doi :10.3969/j.issn.1671-1122.2013.07.010

DPAPI 离线解密方法及其取证应用

钱镜洁，林艺滨，陈江勇

（厦门市美亚柏科信息股份有限公司，福建厦门 361008）

摘　要：数据保护接口是微软从 Windows 2000 开始引入的一种简易程序接口，主要为应用程序和操作系统程序提供高强度的数据加密和解密服务。Windows 系统用户大量的私密数据都采用了 DPAPI 进行加密存储，而这些私密数据在取证过程中往往扮演着十分重要的角色。针对目前大部分取证软件无法快速解密这些私密数据的问题，文章首先分析了 DPAPI 的加密机制，接着给出了 DPAPI 加密数据的离线解密方法，并在此基础上设计实现了 DPAPI 的解密工具，最后以实际例子演示了 DDT 在取证中的应用。

关键词：数据保护接口；主密钥；解密；取证

中图分类号：TP309　　**文献标识码：**A　　**文章编号：**1671-1122（2013）07-0039-05

The Method of Decrypting DPAPI Offline and Its Application in Forensics

QIAN Jing-jie, LIN Yi-bin, CHEN Jiang-yong

(*Xiamen MeiYa Pico Information Co.Ltd.,Xiamen Fujian* 361008,*China*)

Abstract: Data protect application-programming interface (DPAPI) is a simple programming interface that Microsoft has introduced after Windows 2000, mainly providing high intensity data encryption and decryption services for applications and operating system programs. Most private data of system users are stored and encrypted by DPAPI and these private data often play a very important role in forensics. Confronted with the problem that most of the forensic softwares can not decrypt these private data quickly, this paper firstly analyzes the DPAPI encryption mechanism, then touches upon the method of decrypting DPAPI offline and on this basis, designs and implements DPAPI decryption tool (DDT), and finally gives practical examples to demonstrate the application of DDT in forensics.

Key words:data protect application-programming interface;master key;decryption;forensics

0 引言

绝大多数应用程序都有数据加密保护的需求，存储和保护秘密信息最安全的方式就是每次需要加密或解密信息时都从用户那里得到密码，使用后再丢弃[1]。这种方式每次处理信息时都需要用户输入口令，对于绝大多数用户来说，这种方式也是不可取的。因为这要求用户记住很多信息，而用户一般会反复使用同一个密码，从而降低系统的安全性和可用性。因此需要一种加密机制在不需要用户输入任何信息的情况下也能存储秘密数据，而微软数据保护接口（data protect application-programming interface,DPAPI）便是满足这种要求的程序接口。

从Windows 2000 开始，用户程序或操作系统程序就可以直接调用 DPAPI 来加密数据。由于 DPAPI 简单易用而且加密强度大，大量应用程序都采用 DPAPI 加密用户的私密数据，如 IE 浏览器的自动登录密码[2]、远程桌面的自动登录密码、Outlook 邮箱的账号密码、加密文件系统的私钥等。显然，这些私密数据在电子数据取证中发挥着十分重要的作用。DPAPI 内部加密流程异常复杂而且微软官方也未公布过其内部细节，这给理解该接口内部实现机制带来了极大困难。虽然国外研究人员通过各种手段对 DPAPI 内部机制进行了相关研究和讨论[3-5]，但是分析地也都不够准确和全面。本文在已有的研究基础上对 DPAPI 加密机制做了全面剖析，给出了 DPAPI 的离线解密方法并讨论了其在取证过程中的应用。

1 DPAPI 概述

1.1 DPAPI函数

DPAPI 由一个加密函数（CryptProtectData()）和一个解密函数（CryptUnProtectData()）组成[6]，是一组跟 Windows 系统用

收稿日期：2013-06-24

作者简介：钱镜洁（1984-），女，江苏，工程师，硕士，主要研究方向：文件系统与数据恢复；林艺滨（1979-），男，福建，工程师，本科，主要研究方向：手机取证和计算机取证；陈江勇（1986-），男，福建，工程师，硕士，主要研究方向：文件系统、加密解密和软件逆向。

户环境上下文密切相关的数据保护接口。某个系统用户调用 CryptProtectData() 加密后的数据只能由同一系统用户调用 CryptUnProtectData() 来解密，一个系统用户无法调用 CryptUnProtectData() 解密其他系统用户的 DPAPI 加密数据。

CryptoDataProtecttData() 各参数说明如下：

1）pDataIn。DATA_BLOB 结构指针，指向需要加密的数据明文块。

2）szDataDescr。描述字符串。返回的数据包含该字符串，其未被加密。该参数为可选参数，可以为 NULL。

3）pOptionalEntropy。DATA_BLOB 结构指针，指向一个额外熵参数，可以是一个加密密码。该参数为可选参数，可以为 NULL。若加密时设置了额外熵参数，则解密时必须提供同样的熵参数，否则无法解密。

4）pvReserved。保留，必须为 NULL。

5）pPromptStruct。CRYPTPROTECT_PROMPTSTRUCT 结构指针，用于弹出对话框与用户交互，通常为 NULL。

6）dwFlags。加密标识位，通常为 NULL。

7）pDataOut。DATA_BLOB 结构指针，指向经过加密处理后的密文块。

CryptProtectUnData() 的参数跟 CryptProtectData() 的参数类似，详见 MSDN 文档，这里不再说明。

1.2 基本概念介绍

1）加密应用程序编程接口

加密应用程序编程接口（cryptography application programming interface，CryptoAPI）是 Windows 平台提供的一组函数，该函数允许应用程序对用户的秘密信息进行编码、加密和数字签名等操作。CryptoAPI 内部的加密操作是在加密服务提供程序（CSP）的独立模块中执行，DPAPI 是在 CryptoAPI 的基础上实现封装。

