(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 109728909 B
(45)授权公告日 2021.07.27

(21)申请号 201910215113.6

(22)申请日 2019.03.21

(65)同一申请的已公布的文献号
申请公布号 CN 109728909 A

(43)申请公布日 2019.05.07

(73)专利权人 郑建建
地址 100096 北京市昌平区新龙城一期20号楼2单元1101室

(72)发明人 郑建建

(74)专利代理机构 南京苏高专利商标事务所(普通合伙) 32204
代理人 李淑静

(51)Int.Cl.
H04L 9/08(2006.01)
H04L 9/30(2006.01)
H04L 9/32(2006.01)
H04L 29/06(2006.01)

(56)对比文件
曹喆.基于USBKey的身份认证机制的研究与实现.《计算机应用与软件》.2011,全文.
邓明荣.基于AES加密算法的USBKey解锁方案.《电脑编程技巧与维护》.2019,全文.

审查员 谭菲菲

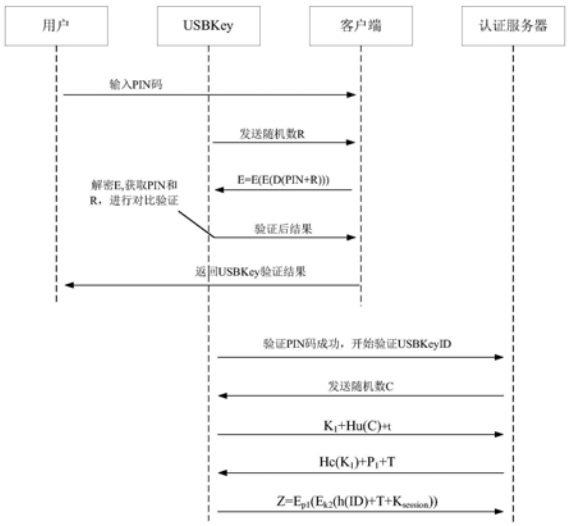
权利要求书3页 说明书6页 附图4页

(54)发明名称

基于USBKey的身份认证方法和系统

(57)摘要

本发明涉及一种改进的基于USBKey的网络环境下的身份认证方法及系统。该方法通过智能安全终端设备USBKey与服务器的双向认证和安全的密钥分发实现了安全通道的建立,客户端代理程序无法触及敏感数据。认证采用软硬件相结合一次一密的强双因子认证模式,很好地解决了安全性与易用性之间的矛盾,并将椭圆曲线ECC算法应用于加解密运算,改进了加解密算法程序设计中的运算问题,降低运算时间复杂度,提高加解密算法整体运行效率。与传统的使用电子证书实现双向认证不同,本发明无需引入第三方证书机构进行电子证书的发放,凭借非对称和对称密钥及相关算法即可实现密钥的协商的安全通道的建立,减小了系统部署的难度。



CN 109728909 B

1.一种基于USBKey的身份认证方法,应用于USBKey,其特征在于,所述方法包括以下步骤:

(1)验证用户的PIN码,包括:

记录用户PIN码,基于客户端请求生成一个随机数R,并发送至客户端;

根据客户端返回的基于PIN码和随机数R的加密数据,解密得到PIN码和随机数R,并与本地存储的PIN码和随机数R进行比对,当一致时,PIN码验证通过,发出开始进行USBKey-ID认证的信号,否则返回出错信息;

(2)验证USBKey-ID,包括:

根据认证服务器传来的随机数C,利用非对称密钥算法产生一对密钥,即公钥K_1和私钥K_2,并利用散列函数h计算该随机数C的散列值$h_u(C)$和公钥K_1的散列值$h_u(K_1)$,生成一个时间戳t,发送$m=\{h_u(C),K_1,t\}$到认证服务器;

根据认证服务器传来的对公钥K_1的散列加密值$h_c(K_1)$、认证服务器根据非对称密钥算法产生的公钥P_1以及时间戳T,验证认证服务器的有效性,当$h_u(K_1)$与$h_c(K_1)$相同且数据发送与接收的时间间隔满足期望时,认为认证服务器身份合法;

根据本地生成的私钥K_2和认证服务器出来的公钥P_1,利用非对称密钥算法对自身ID的散列加密结果h(ID)、接收到的时间戳T和随机生成的会话密钥$K_{session}$进行加密得到$Z=E_{P1}(E_{K2}(h(ID)+T+K_{session}))$,并将加密结果Z发送给认证服务器。

2.一种基于USBKey的身份认证方法,应用于客户端,其特征在于,所述方法包括以下步骤:

基于登录请求向USBKey发送请求随机数的消息,并接收USBKey返回的随机数R;

根据登录时输入的PIN码和接收的随机数R,利用对称密钥算法进行加密,得到加密结果E;

将加密结果E传输至USBKey;

其中所述USBKey上运行如权利要求1所述的身份认证方法。

3.一种基于USBKey的身份认证方法,应用于认证服务器,其特征在于,所述方法包括以下步骤:

基于USBKey发出的开始进行认证的信号,产生一个随机数C传给USBKey,并用散列函数h对该随机数C进行加密得到$h_c(C)$;

根据从USBKey接收的由USBKey根据非对称密钥算法产生的公钥K_1的散列值$h_u(K_1)$、USBKey根据随机数C计算的散列值$h_u(C)$以及时间戳t,验证USBKey的合法性,当$h_u(C)$与$h_c(C)$相同且数据发送与接收的时间间隔满足期望时,认为USBKey身份合法;

