ICS 35.040
L 80

中华人民共和国国家标准

GB/T 15843.6—201X/ISO/IEC 9798-6:2010

信息技术 安全技术 实体鉴别 第6部分 采用人工数据传递的机制

Information technology—Security techniques—Entity authentication—

Part 6: Mechanisms using manual data transfer

(ISO/IEC 9798-6:2010, IDT)

（征求意见稿）

（本稿完成日期：2017年6月16日）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前　言

GB/T 15843《信息技术　安全技术　实体鉴别》分为六个部分：

——第1部分：概述

——第2部分：采用对称加密算法的机制

——第3部分：采用数字签名技术的机制

——第4部分：采用密码校验函数的机制

——第5部分：采用零知识技术的机制

——第6部分：采用人工数据传递的机制

可能还会增加其他后续部分。

本部分为GB/T 15843的第6部分，等同采用ISO/IEC 9798-6:2010《信息技术　安全技术　实体鉴别第6部分：采用人工数据传递的机制》（英文版），仅有编辑性修改，主要有：

——在第1章“范围”中，增加了对附录A的引用；

——ISO/IEC 9798-6:2010正文中所引用的国际标准，凡已被等同采用为国家标准的，以国家标准的标号替换。

本部分的附录A为规范性附录，其它附录为资料性附录。

本标准照GB/T 1.1-2009和GB/T 20000.2-2009给出的规则起草。

本部分由全国信息安全标准化技术委员会（SAC/TC260）提出并归口。

本部分主要起草单位：中国科学院数据与通信保护研究教育中心，北京数字认证股份有限公司，飞天诚信科技股份有限公司。

本部分主要起草人：夏鲁宁，张琼露，林雪焰，朱鹏飞。

引　言

GB/T 15843的本部分规定了采用人工数据传递的实体鉴别机制，包括使用短校验值的机制、使用短摘要值或短密钥的机制以及使用消息鉴别码的机制，并给出了对这些鉴别机制的要求。

在通信网络中，两个设备之间常需通过不安全信道进行实体鉴别。GB/T 15843的其它部分指定了多种实体鉴别机制，这些机制适用于两个设备已经事先共享了秘密密钥，或一个设备拥有另一个设备的公钥的情况。

在本部分中，并不做这种假设，即两个设备之间并没有事先共享的秘密密钥或已知对方公钥。在鉴别过程中，一段短数据被从一个设备人工传递到另一个，或以人工方式对两个设备输出的短数据做比对。

在本部分中，“实体鉴别”这个术语的含义也与其他部分有所不同。本部分的实体鉴别不是由一个设备来验证另一个设备声称的身份，而是针对由同一个用户掌控的两个设备做鉴别，在鉴别过程中这两个设备共享了一个短数据。短数据可能是短校验值、短摘要值、短密钥或消息鉴别码。共享的数据中可以包含两个设备中某个或双方的标识符。

如资料性附录B和C所描述的那样，人工鉴别机制可作为建立秘密密钥共享或可靠交换公钥的基础。此外，人工鉴别机制还可被用作其他秘密或公开安全参数的交换，包括安全策略声明或时间戳等。

本部分凡涉及密码算法的相关内容，按国家有关法规实施。

信息技术 安全技术 实体鉴别 第6部分：采用人工数据传递的机制

1 范围

GB/T15843的本部分规定了8种在设备之间基于人工数据传递进行实体鉴别的机制。本部分指明了这些机制如何被用来支持密钥管理功能，以及如何安全地选择各机制的参数。对于这8种机制，本部分附录A给出了其ASN.1定义，并在附录F对它们的安全性水平和效率进行了分析比较。

这些机制可以适用于多类应用场景。一种典型的应用是在个人网络中，作为设备接入网络的过程的一部分，用户对于自己掌握的两个具备无线通信能力的设备执行二者相互间的实体鉴别。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是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 15843.1-XXXX 信息技术 安全技术 实体鉴别 第1部分：概述（ISO/IEC 9798-1:2010，IDT）

3 术语和定义

GB/T 15843.1-XXXX中界定的，及以下术语和定义适用于本文件。

3.1

校验值 check-value

一个比特串，由某种校验函数计算产生，从通信的发起方传递给通信的接收方，且接收方有能力检验其正确性。

3.2

校验函数 check-value function

函数f，将一个比特串和一个短密钥映射为一个定长为b位的校验值，短密钥可容易地被输入到用户设备或从中读取。校验函数满足以下属性：

——对于任何密钥 k 和任何比特串 d，函数 $f(d,k)$可以被有效计算；

——寻找两个不同的比特串 d 和 d'，使得对于密钥 k 有 $f(d,k)=f(d',k)$，在计算上是不可行地，尽管能够满足上述等式的 k 值在 k 的取值空间中并不是一小部分。

注：在实践中，一个典型的短密钥包含4-6个数字或字母。

3.3

数据起源鉴别 data origin authentication

对于接收到的数据，确认其来源的真实性。

3.4

摘要值 digest-value

一个比特串，由某种摘要函数计算产生，从通信的发起方传递给通信的接收方，且接收方有能力检验其正确性。

3.5

摘要函数 digest function

函数d，将一个比特串和一个长密钥映射为一个定长为b位的摘要值。摘要值可以容易地被输入到用户设备中或从中读取，并满足以下属性：

——对于任何密钥 k 和任何比特串 m，$d(m,k)$可以被有效地计算；

——寻找两个不同的比特串 m 和 m'，使得对于密钥 k 有 $d(m,k)=d(m',k)$，在计算上是不可行地。满足这个等式的密钥占所有可能密钥取值的比例大于$(2^{-b}+\varepsilon)$，b 是摘要值的固定长度，ε是一个相对于 2^{-b}可忽略不计的数。

注 1：实践中，如果密钥 k 的长度是典型的密码杂凑值长度，例如 160 位，那么上述第二个属性应被满足。这个需求产生自杂凑函数密钥长度的理论下界，更多的细节讨论见附录 F。

注 2：附录 D、F 和 G 给出了对密钥和摘要长度的进一步讨论。

3.6

杂凑函数 hash-function

将任意长比特串映射为定长比特串的函数，满足如下属性：

——给定一个输出比特串，寻找一个输入比特串来产生这个输出比特串，在计算上是不可行地；

——给定一个输入比特串，寻找另一个不同的输入比特串来产生相同的输出比特串，在计算上是不可行地。

3.7

人工鉴别证书 manual authentication certificate

一个密钥和一个校验值的组合，由参与鉴别的两个设备之一产生，并具有下列属性：当被输入到另一个设备时，这个证书可被用于在稍晚时刻完成人工鉴别过程。

3.8

消息鉴别码 message authentication code (MAC)

使用消息鉴别算法产生的输出比特串。

3.9

消息鉴别算法 message authentication code (MAC) algorithm

将一个比特串和一个密钥进行计算，得到定长比特串的算法，满足如下属性：

——对于任意密钥和任何输入比特串，都可以有效计算；

——对于某个具体的密钥，在没有任何关于此密钥先验知识的情况下，计算出任何新输入比特串的消息鉴别码在计算上都是不可行地，即便已知之前所有的输入比特串和对应的消息鉴别码。这

意味着即使在观察到前 i-1 个比特串和对应的消息鉴别码后，蓄意选取第 i 个输入比特串使之与前面某个输入比特串相等，二者的消息鉴别码也不会相等或有任何相关性。

3.10

人工实体鉴别 manual entity authentication

在两个设备之间，通过（潜在地非安全）通信通道进行消息交换，同时也采用人工方式传递有限数据，以此实现实体鉴别的过程。

3.11

简单输入接口 simple input interface

允许用户向设备告知某步骤成功或非成功完成的设备接口，例如2个或1个按钮，在给定时间区间内用户选择按下或不按下，从而告知设备成功或失败。

3.12

简单输出接口 simple output interface

允许设备向用户告知某步骤成功或非成功完成的设备接口，例如可以被实现为红绿指示灯或单独一个指示灯，它通过不同的闪亮方式，向用户通知成功或失败。

4 符号和缩略语

A,B 参与鉴别机制的实体的标签

d 摘要函数，用于机制3和5，$d(D,k)$表示使用密钥k对比特串D计算的摘要值

D 设备A和B之间共享的一个比特串，通过执行人工实体鉴别机制来产生

h 杂凑函数，在机制3-6中使用

I_U 实体U的可区分标识符

K 在机制1和2中，被校验函数使用的（短）密钥

k 机制3-6中使用的(长)密钥

K_A, K_{Ai}, K_B, K_{Bi} 机制7和8中使用的随机MAC密钥

MAC 消息鉴别码

R_U 在机制4、6、7和8中使用（短）随机比特串

‖ 在GB/T 15843.1中， $X‖Y$被定义为数据项X和Y根据给定顺序级联的结果。当两个或多个数据项级联的结果在本部分所描述的某个机制中被作为输入，那么这个级联结果应能被唯一地解析成构成它的数据项，也就是说，它可被无歧义地解释。这个特性可以通过多种方式实现，实现方式与具体应用相关，例如可以用下列方法（a）对每个被级联的数据项要求固定长度，并且在机制执行的全过程都保持它们的固定长度，或者（b）对级联后的数据项序列使用一种可以确保唯一性的方法进行编码，例如使用ISO/IEC 8825-1所定义的可辨识编码规则（DER）。

注：附录D和F给出了如何选取适当的短密钥和MAC密钥长度的指南。

5 通用要求

本章指定鉴别机制1-8应满足的通用要求。除这些通用要求外，各鉴别机制还应分别满足第6、7和8章规定具体要求。

a) 执行人工传递鉴别的两个设备之间应存在通信链接（例如：无线链接或互联网链接），这个链接不必是安全的，也就是说，本部分的机制被设计为在攻击者有能力监控甚至篡改被传递数据的情况下，也能够安全地执行；

b) 执行人工传递鉴别的两个设备应同时具有用户数据输入接口和输出接口；

c) 设备的用户数据输入接口至少应具备指示一个鉴别步骤成功或不成功完成的能力（例如，2 个或 1 个按钮，在给定时间区间内用户选择按下或不按下，从而告知设备成功或失败），这种用户数据输入接口以下称为简单输入接口。相比之下，一个标准的输入接口应支持短符号串输入，例如支持数字、十六进制数或字母的键盘。除非另有明确说明，否则每个设备都应具有一个标准的数据输入接口。

d) 设备的用户数据输出接口至少应具备指示一个鉴别步骤成功或失败的能力（例如可以用红色和绿色灯的方式实现），这种用户数据输出接口以下称为简单输出接口。相比之下，一个标准的输出接口应支持短符号串的输出，如数字、十六进制或字母显示屏。除非另有明确说明，否则每个设备都应具有一个标准的数据输出接口。

e) 对于机制 1 和 2，两个执行实体鉴别的设备应就所使用的具体校验函数达成一致，且有能力实现此函数。

注 1：附录 D 给出了用于机制 1 和机制 2 的校验函数、校验值和随机密钥长度的选择指南。附录 E 给出了用于机制 1 和机制 2 的无条件安全校验函数的构造方法。

f) 对于机制 3-6，两个执行实体鉴别的设备应就所使用的具体杂凑函数 h 达成一致，且有能力实现此函数。

注 3：附录 D 给出了用于机制 3－6 的杂凑函数输入输出位长度的选择指南。

g) 对于机制 3 和 5，两个执行实体鉴别的设备应就所使用的具体摘要函数 d 达成一致，且有能力实现此函数。

注 4：附录 D 给出了用于机制 3 和 5 的摘要长度的选取指南，附录 G 则给出了适用于机制 3 和 5 的使用消息鉴别算法和杂凑函数来构造摘要函数的方法。

h) 对于机制 7 和 8，两个执行实体鉴别的设备应就所使用的具体消息鉴别算法达成一致，且有能力实现此算法。

注 5：附录 D 给出了用于机制 7 和 8 的消息鉴别算法、消息鉴别码和随机密钥长度的选择指南。

i) 在执行机制 1-8 之前，两个设备应交换一个数据串 D（结合机制 3-6 中的杂凑值）。D 可由一个设备产生并发送给另一个设备，或两个设备分别产生一个数据串并通过双向通信链路发送给对方，D 是双方产生的数据串的级联。

j) 执行鉴别的两个设备可由同一个用户控制，也可由两个不同的用户控制，如果是后者则这两个用户之间应该存在可信的通信途径。

k) 设备的用户应全程参与鉴别过程以保证正确处理这些机制。执行期间，设备间的人工数据传递不应存在显著延时，设备应按照规范的定义自动触发超时，以排除特定的攻击。

6 使用短校验值的机制

6.1 概述

本节指定了两种使用校验值的人工鉴别机制，适用于多种不同类型的设备。具体地，

——第一种机制（机制1）适用于一个设备具有简单输入接口，另一个设备具有简单输出接口的情况；

——第二种机制（机制2）适用于两个设备都具有简单输入接口的情况。

标准输入或输出接口可被用来模拟简单输入或输出接口。因此，如果两个设备都具有标准输入和输出接口，那么两种机制都是适用的。

这两种机制都以以下的方式执行：一个数据串*D*通过两个设备共享的信道被从一个设备传递到另一个设备（或是两个设备各自产生的数据串的级联），人工实体鉴别机制随之启动。作为鉴别机制的结果，两个设备都确认自己所掌握的数据串*D*与对方所掌握的相同。

