，ASN.1 描述见 5.2.2。

clientVendor：生物特征识别密钥管理器厂商。

clientVersion：生物特征识别密钥管理器版本。

availableBAPKeyManagers：可使用的生物特征识别密钥管理器标识符。

5.2.11 错误码

```
ErrorCode::=INTEGER{
    NO_ERROR(0x0),
    WAIT_USER_ACTION(0x1),
    INSECURE_TRANSPORT(0x2),
    USER_CANCELLED(0x3),
    UNSUPPORTED_VERSION(0x4),
    NO_SUITABLE_BAPKEYMANAGER(0x5),
    PROTOCOL_ERROR(0x6),
    UNTRUSTED_FACET_ID(0x7),
    KEY_DISAPPEARED_PERMANENTLY(0x09),
    BAPKEYMANAGER_ACCESS_DENIED(0x0c),
    USER_NOT_RESPONSIVE(0x0e),
    INSUFFICIENT_BAPKEYMANAGER_RESOURCES(0x0f),
    USER_LOCKOUT(0x10),
    USER_NOT_ENROLLED(0x11),
    UNKNOWN(0xff)
}
```

描述：

NO_ERROR：操作完成，没有错误发生。

WAIT_USER_ACTION：等待用户操作。

INSECURE_TRANSPORT：不安全的传输，例如没有使用 HTTPS。

USER_CANCELLED：用户取消当前操作。

UNSUPPORTED_VERSION：生物特征识别密钥管理器不支持该版本的协议消息。

NO_SUITABLE_BAPKEYMANAGER：没有与身份鉴别服务器策略相匹配的生物特征识别密钥管理器。

PROTOCOL_ERROR：发生协议错误。

UNTRUSTED_FACET_ID：不受信任的依赖方应用程序。

KEY_DISAPPEARED_PERMANENTLY：鉴别私钥丢失并且无法恢复。

BAPKEYMANAGER_ACCESS_DENIED：生物特征识别密钥管理器拒绝访问。

USER_NOT_RESPONSIVE：用户长时间无响应。

INSUFFICIENT_BAPKEYMANAGER_RESOURCES：生物特征识别密钥管理器没有足够的资

源执行操作。

USER_LOCKOUT:由于用户锁定,操作无法执行。

USER_NOT_ENROLLED:用户没有登记。

UNKNOWN:以上没有列出的其他错误。

5.2.12 协议消息

协议消息是通用在线快捷身份鉴别协议的各个参与方交互时传递的消息,ASN.1 描述如下:

```
BAPMessage::=SEQUENCE{
    bapProtocolMessage      OCTET STRING,
    additionalData          OCTET STRING
}
```

bapProtocolMessage:协议消息,例如注册请求消息、注册响应消息、鉴别请求消息、鉴别响应消息、注销请求消息等。

additionalData:附加参数。

5.3 协议流程和要求

5.3.1 注册流程

5.3.1.1 概述

图 4 描述用户注册流程,本流程中用户访问依赖方进行用户名和口令验证的过程作为注册流程的前提条件,不在本流程中进行说明。

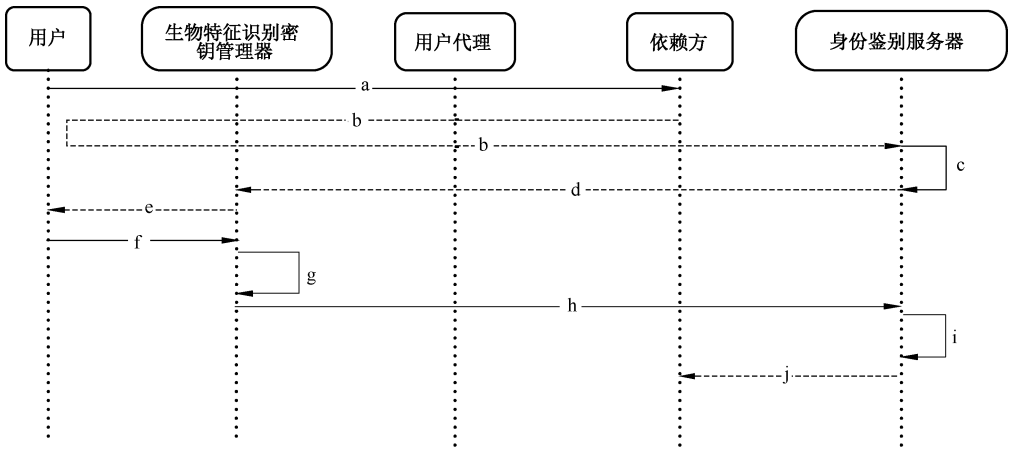

图 4 注册/鉴别详细流程

a) 用户通过用户代理访问依赖方,发起注册流程。

b) 依赖方将用户重定向到身份鉴别服务器。

c) 身份鉴别服务器生成注册请求消息,见 5.3.1.2.1 a)步骤。

d) 身份鉴别服务器通过用户代理将注册请求消息发送至生物特征识别密钥管理器,见 5.3.1.2.1 a)步骤。

e) 生物特征识别密钥管理器首先验证身份鉴别服务器的真实性,然后生物特征识别密钥管理器发起本地用户校验,提示用户使用生物特征识别信息进行身份验证,见 5.3.1.2.2 a)、b)步骤。

f) 用户提交生物识别信息,例如指纹、虹膜等信息。

g) 生物特征识别密钥管理器验证用户提交的生物识别信息,验证通过后,生成一对新的鉴别公

私钥，然后生成注册响应消息，注册响应消息中包含使用厂商私钥对鉴别公钥等信息的签名，见 5.3.1.2.2 c)、d)步骤。

h) 生物特征识别密钥管理器通过用户代理将注册响应消息返回给身份鉴别服务器，见 5.3.1.2.2 e)步骤。

i) 身份鉴别服务器使用厂商公钥验证注册响应消息，验证成功后存储相关信息，否则返回错误信息，见 5.3.1.2.3 a)～c)步骤。

j) 身份鉴别服务器将结果返回给依赖方。

5.3.1.2 注册流程处理规则

5.3.1.2.1 身份鉴别服务器生成注册请求规则

身份鉴别服务器应遵循以下步骤：

a) 创建注册请求消息，并初始化注册请求消息的各个参数。注册请求消息格式如下：

```
RegistrationRequest::=SEQUENCE{
    header          OperationHeader,
    challenge       PrintableString,
    username        PrintableString
}
```

header：操作头，header.op 的值应为“Reg”，ASN.1 描述见 5.2.5。

challenge：身份鉴别服务器提供的挑战值。

username：用户在某依赖方的账号名称；

b) 将注册请求消息发送给生物特征识别密钥管理器。

5.3.1.2.2 生物特征识别密钥管理器处理注册请求规则

生物特征识别密钥管理器应遵循以下步骤：

a) 验证身份鉴别服务器的真实性，验证成功则执行以下步骤，否则拒绝该消息；

b) 解析注册请求消息，判断注册请求消息是否包含必要的参数以及每个参数是否符合要求，若符合要求则执行以下步骤，否则拒绝该消息；

c) 提示用户选择本地生物特征识别器，用户选择后，使用用户选择的本地生物特征识别器验证用户，验证通过后执行以下操作，否则返回错误；

d) 创建注册响应消息，并根据注册请求消息的参数初始化注册响应消息的参数。注册响应消息格式如下：

```
RegistrationResponse::=SEQUENCE{
    header          OperationHeader,
    fcParams        FinalChallengeParams,
    assertion       RegistrationAssertion
}
```

header：操作头，header.op 应为“Reg”，ASN.1 描述见 5.2.5。

fcParams：最终挑战参数 FinalChallengeParams，ASN.1 描述见 5.2.6。该参数使用 UTF8 编码，然后使用[RFC 4627]定义的序列化方法序列化后，再使用 base64url[RFC 4648]对其进行编码后的值。

assertions：注册请求断言，生物特征识别密钥管理器的响应数据，ASN.1 描述如下：

```
RegistrationAssertion::=SEQUENCE{
```

```
    assertionScheme        PrintableString,
    assertion              RegAssertion,
    exts                   SEQUENCE OF Extension OPTIONAL
}
```

assertionScheme：用来编码断言的断言模式名称。例如，该值可为“BAPV1TLV”。

exts：生物特征识别密钥管理器支持的扩展，ASN.1 描述见 5.2.4。

assertion：注册断言，ASN.1 描述如下：

```
RegAssertion::={
    krd            KRD,
    sig            BIT STRING
}
```

krd：密钥注册数据，数据格式如下所示：

```
KRD::=SEQUENCE{
    bkmID              PrintableString,
    assertionInfo      AssertionInfo,
    challengeHash      BIT STRING,
    keyID              PrintableString,
    counters           Counters,
    uauthPubKey        OCTET STRING
}
```