用散列函数h对USBKey产生的公钥K_1加密得到$h_c(K_1)$,生成一个时间戳T,根据非对称密钥算法生成一对密钥,即公钥P_1和私钥P_2,形成数据$M=\{P_1,H_C(K_1),T\}$并发送给USBKey;

根据从USBKey接收到的用私钥K_2和公钥P_1通过非对称密钥算法对ID散列加密结果h(ID)、时间戳T和会话密钥$K_{session}$进行加密得到的$Z=E_{P1}(E_{K2}(h(ID)+T+K_{session}))$,利用私钥$P_2$和公钥$K_1$进行解密,得到明文$Z'=D_{P2}(D_{K1}(Z))=h(ID)+T+K_{session}$,比对T是否一致,如果一致就将h(ID)与预先存储的字段进行匹配以确认身份,并赋予其相关权限;否则断开连接;

其中所述USBKey上运行如权利要求1所述的身份认证方法。

4.一种基于USBKey的身份认证系统,其特征在于,所述系统包括:

客户端，运行认证代理程序，用于验证所输入的用户PIN码是否正确，通过将输入的PIN码与一随机数加密后传输给USBKey端进行判断来实现；

USBKey，用于校验用户输入的PIN码，在校验成功后调用安全存储的私钥进行密码运算，并将运算结果返回给代理客户端和认证服务器，并用于验证认证服务器的合法性；

认证服务器，用于验证USBKey的真实身份，通过验证USBKey传来的USBKey-ID与预存在数据库中的USBKey-ID是否一致来实现USBKey的认证；

所述系统进行认证的过程包括：客户端与USB Key之间进行用户PIN码验证，以及USBKey与认证服务器之间进行USBKey-ID验证；

所述客户端与USB Key之间进行用户PIN码验证包括：

用户插入USBKey，登录到客户端代理程序界面，输入PIN码，客户端代理向USBKey端申请发送随机数；

USBKey生成一个随机数R，标记一次会话，传输到客户端代理程序；客户端代理程序利用对称密钥算法对得到的随机数R和PIN码进行加密得到E，传输给USBKey；

USBKey解密E得到R和PIN码，分别与自身存储的R和PIN码做比对，如果一致说明用户为合法用户，然后向认证服务器发送开始认证的信号，开启USBKey-ID验证程序；否则说明用户为非法，将结果返回给客户端代理程序，显示出错信息；

所述USBKey与认证服务器之间进行USBKey-ID验证包括：认证服务器验证USBKey合法身份，USBKey验证认证服务器的合法身份，以及在双方身份合法性认证通过后认证服务器验证USBKey的ID；

所述认证服务器验证USBKey合法身份过程如下：

认证服务器基于接收到USBKey开始认证的信号，产生一个随机数C传给USBKey，并用散列函数h加密随机数C得到$h_c(C)$；

USBKey接收随机数C后，根据非对称密钥算法产生一对密钥，即公钥K_1和私钥K_2，利用散列函数h加密C得到$h_u(C)$，加密K_1得到$h_u(K_1)$，生成一个时间戳t，发送$m=\{h_u(C),K_1,t\}$到认证服务器；

认证服务器接收m，进行$h_u(C)$与$h_c(C)$的对比并验证时间戳的有效性，当$h_u(C)$与$h_c(C)$相等且数据发送与接收的时间间隔满足期望要求时，接受USBKey的连接，否则拒绝连接；

所述USBKey验证认证服务器的合法身份过程如下：

认证服务器将USBKey产生的公钥K_1用散列函数h加密得到$h_c(K_1)$，生成一个时间戳T，根据非对称密钥算法生成一对密钥，即公钥P_1和私钥P_2，发送$M=\{P_1,H_C(K_1),T\}$给USBKey；

USBKey端接收到M后，进行$h_u(K_1)$与$h_c(K_1)$比较并验证时间戳的有效性，当$h_u(K_1)$与$h_c(K_1)$相等且数据发送与接收的时间间隔满足期望要求时，接受连接，否则断开连接；

认证服务器验证USBKey的ID过程如下：

USBKey确认对方为合法认证服务器身份后，随机生成一个会话密钥$K_{session}$，然后利用认证服务器生成的公钥P_1和自身私钥K_2通过非对称密钥算法加密h(ID)、T和会话密钥$K_{session}$，发送$Z=E_{P1}(E_{K2}(h(ID)+T+K_{session}))$给认证服务器，其中h(ID)是通过散列函数h计算得到的ID的密文，T是之前交互时保存的时间戳信息；

认证服务器得到Z后利用自身私钥P_2和USBKey生成的公钥K_1进行解密，得到明文$Z'=D_{P2}(D_{K1}(Z))=h(ID)+T+K_{session}$，比对T是否一致，如果一致就将h(ID)与数据库中存储的字段进

行匹配以确认身份,以便赋予其相关权限;否则断开连接。

基于USBKey的身份认证方法和系统

技术领域

[0001] 本发明涉及信息安全领域,具体涉及一种基于USBKey的身份认证方法和系统。

背景技术

[0002] 以往PKI/CA为用户签发证书时,一般用户在自己的浏览器中通过非对称算法在内存中运算产生密钥对。这一操作会将用户的私钥读取到PC机的内存中再进行签名运算,将私钥的安全依托于PC机操作系统的安全,这必然是一个很大的漏洞。但是,往往PC机的操作系统很容易受到网络病毒或木马的攻击,攻击者会利用系统漏洞盗取用户的私钥和数字证书,就可以假冒用户进行非法签名等。