6.2 机制 1：一个设备具有简单输入接口，另一个具有简单输出接口

6.2.1 具体要求

本机制应满足如下具体要求。

a) 本机制适用于一个设备（设备 A）具有简单输入接口，另一个设备（设备 B）具有简单输出接口的情况；

b) 设备 A 应具备产生密钥的能力。

6.2.2 数据交互过程

数据交换和操作的过程如下（见图1）。

a) 两个设备都应输出一个信号，确认接收到了数据串 *D*，且已准备好启动鉴别机制。当观察到两个设备都已准备好，用户应输入一个信号给设备 A，通知它机制可以开始。

b) 设备 A 应产生一个随机密钥 *K*，适用于双方使用的校验函数。使用此密钥 *K*，设备 A 应计算数据串 *D* 的校验值，校验值和密钥 *K* 应随后被设备 A 的输出接口输出，用户应通过其输出接口读取校验值和密钥 *K*。

c) 用户应使用设备 B 的输入接口，将设备 A 输出的校验值和密钥 *K* 输入到设备 B。设备 B 应使用密钥 *K* 针对它所存储的数据串 *D* 重新计算校验值，如果两个校验值一致，则设备 B 应通过简单输出接口输出一个成功信号给用户，否则输出失败信号。

d) 用户应将设备 B 输出的成功或失败的结果，通过设备 A 的简单输入接口输入设备 A。

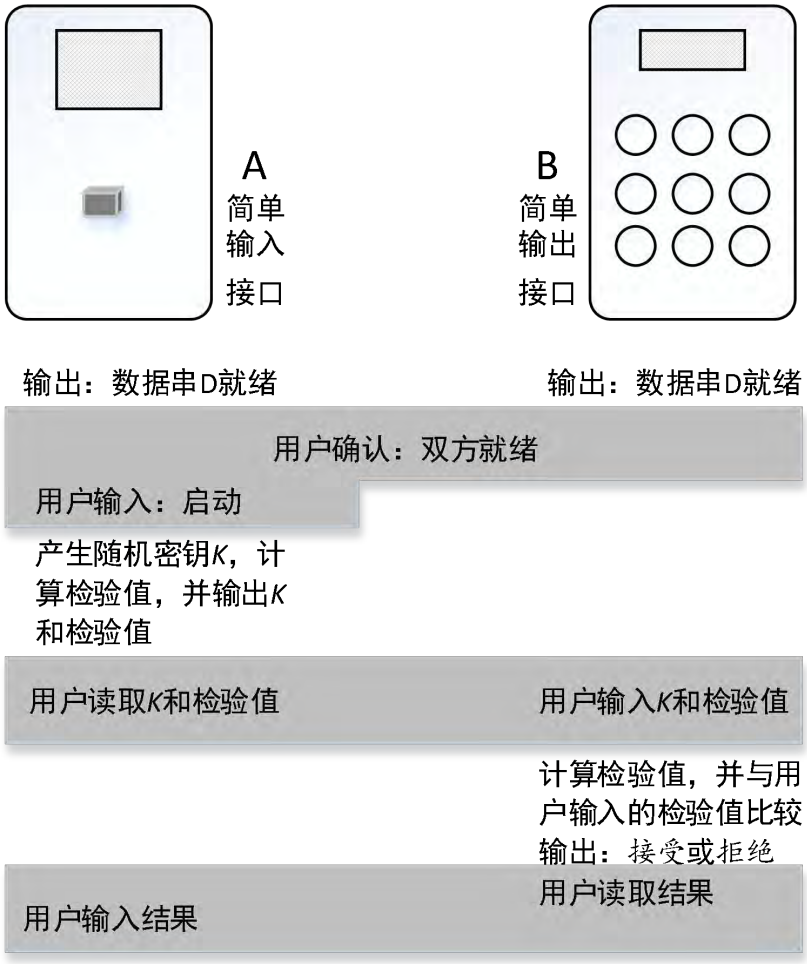

图 1　人工鉴别机制 1

6.2.3　人工鉴别证书

在人工鉴别机制1中，没有任何鉴别信息是通过非安全通道传递的。因此，如果在设备B获得数据串D之前，随机密钥K和校验值就从设备A传递到了设备B，也是不影响机制1的安全性的。随机密钥K和（使用K、D计算的）校验值的组合，被称作人工鉴别证书。利用人工鉴别证书，机制1提供了一种延时鉴别的手段。显然，这种手段只适用于数据串D由设备A产生并发送给设备B的情况。使用人工鉴别证书的鉴别协议如下所述（应满足6.2.1节提出的要求），需注意此协议能够支持数据起源鉴别，但不提供实体身份鉴别能力。

假设设备A产生数据串D，但需稍晚发送给设备B：

a) 设备 A 生成适用于既定校验函数的随机密钥 K，并使用密钥 K 计算数据串 D 的校验值。密钥 K 和校验值随后通过设备 A 的输出接口被输出给用户，并由用户读取。

b) 用户应使用设备 B 的输入接口将设备 A 输出的密钥 K 和校验值输入到设备 B，并由设备 B 本地保存。

c) 一段时间后，当设备 B 接收到来自设备 A 的数据串 *D*，就使用密钥 *K* 重新计算数据串 *D* 的校验值，如果与先前本地存储的校验值一致，则设备 B 接受数据串 *D*，并通过其简单输出接口输出成功信号给用户，否则输出失败信号。

注：数据串*D*可以包含多类数据，例如设备的公钥、身份标识、服务域等。附录B提供了一个例子，表明人工鉴别证书可以用来在两个设备之间建立一个共享密钥。

6.3 机制 2：两个设备都具有简单输入接口

6.3.1 具体要求

本机制应满足如下具体要求。

a) 本节指定的机制适用于两个设备（A 和 B）都具有简单输入接口的情况；

b) 其中一个设备（设备 A）应具有产生密钥的能力。

6.3.2 数据交互过程

数据交换和操作的过程如下（见图2）。

a) 两个设备都应输出一个信号，确认接收到了数据串 *D*，且已准备好启动鉴别机制。当观察到两个设备都已准备好，用户应输入一个信号给设备 A，通知它机制可以开始。

b) 设备 A 产生一个随机密钥 *K*，适用于双方使用的校验函数。使用此密钥 *K*，设备 A 应计算数据串 *D* 的校验值，校验值和密钥 *K* 通过设备 A 的输出接口输出，设备 A 还应通过与设备 B 共享的通信链路将密钥 *K* 传递给设备 B。

c) 设备 B 应使用接收到的密钥 *K* 计算本地存储的数据串 *D* 的校验值，并输出密钥 *K* 和校验值。

d) 用户应比较两个设备输出的校验值和密钥 *K*。如果一致，则用户通过两个设备的简单输入接口向两个设备输入接受信号；如果校验值或密钥值不一致，则鉴别失败，用户应通过两个设备的简单输入接口向两个设备输入拒绝信号。如果两个设备长时间未收到用户的接受信号，则认为鉴别失败（这将需要一个超时机制来实现）。

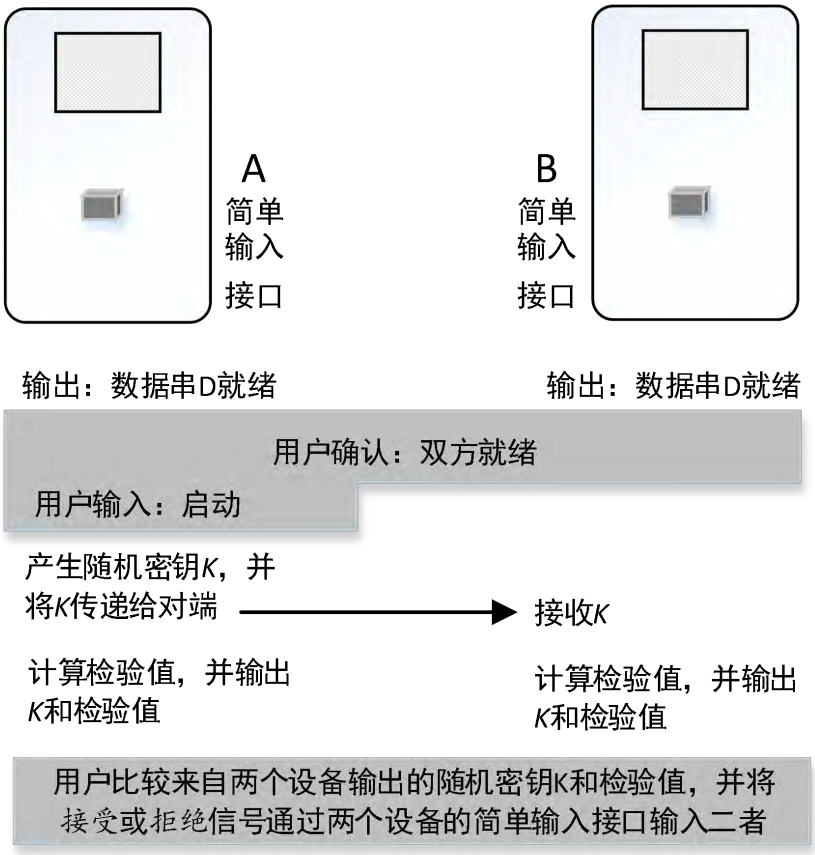

图 2　人工鉴别机制 2

7　使用短摘要值或短密钥的机制

7.1　概述

本节指定了四种人工鉴别机制，涉及短摘要值或短密钥的人工传递。这四种机制适用于不同类型的设备，具体地，

——前两种机制（机制 3 和 4）适用于一个设备具有简单输入接口，另一个设备具有简单输出接口；

——后两种机制（机制 5 和 6）适用于两个设备都具有简单输入接口。

标准输入或输出接口可被用来模拟简单输入或输出接口。因此，如果两个设备都具有标准输入和输出接口，那么四种机制都是适用的。

所有的机制都以以下的方式执行：一个数据串D和一个杂凑值通过双方之间的通信链接被从一个设备传递到另一个（D也可以是两个设备各自产生的数据串的级联），人工实体鉴别机制随之启动。作为鉴别机制的结果，两个设备都确认自己所掌握的数据串D与对方所掌握的相同。