bkmID：生物特征识别密钥管理器标识符。

assertionInfo：断言描述信息，ASN.1 描述见 5.2.8。

challengeHash：服务器挑战的 SM3 杂凑值，SM3 算法应符合 GB/T 32905 的规定。

keyID：鉴别私钥标识符。

counters：生物特征识别密钥管理器计数器值，ASN.1 描述见 5.2.9。

uauthPubKey：生成的鉴别公钥，是符合 GB/T 32918 规定的公钥。

sig：使用厂商私钥，利用 SM2 算法对 krd 进行签名后的签名值，格式应符合 GB/T 35276 的规定，使用的 SM2 算法应当符合密码国家标准、行业标准的相关要求；

e) 将注册响应消息发送给身份鉴别服务器。

5.3.1.2.3 身份鉴别服务器处理注册响应规则

身份鉴别服务器应遵循以下步骤：

a) 解析注册响应消息，判断注册响应消息是否包含必要的参数以及每个参数是否符合要求，若符合要求则执行以下步骤，否则拒绝该消息；

b) 使用厂商公钥验证注册响应消息中签名的正确性；

c) 如果注册响应消息通过验证，将用户鉴别公钥等相关信息保存在身份鉴别服务器，完成注册，返回注册成功；否则返回注册失败。

5.3.2 鉴别流程

5.3.2.1 概述

鉴别流程如图 4 所示，具体流程如下：

a) 用户通过用户代理访问依赖方，发起鉴别流程；

b) 依赖方将用户重定向到身份鉴别服务器；
c) 身份鉴别服务器生成鉴别请求消息，见 5.3.2.2.1 a)步骤；
d) 身份鉴别服务器通过用户代理将鉴别请求消息发送至生物特征识别密钥管理器，见 5.3.2.2.1 b)步骤；
e) 生物特征识别密钥管理器首先验证身份鉴别服务器的真实性，然后生物特征识别密钥管理器发起本地用户校验，提示用户使用生物特征识别信息进行身份验证，见 5.3.2.2.2 a)、b)步骤；
f) 用户提交生物识别信息，例如指纹、虹膜等信息；
g) 生物特征识别密钥管理器验证用户提交的生物识别信息，验证通过后，生成鉴别响应消息，见 5.3.2.2.2 c)、d)步骤；
h) 生物特征识别密钥管理器通过用户代理将鉴别响应消息返回给身份鉴别服务器，见 5.3.2.2.2 e)步骤；
i) 身份鉴别服务器验证鉴别响应消息，见 5.3.2.2.3 a)～c)步骤；
j) 身份鉴别服务器将结果返回给依赖方。

5.3.2.2 鉴别流程处理规则

5.3.2.2.1 身份鉴别服务器生成鉴别请求规则

身份鉴别服务器应遵循以下步骤：

a) 创建鉴别请求消息，并初始化鉴别请求消息的各个参数。鉴别请求消息格式如下：

```
AuthenticationRequest::=SEQUENCE{
    header          OperationHeader,
    challenge       PrintableString
}
```

header：操作头，header.op 的值应为“Auth”，ASN.1 描述见 5.2.5。

challenge：服务器提供的挑战值；

b) 将鉴别请求消息发送给生物特征识别密钥管理器。

5.3.2.2.2 生物特征识别密钥管理器处理鉴别请求规则

生物特征识别密钥管理器应遵循以下步骤：

a) 验证身份鉴别服务器的真实性，验证成功则执行以下步骤，否则拒绝该消息；
b) 解析鉴别请求消息，判断鉴别请求消息是否包含必要的参数以及每个参数是否符合要求，若符合要求则执行以下步骤，否则拒绝该消息；
c) 提示用户选择本地生物特征识别器，用户选择后，使用用户选择的生物特征识别器验证用户，验证通过后执行以下操作，否则返回错误；
d) 创建鉴别响应消息，并根据鉴别请求消息的参数初始化鉴别响应消息的各个参数。鉴别响应消息如下：

```
AuthenticationResponse::=SEQUENCE{
    header          OperationHeader,
    fcParams        FinalChallengeParams,
    assertions      AuthAssertion
}
```

header：操作头，header.op 应为“Auth”，ASN.1 描述见 5.2.5。

fcParams：最终挑战参数 FinalChallengeParams，ASN.1 描述见 5.2.6。该参数使用 UTF8 编码，然后使用[RFC 4627]定义的序列化方法序列化后，再使用 base64url[RFC 4648]对其进行编码后的值。

assertions：AuthAssertion 对象，生物特征识别密钥管理器针对鉴别请求生成的签名断言，ASN.1 描述如下：

```
AuthAssertion::=SEQUENCE{
    assertionScheme     PrintableString,
    assertion           SignAssertion,
    exts                SEQUENCE OF Extension OPTIONAL
}
```

assertionScheme：用来编码断言的断言模式名称。例如，该值可为“BAPV1TLV”。

exts：生物特征识别密钥管理器支持的扩展，ASN.1 描述见 5.2.4。

assertion：生物特征识别密钥管理器针对鉴别请求数据生成的签名断言，ASN.1 描述如下：

```
SignAssertion::={
    signData            SignData,
    sig                 BIT STRING
}
```

signData：待签名数据，数据格式如下所示：

```
SignData::=SEQUENCE{
    bkmID               PrintableString,
    assertionInfo       AssertionInfo,
    nonce               OCTET STRING,
    challengeHash       OCTET STRING,
    keyID               PrintableString,
    counters            Counters
}
```

bkmID：生物特征识别密钥管理器标识符。

assertionInfo：断言描述信息，ASN.1 描述见 5.2.8。

nonce：生物特征识别密钥管理器生成的随机临时参数值。

challengeHash：服务器挑战的 SM3 杂凑值，SM3 算法应符合 GB/T 32905 的规定。

keyID：鉴别私钥标识符。

counters：生物特征识别密钥管理器计数器值，ASN.1 描述见 5.2.9。

sig：使用鉴别私钥，利用 SM2 算法对 signData 进行签名后的签名值，格式应符合 GB/T 35276 的规定，使用的 SM2 算法应当符合密码国家标准、行业标准的相关要求；

e） 将鉴别响应消息发送给身份鉴别服务器。

5.3.2.2.3 身份鉴别服务器处理鉴别响应规则

身份鉴别服务器应遵循以下步骤：

a） 解析鉴别响应消息，判断鉴别响应消息是否包含必要的参数以及每个参数是否符合要求，若符合要求则执行以下步骤，否则拒绝该消息；

b） 使用鉴别公钥验证鉴别响应消息的正确性；

c) 如果验证通过，则鉴别成功，否则失败。

5.3.3 注销流程

5.3.3.1 概述

图 5 描述注销流程：

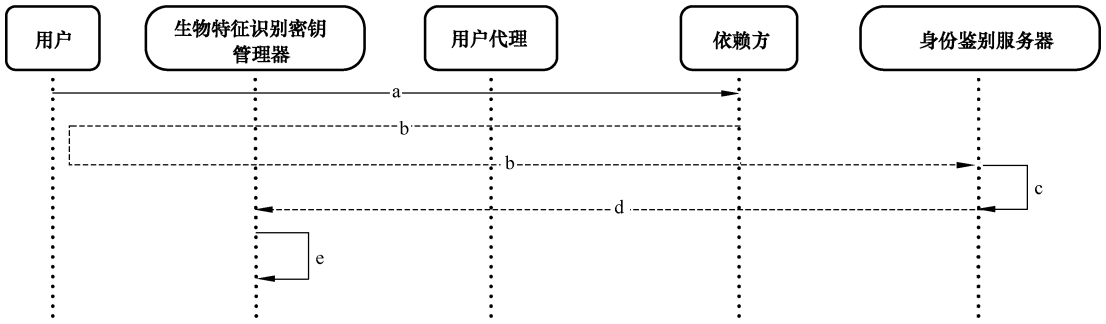

图 5 注销详细流程

a) 用户通过用户代理访问依赖方，发起注销流程；
b) 依赖方将用户重定向到身份鉴别服务器，身份鉴别服务器首先进行用户身份鉴别，鉴别通过后进行后续注销流程，否则拒绝执行注销流程，见 5.3.3.2.1 a)步骤；
c) 身份鉴别服务器生成注销请求消息，删除与该用户相关的数据，见 5.3.3.2.1 b)步骤；
d) 身份鉴别服务器通过用户代理将注销请求消息发送至生物特征识别密钥管理器，见 5.3.3.2.1 c)步骤；
e) 生物特征识别密钥管理器删除用户相关数据，见 5.3.3.2.2 a)、b)步骤。

