第 36 卷　第 5 期　世界科技研究与发展　Vol. 36　No. 5
2014 年 10 月　511 - 515 页　WORLD SCI-TECH R&D　Oct. 2014　pp. 511 - 515

基于一次一密的数字证书部署系统

郭　勇*
（四川大学计算机学院，成都 610065）

摘　要：对数字证书中用于部署的 PKCS#12 证书文件的自身保护机制进行逆向分析和研究。针对其可能遭受到的攻击，提出了一种以一次一密为核心，移动终端、部署证书客户端和服务端这三层为架构模式的身份认证部署系统——基于一次一密的数字证书部署系统（OTPDSYS）。该系统在身份确认方面有良好特性，提供了一种更为安全、稳健的数字证书部署和身份认证解决方案，并可广泛应用于身份认证和软件授权认证等方面。

关键词：数字证书；一次一密；pkcs#12；PFX 口令保护机制；网络身份认证；智能手机终端
中图分类号：TP3 - 0　　**文献标识码**：A　　doi：10.3969/j.issn.1006 - 6055.2014.05.006

Digital Certificate Deployment System Based on One-time-pass

GUO Yong*
（School of Computer Science，Sichuan University，Chengdu 610065）

Abstract：The own protection mechanism of PKCS#12 certificate file is analyzed reversely and researched for the deployment of digital certificates. Based on one-time password and three-tier architecture model of mobile terminal，client and server，a authentication certificate deployment system，the One-time-pass of Authentication Deployment System（OTPDSYS）is proposed. Its good properties in terms of identification provide a more secure，robust deployment of digital certificates and authentication solutions that can be widely used，such as in authentication，certification and software licensing.

Key words：digital certificates；OTP；pkcs#12；PFX Password protection mechanisms；network authentication；smart mobile terminals

1　引言

随着网络攻击方式不断增加，传统数字证书在分发、存储和部署中的安全隐患逐渐增多，已经成为攻击者的攻击目标。本文对 PKCS#12（Public-Key Cryptography Standard）[1]证书文件的自身保护机制进行了分析研究。针对其可能遭受到的攻击提出了基于一次一密码[2]的数字证书部署系统（OTPDSYS），使用手机客户端、部署客户端和验证服务端交互产生一次性验证码 otp，通过认证后，将使用加密传送证书解密密钥至客户端进行证书的解密和安装，以使证书部署与证书私钥保护更加安全、可靠，同时也可应用于软件保护[3]、软件授权管理和对网络安全要求较高的公司与组织的身份认证[4]系统。

2　当前数字证书部署保护机制

目前，主流的数字证书部署主要使用 PFX 数字证书（PKCS#12）。标准的 PKCS#12 是一种用于部署的数字证书文件，通常包含以下几个元素[1]：一个 X.509 证书[5]、所有在证书链的电子商务认证授权机构（Certificate Authority，CA）证书、一个证书独有的私钥。

2.1　pfx 证书的自我保护机制

在公钥体制中，私钥是最重要的部分，是整个加解密体制的核心。PKCS#12 数字证书包含私钥文件，它对于私钥的保护是安全部署的关键。

在需要时，用户根据需求在 CA 申请个人数字证书，CA 通过验证后便会分发 PKCS#12 格式的证书，同时产生一个口令 CPass 用作个人证书私钥的保护口令，与证书一同分发。PKCS#12 文件通过分发时分配的密码对自身进行加密保护，该密码同时用于加密证书文件中的私钥和整个 PKCS#12 文件，是 PKCS#12 证书文件保护的主要手段。

在安装证书时，系统会要求输入密码，在正确输入保护口令 CPass 后，便会将证书以及私钥导入到操作系统里。在 windows 操作系统安装证书时，系统首先判断所安装证书文件是否为 PKCS#12 文件。如果是，系统会在安装 PKCS#12 证书文件时要求输入验证密码，并提供不同级别的私钥保护机制以对安装后的私钥进行存储与使用时的保护。在验证密码是否正确后，将公钥和私钥分开存储在系统的不