[0003] 要想防止这一漏洞,唯一解决的办法就是不能将私钥存储在PC机上,只能保存在其他外部设备中,而且为了防止网络病毒或木马的攻击也不能将私钥读取到PC机的内存中进行运算,还要保证利用私钥能够进行数字签名。在智能卡基础上发展起来的USBKey设备,配合PKI体系就能很好的解决上述问题。既能生成密钥对,存储私钥也可以进行数字签名,整个过程只在USBKey完成,提高了系统的安全性。但是上述解决方案中存在加解密的安全风险,私钥主要用于制作电子签名数据或者是解密相关敏感数据等场景,如制作电子签名场景下,存储在公共操作系统之中的私钥易受到木马或病毒的盗取,从而在本人不知情的情况下制作电子签名,无法体现本人的签名意愿。解密场景下的安全风险亦是如此。此外,通过电子证书的方式实现双向认证是一个常规手段,但是CA服务器的部署和管理,电子证书的审核和发放也较为繁琐,过程较长,成本较高。并且在认证完成后,一般安全通道多建立在服务器和客户端之间,即加密数据所用的会话密码(对称算法密钥)保存在客户端中,极易造成传输数据的泄露。

发明内容

[0004] 发明目的:针对现有技术的问题,本发明提出一种基于USBKey的认证方法,主要是一套基于动态口令认证机制的安全双向身份认证和密钥安全分发协议,能够在满足安全性要求的同时提高易用性。

[0005] 本发明的另一目的在于提供一种相应的基于USBKey的认证系统。

[0006] 技术方案:根据本发明的第一方面,提供一种基于USBKey的认证方法,所述方法适用于USBKey端,包括以下步骤:

[0007] (1)验证用户的PIN码,包括:

[0008] 记录用户PIN码,基于客户端请求生成一个随机数R,并发送至客户端;

[0009] 根据客户端返回的基于PIN码和随机数R的加密数据,解密得到PIN码和随机数R,并与本地存储的PIN码和随机数R进行比对,当一致时,PIN码验证通过,发出开始进行认证的信号,否则返回出错信息;

[0010] (2)验证USBKey-ID,包括:

[0011] 根据认证服务器传来的随机数C,利用非对称密钥算法产生一对密钥,即公钥K_1和

私钥K_2,并利用散列函数h计算该随机数C的散列值$h_u(C)$和公钥K_1的散列值$h_u(K_1)$,生成一个时间戳t,发送m＝{$h_u(C)$,K_1,t}到认证服务器;

[0012] 根据认证服务器传来的对公钥K_1的散列加密值$h_c(K_1)$、认证服务器根据非对称密钥算法产生的公钥P_1以及时间戳T,验证认证服务器的有效性,当$h_u(K_1)$与$h_c(K_1)$相同且数据发送与接收的时间间隔满足期望时,认为认证服务器身份合法;

[0013] 根据本地生成的私钥K_2和认证服务器出来的公钥P_1,利用非对称密钥算法对自身ID的散列加密结果h(ID)、接收到的时间戳T和随机生成的会话密钥$K_{session}$进行加密得到Z＝$E_{P1}(E_{K2}(h(ID)+T+K_{session}))$,并将加密结果Z发送给认证服务器。

[0014] 根据本发明的第二方面,提供一种基于USBKey的认证方法,所述方法适用于客户端,包括以下步骤:

[0015] 基于登录请求向USBKey发送请求随机数的消息,并接收USBKey返回的随机数R;

[0016] 根据登录时输入的PIN码和接收的随机数R,利用对称密钥算法进行加密,得到加密结果E;

[0017] 将加密结果E传输至USBKey。

[0018] 根据本发明的第三方面,提供一种基于USBKey的认证方法,所述方法适用于认证服务器端,包括以下步骤:

[0019] 基于USBKey发出的开始进行认证的信号,产生一个随机数C传给USBKey,并用散列函数h对该随机数C进行加密得到$h_c(C)$;

[0020] 根据从USBKey接收的由USBKey根据非对称密钥算法产生的公钥K_1的散列值$h_u(K_1)$、USBKey根据随机数C计算的散列值$h_u(C)$以及时间戳t,验证USBKey的合法性,当$h_u(C)$与$h_c(C)$相同且数据发送与接收的时间间隔满足期望时,认为USBKey身份合法;

[0021] 用散列函数h对USBKey产生的公钥K_1加密得到$h_c(K_1)$,生成一个时间戳T,根据非对称密钥算法生成一对密钥,即公钥P_1和私钥P_2,形成数据M＝{P_1,$H_C(K_1)$,T}并发送给USBKey;

[0022] 根据从USBKey接收到的用私钥K_2和公钥P_1通过非对称密钥算法对h(ID)、时间戳T和会话密钥$K_{session}$进行加密得到的Z＝$E_{P1}(E_{K2}(h(ID)+T+K_{session}))$,利用私钥$P_2$和公钥$K_1$进行解密,得到明文Z'＝$D_{P2}(D_{K1}(Z))$＝$h(ID)+T+K_{session}$,比对T是否一致,如果一致就将h(ID)与预先存储的字段进行匹配以确认身份,以便赋予其相关权限;否则断开连接。