5.3.3.2 注销流程处理规则

5.3.3.2.1 身份鉴别服务器生成注销请求规则

身份鉴别服务器应遵循以下步骤：

a) 创建注销请求消息，并初始化注销请求消息的各个参数。注册请求消息格式如下：

```
DeregistrationRequest::=SEQUENCE{
    header              OperationHeader,
    bapKeyManagers SEQUENCE OF DeregisterBAPKeyManager
}
```

header：操作头，header.op 应为“Dereg”，ASN.1 描述见 5.2.5。

bapKeyManagers：要注销的生物特征识别密钥管理器(DeregisterBAPKeyManager)列表，DeregisterBAPKeyManager 的 ASN.1 描述如下：

```
DeregisterBAPKeyManager::=SEQUENCE{
    bkmaID          PrintableString,
    keyID           PrintableString
}
```

bkmaID：要注销的生物特征识别密钥管理器标识符(bkmID)。

keyID：与鉴别私钥相关联的唯一的密钥标识符(KeyID)；

b) 删除身份鉴别服务器上与该用户相关的数据；

c) 将注销请求消息发送到生物特征识别密钥管理器。

5.3.3.2.2 生物特征识别密钥管理器处理注销请求规则

生物特征识别密钥管理器应遵循以下步骤：

a) 解析注销请求消息，判断注销响应消息是否包含必要的参数以及每个参数是否符合要求，若符合要求则执行以下步骤，否则拒绝该消息；

b) 删除对应 KeyID 的密钥对等相关数据，注销成功。

6 双因素在线快捷身份鉴别协议

6.1 协议架构

6.1.1 概述

针对已部署使用用户名/口令等方式鉴别用户的依赖方，双因素在线快捷身份鉴别协议将鉴别器作为用户身份鉴别的第二因素，提升用户身份鉴别的安全性。协议规定了鉴别器应完成的功能，并且规定了协议架构、协议消息框架、协议流程和要求以及协议接口，其中，安全风险及措施建议见附录 A，关于协议接口见附录 C。在实际部署过程中使用的商用密码产品，应当由具备资格的商用密码检测、认证机构检测认证合格后，方可部署到实际系统中。

在双因素在线快捷身份鉴别协议中，包括五个实体：鉴别器、鉴别客户端、用户代理、依赖方、身份服务提供方。鉴别器通过用户代理与身份服务提供方通信，负责生成密钥对和签名，将公钥发送给身份服务提供方用于用户身份的绑定和身份鉴别，将私钥存储在鉴别器中并保证其安全性。身份服务提供方通过用户代理与鉴别器通信，负责鉴别用户身份，提供用户鉴别结果。鉴别客户端负责构造发送给鉴别器的请求消息，接收鉴别器发送的响应消息。用户代理需支持鉴别客户端接口。

双因素在线快捷身份鉴别协议由注册、鉴别和注销三种会话组成。

注册：用户将鉴别器生成的鉴别公钥发送给身份服务提供方，身份服务提供方将鉴别公钥与用户身份进行绑定。

鉴别：在使用用户名口令鉴别用户身份的基础之上，同时使用已注册的鉴别公私钥进行身份鉴别。

注销：用户删除与身份服务提供方关联的鉴别公私钥对。

三种会话的协议流程中，协议参与方应保护协议消息数据的机密性。用户代理与身份服务提供方的通信宜采用传输层密码协议(TLCP，GB/T 38636)。鉴别客户端与鉴别器宜通过 USB、NFC、蓝牙等传输协议进行通信。

本协议中有关密钥等关键安全参数的全生命周期管理或安全管理的要求应符合 GB/T 37092 的规定。

6.1.2 注册

在用户向依赖方进行注册时，鉴别器生成一对鉴别公私钥对，将鉴别公钥发送给依赖方。依赖方保存鉴别公钥，并与用户身份绑定，完成注册过程。会话流程(注册流程)如图 6 所示，图中采用 HTTP 重定向方式对协议交互情况进行说明。

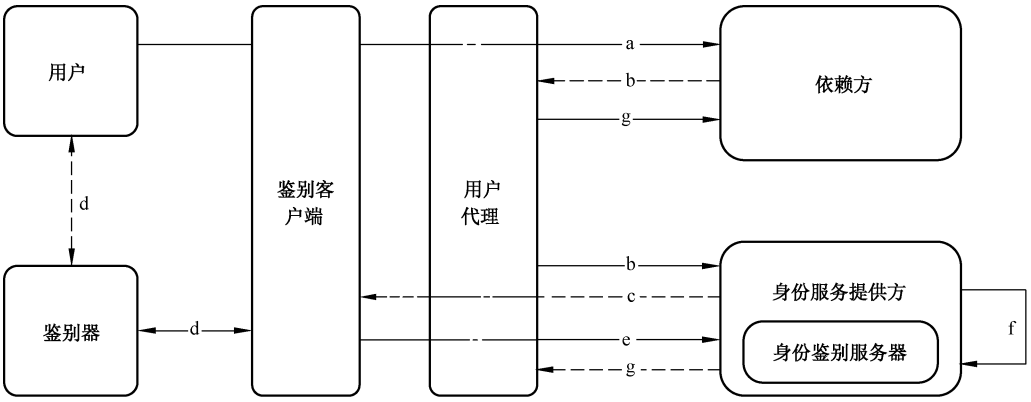

图 6　会话流程(注册流程/鉴别流程)

a)　用户通过用户代理访问依赖方。

b)　当用户需要进行身份鉴别注册时,依赖方将用户定向到身份鉴别服务器。可使用 HTTP 重定向方式将用户设备重定向到身份鉴别服务器,或者使用消息转发方式。

c)　身份鉴别服务器验证用户名和口令成功后,构造注册请求参数,通过用户代理调用鉴别客户端的注册接口。

d)　鉴别客户端验证身份鉴别服务器的身份,验证通过后构造注册请求消息,将注册请求消息发送给鉴别器。鉴别器与用户进行交互验证用户身份,其中,图中 d 虚线部分代表用户与鉴别器的交互,比如用户使用指纹或 PIN 码等。

e)　鉴别器验证用户身份成功后,生成一对鉴别公私钥,构造包含鉴别公钥和密钥句柄等内容的注册响应消息,通过鉴别客户端和用户代理将注册响应消息发送给身份鉴别服务器。

f)　身份鉴别服务器验证鉴别器身份并返回验证结果,若验证通过,则保存对应的鉴别公钥和密钥句柄。

g)　身份鉴别服务器将注册结果返回给依赖方。

6.1.3　鉴别

在鉴别流程中,鉴别器使用鉴别私钥对挑战等信息进行签名,向依赖方证明其拥有该鉴别私钥,完成身份鉴别过程。会话流程(鉴别流程)如图 6 所示,图中采用 HTTP 重定向方式对协议交互情况进行说明。

a)　用户通过用户代理访问依赖方。

b)　当用户需要进行身份鉴别时,依赖方将用户定向到身份鉴别服务器。可使用 HTTP 重定向方式将用户设备重定向到身份鉴别服务器,或者使用消息转发方式。

c)　身份鉴别服务器验证用户名和口令成功后,构造鉴别请求参数,通过用户代理调用鉴别客户端的签名接口。

d)　鉴别客户端验证身份鉴别服务器的身份,验证通过后构造鉴别请求消息,将鉴别请求消息发送给鉴别器。鉴别器与用户进行交互验证用户身份,其中,图中 d 虚线部分代表用户与鉴别器的交互,比如用户使用指纹或 PIN 码等。

e)　鉴别器验证用户身份成功后,选择对应的鉴别私钥对挑战等信息进行签名,构造鉴别响应消息,并将鉴别响应消息通过鉴别客户端和用户代理发送给身份鉴别服务器。

f)　身份鉴别服务器使用对应的鉴别公钥对签名进行验证,验证成功则用户鉴别成功,否则用户鉴别失败。

g） 身份鉴别服务器将鉴别结果返回给依赖方。

6.1.4 注销

在注销流程中，依赖方和鉴别器分别删除相应的鉴别公钥和鉴别私钥。注销流程如图 7 所示，图中采用 HTTP 重定向方式对协议交互情况进行说明：

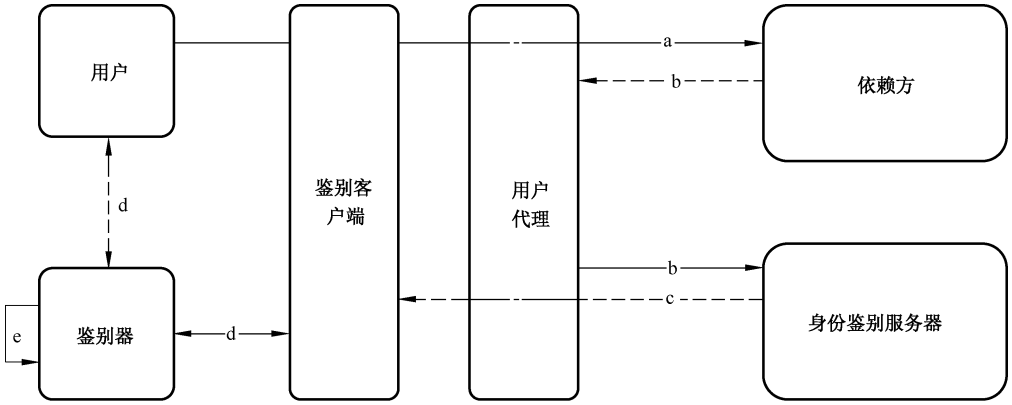

图 7 注销流程

a） 用户在身份鉴别成功后，通过用户代理向依赖方发起注销流程；

b） 依赖方将用户定向到身份鉴别服务器，可使用 HTTP 重定向方式将用户设备重定向到身份鉴别服务器，或者使用消息转发方式，首先进行用户的身份鉴别，鉴别通过后执行后续注销流程，否则拒绝执行注销流程；

c） 身份鉴别服务器删除相应的鉴别公钥，构造注销请求参数，通过用户代理调用鉴别客户端的注销接口；

d） 鉴别客户端构造注销请求消息，发送给鉴别器；

e） 鉴别器收到注销请求消息后，验证身份鉴别服务器的真实性，验证通过则删除相应的鉴别私钥，否则拒绝该消息。

6.2 协议消息框架

6.2.1 概述

双因素在线快捷身份鉴别协议主要包括请求消息和响应消息。请求消息是由鉴别客户端创建，由鉴别器处理；响应消息是由鉴别器创建，由依赖方使用。本文件规定了请求和响应消息的编码方式。本文件不规定用户代理与鉴别器之间的通信方式，可使用 USB、NFC 或蓝牙等协议传输。本文件基于 GB/T 16649.4 规定鉴别客户端与鉴别器之间通信的消息封装框架。