7.2　机制 3：一个设备具有简单输入接口，另一个具有简单输出接口

7.2.1　具体要求

本机制应满足如下具体要求。

a) 本机制适用于一个设备（设备 A）具有简单输入接口，另一个设备（设备 B）具有简单输出接口的情况；

b) 设备 A 应具备产生（长）随机密钥的能力。

7.2.2 数据交互过程

数据交换和操作的过程如下（见图3）。需注意步骤a）和b）可能并行发生，步骤d）和e）也是。

a) 设备 A 和设备 B 应暂时共享数据串 D，例如可通过在双方之间的通信链路上执行未受保护的消息交换来实现。

b) 设备应产生并安全保存随机密钥 k，k 应适用于双方使用的摘要函数 d。设备 A 应计算 $h(k)$并将其传递给设备 B，传递方式不必是安全的，例如可通过双方之间的通信链路传递。

c) 两个设备都应通过他们的标准输出接口输出信号来确认它们已经完成步骤 a）和 b），而且它们已经准备好启动鉴别机制。观察到双方输出的确认信号后，用户应通过设备 A 的简单输入接口输入一个信号，告知设备 A 鉴别机制可以开始。

d) 设备 A 应计算短摘要值 $d(D,k)$，并将其通过自身的标准输出接口输出。用户应从设备 A 的标准输出接口读取短摘要值，并使用设备 B 的标准输入接口将其输入设备 B。

e) 设备 A 应通过双方之间的通信链路将密钥 k 传递给设备 B。接收到 k 后，设备 B 应计算 $h(k)$并检验是否与步骤 b）中接收到的值相等。如果相等，则设备 B 执行步骤 f）；否则设备 B 应给出失败信号，且应执行一种策略以确保短时间内拒绝开始新的机制 3 实例。

f) 设备 B 应使用密钥 k 及其存储的数据串 D 重新计算短摘要值 $d(D,k)$。如果计算出的摘要值与设备 B 在步骤 d）中接收的副本相等，则设备 B 通过它的简单输出接口向用户输出成功信号；否则应输出失败信号，并执行一种策略以确保短时间内拒绝开始新的机制 3 实例。

g) 用户应将设备 B 输出的成功或失败的结果，通过设备 A 的简单输入接口输入设备 A。如果设备 A 没有收到任何信号，则应视为失败（这需要实施一种超时机制）。

注 1：步骤 e)和 f)中用到的延时策略可以防止这样一类中间人攻击：攻击者尝试在设备 B 给出失败信号后假冒设备 A 的身份立即再次启动机制 3，此时由于设备 A 仍在等待 g)步骤中用户要输入的信号，所以有可能被成功攻击。

注 2：在本机制中，设备 B 信任设备 A，因为是设备 A 产生的随机密钥 k 并因此首先决定了 $d(D,k)$的正确值。如果设备 B 也具备产生随机密钥的能力，则设备 A 和设备 B 就不必一定相互信任。但如果这样，可能会使本机制变得更为复杂，因为需要更多的网络通信和同步开销。机制 4-6 也存在这种情况。

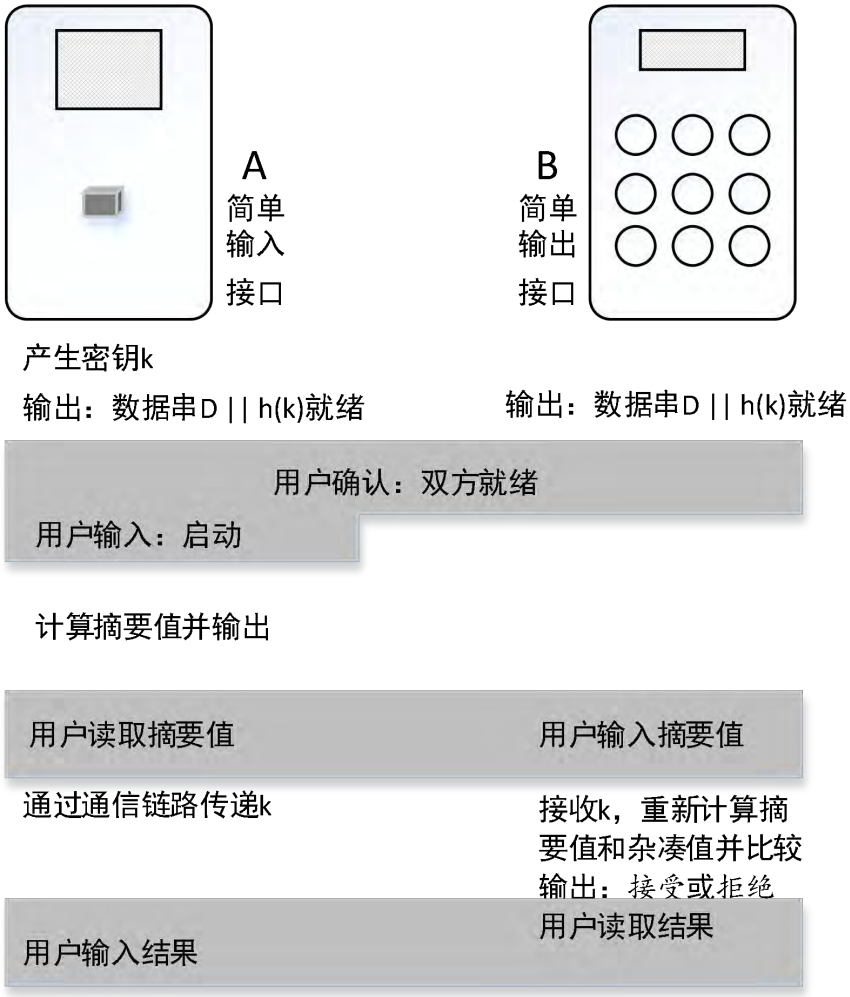

图 3　人工鉴别机制 3

7.3　机制 4：一个设备具有简单输入接口，另一个具有简单输出接口

7.3.1　具体要求

本机制应满足如下具体要求。

a)　本机制适用于一个设备（设备 A）具有简单输入接口，另一个设备（设备 B）具有简单输出接口的情况；

b)　设备 A 应具有产生（长）随机密钥和短随机比特流的能力。

7.3.2　数据交换规范

数据交换和操作的过程如下（见图4）。需注意步骤a）和b）可能并行发生，步骤d）和e）也是。

a)　设备 A 和设备 B 应暂时共享数据串 D，例如可通过在双方之间的通信链路上执行未受保护的消息交换来实现。

b)　设备 A 应产生并安全保存（长）随机密钥 k 和（短）随机比特流 R。设备 A 应计算 $h(D\|k\|R)$并传递给设备 B，传递方式不必是安全的，例如可通过双方之间的通信链路传递。

c) 两个设备应通过它们的标准输出接口输出信号，指示它们已经完成了步骤 a）-b），且已准备好启动鉴别机制。观察到这个信号后，用户应通过设备 A 的简单输入接口向设备 A 输入信号，告知设备 A 鉴别机制可以开始。

d) 设备 A 应通过它的标准输出接口输出短随机比特流 R，并由用户读取。用户随后应使用设备 B 的标准输入接口将 R 输入到设备 B。

e) 设备 A 应通过双方之间的通信链路将密钥 k 传递给设备 B。

f) 在设备 B 经过步骤 d)和 e)接收到短随机比特流 R 和密钥 k 后，应使用它们和本地存储的数据串 D 重新计算 $h(D\|k\|R)$。如果杂凑值与 b)步骤中接受自设备 A 的杂凑值相等，则设备 B 应通过自己的简单输出接口向用户输出成功信号，否则输出失败信号。

g) 用户应通过设备 A 的简单输入接口向设备 A 输入鉴别结果，即 f)步骤中由设备 B 获得的成功或失败信号。如果设备 A 没有收到任何信号，则应视为失败（这需要实施一种超时机制）。

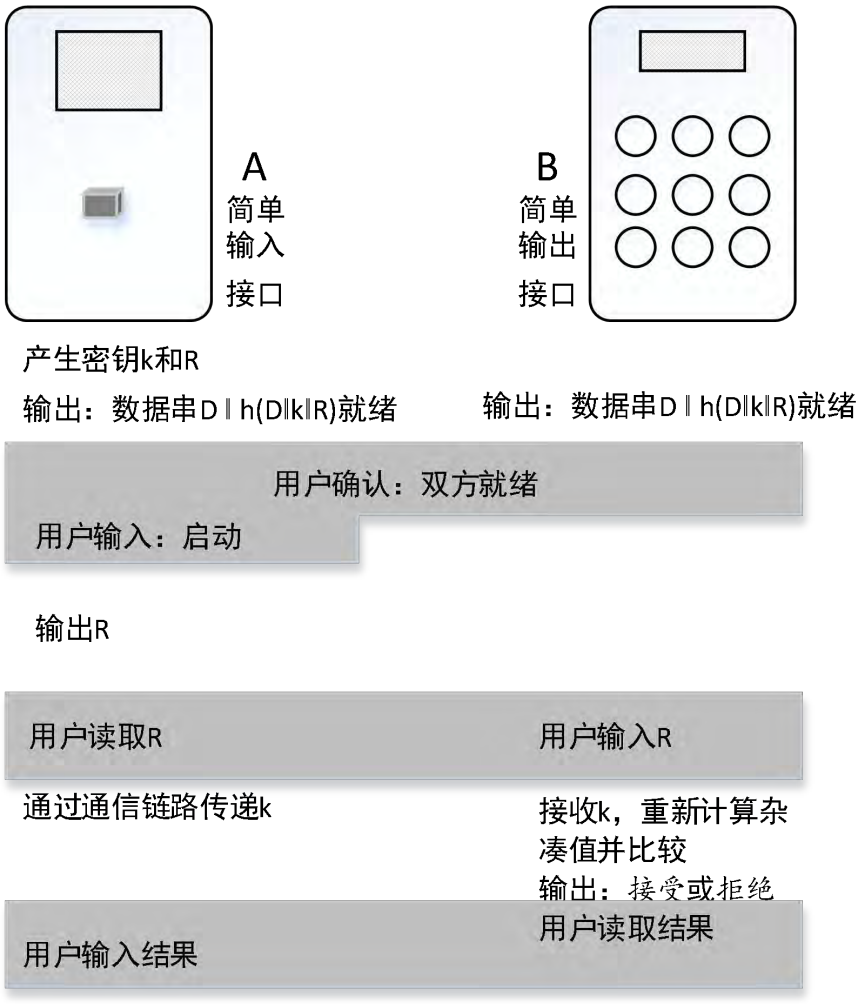

图 4　人工鉴别机制 4

7.4　机制 5：两个设备都具有简单输入接口

7.4.1　具体要求

本机制应满足如下具体要求。

a) 本节指定的机制适用于两个设备（A 和 B）都具有简单输入接口的情况；

b) 其中一个设备（设备 A）应具备产生（长）随机密钥的能力。

7.4.2 数据交换规范

数据交换和操作的过程如下（见图5），注意：步骤a）和b）可能并行发生，步骤d）和e）也是。

a) 设备 A 和设备 B 应暂时共享数据串 D，例如可通过在双方之间的通信链路上执行未受保护的消息交换来实现。

b) 设备 A 应产生并安全保存随机密钥 k，k 应适用于双方使用的摘要函数 d。设备 A 应计算 $h(k)$ 并将其传递给设备 B，传递方式不必是安全的，例如可通过双方之间的通信链路传递。

c) 两个设备应通过它们的标准输出接口输出信号，指示它们已经完成了步骤 a）-b），且已准备好启动鉴别机制。观察到这个信号后，用户应通过设备 A 的简单输入接口向设备 A 输入信号，告知设备 A 鉴别机制可以开始。

d) 设备 A 应通过双方之间的通信链路将密钥 k 传递给设备 B。

e) 设备 A 应计算短摘要值 $d(D,k)$，并通过其标准输出接口输出。

f) 在设备 B 经过步骤 d)接收到密钥 k 后，应重新计算杂凑值 $h(k)$，并使用其本地存储的数据串 D 重新计算短摘要值 $d(D,k)$。如果算得的杂凑值与步骤 b)中接受自设备 A 的杂凑值相等，则设备 B 应通过其标准输出接口将短摘要值输出；否则设备 B 应给出失败信号，且应执行一种策略以确保短时间内拒绝开始新的机制 5 实例。

g) 用户应将设备 A 和设备 B 在 e)-f)步骤中输出的两个短摘要值进行比较，如果相等则用户应通过两个设备的简单输入接口向它们输入接受信号，否则鉴别失败，用户应输入拒绝信号。如果两个设备长时间未收到用户的接受信号，则认为鉴别失败（这将需要一个超时机制来实现）。