[0023] 根据本发明的第四方面,提供一种基于USBKey的身份认证系统,该系统包括:

[0024] 客户端,运行认证代理程序,用于验证所输入的用户PIN码是否正确,通过将输入的PIN码与一随机数加密后传输给USBKey端进行判断来实现;

[0025] USBKey,用于校验用户输入的PIN码,在校验成功后调用安全存储的私钥进行密码运算,并将运算结果返回给代理客户端和认证服务器,并用于验证认证服务器的合法性;

[0026] 认证服务器,用于验证USBKey的真实身份,通过验证USBKey传来的USBKey-ID与预存在数据库中的USBKey-ID是否一致来实现USBKey的认证。

[0027] 进一步地,根据上述身份认证系统进行认证的过程包括:客户端与USB Key之间的用户PIN码验证,以及USBKey与认证服务器之间的USBKey-ID验证。

[0028] 用户PIN码验证过程如下:

[0029] 用户插入USBKey,登录到客户端代理程序界面,输入PIN码,客户端代理向USBKey端申请发送随机数;

[0030] USBKey生成一个随机数R,标记一次会话,传输到客户端代理程序;客户端代理程序利用对称密钥算法对得到的随机数R和PIN码进行加密得到E,传输给USBKey;

[0031] USBKey解密E得到R和PIN码,分别与自身存储的R和PIN码做比对,如果一致说明用户为合法用户,然后开启USBKey-ID验证程序;否则说明用户为非法,将结果返回给客户端代理程序,显示出错信息。

[0032] USBKey-ID验证包括认证服务器验证USBKey合法身份,USBKey验证认证服务器的合法身份,以及认证服务器验证USBKey的ID。

[0033] 认证服务器验证USBKey合法身份过程如下:

[0034] 认证服务器基于接收到USBKey开始认证的信号,产生一个随机数C传给USBKey,并用散列函数h加密随机数C得到$h_c(C)$;

[0035] USBKey接收随机数C后,根据非对称密钥算法产生一对密钥,即公钥K_1和私钥K_2,利用散列函数h加密C得到$h_u(C)$,加密K_1得到$h_u(K_1)$,生成一个时间戳t,发送$m=\{h_u(C),K_1,t\}$到认证服务器;

[0036] 认证服务器接收m,进行$h_u(C)$与$h_c(C)$的对比并验证时间戳的有效性,当$h_u(C)$与$h_c(C)$相等且数据发送与接收的时间间隔满足期望要求时,接受USBKey的连接,否则拒绝连接;

[0037] USBKey验证认证服务器的合法身份过程如下:

[0038] 认证服务器将USBKey产生的公钥K_1用散列函数h加密得到$h_c(K_1)$,生成一个时间戳T,根据非对称密钥算法生成一对密钥,即公钥P_1和私钥P_2,发送$M=\{P_1,H_C(K_1),T\}$给USBKey;

[0039] USBKey端接收到M后,进行$h_u(K_1)$与$h_c(K_1)$比较并验证时间戳的有效性,当$h_u(K_1)$与$h_c(K_1)$相等且数据发送与接收的时间间隔满足期望要求时,接受连接,否则断开连接;

[0040] 认证服务器验证USBKey的ID过程如下:

[0041] USBKey确认对方为合法认证服务器身份后,随机生成一个会话密钥$K_{session}$,然后利用P_1和K_2通过非对称密钥算法加密h(ID)、T和会话密钥$K_{session}$,发送$Z=E_{P1}(E_{K2}(h(ID)+T+K_{session}))$给认证服务器;

[0042] 认证服务器得到Z后利用P_2和K_1进行解密,得到明文$Z'=D_{P2}(D_{K1}(Z))=h(ID)+T+K_{session}$,比对T是否一致,如果一致就将h(ID)与数据库中存储的字段进行匹配以确认身份,以便赋予其相关权限;否则断开连接。

[0043] 进一步地,上述非对称密钥算法采用椭圆曲线密码(Elliptic curve cryptography,ECC)算法。

[0044] 有益效果:

[0045] 1、本发明通过密码算法和安全协议实现了智能安全终端设备(如USBKey)与服务器的双向认证、密钥分发及安全通道的建立,客户端代理程序(如部署在PC端的)无法触及业务数据,实现在非安全的PC环境的敏感数据的传输。

[0046] 2、与传统的使用电子证书实现双向认证不同,本发明无需引入第三方证书机构进行电子证书的发放,凭借非对称和对称密钥及相关算法即可实现密钥的协商的安全通道的建立,减小了系统部署的难度。

[0047] 3、本发明将椭圆曲线ECC算法应用于加解密运算,改进了加解密算法程序设计中的运算问题,降低运算时间复杂度,提高加解密算法整体运行效率。

附图说明

[0048] 图1是根据本发明实施例的基于USBKey的认证系统结构框图；

[0049] 图2是根据本发明实施例的基于USBKey的认证方法流程图；

[0050] 图3是根据本发明实施例的客户端代理认证机制流程图；

[0051] 图4是根据本发明实施例的认证服务器端认证机制流程图。

具体实施方式

[0052] 现在参照附图对本发明的技术方案做出进一步描述。应当了解，以下提供的实施例仅是为了详尽地且完全地公开本发明，并且向所属技术领域的技术人员充分传达本发明的技术构思，本发明还可以用许多不同的形式来实施，并且不局限于此处描述的实施例。对于表示在附图中的示例性实施方式中的术语并不是对本发明的限定。