6.2.2 协议消息框架

6.2.2.1 请求消息框架

请求消息被封装为命令 APDU，封装方式如下：

CLA INS P1 P2 [Lc 〈request-data〉] [Le]

其中：

CLA：预留字节，鉴别客户端将此字节设置为 0。

INS：双因素在线快捷身份鉴别协议命令代码。

P1，P2：参数 1 和 2，由每个命令定义。

Lc：请求数据 request-data 的长度。如果没有请求数据，则省略 Lc。

request-data：请求消息，表示注册请求消息或鉴别请求消息，具体格式分别见注册流程 6.3.1.2.1 与鉴别流程 6.3.2.2.1。

Le：响应数据的最大期望长度。如果没有响应数据，则省略 Le。

APDU 有两种不同的编码：短编码和扩展长度编码。

6.2.2.1.1 短编码

在短编码中，请求数据的最大长度为 255 字节。其中，Lc 的编码方式如下：

令 Nc=|〈request-data〉|。若 Nc 值为 0，则省略 Lc；若 Nc 值不为 0，则将 Lc 编码为表示 Nc 值的单个字节。

如果指令不产生响应数据，则省略 Le。否则，在短编码中，Le 的编码方式如下：

令 Ne 等于期望响应数据的最大长度，将 Le 编码为 Ne 值。

6.2.2.1.2 扩展长度编码

在扩展长度编码中，请求数据的最大长度为 65535 字节。其中，Lc 的编码方式如下：

令 Nc=|〈request-data〉|。若 Nc 值为 0，则省略 Lc。否则，将 Lc 编码为：

0 MSB(Nc)LSB(Nc)

即，在请求数据前有三个长度字节，其中：

第一个字节值为 0；

第二个字节为 MSB(Nc)，表示 Nc 的最高有效字节；

第三个字节为 LSB(Nc)，表示 Nc 的最低有效字节。

如果指令不产生响应数据，则省略 Le。否则，在扩展长度编码中，Le 按如下方式编码：

令 Ne 等于期望响应数据的最大长度(在扩展长度编码中，Ne 的最大值为 65536 字节)。

若 Ne 在 1～65535(含 1 和 65535)，令 Le1＝MSB(Ne)，表示 Ne 的最高有效字节；令 Le2＝LSB(Ne)，表示 Ne 的最低有效字节。

若 Ne＝65536 时，令 Le1＝0，Le2＝0。

当 Lc 存在，即 Nc＞0 时，Le 编码为：

Le1 Le2

当 Lc 不存在，即 Nc＝0 时，Le 编码为：

0 Le1 Le2

即，当 Lc 不存在时，Le 以 0 作为单字节前缀。

6.2.2.2 响应消息框架

响应消息封装为响应 APDU，封装方式如下：

〈response-data〉 SW1 SW2

其中：

response-data：响应消息，包括注册响应消息、鉴别响应消息，具体格式分别见注册流程 6.3.1.2.3 与鉴别流程 6.3.2.2.3。

SW1、SW2：分别是状态码字节 1 和状态码字节 2，形成一个 16 位状态码。状态码见 6.2.2.3。

6.2.2.3 状态码

以下 ISO 7816-4 定义的状态码在双因素在线快捷身份鉴别协议中有特殊含义：

‘9000’：命令成功完成。

‘6985’:用户验证失败,请求被拒绝。

‘6984’:由于密钥句柄无效,请求被拒绝。

协议实现时可定义其他状态码,提供其他错误信息。用户代理只处理上面列出的错误代码,其他错误将被视为一般错误。

6.3 协议流程和要求

6.3.1 注册流程

6.3.1.1 概述

注册详细流程如图 8 所示:

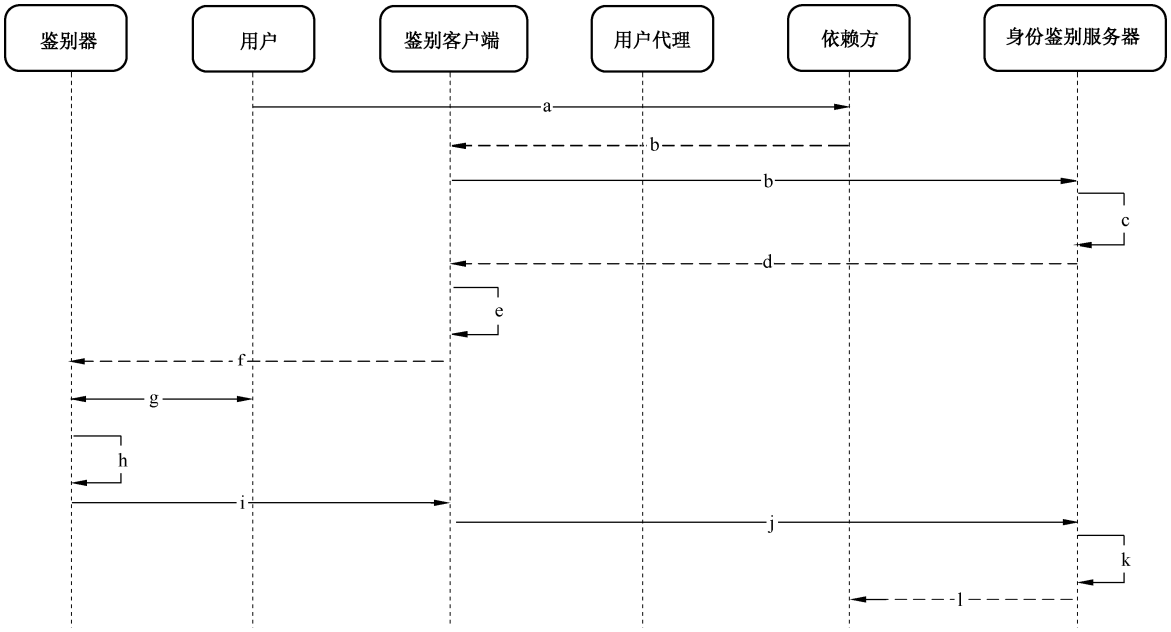

图 8 注册/鉴别详细流程

a) 用户使用用户名和口令通过用户代理访问依赖方,发起注册流程;

b) 依赖方将用户重定向到身份鉴别服务器;

c) 身份鉴别服务器验证用户名和口令;

d) 用户名和口令验证成功后,身份鉴别服务器构造注册请求参数,通过用户代理调用鉴别客户端的注册接口,见 6.3.1.2.1 a)、b)步骤;

e) 鉴别客户端根据注册请求参数,构造注册请求消息,见 6.3.1.2.2 a)步骤;

f) 鉴别客户端将注册请求消息发送给鉴别器,见 6.3.1.2.2 b)步骤;

g) 鉴别器与用户进行交互以验证用户身份,见 6.3.1.2.3 a)步骤;

h) 鉴别器验证用户身份通过后,生成一对仅对应于该依赖方的鉴别公私钥和密钥句柄等,构造注册响应消息,见 6.3.1.2.3 b)、c)步骤;

i) 鉴别器将注册响应消息发送给鉴别客户端,见 6.3.1.2.3 d)步骤;

j) 鉴别客户端通过用户代理将注册响应消息发送给身份鉴别服务器;

k) 身份鉴别服务器验证鉴别器身份,若验证通过则存储对应的鉴别公钥和密钥句柄,否则注册失败,见 6.3.1.2.4 a)～c)步骤;

l) 身份鉴别服务器将验证结果返回给依赖方。

6.3.1.2 注册流程处理规则

6.3.1.2.1 身份鉴别服务器调用注册接口规则

身份鉴别服务器调用注册接口应遵循以下步骤。

a) 身份鉴别服务器验证用户名和口令成功后，构造注册请求参数，参数包括：

1) appid：该依赖方的应用程序标识符；

2) registerRequests：注册请求，格式如下：

```
registerRequest::=SEQUENCE{
    version         PrintableString,
    challenge       PrintableString
}
```