注：步骤f)中用到的延时策略可以防止这样一类中间人攻击：攻击者尝试在设备B给出失败信号后假冒设备A的身份立即再次启动机制5，此时由于设备A仍在等待g)步骤中用户要输入的信号，所以有可能被成功攻击。

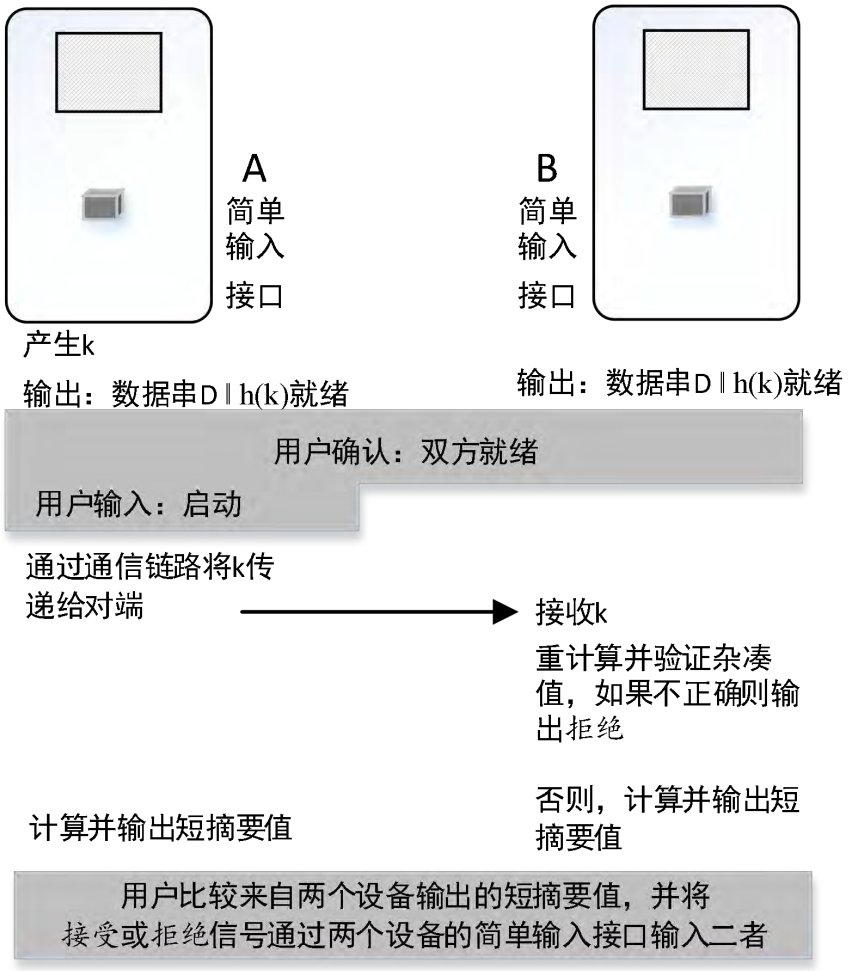

图 5　人工鉴别机制 5

7.5　机制 6：两个设备都具有简单输入接口

7.5.1　具体要求

本机制应满足如下具体要求。

a)　本节指定的机制适用于两个设备（A 和 B）都具有简单输入接口的情况；

b)　其中一个设备（设备 A）应具备产生（长）随机密钥和短随机比特流的能力。

7.5.2　数据交换规范

数据交换和操作的过程如下（见图6），注意：步骤a）和b）可能并行发生，步骤d）和e）也是。

a)　设备 A 和设备 B 应暂时共享数据串 D，例如可通过在双方之间的通信链路上执行未受保护的消息交换来实现。

b)　设备 A 应产生并安全保存（长）随机密钥 k 和（短）随机比特流 R，并计算 $h(D\|k\|R)$，然后将此杂凑值传递给设备 B。传递方式不必是安全的，例如可通过双方之间的通信链路传递。

c) 两个设备应通过它们的标准输出接口输出信号，指示它们已经完成了步骤 a）-b），且已准备好启动鉴别机制。观察到这个信号后，用户应通过设备 A 的简单输入接口向设备 A 输入信号，告知设备 A 鉴别机制可以开始。

d) 设备 A 应通过双方之间的通信链路将（长）密钥 k 和（短）随机比特流 R 传递给设备 B。

e) 设备 A 应通过其标准输出接口输出短随机比特流 R。

f) 在设备 B 经过步骤 d)接收到 k 和 R 后，应使用其本地存储的数据串 D 重新计算杂凑值 $h(D||k||R)$。如果算得的杂凑值与步骤 b)中接受自设备 A 的杂凑值相等，则设备 B 应通过其标准输出接口将短随机比特流 R 输出；否则设备 B 应给出失败信号，且应执行一种策略以确保短时间内拒绝开始新的机制 6 实例。

g) 用户应将设备 A 和设备 B 在 e)-f)步骤中输出的两个短随机比特流进行比较，如果相等则用户应通过两个设备的简单输入接口向它们输入接受信号，否则鉴别失败，用户应输入拒绝信号。如果两个设备长时间未收到用户的接受信号，则认为鉴别失败（这将需要一个超时机制来实现）。

注：步骤f)中用到的延时策略可以防止这样一类中间人攻击：攻击者尝试在设备B给出失败信号后假冒设备A的身份立即再次启动机制6，此时由于设备A仍在等待g)步骤中用户要输入的信号，所以有可能被成功攻击。

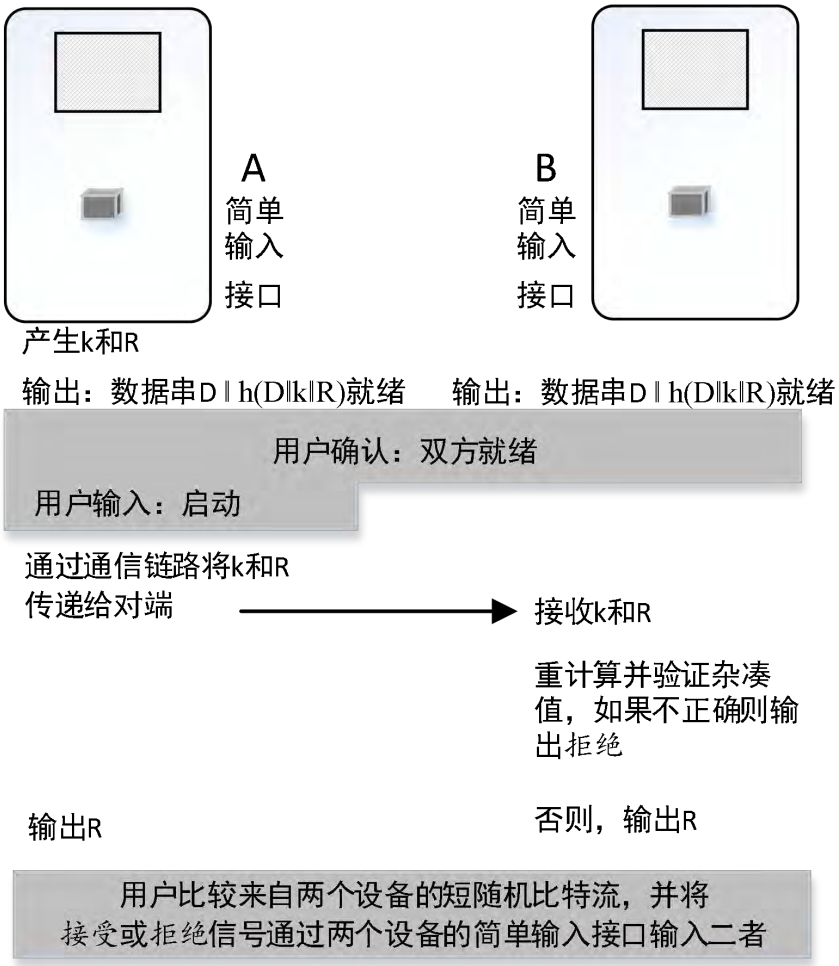

图 6　人工鉴别机制 6

8　使用消息鉴别码（MAC）的机制

8.1　概述

本节指定了两类基于消息鉴别码的人工鉴别机制， 适用于多种不同类型的设备。具体地，

——第一种机制（机制7）适用于两个设备都具有简单输出接口；

——第二种机制（机制8）适用于一个设备具有简单输入接口，另一个设备具有简单输出接口的情况。

标准输入或输出接口可被用来模拟简单输入或输出接口。因此，如果两个设备都具有标准输入和输出接口，那么两种机制都是适用的。

这两种机制都以以下的方式执行：一个数据串D通过两个设备共享的信道被从一个设备传递到另一个设备（或是两个设备各自产生的数据串的级联），人工实体鉴别机制随之启动。作为鉴别机制的结果，两个设备都确认自己所掌握的数据串D与对方所掌握的相同。

8.2　机制 7：两个设备都具有简单输出接口

8.2.1　概述

本机制有两个变种（7a和7b），8.2.3节定义的机制7a要求两个设备之间做少量交互，而8.2.4节定义的机制7b则要求少量由用户执行的人工交互。

8.2.2　具体要求

本机制应满足如下具体要求。

a)　本机制的两个变种适用于两个设备（A 和 B）都具有简单输出接口的情况；

b)　两个设备应具备产生随机 MAC 密钥的能力，用户应具有产生短随机比特串的能力；

c)　两个设备在启动机制之前就知道彼此的身份。

注：如果用户不严格地去随机选择比特串，例如总是使用一个相同的值，则鉴别机制遭受攻击的风险大大提高。

8.2.3　机制 7a 的数据交互过程

数据交换和操作的过程如下（见图7），注意：步骤b）和c）可能并行发生，步骤的d）和e）、f）和g）也是如此。

a)　两个设备应通过各自的简单输出接口输出一个信号，以表明他们已经收到数据串 D 且已经准备好启动鉴别机制。当观察到两个设备都已准备好，用户应生成短随机比特串 R，并将 R 输入到两个设备，随后向设备 A 输入信号告知机制 7a 可以开始。

b)　设备 A 应生成随机密钥 K_A，适用于双方商定的消息鉴别函数。设备 A 应使用 K_A 对 I_A（设备 A 的身份标识）、数据串 D 和随机比特串 R 级联而成的比特串计算一个 MAC，记为 MAC_A，并将 MAC_A 通过与设备 B 之间的通信链路传递给设备 B。

c)　设备 B 应生成随机密钥 K_B，适用于双方商定的消息鉴别函数。设备 B 应使用 K_B 对 I_B（设备 B 的身份标识）、数据串 D 和随机比特串 R 级联而成的比特串计算一个 MAC，记为 MAC_B，并将 MAC_B 通过与设备 A 之间的通信链路传递给设备 A。

d)　设备 A 收到 MAC_B 后，应发送 K_A 给设备 B。

e)　设备 B 收到 K_A 后，使用 K_A 对自己存储的 R、D、I_A 计算 MAC 值，并验证是否与收到的 MAC_A 相同，如果相同，设备 B 输出成功指示。

f) 设备 B 收到 MAC_A 后，应发送 K_B 给设备 A。

g) 设备 A 收到 K_B 后，使用 K_B 对自己存储的 R、D、I_B 值计算 MAC 值，并验证是否与收到的 MAC_B 相同，如果相同，设备 A 输出成功指示。

h) 用户应确认两个设备是否都输出了成功指示，如果都成功，则用户应向两个设备输入成功确认信号；如果两个设备中的一个或两个输出了失败指示，则用户应输入失败信号给两个设备。如果用户在指定的时间内没有向设备输入成功信号，则应被设备解释为失败。