2014-03-14 收稿，2014-04-28 接受
* 通讯作者，E-mail：wbscn@sohu.com

同位置。当用户需要导出或使用证书进行网络通信时,操作系统会将指定证书的公私钥及证书的其它信息逐一解密后读出供用户使用。

2.2 口令验证过程

安装证书时,输入口令 pass,点击下一步后,如果系统判断证书文件为 PKCS#12 标准,便会调用相应验证机制进行验证。本次实验证书正确口令为 123,具体验证过程描述如下:

第 1 步:填充扩展。首先将固定前缀 pre_data:0x03 扩展到 64 字节得到 pre_data_64,并将证书文件结尾处 20 字节数据 f_data_salt 按顺序扩展至 64 字节得到 f_data_salt_64,同时将解密口令 pass 的 unicode 编码按顺序扩展至 64 位 upass_64。

第 2 步:产生基本散列。扩展后的数据共 192 字节,基于安全散列算法对其进行 sha1 哈希运算[6]得到 hs1,其中"||"表示数据连接。

hs1 = sha1(pre_data_64 ‖ f_data_salt_64 ‖ upass_64

第 3 步:生成基本比对密钥。在生成基本散列 hs1 后,对其进行 2000 次 sha1 运算,生成 20 字节基本比对密钥 K1,用作随后的比对散列值的运算,同时经过另一系列的算法运算后产生 128 字节数据用于其它运算。

K1 = loop[sha1(ha1)]

第 4 步:生成比对散列值。将 K1 与读取证书文件对应位置数据 f_data 进行 hmac_sha1 运算[6],得出比对散列值 SC。

SC = hmac_sha1(K1,fdata)

第 5 步:比对。使用证书文件保存的 20 字节验证散列值 f_data_hpass,与生成的 SC 逐字节比对,然后将比对结果作为口令验证结果。如果正确,则使用输入口令产生解密密钥,对证书链和私钥分别进行 ASN.1 解码[7]并解密。本实验中 f_data_hpass 与 SC 相同,验证完成。

2.3 算法分析

PKCS#12 证书保护机制的算法流程、加密强度及算法核心特点有以下几个方面:

1)基于 sha1 和 hamc_sha1 算法的安全性。安全散列算法和键控消息认证码基础的安全散列算法是现在比较安全的两种消息散列算法,在具体算法流程中使用 sha1 算法生成 hmac_sha1 算法的 key,使得整个流程衔接流畅,保证了整个验证机制抗中间衔接环节攻击的强度。

2)在运算生成验证散列值时,加入 20 字节随机产生的盐(salt)。盐的加入保证了即使不同证书在加密时使用了同样的密码,也不会影响整个加密结果,保障了证书文件的唯一性。

3)将加密和进行 ASN.1 编码后的证书文件数据作为运算要素加入口令验证过程,使得同样的加密口令产生不同的密钥和验证散列值。同时增加运算量,增强了整个体制的抗穷尽性。

4)使用大量的循环。在产生 hmac_sha1 算法密钥时,采用 2000 次 sha1 循环,使得算法更加不可逆,增加整个体制的抗穷尽性。

2.4 存在的隐患

综上所述,整个 PKCS#12 证书文件口令保护机制整体很强健,在算法的可逆性和抗穷尽等方面都比较优秀。但是由于 PKCS#12 证书文件用作证书的分发、保存的性质,决定了它的便携性和私钥随文件携带等特性,这些都带来了一些隐患:

首先是容易对算法进行攻击。算法体系中随机产生的盐和验证散列值都保存在证书文件中。这些算法中的关键因素在为证书文件的独立性和可移植性提供便捷的同时,也为攻击提供了基础。攻击者可以通过还原算法,对其进行无限制次数的口令穷尽攻击。

其次可以利用失误进行攻击。用户若在证书安装时选择私钥可导出,而未选择强私钥保护,就可能使得攻击者重新导出指定 PKCS#12 格式证书,在另外的攻击机上重新安装证书,从而合法使用该证书。