version：待注册的鉴别器应声明的协议版本。本文件中该参数值为“TAP_V1”。

challenge：身份鉴别服务器生成随机挑战，该随机挑战值的编码方式为 websafe-base64；

3) registeredKeys：已经注册的鉴别器的密钥句柄 RegisteredKey 的序列，RegisteredKey 格式如下：

```
RegisteredKey::=SEQUENCE{
    version         PrintableString,
    keyHandle       PrintableString,
    transports      SEQUENCE OF Transport,
    appID           PrintableString
}
```

version：待注册的鉴别器应声明的协议版本。本文件中该参数值为“TAP_V1”。

keyHandle：密钥句柄，以 websafe-base64 形式编码。

transports：该鉴别器支持的传输方式，该值可为空。

Transport 用来向鉴别客户端指示该密钥句柄使用的传输方式，定义如下：

```
Transport ::=PrintableString
```

本文件有以下四种传输方式：

bt：经典蓝牙传输方式，Transport 值为“bt”。

ble：低功耗蓝牙传输方式，Transport 值为“ble”。

nfc：近距离无线通信传输方式，Transport 值为“nfc”。

usb：USB 人机交互设备传输方式，Transport 值为“usb”。

appID：注册请求中的 appid 参数，该值可为空；

4) callback：身份鉴别服务器的回调函数句柄，该回调函数的参数为 RegisterResponse 和 Error。鉴别客户端收到鉴别器的注册响应消息后，调用该回调函数将注册响应消息发送给身份鉴别服务器。

RegisterResponse 格式如下：

```
RegisterResponse::=SEQUENCE{
    version             PrintableString,
    registrationData    PrintableString,
    clientData          PrintableString
}
```

version：注册的鉴别器声明的协议版本，本文件中该参数值为“TAP_V1”。

registrationData：原始注册响应，编码方式为 websafe-base64。

clientData：由用户代理创建的 websafe-base64 编码的鉴别客户端数据。

Error 格式如下：

```
Error ::=SEQUENCE{
    errorCode       ErrorCode,
    errorMessage    PrintableString Optional
}
```

errorCode：错误编码，格式如下：

```
ErrorCode::=ENUM{
    const short OK=0,
    const short OTHER_ERROR=1,
    const short BAD_REQUEST=2,
    const short CONFIGURATION_UNSUPPORTED=3,
    const short DEVICE_INELIGIBLE=4,
    const short TIMEOUT=5
}
```

OK：成功。

OTHER_ERROR：未能被列举出的其他错误。

BAD_REQUEST：不能被处理的请求。

CONFIGURATION_UNSUPPORTED：鉴别客户端不支持。

TIMEOUT：超时。

errorMessage：错误结果的描述信息；

5) opt_timeoutSeconds：可选，请求超时时间。

b) 身份鉴别服务器调用鉴别客户端的注册接口，通过用户代理向鉴别客户端传递以上请求参数。

6.3.1.2.2 鉴别客户端生成注册请求规则

鉴别客户端生成注册请求消息应遵循以下步骤。

a) 鉴别客户端构造注册请求消息，包含挑战参数 challenge 和应用程序参数 application 两部分，注册请求消息格式如下：

```
REGISTER::=SEQUENCE{
    challenge       BIT STRING,
    application     BIT STRING
}
```

1) challenge：挑战参数。挑战参数是鉴别客户端生成的客户端数据 ClientData 的 SM3 散列值，SM3 算法应符合 GB/T 32905 的规定，ClientData 为 JSON 数据结构，且包含来自依赖方的挑战，描述形式如下：

```
ClientData::=SEQUENCE{
    typ             PrintableString,
    challenge       PrintableString,
    origin          PrintableString
}
```

typ：操作类型。typ 是常量，注册时值为 navigator.id.finishEnrollment，身份鉴别时值为

navigator.id.getAssertion。

challenge：依赖方生成的随机挑战。该随机挑战值的编码方式为 websafe-base64。

origin：依赖方的 AppID；

2） application：应用程序参数。应用程序参数是依赖方的 AppID 的 UTF-8 编码的 SM3 散列值，SM3 算法应符合 GB/T 32905 的规定。

b） 鉴别客户端将注册请求消息发送给鉴别器。

6.3.1.2.3 鉴别器处理注册请求规则

鉴别器处理注册请求应遵循以下步骤：

a） 鉴别器与用户进行交互，可使用指纹、PIN 码等验证用户，以确保用户允许鉴别器的注册操作；

b） 解析注册请求消息。判断鉴别请求消息是否包含必要的参数，以及每个参数是否符合要求；

c） 构造注册响应消息。注册响应消息分为用户验证失败响应消息和注册成功响应消息两种：

1） 用户验证失败响应消息：

当鉴别器无法完成用户交互验证时，由鉴别器构造用户验证失败响应消息，用户验证失败响应消息的状态码应为 SW_CONDITIONS_NOT_SATISFIED；

2） 注册成功响应消息：

当鉴别器完成用户验证后，鉴别器根据 GB/T 32918 规定的 SM2 算法创建新鉴别密钥对，鉴别私钥保存在鉴别器中，鉴别公钥通过注册成功响应消息发送给依赖方。注册成功响应消息格式如下：

```
REGISTER_SUCCESS::=SEQUENCE{
    reversed_byte               BIT STRING,
    user_public_key_length      BIT STRING,
    user_public_key             BIT STRING,
    key_handle_length           BIT STRING,
    key_handle                  BIT STRING,
    certificate                 BIT STRING,
    signature                   BIT SRTING
}
```

reversed_byte：保留字节，值为 0x05。

user_public_key_length：鉴别公钥长度字节，值为 0x40。

user_public_key：鉴别公钥。

key_handle_length：密钥句柄长度字节。指定密钥句柄的长度，且是无符号的（范围 0～255）。

key_handle：密钥句柄。由鉴别器来标识生成的密钥对，鉴别器可封装所生成的鉴别私钥及应用程序标识符，并将其作为密钥句柄输出。

certificate：厂商公钥证书，证书格式为 X.509 DER。

signature：签名值，是对以下字节字符串使用厂商私钥进行签名后的签名值：

RFU[1 字节]：保留字节，值为 0x00。

application[32 字节]：来自注册请求消息的应用程序参数。

challenge[32 字节]：来自注册请求消息的挑战参数。

key_handle_length[1 字节]：密钥句柄长度字节。

key_handle[可变长]：密钥句柄。密钥句柄长度不包括在签名基本字符串中。

user_public_key_length[1 字节]：鉴别公钥长度字节，值为 0x40。

user_public_key[64 字节]：鉴别公钥。

d) 将注册响应消息发送给鉴别客户端，通过用户代理将注册响应消息发送给身份鉴别服务器。

6.3.1.2.4 身份鉴别服务器处理注册响应规则

身份鉴别服务器处理注册响应消息应遵循以下步骤：

a) 解析注册响应消息，判断注册响应消息中各参数是否符合规范。若符合则执行以下步骤，否则返回注册失败；

b) 首先验证注册响应消息中的证书由可信第三方签发，然后使用证书中的厂商公钥验证签名。若验证通过则执行以下步骤，否则返回注册失败；

c) 身份鉴别服务器存储注册响应消息中鉴别公钥和密钥句柄等特定信息，并与用户身份绑定，完成注册，返回注册成功。

6.3.2 鉴别流程

6.3.2.1 概述

鉴别详细流程如图 8 所示。

a) 用户使用用户名和口令通过用户代理访问依赖方，发起鉴别流程；

b) 依赖方将用户重定向到身份鉴别服务器；

c) 身份鉴别服务器验证用户名和口令；

d) 用户名和口令验证成功后，身份鉴别服务器构造签名请求参数，通过用户代理调用鉴别客户端的签名接口，见 6.3.2.2.1 a)、b)步骤；

e) 鉴别客户端根据签名请求参数，构造鉴别请求消息，见 6.3.2.2.2 a)步骤；

f) 鉴别客户端将鉴别请求消息发送给鉴别器，见 6.3.2.2.2 b)步骤；

g) 鉴别器与用户进行交互以验证用户身份，见 6.3.2.2.3 a)、b)步骤；

h) 鉴别器验证用户身份通过后，使用鉴别器中存储的对应鉴别私钥对消息进行签名，构造鉴别响应消息，见 6.3.2.2.3 c)步骤；

i) 鉴别器将鉴别响应消息发送给鉴别客户端，见 6.3.2.2.3 d)步骤；

j) 鉴别客户端通过用户代理将鉴别响应消息发送给身份鉴别服务器；

k) 身份鉴别服务器使用对应的鉴别公钥验证鉴别响应消息，见 6.3.2.2.4 a)、b)、c)步骤；

l) 身份鉴别服务器将验证结果返回给依赖方。

6.3.2.2 鉴别流程处理规则

6.3.2.2.1 身份鉴别服务器调用签名接口规则

身份鉴别服务器调用签名接口应遵循以下步骤。

a) 身份鉴别服务器构造签名请求参数，包括：

1) appid：该依赖方的应用程序标识符；

2) challenge：身份鉴别服务器生成随机挑战，该随机挑战值的编码方式为 websafe-base64；

3) registeredKeys：在注册时，身份鉴别服务器存储的用户密钥句柄列表，格式见 6.3.1.2.1；

4) callback：身份鉴别服务器的回调函数句柄，该回调函数的参数为 SignResponse 和 Error，Error 格式见 6.3.1.2.1。鉴别客户端收到鉴别器的鉴别响应消息后，调用该回调函数将鉴别响应消息发送给身份鉴别服务器。