注：在此机制中，步骤g）用于防止替换攻击，即攻击者试图伪装成设备A来欺骗设备B。

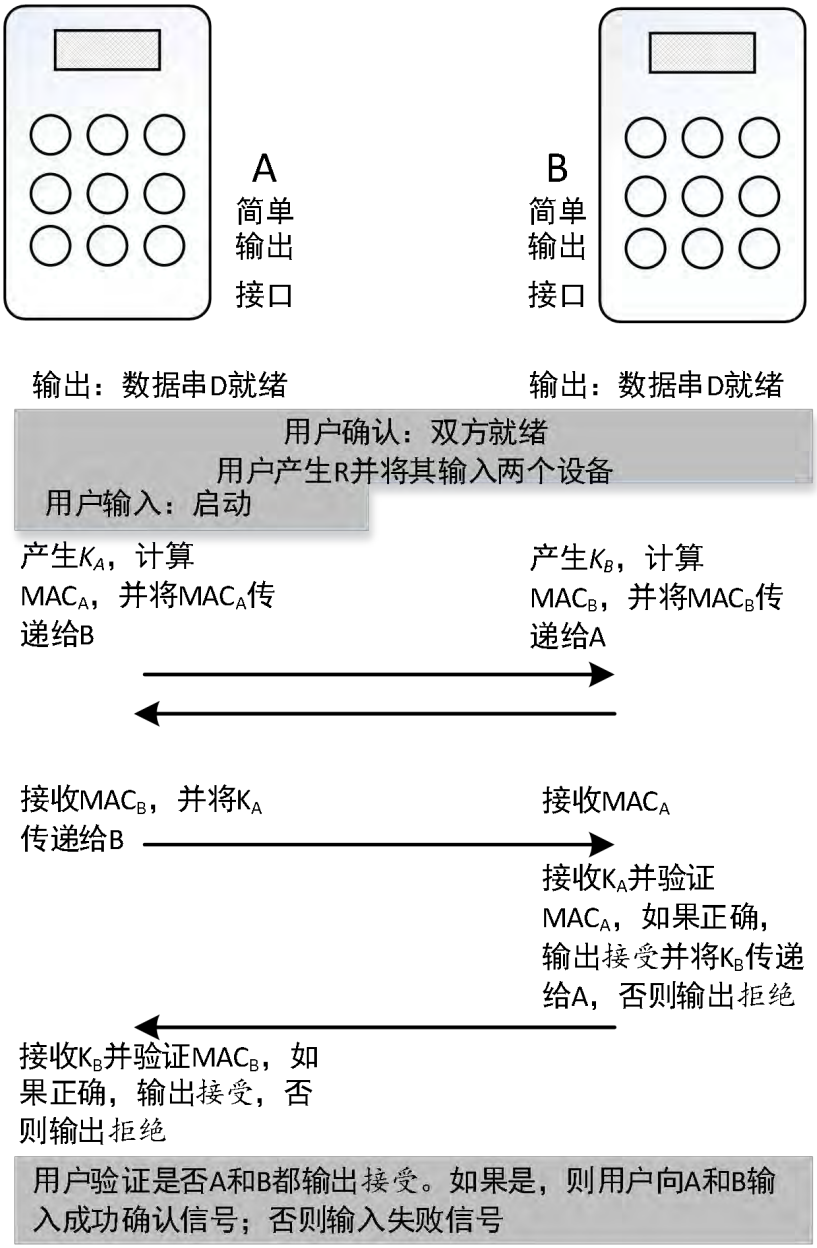

图 7　人工鉴别机制 7a

8.2.4 机制 7b 的数据交互过程

数据交换和操作的过程如下（见图8）。

a) 两个设备应通过各自的简单输出接口输出一个信号，以表明他们已经收到数据串 D 且已经准备好启动鉴别机制。当观察到两个设备都已准备好，用户应生成短随机比特串 $R=(r_1,r_2,\ldots,r_n)$，其中 r_i 是一个比特，n 是 R 中的比特数量。用户应将 R 输入到两个设备，随后向设备 A 输入信号告知机制 7b 可以开始。

b) 对于 $i=1,2,\ldots n$，下列 1)-5)步操作应被执行，其中步骤 1)和 2)可以被并行执行。

1) 设备 A 应生成随机密钥 K_{Ai}，适用于双方商定的消息鉴别函数。设备 A 应使用 K_{Ai} 对 I_A（设备 A 的身份标识）、数据串 D 和随机比特 r_i 级联而成的比特串计算一个 MAC，记为 MAC_{Ai}，并将 MAC_{Ai} 通过与设备 B 之间的通信链路传递给设备 B。

2) 设备 B 应生成随机密钥 K_{Bi}，适用于双方商定的消息鉴别函数。设备 B 应使用 K_{Bi} 对 I_B（设备 B 的身份标识）、数据串 D 和随机比特串 r_i 级联而成的比特串计算一个 MAC，记为 MAC_{Bi}，并将 MAC_{Bi} 通过与设备 A 之间的通信链路传递给设备 A。

3) 设备 A 收到 MAC_{Bi} 后，应发送 K_{Ai} 给设备 B。

4) 设备 B 收到 MAC_{Ai} 和 K_{Ai} 后，使用 K_{Ai} 对自己存储的 r_i、D、I_A 计算 MAC 值，并验证是否与收到的 MAC_{Ai} 相同，如果相同，设备 B 将 K_{Bi} 传递给设备 A，否则终止鉴别协议。

5) 设备 A 收到 K_{Bi} 后，使用 K_{Bi} 对自己存储的 r_i、D、I_B 值计算 MAC 值，并验证是否与收到的 MAC_{Bi} 相同，如果不相同，则设备 A 终止鉴别协议。

注：在$i=n$时，如果步骤4）和5）的验证结果都是相同，则设备B和设备A可以各自输出一个成功标志。虽然“输出成功标志”在本协议中未做要求，但可以选择这样做，以便起到“告知用户鉴别过程成功完成”的作用。