[0053] 参照图1和图2，系统总体构架包括USBKey，客户端，认证服务器。客户端通过网关连接至认证服务器，用户通过客户端上运行的认证代理程序与服务器和USBKey交互。

[0054] 客户端用于验证用户输入PIN码是否正确，输入的PIN码会与一个随机数(系统自动生成)由3DES加密程序加密传输给USBKey端，由USBKey端来解密，与存储在USBKey端的PIN码和随机数比对，一致则通过认证，不一致则重新输入。

[0055] USBKey用于校验用户输入的PIN码：在校验PIN码的过程通过随机数R，保证校验指令的数据是一次一密，防止PIN码在传输过程被窃听。校验成功后调用安全存储的私钥进行密码运算，并将运算结果返回给代理客户端和服务器。此外，在认证过程中，USBKey还对服务器的合法性进行验证。在会话密钥分发过程中，USBKey随机生成会话密钥，并使用该密钥对传输数据进行加密，实现UBSKey与服务器之间安全通道的建立。代理客户端和传输链路无法获取传输数据的明文。

[0056] 认证服务器的任务是验证USBKey的真实身份，通过验证USBKey传来的USBKey-ID来实现，与预存在数据库中的USBKey-ID对比，如果一致则身份认证通过，否则切断连接，返回错误消息。

[0057] 整个认证机制的流程分为三个阶段：注册阶段，认证阶段和注销一更新阶段。

[0058] 在这三个阶段中：

[0059] ID为USBKey的ID号；PIN为身份认证码；R为随机数，由随机数产生器生成；t为用户登录时间；$h_c()$为认证服务器中SHA-1消息函数；$h_u()$为USBKey中SHA-1消息函数；K,KR,KL为3DES密钥；h()为单项散列函数。

[0060] 1、注册阶段

[0061] 注册用户认证信息在身份认证服务器端完成，只有注册过的用户才可能参与后续的身份认证。

[0062] 为注册用户认证信息，认证服务器为申请用户分配一个USBKey，并记录该Key唯一的ID序列号。认证服务器在本地数据库中为该用户创建一条包含如下字段的记录：

[0063] <ID号，服务器端上次认证信息、服务器端本次认证信息>

[0064] 其中，ID号取为用户USBKey的ID序列号，服务器上次认证信息和服务器本次认证信息都为空。USBKey进行PIN码和ID的注册，将两者写入到USBKey硬件的保护区域，并且USBKey存储该用户ID所对应的SHA-1,3DES,ECC等加密算法的密文形式，作为一个比对校验

因子。

[0065] 2、认证阶段

[0066] (1)用户PIN码验证

[0067] 用户PIN码验证主要通过USBKey与客户端之间的交互完成，客户端代理程序的流程如图2所示，实现过程如下：

[0068] 用户插入USBKey，登录到客户端代理程序界面，输入PIN码，默认规定为6位数字，点击登录后，系统这时触发一个发送随机数的事件，客户端代理向USBKey端申请发送随机数。

[0069] USBKey生成一个随机数R，标记一次会话，传输到客户端代理程序。客户端代理程序将得到的R和PIN码进行3次DES加密得到E，计算：密钥$K=(K_L||K_R)$，加密$E=DES(K_L)[DES^{-1}(K_R)[DES(K_L[R+PIN])]]$，传输给USBKey。

[0070] USBKey解密E得到R和PIN码，分别与自身存储的R和PIN码做比对，如果一致说明用户为合法用户，然后开启USBKey-ID验证程序；否则说明用户为非法，将结果返回给客户端的认证代理程序，显示客户出错信息。

[0071] (2)USBKey-ID验证

[0072] USBKey-ID验证主要通过USBKey与认证服务器端的交互来完成，图3示出了其处理流程，实现过程如下：

[0073] 认证服务器接到USBKey开始进行认证的信号后，产生一个随机数C传给USBKey，并用SHA-1加密C得到$h_c(C)$。

[0074] USBKey接受C后，根据ECC算法产生一对密钥(公钥K_1和私钥K_2)，SHA-1加密C得到$h_u(C)$，生成一个时间戳t，发送$m=\{h_u(C),K_1,t\}$到认证服务器，最后SHA-1加密K_1，得到$h_u(K_1)$。

[0075] 认证服务器接收m，提取$h_u(C)$，与$h_c(C)$对比，如果不相等拒绝对方的连接。

[0076] 认证服务器验证t'与t之间的有效性。如果$(t'-t)\geq\Delta t$，认证服务器将断开连接，否则接受。其中t'是认证服务器的当前时间戳，Δt是期望的有效时间间隔。

[0077] 认证服务器在确认对方身份后还要进一步确认对方的ID，将K_1用SHA-1加密得到$h_c(K_1)$，生成一个时间戳T，根据ECC算法生成一对密钥(公钥P_1和私钥P_2)，发送$M=\{P_1,H_C(K_1),T\}$给USBKey。

[0078] USBKey端接收到M后，比较$h_u(K_1)$与$h_c(K_1)$，如果不相等断开连接。

[0079] USBKey验证T'与T之间的有效性。如果$(T'-T)\geq\Delta T$，将断开与认证服务器的连接，否则接受。T'是USBKey的当前时间戳，ΔT是期望的有效时间间隔。