SignResponse 格式如下：

SignResponse::=SEQUENCE{

```
    keyHandle       PrintableString,
    signatureData   PrintableString,
    clientData      PrintableString
}
```

keyHandle：用于签名的 RegisteredKey 的密钥句柄。

signatureData：来自鉴别器的 websafe-base64 编码的原始响应。

clientData：由用户代理创建的 websafe-base64 编码的鉴别客户端数据。

如果有多个鉴别器响应了身份验证请求，则用户代理将选择其中一个响应并将其传递给调用者；

5) opt_timeoutSeconds：可选，请求超时时间。

b) 身份鉴别服务器调用鉴别客户端的签名接口，通过用户代理向鉴别客户端传递以上请求参数。

6.3.2.2.2 鉴别客户端生成鉴别请求规则

鉴别客户端生成鉴别请求应遵循以下步骤。

a) 鉴别客户端构造鉴别请求消息。鉴别请求消息格式如下：

```
AUTHENTICATE::=SEQUENCE{
    control                 BIT STRING,
    challenge               BIT STRING,
    application             BIT STRING,
    key_handle_length       BIT STRING,
    key_handle              BIT STRING
}
```

control：控制字节(P1)，控制字节由用户代理决定，依赖方不能指定其值。

1) 控制字节为 0x07(“check-only”)，表示鉴别器应检查请求中所提供的密钥句柄是否由该鉴别器创建，以及是否为依赖方的 AppID 对应的应用创建。在注册期间，可发送控制字节为 0x07(“check-only”)的鉴别请求消息，以验证鉴别器是否注册。如果检查通过，鉴别器返回鉴别响应消息 error：test of user-press-required，则该鉴别器已注册。如果密钥句柄不是由该鉴别器创建，或者不是依赖方的 AppID 对应的应用创建，鉴别器返回鉴别响应消息 error：bad-key-handle。

2) 控制字节为 0x03(“force-user-press-and-sign”)，要求鉴别器与用户完成交互验证后，才可提供签名操作。鉴别器使用对应私钥签名后，返回鉴别响应消息 success 或其他错误响应。在鉴别期间，鉴别客户端应发送控制字节为 0x03(“force-user-press-and-sign”)的鉴别请求消息。

challenge：挑战参数，挑战参数是由鉴别客户端生成的客户端数据 ClientData 的 SM3 散列值，SM3 算法应符合 GB/T 32905 的规定，ClientData 包含来自依赖方的挑战，ClientData 应为 JSON 数据结构，描述形式如下：

```
ClientData::=SEQUENCE{
    typ             PrintableString,
    challenge       PrintableString,
    origin          PrintableString
}
```

typ：操作类型。typ 是常量，注册时为 navigator.id.finishEnrollment，身份鉴别时为 navi-

gator.id.getAssertion。

challenge：依赖方生成随机挑战。该随机挑战值的编码方式为 websafe-base64。

origin：依赖方的 AppID。

application：应用程序参数。应用程序参数是依赖方的 AppID 的 UTF-8 编码的 SM3 散列值，SM3 算法应符合 GB/T 32905 的规定。

key_handle_length：密钥句柄长度。指定密钥句柄的长度，该值是无符号的(范围 0～255)。

key_handle：密钥句柄。由依赖方提供，依赖方在用户注册时获得。

b) 鉴别客户端将鉴别请求消息发送给鉴别器。

6.3.2.2.3 鉴别器处理鉴别请求规则

鉴别器处理鉴别请求应遵循以下步骤。

a) 鉴别器与用户进行交互，可使用指纹、PIN 码等验证用户，以确保用户允许鉴别器的签名操作。

b) 解析鉴别请求消息。判断鉴别请求消息是否包含必要的参数，以及每个参数是否符合要求。

c) 构造鉴别响应消息。鉴别响应消息分为用户验证失败响应消息、密钥句柄错误响应消息、鉴别成功响应消息，详细格式如下：

1) 用户验证失败响应消息：

当鉴别器无法完成用户交互验证时，由鉴别器输出用户验证失败响应消息。用户验证失败响应消息的状态码为 SW_CONDITIONS_NOT_SATISFIED；

2) 密钥句柄错误响应消息：

如果鉴别请求消息中的密钥句柄不是由该鉴别器创建，或不是依赖方的 AppID 对应的应用创建，由鉴别器输出密钥句柄错误响应消息，状态码为 SW_WRONG_DATA；

3) 鉴别成功响应消息：

解析鉴别请求消息成功后，鉴别器使用存储的对应鉴别私钥进行签名，由鉴别器返回鉴别成功响应消息。鉴别成功响应消息格式如下：

```
AUTHENTICATE_SUCCESS::=SEQUENCE{
        user_presence     BIT STRING,
        counter           INTEGER,
        signature         BIT SRTING
}
```