最后可以利用恶意软件进行攻击。windows 安装 pkcs#12 证书部件对键盘记录几乎无防备。在安装证书文件输入口令时,攻击者可以通过相关恶意软件记录下使用者输入的口令,再通过拷贝证书文件到任意攻击机安装,从而合法使用。

3 OTPDSYS 系统设计

针对 2.4 小节指出的安全隐患,可通过以下途径解决:

首先,基于一次一密的验证方式,部署客户端在安装证书时要求输入一次一密验证码,而验证码是安装在用户的移动终端根据时间而改变的,避免了可能丢失的部署客户端被口令穷尽的攻击。

其次,在认证时确保同一时间只能由一个用户认证和认证通过,即使一次一密保存时间未过期也要销毁,从而保证了攻击者无法利用键盘记录和时

间差来通过合法性验证。为此,提出 OTPDSYS 系统进行数字证书部署。

3.1　一次一密技术(OTP)

一次性密码的主要思路是用户在每次登陆时在一次一密生成算法中加入不确定因素,从而生成在一定时间内唯一有效的密码,用户每次登录时发生更改。

产生一次一密码有两种不确定因素的生成方式:与时间同步;与计数器同步。两种方法通常都要求用户携带一个与服务器同步的小型硬件设备或手机,通过算法来生成密码。

OTPDSYS 系统使用时间同步作为不确定因素 ST,同时为了确保用户的唯一性,加入确定因素 SP。将用户在申请证书时注册的手机号码作为一次一密的 SP,在保障变化性的同时也保障了唯一性。

OTPDSYS 的 OTP 算法以密钥哈希消息身份验证代码(HMAC-SHA1,以下简称 HAMC)为基础。HMAC 算法是基于密钥的一种加密哈希,可接受任意消息和密钥,并将消息映射成固定长度的摘要值(如 20 字节),从而确保只有具有相同密钥的人才能从相同的消息生成相同的摘要值。

该算法以时钟作为不确定因素 SP,以手机号码这一不变因素 SP 作为算法的密钥。具体步骤如图 1 所示。

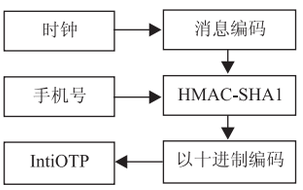

图 1　OTP 算法步骤

Figure 1　OTP Algorithm steps

1)首先读取时钟,并将其编码为 HAMC 计算的输入消息。读取时钟 time 并让其整除时间间隔 interval,将时间间隔设计为 30s,即产生的 OTP 在 30 秒内是有效的。st = time/interval。这样在 30s 内产生的 st 是相同的,其余时间则不同,从而确保每次算法的输入消息不同。最终输入的 ST 编码为 8 字节以适应 HAMC 算法输入。

2)以用户手机号码作为 HAMC 的密钥,相当于为每一位用户配给了独一无二的密钥,这样确保算法的唯一性和确定性。

3)将 HMAC 算法的输出进行十进制编码。在编码中尽可能令 HAMC 产生的哈希值计算位不超过 OTP 结果的长度(在本系统为六位数),尽可能避免丢失位而产生的计算暴露攻击。本系统中取哈希值的最后一位数值作为选取哈希位值的位置,并依次选取所指向四字节中的数值,最终得出 initOTP。具体如图 2 所示。

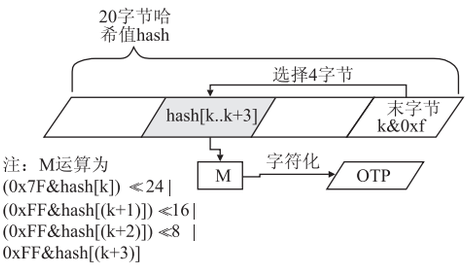

图 2　OTP 算法框图

Figure 2　OTP Algorithm Block Diagram

最终 initOTP 尾部续上服务器返回 Rnum(下文有定义),形成 OTP。

3.2　系统设计

OTPDSYS 的目的是设计出一种安全便捷的网络身份认证系统,因此,采用手机客户端 MC、部署客户端 DC 和服务器端。同时,服务端分为注册部署服务器 RS 和验证服务器 AS,如图 3 所示。