[0080] USBKey确认对方为合法认证服务器身份后，随机生成会话密钥$K_{session}$，然后用P_1和K_2通过ECC算法加密h(ID)、T和会话密钥$K_{session}$，发送$Z=E_{P1}(E_{K2}(h(ID)+T+K_{session}))$给认证服务器。其中，h(ID)是通过单向杂凑算法SHA-1计算得到的ID的密文，T是之前交互时保存的时间戳信息。

[0081] 认证服务器得到Z后利用P_2和K_1进行解密，得到明文$Z'=D_{P2}(D_{K1}(Z))=h(ID)+T+K_{session}$，比对T是否一致，如果一致就将h(ID)与数据库中存储的字段进行匹配以确认身份，以便赋予其相关权限；否则断开连接。身份确认的同时安全通道也建立成功，即可使用会话密钥$K_{session}$进行敏感数据的安全传输，同时客户端代理程序无法解密获取该数据。

[0082] 3、注销一更新阶段

[0083] 用户忘记PIN码或丢失USBKey硬件设备,需立即联系生产厂商进行索取PIN码或者注销,发放USBKey的硬件厂商将定期通过与企业建立的安全信道进行更新USBKey-ID。

[0084] 良好的认证机制是认证系统实现安全性和防止各种攻击的关键。本发明身份认证机制采用动态口令认证机制中的请求/响应方式,实现了客户端和认证服务器身份的双向认证,同时保证了客户端和服务器认证信息认证同步。所实现的安全通道建立在密码设备(USBKey)和认证服务器之间,客户端是透传的,提高了安全性。