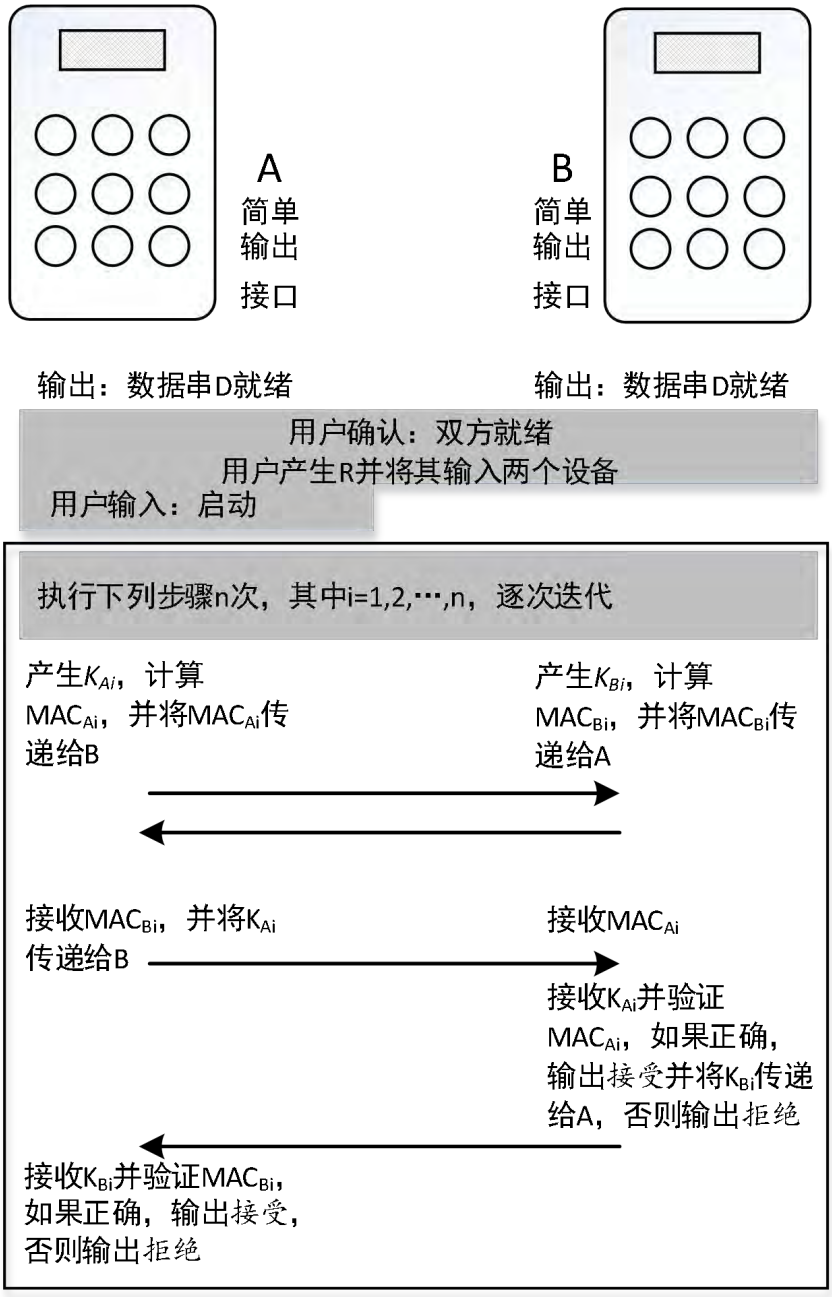

图 8　人工鉴别机制 7b

8.3　机制 8：一个设备具有简单输入接口，另一个具有简单输出接口

8.3.1　概述

本机制有两个变种（8a和8b），8.3.3节定义的机制8a要求两个设备之间做少量交互，而8.3.4节定义的机制8b则要求少量由用户执行的人工用户交互。

8.3.2　具体要求

本机制应满足如下具体要求。

a) 本机制的两个变种适用于一个设备（设备 A）具有简单输入接口，另一个设备（设备 B）具有简单输出接口的情况；

b) 两个设备都应具有产生随机 MAC 密钥的能力；

c) 两个设备在启动机制之前就应知道彼此的身份。

8.3.3 机制 8a 的数据交互过程

数据交换和操作的过程与机制7a相似（如8.2.3节所述），但有以下不同：

——步骤 a)中设备 A 产生随机比特串并显示给用户，用户将其复制到设备 B，因此本机制中用户无需产生随机比特串。

机制 8a 如图 9 所示。

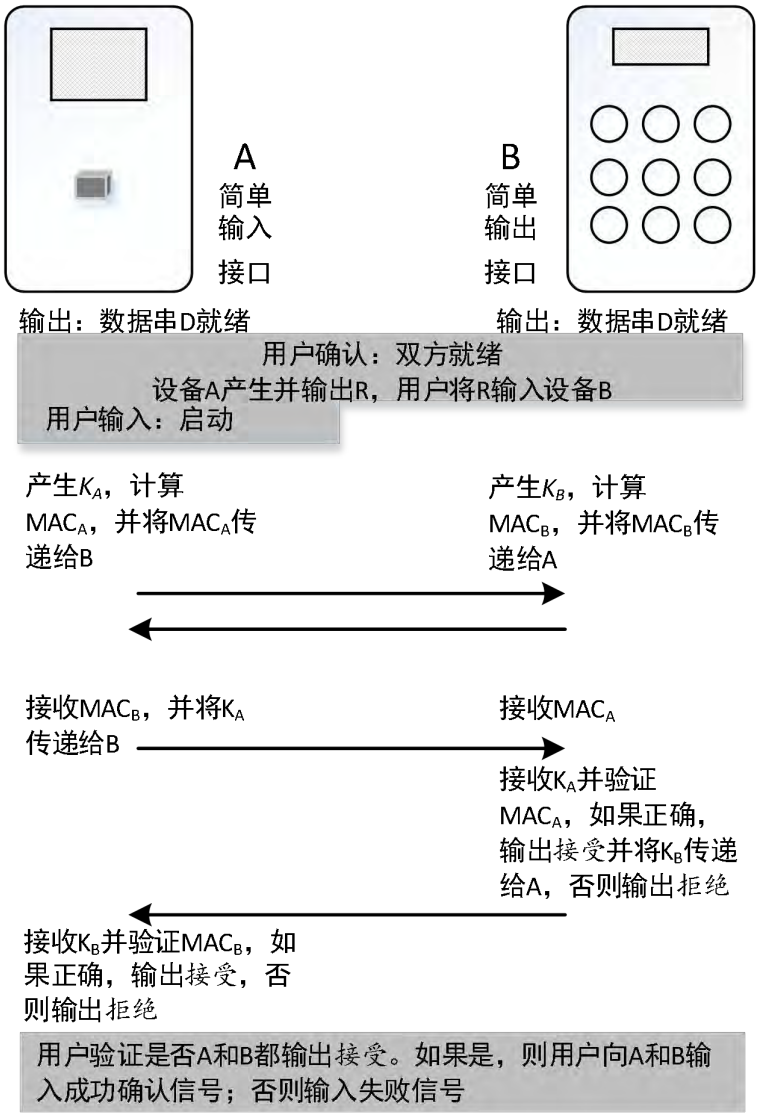

图 9 人工鉴别机制 8a

8.3.4 机制 8b 的数据交互过程

数据交换和操作与机制7b类似（如8.2.4节所述），但有如下不同：

——步骤 a)中设备 A 产生随机比特串并显示给用户，用户将其复制到设备 B，因此本机制中用户无需产生随机比特串。

附 录 A
（规范性附录）
ASN.1 定义

```
EnityAuthenticationMechanisms-6 {

iso(1) standard(0) e-auth-mechanisms(9798) part6(6)
asn1-module(0) object-identifiers(0) }
    DEFINITIONS EXPLICIT TAGS ::= BEGIN
-- EXPORTS All; --
-- IMPORTS None; --
OID ::= OBJECT IDENTIFIER -- alias
-- Synonyms --
is9798-6 OID ::= { iso(1) standard(0) e-auth-mechanisms(9798) part6(6) }
mechanism OID ::= { is9798-6 mechanisms(1) }
-- Mechanisms using manual transfer of a short key and a short check-value --
mdt-kc-siso OID ::= {mechanism mdt-kc-siso(1)}
mdt-kc-sisi OID ::= {mechanism mdt-kc-sisi(2)}

-- Mechanisms using manual transfer of a short digest-value or a short key --
mdt-c-siso-one OID ::= {mechanism mdt-c-sisoone(3)}
mdt-c-siso-two OID ::= {mechanism mdt-c-sisotwo(4)}
mdt-c-sisi-one OID ::= {mechanism mdt-c-sisione(5)}
mdt-c-sisi-two OID ::= {mechanism mdt-c-sisitwo(6)}
-- Mechanisms using a MAC --
mac-k-soso OID ::= {mechanism mac-k-soso(7)}
mac-k-siso OID ::= {mechanism mac-k-siso(8)}
END -- EntityAuthenticationMechanisms-6 --
```

附 录 B
（资料性附录）
使用人工鉴别协议来执行密钥交换

B.1 概述

本节描述使用本部分中的人工鉴别机制来执行密钥交换的方法，以实现两个通信实体共享一个秘密密钥。

B.2 经鉴别的Diffie-Hellman密钥协商

本节讨论的是Diffie-Hellman密钥协商机制，遵循ISO/IEC 11770-3中的密钥协商机制4（参见ISO/IEC 11770-3的附录B.5），本附录在其中加入了人工鉴别的环节，来对其进行鉴别。以下是对这种机制的扼要描述，全面而详尽的描述见ISO/IEC 11770-3。

Diffie-Hellman密钥协商机制涉及一个一般群G，以及G中的一个元素g，g有足够大的阶。密钥协商的步骤如下：

a) 设备 A 秘密产生随机整数 x，计算 g^x 并将其传递给设备 B；
b) 设备 B 秘密产生随机整数 y，计算 g^y 并将其传递给设备 A；
c) 设备 A 和设备 B 针对数据串 $D=(g^x \| g^y \| text)$执行本标准所定义的任一种人工鉴别机制，其中“*text*”是两个设备期望取得一致的任意数据，例如彼此的身份标识；
d) 如果人工鉴别的结果是成功，则双方可以计算共享 Diffie-Hellman 密钥 $S=g^{xy}$。

如果将S作为双方的共享秘密使用，两个设备也可以从S中导出所需长度和格式的各类密钥。

B.3 使用人工鉴别证书的经鉴别的Diffie-Hellman密钥协商

B.3.1 概述

本节讨论的是使用人工鉴别证书来执行鉴别的Diffie-Hellman密钥协商机制。由于人工鉴别证书是人工鉴别机制1支持的一种特性，所以6.2.1节提出的具体要求在这里要得到满足。设备A使用随机密钥K以及使用K对D计算的校验值对设备B执行鉴别。需要注意的是，本机制要求两个设备商定并实施一个对称加密机制e，其中$e_L(M)$表示对数据M使用密钥L进行加密。对称加密技术在ISO/IEC 18033-3和ISO/IEC 18033-4中做了标准化。

Diffie-Hellman密钥协商机制涉及一个一般群G，以及G中的一个元素g，g有足够大的阶。密钥协商步骤分为两个阶段，阶段1和阶段2。

在阶段1，设备A产生自己的Diffie-Hellman私钥，计算相应的公钥，针对包含此公钥的数据串D产生一个人工鉴别证书，并将证书传递给设备B。在阶段2，设备B接收设备A的公钥并做验证，之后产生自己的Diffie-Hellman私钥和公钥。而后，两个设备计算共享的Diffie-Hellman秘密，并从中导出加密密钥。最后，作为人工鉴别过程的一部分，设备A验证密钥K被共享的Diffie-Hellman秘密加密后的密文。上述操作的结果是双方共享的Diffie-Hellman秘密得到鉴别。

B.3.2 阶段1

a) 设备 A 秘密产生随机整数 x，并计算 g^x，数据串 D 由 g^x 和其它需被可靠传递给设备 B 的数据组成。设备 A 随后创建一个人工鉴别证书，其中的校验值针对 D 计算。人工鉴别证书（K,校验值）被人工传递给设备 B，并由设备 B 保存。设备 A 保存 x 和 g^x，以及 D 所包含的其它数据项。

B.3.3 阶段2（稍晚时刻由任一设备发起）

b) 设备 A 将数据串 D，即 g^x 和其它数据项，通过双方共享的通信链接发送给设备 B。设备 B 基于自己保存的人工鉴别证书验证 g^x 的真实性。

c) 设备 B 秘密产生随机整数 y，并计算 g^y。设备 B 计算双方的 Diffie-Hellman 共享秘密 $S=(g^x)^y$，并使用 S 来加密密钥 K，即包含在人工鉴别证书中的密钥 K。设备 B 将使用 S 加密的 K 密文 $e_S(K)$和它的 Diffie-Hellman 公钥 g^y 一并发送给设备 A。

d) 设备 A 计算自己的共享秘密 $S=(g^y)^x$，然后解密 $e_S(K)$，从而验证 K 的正确性。如果验证通过，则设备 A 认可 S。

设备A和B随后可以从S之中各自导出所需长度和格式的密钥。

注：上述描述中，步骤c)和d)用到了人工鉴别证书中的密钥K，其目的是让设备A验证设备B计算的S与自己计算的版本是相同的。这通过设备B使用S作为密钥来加密K，并传递给设备A供其验证来实现。事实上，也可以使用其它双方已经共享的值来代替K来完成这种验证，例如使用校验值或阶段1中被通信双方协商一致的其它数据。

B.4 两个以上组件

本节描述使用人工鉴别技术来实现两个以上设备共享一个通用密钥的方法。

a) 一个设备扮演“主设备”的角色。

b) 主设备与每个其它设备之间执行附录 B.2 中描述的机制，从而与每个其它设备建立一个共享的加密密钥。

c) 之后，任选一个设备产生通用密钥，并将这个通用密钥以主设备为桥梁传递给所有设备。由于主设备与任一设备之间都已经有了共享的加密密钥，因此通用密钥传播的过程是经加密并做完整性保护的。

如果设备的数量为n，主设备需执行n次在群G中的幂运算，而任一其它设备均需执行2次在群G中的幂运算计算。因此建议选择主设备的时候，应选取具备足够计算能力的。

附 录 C
（资料性附录）
使用人工鉴别协议来交换公钥

C.1 概述

本附录描述了使用本部分定义的人工鉴别机制执行可靠公钥交换的方法。公钥交换发生在CA和客户端之间，CA需要将它的公钥可靠地传递给客户端，客户端也需要将自己的公钥可靠传递给CA。

这里描述了两类不同的情况，区别在于客户端私钥由谁产生。一种情况是客户端自己产生私钥，另一种是由密钥管理设施产生私钥并导入到客户端。

C.2 需求

CA应配备标准输出接口，例如显示器；还应配备简单输入接口，用于接收命令。客户端应拥有标准输入接口和简单输出接口，例如音频输出，以便指示鉴别成功或失败。

注：事实上，如果CA和客户端拥有不同于上述的接口，本附录描述的步骤也可被使用，只需更换合适的手工鉴别协议即可。

客户端和CA需有共享的通信信道。

C.3 私钥在设备内产生的情况

操作步骤如下。

a) CA 应可靠地获知客户端的真实身份，例如可由用户通过 CA 的输入接口键入客户端身份标识。但通过人工鉴别协议的执行，也可确保 CA 可靠获知客户端真实身份（见下列描述）。

b) CA 将自己的公钥 P_{CA} 传递给客户端，客户端也传递自己的公钥 P_M 给 CA。传递是经由双方共享的（非可信）通信信道实施的。除 P_M 外，客户端还可将希望包含在公钥证书之中的其它信息，例如客户端的身份标识，传递给 CA,以便 CA 在产生公钥证书时正确处理。

c) CA 和客户端现在执行 6.2 节描述的人工鉴别机制，来验证被交换的公钥是正确的。CA 扮演设备 A 的角色，客户端扮演设备 B，数据串 D 由 P_{CA}、P_M 和其它由客户端和 CA 提供的数据组成，例如双方的唯一性身份标识符。

d) 当（且仅当）客户端（设备 B）给出成功指示，用户才命令 CA（设备 A）产生相应的公钥证书，公钥证书随后可以通过双方之间的通信信道被传递给客户端，传递过程不必受到保护。

e) 客户端（设备 B）在认可收到的证书前执行两个检验。第一，客户端使用 CA 的公钥 P_{CA} 验证证书上的 CA 签名；第二，客户端验证证书的各数据域（包括公钥 P_M 和客户端身份标识）都是期望的值。验证通过后，整个过程结束。

C.4 私钥在设备外产生的情况

如果客户端的私钥是由CA或其它的可信密钥产生设备产生，私钥则应被安全地传递给客户端。

从CA向客户端传递私钥的步骤如下所述。

a) CA 和客户端使用附录 B 描述的密钥协商机制建立共享密钥。

b) 使用步骤 a)建立的共享密钥，CA 将客户端的私钥做加密和完整性保护，并传递给客户端，客户端安全地保存私钥。CA 还应将自己的公钥 P_{CA} 做完整性保护后传递给客户端，也是使用步骤 a)建立的共享密钥来实现。相关的对称加密技术在 ISO/IEC 18033-3 和 ISO/IEC 18033-4 中被标准化。

c) 客户端使用步骤 a)建立的共享密钥，将希望包含在公钥证书中的其它信息做完整性保护后发送给 CA。

d) CA 产生客户端的公钥证书，随后通过双方之间的通信信道传递给客户端，传递过程不必受到保护。

e) 客户端在认可收到的证书前执行两个检验。第一，客户端使用 CA 的公钥 P_{CA} 验证证书上的 CA 签名；第二，客户端验证证书的各数据域（包括公钥 P_M 和客户端身份标识）都是期望的值。验证通过后，整个过程结束。

附 录 D
（资料性附录）
机制安全性和参数长度选择

D.1 概述

本附录针对8个人工鉴别的安全性进行探讨，并对校验值、摘要值、消息鉴别码、随机比特串以及密钥的长度选择提供指导性建议。

D.2 机制 1 和机制 2 的使用

所有在两个设备之间通信信道上传递的数据都是公开的，即便在有些情况下数据串D需保密。这两个人工鉴别机制的安全目标是保护数据的完整性，而非机密性。所需的完整性保护是使用基于校验值的检验步骤来实现的。