user_presence：用户存在性字节，若为 0 表示用户交互验证失败，若为 1 表示用户交互验证通过。

counter：计数器值，鉴别器每次执行身份鉴别操作时增加计数器值。

signature：签名值，是对以下字节字符串使用鉴别私钥进行签名后的签名值：

application[32 字节]：鉴别请求消息中的依赖方的 AppID。

user_presence[1 字节]：用户存在性字节。

counter[4 字节]：计数器值。

challenge[32 字节]：鉴别请求消息中的挑战参数。

d) 鉴别器将鉴别响应消息发送给鉴别客户端，通过用户代理将鉴别响应消息发送给身份鉴别服务器。

6.3.2.2.4 身份鉴别服务器处理鉴别响应规则

身份鉴别服务器处理鉴别响应消息应遵循以下步骤：

a) 解析鉴别响应消息，判断鉴别响应消息是否包含必要的参数，以及每个参数是否符合要求。若

符合要求则执行以下步骤，否则返回鉴别失败；

b） 使用该用户绑定的鉴别公钥，验证签名；

c） 如果验证通过，则返回鉴别成功，否则返回鉴别失败。

6.3.3 注销流程

6.3.3.1 概述

注销详细流程图如图 9 所示：

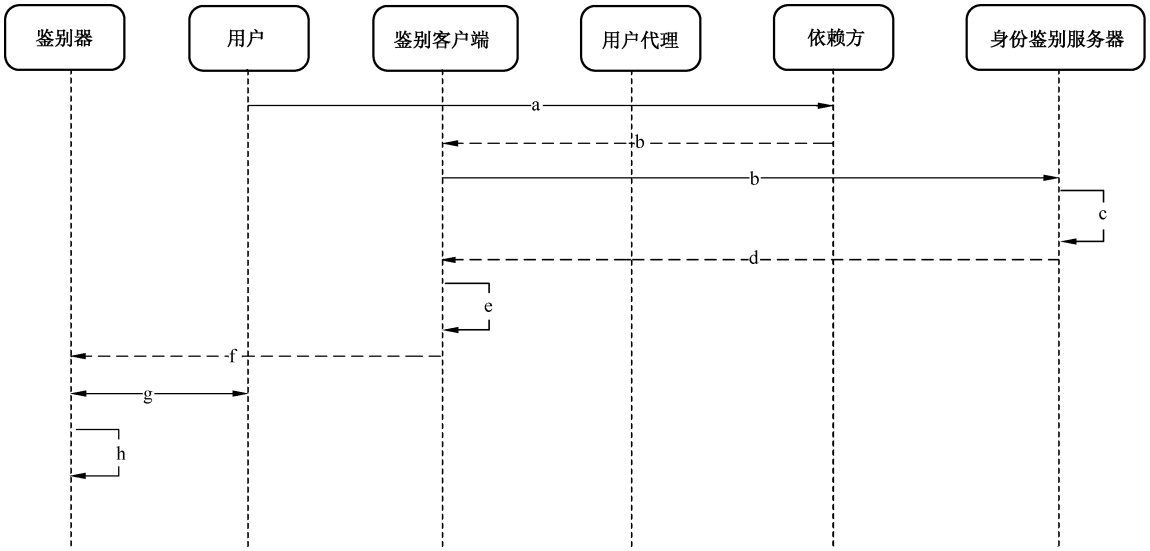

图 9 注销详细流程

a） 用户使用用户名和口令通过用户代理访问依赖方，发起注销流程；

b） 依赖方将用户重定向到身份鉴别服务器；

c） 身份鉴别服务器验证用户名和口令，用户名和口令验证成功后，身份鉴别服务器删除对应用户绑定的鉴别公钥和密钥句柄等信息；

d） 身份鉴别服务器构造注销请求参数，通过用户代理调用鉴别客户端的注销接口，见 6.3.3.2.1 a)、b)步骤；

e） 鉴别客户端根据注销请求参数，构造注销请求消息，见 6.3.3.2.2 a)步骤；

f） 鉴别客户端将注销请求消息发送给鉴别器，见 6.3.3.2.2 b)步骤；

g） 鉴别器与用户交互验证用户身份，见 6.3.3.2.3 a)、b)步骤；

h） 用户交互验证通过后，鉴别器删除鉴别私钥，见 6.3.3.2.3 c)步骤。

6.3.3.2 注销流程处理规则

6.3.3.2.1 身份鉴别服务器调用注销接口规则

身份鉴别服务器调用注销接口应遵循以下步骤：

a） 身份鉴别服务器构造注销请求参数，参数包括：

appid：该依赖方的应用程序标识符。

registeredKeys：在注册时，身份鉴别服务器存储的用户密钥句柄列表，RegisteredKey 格式见 6.3.1.2.1；

b） 身份鉴别服务器通过用户代理调用鉴别客户端的注销接口。

6.3.3.2.2 鉴别客户端生成注销请求消息规则

鉴别客户端生成注销请求消息应遵循以下步骤：

a) 鉴别客户端构造注销请求消息，注销请求消息格式如下：

```
UNREGISTRATION::=SEQUENCE{
        registeredKeys      SEQUENCE OF RegisteredKey
}
```

registeredKeys：在注册时，依赖方存储的用户密钥句柄列表，RegisteredKey 格式见 6.3.1.2.1；

b) 鉴别客户端将注销请求消息发送给鉴别器。

6.3.3.2.3 鉴别器处理注销请求消息规则

鉴别器处理注销请求应遵循以下步骤：

a) 鉴别器与用户进行交互，可使用指纹、PIN 码等验证用户，以确保用户允许鉴别器的注销操作。若用户交互验证成功则执行以下步骤，否则拒绝该消息；

b) 解析注销请求消息，判断注销请求消息是否包含必要的参数，以及每个参数是否符合要求。若符合要求执行以下步骤，否则拒绝该消息；

c) 鉴别器删除对应 keyID 的鉴别私钥，注销成功。

附 录 A
（资料性）
安全风险及措施建议

A.1 恶意依赖方

攻击者可能伪装成合法依赖方，攻击者获取到用户隐私信息后，可伪装成该用户，以该用户的身份向真正的依赖方注册鉴别器。

依赖方需加强注册期间的身份鉴别及用户口令的初始强度。用户在与依赖方通信时，需验证对方的真实性。

A.2 客户端恶意软件

攻击者可能在用户设备上安装恶意软件，恶意篡改生物特征识别密钥管理器数据或者运算执行过程。

需保证生物特征识别密钥管理器在可信环境中执行，采用操作系统特权分离机制，避免受到可信环境之外的任何恶意修改。

A.3 用户设备丢失

攻击者可能获得对用户设备的物理访问权，访问依赖方。

通过鉴别器保护鉴别密钥，防止鉴别密钥的滥用。在使用鉴别密钥之前，鉴别器要求用户使用指定的方式来交互验证用户身份。攻击者将无法通过身份验证，因此无法使用鉴别密钥访问依赖方。如果用户申请对丢失设备上的公钥进行远程注销，需要向依赖方核验身份，若依赖方可采用身份核验措施确认该用户合法拥有账户，或者采用备用鉴别手段（如短信验证码等）确认用户合法拥有账户，则可以将该账户绑定的公钥删除或作废，用户即可重新在新设备上完成注册过程。

A.4 服务器验证错误

客户端无法正确验证远程服务器标识，攻击者可修改依赖方的身份鉴别策略，或者将用户鉴别方式修改为易于攻击的鉴别方式。攻击者还可截获发送给依赖方的消息，进行并行会话攻击、转发攻击、伪造等。

依赖方和客户端应验证彼此身份，在通信时可使用 TLCP 等安全传输协议。依赖方需验证与客户端之间安全通道的连续性，并且在消息中加入随机数 Nonce、随机数挑战等，抵抗并行会话攻击、转发攻击等。

A.5 攻击依赖方应用

攻击者可能通过跨站点脚本攻击等手段，在依赖方的安全上下文中执行恶意操作，或者在用户身份验证成功后滥用会话标识。

鉴别器需通过安全用户界面来可信显示依赖方名称和用户需要确认的事务数据。

A.6 中间人攻击

攻击者可在通信各方毫不知情的情况下，通过拦截鉴别器、用户代理和依赖方之间的正常网络通信数据，进行数据篡改，实施中间人攻击。

可使用鉴别器实现的安全用户界面来可信显示依赖方名称和用户需要确认的事务数据。

A.7 重放攻击

攻击者可重复发送消息以验证用户身份。

服务器在消息中加入随机数 Nonce 来检测消息的重放，服务器随机生成挑战发送给客户端，验证与客户端返回响应中的挑战是否一致，以此来保证事务数据的完整性。

A.8 滥用鉴别密钥

恶意应用声称访问一个依赖方，可实际上访问另一个依赖方，试图滥用在其声称访问的依赖方注册的鉴别密钥。

鉴别器需向用户提供展示界面，显示依赖方的名称，用于用户对操作的确认。鉴别器需限制注册密钥的使用安全边界。依赖方和鉴别器宜采用 AppID 分离机制，应用程序仅能访问与其应用程序标识符相关联的注册密钥。

A.9 假冒客户端

攻击者可欺骗用户安装假冒的应用程序。

鉴别器实现的安全用户界面可显示真实的依赖方名称和用户需要确认的事务数据。

A.10 恶意鉴别器

恶意鉴别器可能是由攻击者仿制，也可能是由厂商生产的留有后门的不合格鉴别器。用户可能无法识别恶意鉴别器。

依赖方需只允许注册合格厂商生产的经过认证的鉴别器，并且定期更新相关数据，识别合法的鉴别器。

A.11 鉴别密钥泄露

攻击者窃取鉴别密钥，使用克隆的鉴别器冒充用户。

鉴别器需保护鉴别密钥，防止误用，在使用密钥之前，要求用户使用鉴别器指定的方式来交互验证用户身份。

鉴别器需使用单调递增的签名计数器，每次执行鉴别操作时增加计数器的值，签名中包含计数器值，将签名发送给依赖方。依赖方可通过检查签名计数器值来检测是否有克隆的鉴别器存在。

当鉴别器进行状态检查发现异常时，或鉴别器制造商发布鉴别器破坏的情况时，依赖方会收到鉴别器鉴别密钥泄露或受损的情况，需执行相关的操作。

A.12 用户验证数据注入攻击

攻击者可能会将预先捕获的用户验证数据注入鉴别器。例如，如果使用口令作为用户验证方法，攻击者可捕获用户输入的口令，将正确的口令发送到鉴别器(绕过键盘/PIN 面板等)，仿冒用户身份登录。口令可在攻击前获取，例如说服用户将口令输入恶意应用程序，或直接或间接监视口令数据。

依赖方或鉴别器可使用物理安全的验证输入，如使用指纹感知器，或者不能被恶意软件绕过的可信用户接口。

A.13 用户验证数据泄露

攻击者读取捕获的用户验证数据，获取到个人识别信息(PII)。

设备厂商需仅限鉴别器访问验证数据，可有效降低该风险。

A.14 生物特征识别器与生物特征识别密钥管理器之间通信安全问题

攻击者可能截获生物特征识别器与生物特征识别密钥管理器之间通信的数据，获得相关的用户数据。

设备生产厂商需保证生物特征识别器与生物特征识别密钥管理器之间的通信安全，防止通信数据的泄露。

A.15 依赖方与身份鉴别服务器之间通信安全问题

攻击者可能截获依赖方与身份鉴别服务器之间通信的数据，获得相关的用户数据。

依赖方与身份鉴别服务器通信之前，需采用数字证书等方式验证双方的身份，验证通过再进行通信。

A.16 功能滥用

攻击者可能向鉴别器发送无效的命令或带有无效参数的命令，导致绕过用户验证过程。

鉴别器厂商需提升鉴别器固件的健壮性。

A.17 固件回滚

攻击者可能可以安装旧版本的或存在 bug 的固件，导致成功的攻击。

鉴别器厂商需验证固件的稳健性。

附 录 B
（资料性）
可信环境实现方式

B.1 概述

目前针对可信环境的实现方式包括 TPM、TEE、SecureElement。

B.2 TPM

需通过调用 TPM、API 实现本文件中指定的鉴别器的功能，并且支持相关的用户鉴别技术，如生物特征识别、PIN 码或其他等。

B.3 TEE

需在 TEE 内部设计一个特定的 trustlet，即在 TEE 内部运行的可信应用程序，用于实现本文件中指定的鉴别器的功能，并且支持相关的用户鉴别技术，如生物特征识别、PIN 码或其他等。

B.4 Secure Elements

需在 Secure Elements 内部设计一个特定的 Applet，即在 SE 内部运行的可信应用程序，用于实现本文件中指定的鉴别器的功能，并且支持相关的用户鉴别技术，如生物特征识别、PIN 码或其他等。