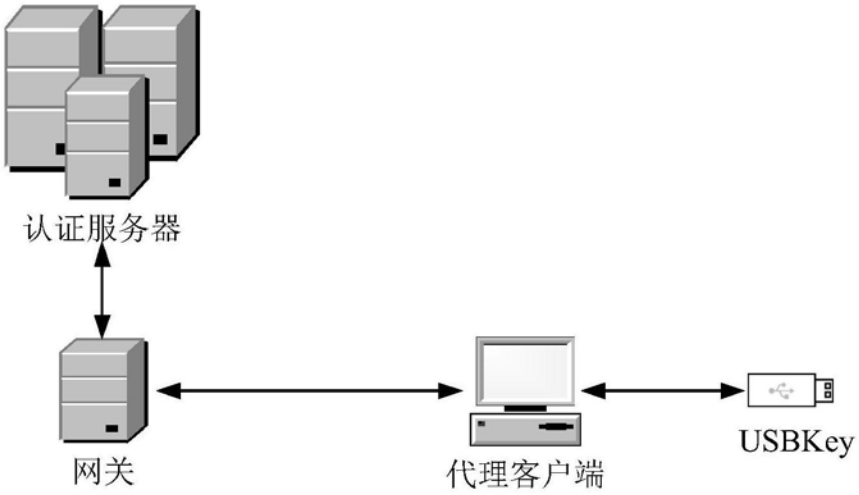


图1

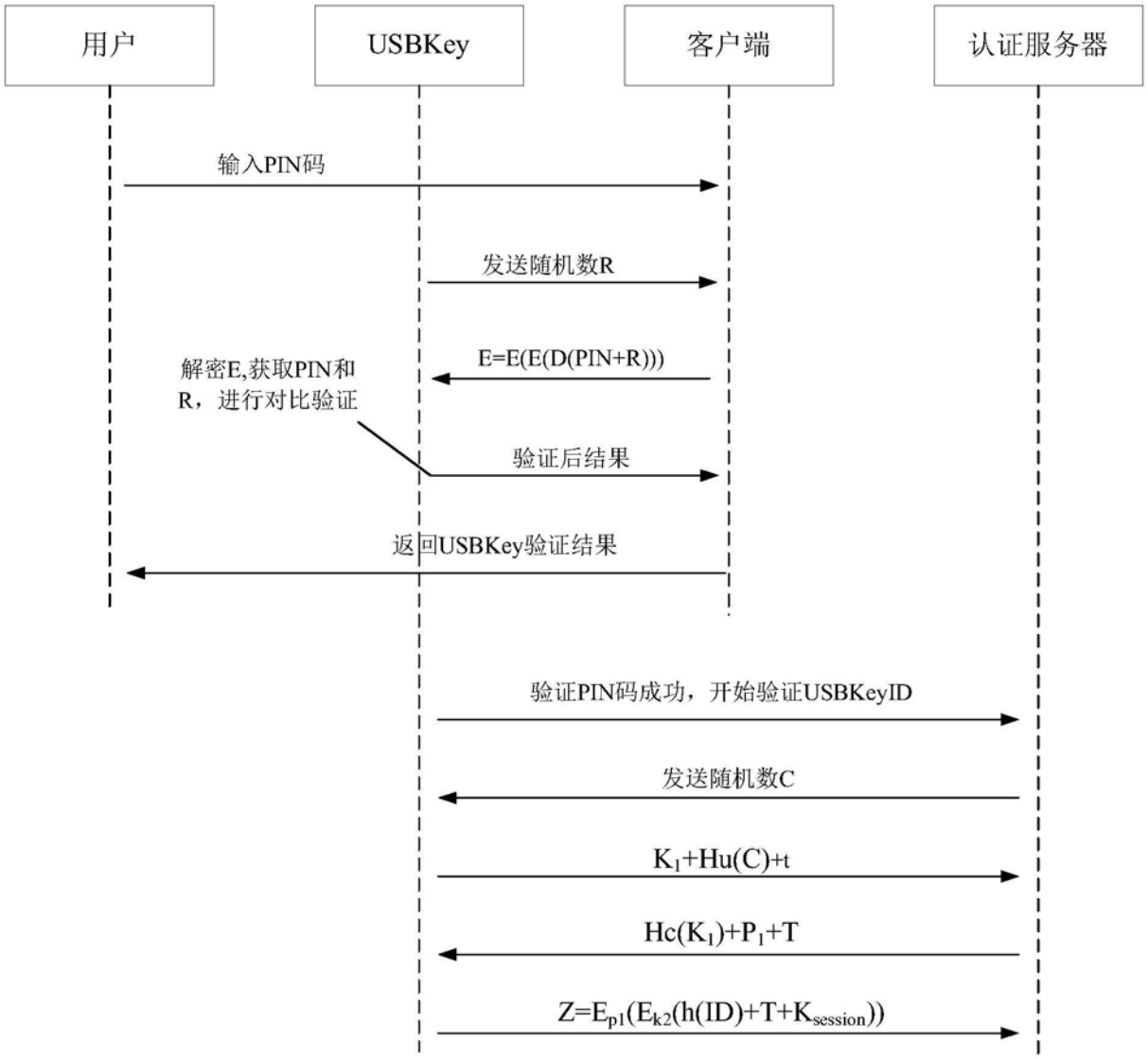

用户
USBKey
客户端
认证服务器
输入PIN码
发送随机数R
E=E(E(D(PIN+R)))
解密E,获取PIN和
R，进行对比验证
验证后结果
返回USBKey验证结果
验证PIN码成功，开始验证USBKeyID
发送随机数C
K1+Hu(C)+t
Hc(K1)+P1+T
Z=Ep1(Ek2(h(ID)+T+Ksession))