校验函数是一个映射f，将D所在的数据空间和K所在的密钥空间映射到一个校验值空间C：

$$f : D \times K \rightarrow C,\ c = f(d,k)$$

在机制1和机制2中，校验值用于保护数据的完整性。因此，这两个机制的安全性是基于校验函数的无条件安全，而非计算安全性。校验函数的无条件安全基于消息鉴别理论的结果，示例可见参考文献[28]的4.5节。一般会考虑两类主要的攻击：

——假冒攻击

——替换攻击

在假冒攻击中，攻击者在接收方未观察到发、收双方此前交换的数据时，尝试让接收者相信数据来自合法的发送方。在替换攻击中，攻击者首先观察到某个数据d，然后使用$\hat{d} \neq d$来替换d。攻击者成功执行假冒攻击和替换攻击的可能性用P_I和P_S分别表示，可以表达为：

$$P_I \triangleq \max_{c \in C, d \in D} P_k(c = f(d,k)),$$

$$P_S \triangleq \max_{\substack{c,\hat{c} \in C \\ d,\hat{d} \in D, d \neq \hat{d}}} P_k(\hat{c} = f(\hat{d},k) \mid c = f(d,k))$$

这两个机制的安全性依赖于攻击者使用数据$\hat{d} \neq d$来成功替换数据d的可能性。如果鉴别双方接受了$\hat{d}$并视为有效数据，则攻击成功了。鉴于我们假定两个设备在物理上彼此接近，且除非两个设备都指示他们已就绪，否则我们不会接受任何数据，因而假冒攻击并不适用于机制1和机制2场景下的人工鉴别。另外，如果使用消息鉴别码做完整性保护，正常情况下数据及其消息鉴别码都会被传递且能够被被攻击者观察到。而在机制1和机制2中情况不同，这两个机制中使用校验值而非消息鉴别码来保护完整性，这样只有数据是在公开信道上传递的，攻击者在数据串D被传递之前无法获知校验函数的输出（事实上，在机制1中攻击者根本无法接触到校验函数的输出）。这简化了安全性分析和对成功的替换攻击的表达，因此针对机制1和机制2成功实施替换攻击的可能性可以表达为：

$$P_S = \max_{\substack{d,\hat{d}\in D \\ d\neq\hat{d}}} P(f(d,k) = f(\hat{d},k) \mid d\text{被观察到})$$

这样，在密钥k是在密钥空间K中被随机选取的前提下，上述的可能性可以表达为：

$$P_S = \max_{d,\hat{d}\in D, d\neq\hat{d}} \frac{\left|\{k \in K : f(d,k) = f(\hat{d},k)\}\right|}{\left|K\right|}$$

$|K|$表示集合K的势。从这个公式可以看出，为了获得高安全性，校验值函数的碰撞概率就必须很低。这可以通过使用纠错码来构造校验函数来保证，附录E给出了一个纠错码构造校验函数的方案。

基于上述分析，推荐使用16-20位的密钥长度和16-20位的校验值长度。附录E给出了一个表，列举了对于16位和20位的长度，成功施行攻击的可能性。

D.3 机制 3 和机制 5 的使用

机制3和机制5的安全性原理与机制1和机制2基本类似，不同的是使用摘要函数和杂凑函数来替代校验函数。杂凑函数在ISO/IEC 10118[11]中予以标准化，该标准中定义的杂凑函数均可用于机制3和机制5。

因为长随机密钥k被用于保证摘要函数要满足的第二个条件（即摘要值碰撞），所以建议使用160位的k值。

鉴于摘要值是在设备A和设备B之间手工传递或手工比对的，建议使用输出长度为b=16-20比特的短摘要函数。短摘要函数与消息鉴别算法类似，只是摘要值要短于消息鉴别码的典型长度。短摘要值的可能构造方法包括使用消息鉴别码或标准密码杂凑算法输出的前b个比特，文献[25]的附录G给出了建议。

有关机制3和机制5的安全性证明在文献[24]中已经给出。

D.4 机制 4 和机制 6 的使用

机制4和机制6的安全性原理与机制1和机制2不同，使用杂凑函数来替代校验函数。杂凑函数在ISO/IEC 10118[11]中予以标准化，该标准中定义的杂凑函数均可用于机制4和机制6。

对于机制3和机制5，建议使用160比特位的k值。由于机制3和机制5涉及设备A与设备B之间短比特流R的手工传递或者对比，因此建议使用16-20位长度的比特流R。

有关机制4和机制6的安全性证明在文献[24]中给出。

D.5 机制 7 和机制 8 的使用

机制7和机制8的安全性原理与机制1和机制2不同，使用消息鉴别函数来替代校验函数。消息鉴别函数在ISO/IEC 9797中予以标准化，该标准中定义的消息鉴别函数均可用于机制7和机制8。

16-20位的随机比特串适用于机制7和机制8，但是消息鉴别码位数应更长一些。用于机制7和机制8的消息鉴别函数的输出应在128-160比特位范围内。类似地，用于消息鉴别函数的随机密钥K_A和K_B（K_{Ai}和K_{Bi}）长度也应该在128-160比特位范围内。超时程序则用于探测机制7和机制8可能出现的中断。

附 录 E
（资料性附录）
一种产生短校验值的方法

E.1 概述

本附录针对适用于机制1和2的校验函数进行了说明，同时也探讨了使用校验值模式时，两种机制被攻破的概率。使用附录D中的攻击表达式，一个直接的攻击方法就是使用产生自编码理论的校验函数。文献[15]中讨论了纠错码与校验值之间的关系。

在探讨具体实例前，首先给出两个编码理论的基本定义。为简单起见，本部分只探讨有限域F_q上定义的编码。V表示有限域F_q上的多进制代码。假设编码字长度为n，多进制码是消息映射成的编码字，消息和编码字一一对应。多进制码V包括所有向量$v \in V = \{v^{(d)} : d \in D\}$，$v^{(d)} = (v_1(d), v_2(d), \ldots, v_n(d))$，$v_i(d) \in F_q$。

此外，还需要以下两个定义。

定义：如果x和y是两个长度为n的多进制元组，那么它们的海明距离是

$$d_H(x,y) \triangleq |\{i \in \{1,\ldots,n\} : x_i \neq y_i\}|$$

定义：编码V的最短距离是

$$d_H(V) \triangleq \min_{x,y \in V, x \neq y} d_H(x,y)$$

接下来我们将展示如何根据多进制编码构建适用于机制1和机制2的校验函数。构建方法十分简单，消息和密钥空间可以映射为

$$f(d,k) = v_k(d), \quad k \in K = \{1,\ldots,n\}$$

因此，校验函数可通过长度为n的密钥和与编码长度相同的消息获得。

针对上述构建方法，替换攻击成功的概率描述如下。采用附录D中P_S的表达式，P_S应遵循：

$$P_S = \max_{d,\hat{d} \in D, d \neq \hat{d}} \frac{\left|\{k \in K : f(d,k) = f(\hat{d},k)\}\right|}{|K|} = \max_{d,\hat{d} \in D, d \neq \hat{d}} \frac{\left|\{k \in K : v_k(d) = v_k(\hat{d})\}\right|}{|K|}$$

$$= \max_{d,\hat{d} \in D, d \neq \hat{d}} \frac{n - d_H\left(v^{(d)}, v^{(\hat{d})}\right)}{n} = 1 - \frac{d_H(V)}{n}$$

在给出了成功替换攻击概率的精确表达式后，现在探讨具体的构建方法。著名的里德索罗门(RS)码[25]满足具有最小距离且足够长编码这一属性。里德索罗门码可以在任意有限域F_q上构建，编码字的

计算很简单，涉及有限域上多项式求值。将数据（消息）编码成有限域F_q上长度为t的多元组，$d = d_0, d_1, \ldots, d_{t-1}, d_i \in F_q$，那么，生成的里德索罗门码编码多项式如下

$$p^{(d)}(x) = d_0 + d_1 x + d_2 x^2 + \ldots + d_{i-1} k^{i-1}$$

校验函数可直接通过在任意点$k \in F_q$上评估多项式产生

$$f(d,k) = v_k(d) = p^{(d)}(k) = d_0 + d_1 k + d_2 k^2 + \ldots + d_{i-1} k^{i-1}$$

里德索罗门码一般具有以下属性（[25]）：

$n = q = |K|$，

$|D| = q^t = n^t$，

$d_H(V) = n - t + 1$。

以上属性表明校验值的$P_S = (t-1)/n$源于里德索罗门码。随着消息D长度的增加，概率也随之增大。因此，一个较好的获得较低概率的方法是首先对数据使用好的单向杂凑函数，例如，使用在ISO/IEC 10118-3中规定的专用杂凑函数，而后将单向杂凑函数的输出作为里德索罗门码的输入。通过此方法，128比特位（缩短的SHA-1）的消息能够提供足够的安全性。同时，这也表示我们能够无需再增加密钥长度或者是校验函数的输出长度就可以维持较低的概率。表E.1给出了两种构建方法的实例并给出了成功攻击的概率。

表F.1　里德索罗门(RS)码校验值：成功替换攻击的概率，P_s

$\log_2\|D\|$	$\log_2(n)$	P_S
128	16	2^{-13}–2^{-16}
256	16	2^{-12}–2^{-16}
128	20	2^{-17}–2^{-20}
256	20	2^{-16}–2^{-20}

如表所示，编码使用4个十六进制位长度的密钥和校验值进行加密，被攻破的概率不超过2^{-12}，如果将密钥和校验值长度增加至5个十六进制位，概率会减少至2^{-17}或者更低。

附 录 F
（资料性附录）
对机制 1-8 的安全性及效率的比较分析

继2005年机制1、机制2、机制7和机制8（机制7和机制8当时被标记为机制3和机制4）标准化后，又出现了大量更好的机制，这些机制用户输入更少且能够实现对攻击者攻破概率的严格界限。这些机制中的一部分已经得到了安全性证明，而机制1、机制2、机制7和机制8并没有得到充分的安全性证明（尽管这些机制并不明确是否存在安全问题）。

尤其是，文献[19]、[25]、[30]、[1]、[32]、[33]、[6]、[7]、[20]、[21]、[23]和[24]中学者们的相关工作值得我们关注。目前已有大量不同的机制，但它们都可以归为文献[24]提出的两种处理认证数据D的加密函数类型中的一种。两种加密函数类型在本标准中所应用的新机制如下：（1）机制3和机制5使用摘要函数（短输出）将D与长随机密钥k绑定；（2）机制4和6使用杂凑函数（长输出或者加密）将D与随机密钥k和短随机比特流R绑定。

为了使本标准的用户获得最佳可用技术，第7节中提出了四种新机制（机制3-6）。相比于机制1、机制2、机制7和机制8，机制3-6更好且更高效，并且它们无需改变设备的输入和输出接口。

机制3-6已发布，其安全性证明参见文献[21]、[22]、[23]、[24]、[32]和[33]。

四种新机制的主要特点如下：

——机制 3-6 减半了机制 1-2 中人工数据传递和对比的数据量。机制 1 和机制 2 中，用户必须从一个设备向另一个设备同时传递短密钥和校验值，或者对比设备显示的输出值。机制 3-6 减半人工交互的数据量，即用户只需要传递或者对比短摘要值或短随机比特流，二者长度与机制 1-2 中的短密钥或校验值相同。

注 1：短随机密钥、校验值或摘要值的手工传递比执行协议的按键操作、阅读 1 个比特的结果（接受/拒绝）操作或任何其他信息的比较操作等行为更为重要。因此，这些人工交互的后者操作行为在此处和表 F.1 中不做分析。

——尽管机制 3-6 只需要减半的人工数据传递量，但是其安全性高于机制 1-2。文献[21]和[24]展示了机制 1-2 由于计算校验值时使用短比特位长度的密钥 K，其提供的安全性强度低于理想状态。相反地，机制 3-6 中使用的密钥 k 能够通过（高带宽）共享通信链接传递。因此，密钥 k 可大大延长，例如，附录 D 中规定的 160 比特。这一属性来源于一般性杂凑函数的密钥长度理论下界，更进一步地研究请参见文献[2]、[5]、[27]和[31]。