附 录 C
（资料性）
协议接口

C.1 通用在线快捷身份鉴别协议接口

C.1.1 概述

通用在线快捷身份鉴别协议的主要接口是生物特征识别密钥管理器接口，如图 C.1 所示：

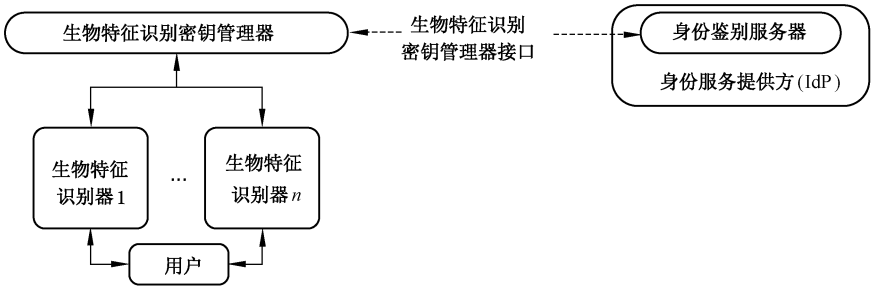

图 C.1 协议接口

生物特征识别密钥管理器接口是生物特征识别密钥管理器提供给身份鉴别服务器的接口，身份鉴别服务器可通过用户代理调用该接口完成通用在线快捷身份鉴别协议规定的操作。

C.1.2 生物特征识别密钥管理器接口

C.1.2.1 概述

本附录定义了生物特征识别密钥管理器的接口，该接口包含六个方法，分别是发现方法、执行操作方法、通知结果方法、发现回调、响应回调、异常回调。生物特征识别密钥管理器的接口定义如下：

```
interface bap{
    void discover (DiscoveryCallback completionCallback,ErrorCallback errorCallback);
     void processBAPOperation (BAPMessage message, BAPResponseCallback completionCallback,ErrorCallback errorCallback);
    void notifyBAPResult (int responseCode,BAPMessage BAPResponse);
}。
```

C.1.2.2 发现方法

void discover (DiscoveryCallback completionCallback,ErrorCallback errorCallback)

在执行在通过线快捷身份鉴别协议之前，身份鉴别服务器调用该方法，检查用户设备是否支持通用在线快捷身份鉴别协议，发现方法参数见表 C.1：

表 C.1 发现方法参数

参数	类型	可为空	可选	描述
completionCallback	DiscoveryCallback	否	否	用于接收生物特征识别密钥管理器发现数据的回调
errorCallback	ErrorCallback	否	否	用于接收错误码和错误信息的回调

返回类型：void。

C.1.2.3 执行操作方法

void processBAPOperation（BAPMessage message，BAPResponseCallback completionCallback，ErrorCallback errorCallback）

身份鉴别服务器调用该方法，执行通用在线快捷身份鉴别协议的注册操作、鉴别操作或者注销操作。执行操作方法参数见表 C.2：

表 C.2 执行操作方法参数

参数	类型	可为空	可选	描述
message	BAPMessage	否	否	生物特征识别密钥管理器将处理的 BAPMessage
completionCallback	BAPResponseCallback	否	否	用于接收生物特征识别密钥管理器发送给身份服务提供方的身份鉴别服务器的响应消息
errorCallback	ErrorCallback	否	否	用于接收错误码和信息的回调

返回类型：void。

C.1.2.4 发现回调

发现回调用于生物特征识别密钥管理器在异步执行完成发现过程后将发现数据返回给身份鉴别服务器，方法参数如下：

callback DiscoveryCallback＝void（DiscoveryData data）；

参数：

data：DiscoveryData 类型，描述当前身份鉴别服务器可使用的生物特征识别密钥管理器和生物特征识别密钥管理器当前的状态。

C.1.2.5 通知结果方法

void notifyBAPResult（int responseCode，BAPMessage bapResponse）

当身份鉴别服务器接收并处理协议消息后，应调用该方法，将身份鉴别服务器响应状态码返回给生物特征识别密钥管理器，通知结果方法参数见表 C.3：

表 C.3 通知结果方法参数

参数	类型	可为空	可选	描述
responseCode	int	否	否	服务器响应状态码
bapResponse	BAPMessage	否	否	该响应状态码对应的响应消息

返回类型：void。

C.1.2.6 响应回调

响应回调用于生物特征识别密钥管理器在异步执行完成操作（例如注册、鉴别）后将协议消息返回给身份鉴别服务器。方法参数如下：

参数：

callback BAPResponseCallback＝void（BAPMessage bapResponse）；

参数：

bapResponse：BAPMessage 类型，生物特征识别密钥管理器返回的响应消息。

C.1.2.7 异常回调

异常回调用于生物特征识别密钥管理器在异步执行操作时返回操作的错误信息。方法参数如下：

callback ErrorCallback＝void (ErrorCode code)；

参数：

code：ErrorCode 类型，用于描述操作的结果。

C.2 双因素在线快捷身份鉴别协议接口

C.2.1 概述

本文件规定依赖方使用用户代理与鉴别客户端进行交互的接口，不规定鉴别客户端传递消息给鉴别器的方式，可使用 USB、NFC 和蓝牙等协议传输。

C.2.2 双因素在线快捷身份鉴别客户端接口

Interface tap{

void register (DOMString appID, sequence〈RegisterRequest〉 registerRequests, sequence〈RegisteredKey〉 registeredKeys, function (RegisterResponse or Error) callback, optional unsigned long opt_timeoutSeconds)；

void sign (DOMString appID, DOMString challenge, sequence〈RegisteredKey〉 registeredKeys, function(SignResponse or Error) callback, optional unsigned long opt_timeoutSeconds)；

void unregister (DOMString appID, sequence〈RegisteredKey〉 registeredKeys)；

}。

C.2.2.1 注册方法

void register (DOMString appID, sequence〈RegisterRequest〉 registerRequests, sequence〈RegisteredKey〉 registeredKeys, function(RegisterResponse or Error) callback, unsigned long opt_timeoutSeconds)

注册方法参数见表 C.4：

表 C.4 注册方法参数

参数	类型	可空	可选	描述
appID	DOMString	否	否	对于请求的应用标识符
registerRequests	sequence〈RegisterRequest〉	否	否	依赖方接受的双因素在线快捷身份鉴别协议各个版本的注册请求序列
registeredKeys	sequence〈RegisteredKey〉	否	否	已经注册的鉴别器的密钥句柄序列
callback	function(RegisterResponse or Error)	否	否	响应句柄
opt_timeoutSeconds	unsigned long	是	是	鉴别客户端处理请求的超时数据，以秒计

返回类型：void

C.2.2.2 身份鉴别签名方法

void sign（DOMString appID，DOMString challege，sequence〈RegisteredKey〉 registeredKeys，function(SignResponse or Error) callback，unsigned long opt_timeoutseconds)

身份鉴别签名方法参数见表 C.5：

表 C.5 身份鉴别签名方法参数

参数	类型	可空	可选	描述
appID	DOMString	否	否	对于请求的应用标识符
challenge	DOMString	否	否	websafe-base64 编码的挑战
registeredKeys	sequence〈RegisteredKey〉	否	否	已经注册的鉴别器的密钥句柄序列
callback	function(SignResponse or Error)	否	否	响应句柄
opt_timeoutSeconds	unsigned long	是	是	鉴别客户端处理请求的超时数据，以秒计

返回类型：void

鉴别客户端接口应使用响应对象或错误对象调用提供的回调。通过测试非零 errorCode 键可检测到错误。

EXAMPLE 1

```
TAP.sign(reqs,function(response){
                if (response.errorCode){
                // response is an Error
                ...
                }else{
                //response is a SignResponse
                ...
                }
        })。
```

C.2.2.3 注销方法

void unregister (DOMString appID，sequence〈RegisteredKey〉 registeredKeys)

注销方法参数见表 C.6：

表 C.6 注销方法参数

参数	类型	可空	可选	描述
appID	DOMString	否	否	对于请求的应用标识符
registeredKeys	sequence〈RegisteredKey〉	否	否	已经注册的鉴别器的密钥句柄序列

参 考 文 献

[1] FIDO UAF Protocol Specification: FIDO Alliance Review Draft 05 October 2016 [S/OL]. [2016-10-05] https://fidoalliance.org/specs/fido-uaf-v1.1-rd-20161005/fido-uaf-protocol-v1.1-rd-20161005.html

[2] FIDO U2F Protocol Specification: FIDO Alliance Review Draft 15 September 2016 [S/OL].[2016-09-15] https://fidoalliance.org/specs/fido-u2f-v1.1-id-20160915/FIDO-U2F-COMPLETE-v1.1-id-20160915.pdf

[3] RFC 4627 The application/json Media Type for JavaScript Object Notation (JSON). https://tools.ietf.org/html/rfc4627

[4] RFC 4648 The Base16,Base32,and Base64 Data Encodings.https://tools.ietf.org/html/rfc4648