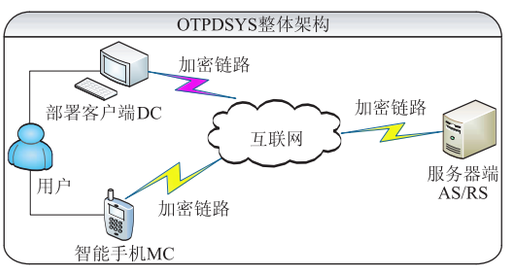

图 3　OTPDSYS 整体架构

Figure 3　OTPDSYS Overall architecture

用户在首次安装个人数字证书时,应配合使用智能手机 MC 产生 OTP,配合部署客户端通过加密链路与验证服务器进行交互后,安装个人数字证书。

为了提高加解密速率,OTPDSYS 系统采用密钥长度可变的流加密算法 RC4[8](加解密使用相同的密钥,属于对称加密算法)为整个通信链路中的信源加解密,使双方可以根据事先约定独自计算得出密钥,从而减少网络传输产生的隐患。

以下分别介绍这三个模块的架构与功能。

3.2.1　服务器端

服务器端作为整个身份认证系统的核心与中转站,是系统安全的关键。OTPDSYS 系统的服务器端包括两个部分:注册部署服务器和验证服务器,分别完成用户注册、生成部署客户端和用户部署数字证书时进行身份验证的工作。

1)注册部署服务器

注册部署服务器 RS 的功能主要用于用户注册,对用户注册信息进行验证、存储,以及生成部署客户端等。主要工作流程如下:

(1)用户在申请目标机构或网站的个人数字证书时,需要填写相关信息,如用户名、手机号和注册密码哈希值。RS 自动记录对应证书编号、加密证书密钥和证书安装密码。

(2)对用户填写手机号进行短信确认,在确认成功后,将该次申请成功证书编号 IDSN 和证书密码 Kpass 发送至手机。

(3)使用加密证书密钥 Ckey 加密数字证书,并与证书 IDSN 一起封装至部署客户端。

(4)提供该用户本次申请成功部署客户端下载链接以供用户下载。下载时需输入发送至手机的证书密码 Kpass 进行验证。

(5)将该用户信息存储至后台数据服务器,以便验证服务器 AS 进行数据的查询。

2)验证服务器

验证服务器 AS 是用于用户在本机进行数字证书部署时的验证工作。AS 拥有的模块主要有:

(1)时钟同步系统[9]。作为一次一密算法中的不确定因子,验证服务端 AS 与手机移动端 MC 的时钟能否同步与一致直接决定验证过程成功与否。

(2)OTP 算法与加密链路加解密模块。

(3)读取用户存储信息与存储临时数据功能模块。

AS 的验证模块也分为两个部分:智能手机验证与部署客户端验证。

(1)与 MC 端的验证过程

(a)接受连接请求。

(b)解析查询:当收到 201 未开始的 OTP 验证请求时,首先解析请求包中用户名 Uname,并使用 Uname 查询该用户注册手机号码 Pnum、注册密码哈希值 HRpass 和对应证书编号 IDSN。

(c)解密数据包:调用 OTP 模块,使用时钟 st 和 Pnum 生成初始 OTP:initOTP = OTP(pnum. st);随后使用 intiOTP 作为初始密钥,解密 MC 发来数据包 mccap。并返回解密后的数据 cryptmc = RC4(mccap. intiOTP)。

(d)验证阶段:将 cryptmc 中的手机号 mcPnum 和注册密码 ha 哈希值 mcHRpass 与第二阶段读取出的 Pnum 和 HRpass 进行对比。若匹配成功,返回验证成功标识 200 和被加密模块加密的随机数 Rnum。将最终 initOTP|Rnum 生成的 OTP 和 IDSN 保存至有效期为 30 s 的临时表 Ttable 中。不成功则返回同步要求 101,要求 MC 进行时钟同步。