图2

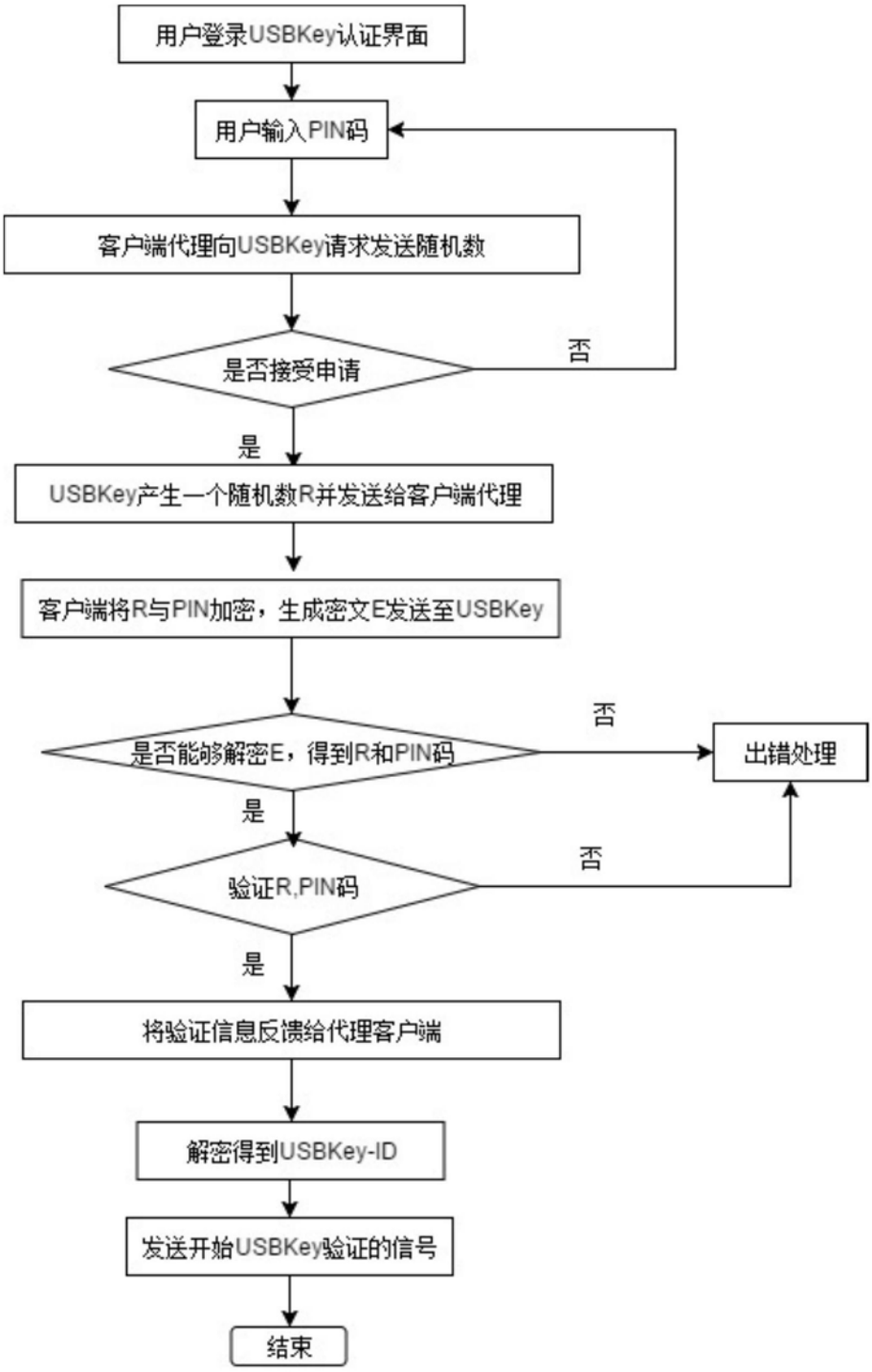

用户登录USBKey认证界面
用户输入PIN码
客户端代理向USBKey请求发送随机数
是否接受申请
否
是
USBKey产生一个随机数R并发送给客户端代理
客户端将R与PIN加密，生成密文E发送至USBKey
是否能够解密E，得到R和PIN码
否
出错处理
是
验证R,PIN码
否
是
将验证信息反馈给代理客户端
解密得到USBKey-ID
发送开始USBKey验证的信号
结束

图3

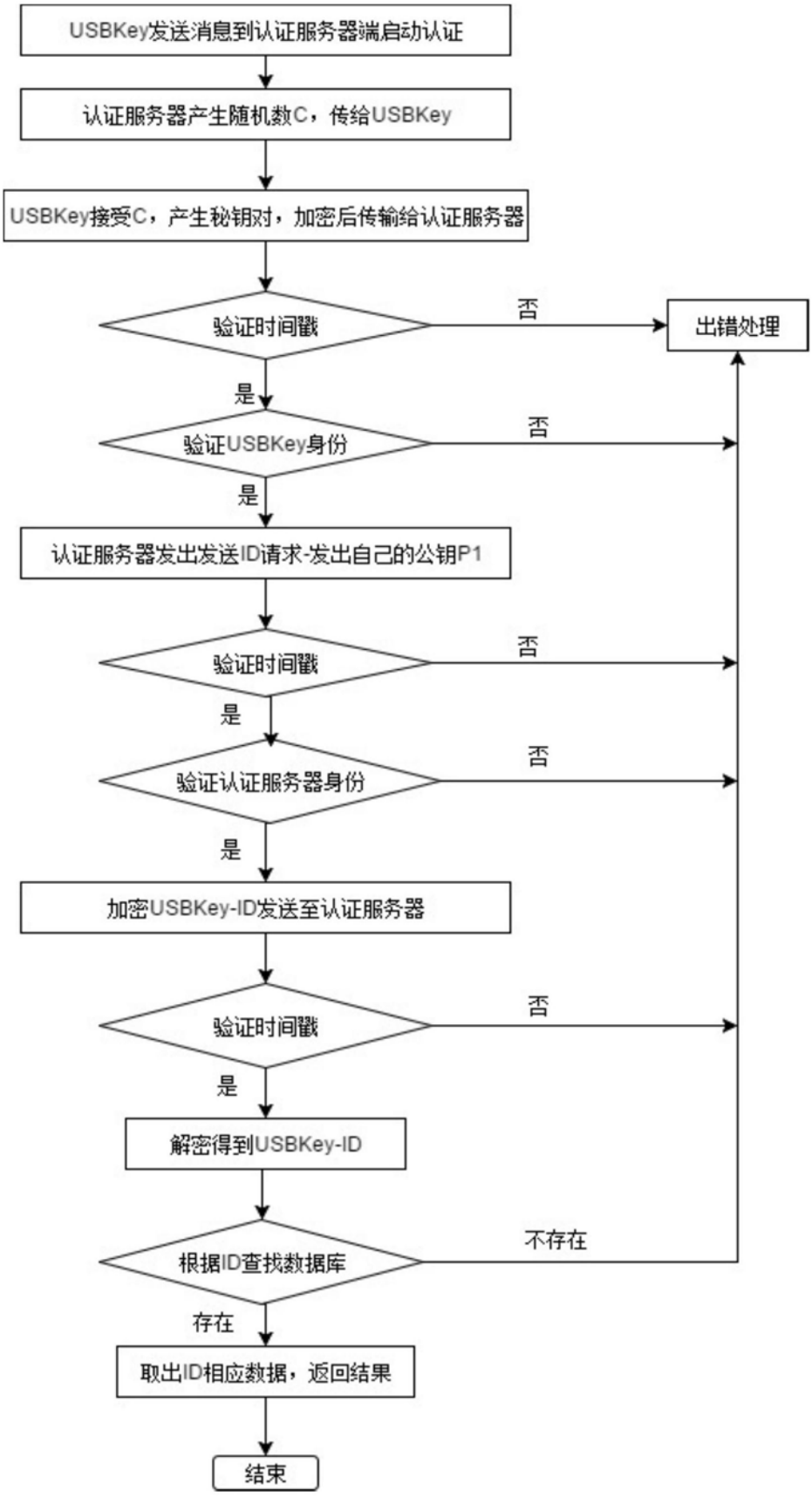

USBKey发送消息到认证服务器端启动认证
认证服务器产生随机数C，传给USBKey
USBKey接受C，产生秘钥对，加密后传输给认证服务器
验证时间戳
否
出错处理
是
验证USBKey身份
否
是
认证服务器发出发送ID请求-发出自己的公钥P1
验证时间戳
否
是
验证认证服务器身份
否
是
加密USBKey-ID发送至认证服务器
验证时间戳
否
是
解密得到USBKey-ID
根据ID查找数据库
不存在
存在
取出ID相应数据，返回结果
结束

图4