——文献[21]和[24]对机制 3-6 给出安全性证明。尽管机制 3-6 互不相同，但对于相同的人工数据传递量而言，它们提供同等的安全性级别。

——机制 7-8 提供的安全级别与机制 3-6 相同，但它们的安全性依赖于短随机密钥 R 这一秘密人工数据的传递。不同于机制 1-6，机制 7-8 中的短随机密钥应是保密的，即只有设备和用户知晓。因此，在人工数据传递过程中应注意避免随机密钥被监听，例如，通过隐藏相机。如果短随机密钥被破坏，入侵者有可能发起中间人攻击。

表F.1总结了机制1-8中人工数据传递间的不同点和安全性。成功的攻击表明当设备计算的比特串与人工传递比特串是同一比特串时协议虽然奏效了，但是敌手已经成功操纵了数据D，所以当数据D通过共享（不安全）通信链接交换以便设备获得不同版本的数据D时，依旧处于被攻破的状态。

注 2：机制 1、机制 2、机制 3、机制 5、机制 7 和机制 8 中，每一个协议会话使用的校验值函数、摘要函数或消息鉴别函数使用的密钥都是随机和新生的。因此，依赖于附录 D.2 中提到的多信息输入的单个密钥再利用的替换攻击互不相关。更多关于成功攻击的安全性证明和定义参见文献[21]和[24]。

为了对比人工数据传递的数据量，机制3和机制5使用的摘要值长度，机制4、机制6、机制7和机制8使用的短随机比特流R的长度，机制1和机制2校验函数使用的校验值和短密钥长度都采用b比特位。

以上机制涉及两种类型的人工交互方式：（1）人工传递信息位（例如摘要值和短随机比特流）；（2）人工对比两种比特流。表F.1中使用(m, c)来表示人工数据传递比特数(m)和对比比特位数值(c)。

表 F.1　机制 1-8 的对比情况

机制	设备 A 和 B 的接口类型	人工交互 （公开/秘密比特位）	加密函数	成功攻击概率
1（旧）	*A*：简单输入接口 *B*：简单输出接口	$(2b, 0)$（公开数据）	校验函数	$> 2^{-b}$
2（旧）	*A*、*B*：简单输入接口	$(0, 2b)$（公开数据）	校验函数	$> 2^{-b}$
3（新）	*A*：简单输入接口 *B*：简单输出接口	$(b, 0)$（公开数据）	杂凑函数&摘要函数	$2^{-b} + \varepsilon$
4（新）	*A*：简单输入接口 *B*：简单输出接口	$(b, 0)$（公开数据）	杂凑函数	$2^{-b} + \varepsilon$
5（新）	*A*、*B*：简单输入接口	$(0, b)$（公开数据）	杂凑函数&摘要函数	$2^{-b} + \varepsilon$
6（新）	*A*、*B*：简单输入接口	$(0, b)$（公开数据）	杂凑函数	$2^{-b} + \varepsilon$
7（旧）	*A*、*B*：简单输出接口	$(2b, 0)$（秘密数据）	消息鉴别算法	$2^{-b} + \varepsilon$
8（旧）	*A*：简单输入接口 *B*：简单输出接口	$(b, 0)$（秘密数据）	消息鉴别算法	$2^{-b} + \varepsilon$

注 3：机制 1、机制 2、机制 7 和机制 8 在 2005 年予以标准化，因此在表 F.1 中标记为“（旧）”。相反，新加入的机制 3-6 则标记为“（新）”。

注 4：公开数据的人工传递表明在机制 1-6 中传递的数据可以被任何人获知。与此相反，机制 7-8 则对除了设备和用户以外的任何人都保密。

注 5：在对比机制 3-6 的计算复杂性（见附录 F.2）时，机制所需的加密函数信息十分重要，因此，我们根据机制所需的加密函数对机制进行分类。

注 6：表 F.1 中，ε 相对于 2^{-b} 可忽略不计，而“$>2^{-b}$”则通过不可忽略的 ε 值表示攻击成功的概率大于 2^{-b}。

在这些新加入的机制中，机制3和机制5在计算效率方面优于机制4和机制6，文献[16]、[21]、[22]和[23]对此方面做出了解释。机制3和机制5采用短输出摘要函数（例如16-20比特位）对要鉴别的数据D进行处理，而非机制4和机制6采用长输出杂凑函数（例如160或更多比特位）对数据D进行处理的方式。在实际中，一般数据D都比较大，例如，它有可能包含图片或DVD，其长度明显长于随机密钥k，计算数据D的短摘要值显然要快于长杂凑值。因此，在计算成本方面，机制3和机制5比机制4和机制6高效，特别适用于小设备和轻量级加密应用。因为小设备和轻量级加密应用中，优化计算效率是一个重要的衡量标准。

附 录 G
（资料性附录）
生成短摘要值的方法

本附录针对适用于机制3和机制5的两种摘要函数构建方法进行说明。第一种构建方法产生自消息鉴别函数，已在ISO/IEC 9797中予以标准化；第二种构建方法产生自杂凑函数，已在ISO/IEC 10118中予以标准化。

注1：存在许多其他的机制也可用于计算摘要函数，例如：在文献[17]、[23]和[21]中提到的基于拓普利兹矩阵乘法或整数乘法的机制。但是，以上文献中提到的构建方法并未进行充分的分析和检测，因此，本附录中不做描述。

在以下的定义中，$trunc_b(x)$函数输出了；比特串x前（从左侧开始）b个比特位。

定义1：使用密钥k计算消息m的摘要值，计算

$$d(m,k) = trunc_b(MAC_k(m))$$

定义2：使用密钥k计算消息m的摘要值，计算

$$d(m,k) = trunc_b(h(m \parallel k))$$

注2：文献[4]说明了定义1中的摘要值计算方法，文献[22]说明了定义2中的摘要值计算方法。

参 考 文 献

[1] M. Cagalj, S. Capkun, and J. Hubaux, 'Key agreement in peer-to-peer wireless networks', in: Proceedings of the IEEE, Special Issue on Security and Cryptography 94(2) (2006), 467–478

[2] J.L. Carter and M.N. Wegman, 'Universal Classes of Hash Functions', Journal of Computer and System Sciences 18(2) (1979), 143–154

[3] C. Gehrmann and K. Nyberg, 'Enhancements to Bluetooth baseband security', in: Proceedings of Nordsec 2001, Copenhagen, Denmark, November 2001

[4] C. Gehrmann, C.J. Mitchell and K. Nyberg, 'Manual authentication for wireless devices', Cryptobytes 7(1) (2004), 29–37

[5] P. Gemmell and M. Naor, 'Codes for Interactive Authentication', in: Advances in Cryptology – Crypto 1993, LNCS, Vol. 773, D.R. Stinson, ed., Springer, 1993, pp. 355–367

[6] J.-H. Hoepman, 'Ephemeral Pairing on Anonymous Networks', in Proceedings of the Second International Conference on Security in Pervasive Computing (SPC 2005), LNCS, Vol. 3450, D. Hutter and M. Ullmann, eds., Springer, 2005, pp. 101–116

[7] J.-H. Hoepman, 'Ephemeral Pairing Problem', in: Proceeding of the 8th International Conference on Financial Cryptography, LNCS, Vol. 3110, A. Juels, ed., Springer, 2004, pp. 212–226

[8] ISO 7498-2:1989, 信息处理系统—开放系统互联—基本参考模型—第2部分：安全架构（英文版）

[9] ISO/IEC 9797 (所有部分), 信息技术—安全技术—消息鉴别码（英文版）

[10] ISO/IEC 8825-1, 信息技术—ASN.1编码规则—第1部分：基本编码规则（BER）、正则编码规则（CER）和可辨识编码规则（DER）规范（英文版）

[11] ISO/IEC 10118 (所有部分)，信息技术—安全技术—杂凑函数（英文版）

[12] ISO/IEC 11770-3:2008, 信息技术—安全技术—密钥管理—第3部分：采用非对称技术的机制（英文版）

[13] ISO/IEC 18033-3:2005, 信息技术—安全技术—加密算法—第3部分：分组密码（英文版）

[14] ISO/IEC 18033-4:2005, 信息技术—安全技术—加密算法—第4部分：序列密码（英文版）

[15] G. Kabatianskii, B. Smeets and T. Johansson, 'On the cardinality of systematic authentication codes via error correcting codes', IEEE Transactions on Information Theory 42(2) (1996), 566–578

[16] K. Khoo, F.-L. Wong and C.-W. Lim, 'On a construction of short digests for authenticating ad-hoc networks', in: Proceedings of ICCSA 2009, LNCS vol. 5593, pp. 863-876

[17] H. Krawczyk, 'New Hash Functions For Message Authentication', in: Advances in Cryptology – Eurocrypt 1995, LNCS, Vol. 921, L.C. Guillou and J.-J. Quisquater, eds., Springer, 1995, pp. 301–310

[18] J.-O. Larsson, 'Higher layer key exchange techniques for Bluetooth security', in: Opengroup Conference, Amsterdam, October 2001

[19] S. Laur and K. Nyberg, 'Efficient Mutual Data Authentication Using Manually Authenticated Strings', LNCS, Vol. 4301, D. Pointcheval, ed., Springer, 2006, pp. 90–107

[20] A.Y. Lindell. 'Comparison-Based Key Exchange and the Security of the Numeric Comparison Mode in Bluetooth v2.1', in: Proceedings of the Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology, LNCS, Vol. 5473, M. Fischlin, ed., Springer, 2009, pp. 66–83

[21] L. H. Nguyen and A. W. Roscoe, 'Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey', Journal of Computer Security (to appear). See: http://eprint.iacr.org/2010/206.pdf

[22] L. H. Nguyen and A. W. Roscoe, 'Authenticating ad hoc networks by comparison of short digests', Information & Computation 206(2–4) (2008), 250–271

[23] L. H. Nguyen and A. W. Roscoe, 'Efficient group authentication protocol based on human interaction' in: Proceedings of Workshop on Foundation of Computer Security and Automated Reasoning Protocol Security Analysis, 2006, pp. 9–31. See: http://eprint.iacr.org/2009/150

[24] L. H. Nguyen and A. W. Roscoe, 'Separating two roles of hashing in one-way message authentication', in Proceedings of Workshop on Foundation of Computer Security, Automated Reasoning Protocol Security Analysis, and Issues in the Theory of Security, 2008, pp. 195–209. See: http://eprint.iacr.org/2009/003

[25] S. Pasini and S. Vaudenay, 'SAS-based Authenticated Key Agreement', in Proceedings of International Conference on Practice and Theory in Public Key Cryptography (PKC 2006), LNCS, Vol. 3958, M. Yung, Y. Dodis, A. Kiayias and T. Malkin, eds., Springer, 2006, pp. 395–409

[26] I.S. Reed and G. Solomon, 'Polynomial codes over certain finite fields', SIAM Journal 8 (1960), 300–304

[27] D.R. Stinson, 'Universal Hashing and Authentication Codes', in Advances in Cryptology – Crypto 1991, LNCS, Vol. 576, J. Feigenbaum, ed., Springer, 1992, pp. 74–85

[28] D.R. Stinson, 'Cryptography – Theory and Practice', CRC Press, 2002, 2nd edition

[29] SHAMAN Project Deliverable D13 (Annex 2), Final technical report – Workpackage 2 – Security for distributed terminals, 2003. Available at http://www.ist-shaman.org/

[30] S. Vaudenay, 'Secure Communications over Insecure Channels Based on Short Authenticated Strings', in: Advances in Cryptology – Crypto 2005, LNCS, Vol. 3621, V. Shoup, ed., Springer, 2005, pp. 309-326

[31] M.N. Wegman and J.L. Carter, 'New Hash Functions and Their Use in Authentication and Set Equality', Journal of Computer and System Sciences 22(3) (1981), 265–279

[32] F.-L. Wong and F. Stajano, 'Multi-channel Protocols', in Proceedings of the 13th International Workshop on Security Protocols, LNCS, Vol. 4631, B. Christianson, B. Crispo, J.A. Malcolm and M. Roe, eds., Springer, 2005, pp. 128–132

[33] F.-L. Wong and F. Stajano, 'Multi-channel Security Protocols', IEEE Pervasive computing 6(4) (2007), 31–39