(e)时钟同步:当收到 MC 进行时钟同步的请求 100 后,解析数据包,根据用户名查询 HRpass 解密并验证数据包,验证通过后使用 HRpass 作为密钥加密发送至 MC 端。

(2)与 DC 端的验证过程

(a)接受连接请求。

(b)解析验证:收到 301 开头的 MC 验证数据包后,提取数据包中的证书编号 IDSN 与 OTP。查询临时表 Ttable 中 IDSN 对应的 OTP 是否存在或匹配。若验证通过则返回 300,验证通过标识和证书加密密钥 key(key 使用 OTP 和证书口令 Kpass 加密传输),并删除临时表格 Ttable 中本 IDSN 的数据;若验证没通过则返回 302 验证错误。

3.2.2 手机移动端

OTPDSYS 的手机移动端 MC 是用户配合部署客户端使用的移动终端(目前只支持 Android 系统),主要负责为用户提供 OTP。MC 端模块主要有:通讯模块、时钟模块与初始一次一密 initOTP 生成模块。具体工作流程如下:

1)要求用户输入用户名和注册密码 rpass,并建立与验证服务端 AS 的通讯。

2)获取加密因子并生成 initOTP:initOTP 生成模块读取自身时钟系统时钟 st,同时获取本机手机号 Pnum。利用获取的 st 和 Pnum 作为 initOTP 算法输入因子,产生 initOTP。

3)加密发包:用 initOTP 作为加密链路密钥加密手机号码 Pnum 和 rpass 的哈希值。并在数据包前填充 201 OTP 请求标识和用户名,发送至 AS 端,等待验证结果。

4)当收到验证成功标识 200 后,将验证端发送来的数据用 initOTP 解密并提取验证数 Rnum,与 initOTP 结合并压缩至 6 位数字成为最终 OTP。显示至手机端,提供给用户。

当收到 101 时钟同步要求标识后,使用注册密码 Rpass 加密时钟同步字符串,在数据包前填充时钟同步请求标识 100 和用户名然后发送至验证服务端。

当收到时钟同步标识 102 后,解密时钟数据并同步本地时钟,否则断开验证服务器 AS 的连接。通讯过程如图 4 所示。

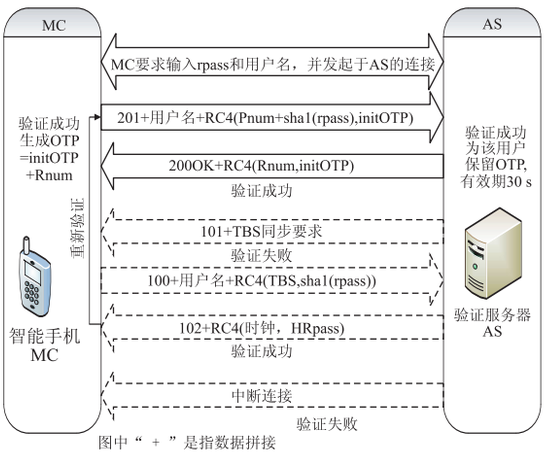

图 4　手机移动端 MC 通讯过程

Figure 4　MC Communication process

3.2.3　部署客户端

部署客户端 DC 是用户使用的部署终端，为用户提供最终的数字证书验证和部署服务，确保用户的合法性、唯一性与安全性，主要有通讯验证模块、解密安装数字证书模块等。具体工作流程如下：

1）提示用户输入证书密码 kpass 和手机客户端显示的 OTP，并建立与验证服务端 AS 的连接。

2）读取 DC 中的证书编号 IDSN 和用户输入 OTP 并将其填充至 301 开始的数据包中，发送至验证服务端 AS，等待验证结果。

3）当收到 300 验证成功标识后，使用 OTP 和用户输入 kpass 解密数据包，验证解密后数据包是否正确，若正确则调用解密模块使用解密后证书加密密钥 key，解密证书并安装，安装证书默认采用证书私钥不可导出，以确保用户的信息安全，通讯过程如图 5 所示。

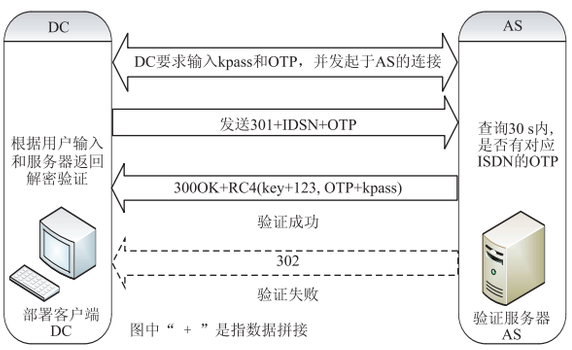

图 5　部署客户端 DC 通讯过程

Figure 5　DC Communication process

3.3　主要性能对比

将 OTPDSYS 系统和传统 PKCS#12 数字证书部署方案进行对比，结果如表 1 所示。

表 1　OTPDSYS 和 PKCS#12 的性能对比

Table 1　Performance comparation of OTPDSYS and PKCS # 12

	PKCS#12	OTPDSYS
对抗恶意软件[10]	无	一次一密对抗
抗穷尽	无	有时间进行约束
使用透明性	用户可选择，但会引起密钥的不安全	用户透明使用，系统默认安装
灵活性	灵活性高	灵活性较差，有网络
可扩展性	不可	可用作身份认证、软件授权保护等

4　结束语

本文对数字证书，尤其是 PKCS#12 格式的 PFX 数字证书文件口令保护机制进行研究，阐述了其验证算法的验证过程，分析了其算法强度，指出了其机制可能存在的安全隐患。重点研究了基于一次一密的文件数字证书部署方案 OTPDSYS 系统，从三个组成模块设计对系统进行了研究。从与 pkcs#12 部署方案的对比结果可以看出，OTPDSYS 系统在恶意软件对抗、抗穷尽破译和可扩展性等方面具有很大的优势。除此之外，OTPDSYS 以其良好的可扩展性还可应用于身份认证、软件授权保护等领域。

参考文献

[1] RSA Laboratories. PKCS #12v1. 1：Personal information Exchange Syntax[EB/OL]. 2012. http://www. emc. com/collateral/white-papers/h11301-pkcs-12v1-1-personal-information-exchange-syntax-wp. pdf.

[2] 王子成，赵晓航，王宏，等. 基于 DNA 密码的一次一密加密算法[J]. 计算机工程与应用，2013，49(6)：146-148.

[3] 赵路华. 基于 USB Key 的软件保护体系研究[J]. 计算机安全，2013，150(8)：18-20.

[4] 李欣，吴旭东. 一种基于证书的统一身份管理技术研究[J]，信息网络安全，2011，129(9)：26-28.

[5] 陈旭东，曹斌，闾凡兵，等. 基于 X. 509 标准的证书交换接口的安全性研究[J]. 贵州大学学报(自然科学版)，2013，30(1)：84-87，110.

[6] 徐名扬，张衡. 基于 SHA-1 算法的加密认证系统设计[J]. 中国集成电路，2011，151(12)：37-44.

[7] 刘雪飞，吴伯桥，凌涛. ASN. 1 在网络管理中的应用研究[J]. 信息安全与技术，2013，95(6)：96-97.

[8] 胡亮，迟令，袁巍，等. RC4 算法的密码分析与改进[J]. 吉林大学学报，2012，50(3)：511-516.

[9] 徐卫军，焦蓉. 无线网状网信标时钟同步系统设计[J]. 计算机工程与设计，2013，34(12)：4 126-4 130.

[10] 杨洪深，赵宗渠，俊峰. 基于中间代码的恶意软件检测技术研究[J]. 四川大学学报(自然科学版)，2013，50(6)：1 216-1 222.