2）加密服务提供程序

加密服务提供程序（cryptographic service provider，CSP）是一组实现标准加密和签名算法的硬件和软件的组合。每个 CSP 都包含一组它们自己定义并实现的函数。不同的 CSP 提供的安全算法不同，且 CSP 是平台相关的。不同版本的 Windows 操作系统提供的 CPS 的个数和类型也不同，每个 CSP 都有其对应的名称和类型，名称必须是唯一的。目前常用的 CSP 类型有 9 种，要指定采用哪种 CSP，只需在 CryptAcquirecContext() 中指定即可，DPAPI 默认使用 PROV_RSA_FULL 类型。

3）算法标识

算法标识（ALG_ID）是微软定义的一系列 32 位整型值，用于指明 CryptoAPI 所采用的加密或散列算法类型。其中以 0x66 开头的标识通常表示对称加密算法，以 0x88 开头的标识通常表示散列算法。例如，CALG_3DES 对应的值为 0x6603，表示为三重数据加密标准。

4）安全散列算法

安全散列算法 (secure hash algorithm，SHA)[7] 是散列算法中的一种，又叫摘要算法，用于产生消息摘要。在数字签名标准（digital signature standard，DSS）中，安全散列算法通常和数字签名算法 (digital signature algorithm，DSA) 一起用于对消息进行数字签名。每一个安全散列算法都有其对应的算法标识。在 CryptoAPI 中，SHA 对应的算法标识为 CALG_SHA。目前，安全散列算法有 4 种：SHA-1、SHA-256、SHA-384 和 SHA-512，可分别产生 160 位、256 位、384 位和 512 位长的消息摘要。

5）会话密钥

会话密钥 (session key) 是随机产生的密钥，使用一次后，立刻被丢弃而不会被保存。在 CryptoAPI 中，会话密钥通常是对称加密算法的密钥。会话密钥由 40~2000 位随机数组成，可以通过调用 CryptoAPI 中的 CryptDeriveKey() 并传递一个散列值来生成。

6）干扰值

干扰值（salt value）也称做“盐”，通常是随机数，一般可以看做是会话密钥的一部分。干扰值被添加到会话密钥后，通常是以明文的形式被放置在加密数据的前端。加入干扰值可以有效地防止对称加密算法被预先计算好的彩虹表攻击。在 CryptoAPI 中，干扰值通过 CryptGenRandom() 来生成。

7）基于口令的密钥派生函数

基于口令的密钥派生函数（password-based key derivation function，PBKDF）[8] 通过对干扰后的用户输入口令计算多次散列来缓和字典攻击。攻击者若想确定口令的正确性，需执行上百万条指令，导致完成一次字典攻击就需要花费大量时间。PBKDF 目前有两个版本：PBKDF1 和 PBKDF2，两个函数均以口令、干扰值和内部函数的迭代次数等作为输入。在 DPAPI 中，采用的是 PBKDF2 版本，且内部做了部分改动。

8）加密散列函数

加密散列函数又叫基于散列的消息认证代码（hash-based message authentication code，HMAC）。使用加密散列函数需要一个密钥，同时还需指定一个散列函数，可以是 MD5 或 SHA-1 等。在 DPAPI 中，HMAC 主要用于数据认证。

9）密钥分组链接

密钥分组链接（cipher-block chaining，CBC）是一种加密模式。在 CBC 模式中，每个分组完的明文块都需要与前一个经过加密后的密文块进行异或操作，然后进行加密操作。然而，由于第一个明文块之前没有对应的加密块，因此需要使用初始化向量。CBC 加密模式是微软默认使用的加密模式，

DPAPI 内部所有对称加密算法默认都采用 CBC 加密模式。

10）填充

填充（padding）是明文根据加密函数进行数据分组后，由于最后一个明文块不满足分组数据长度要求而在尾部额外添加的数据。填充的数据解密后一般会被自动移除，DPAPI 内部所有对称加密算法默认都采用 PKCS 填充方式。

2 DPAPI 加密机制分析

CryptProtectData() 是对 CryptoAPI 的封装，其加密过程如图 1 所示。整个加密过程大致可以分成 3 个阶段，分别为生成主密钥、解密主密钥以及使用主密钥加密数据。

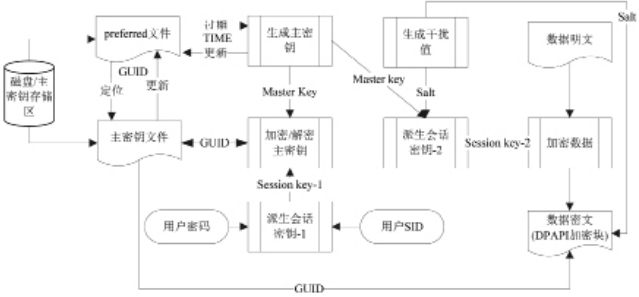

图1 DPAPI加密过程

2.1 生成主密钥

当应用程序调用 CryptProtectData() 时，DPAPI 会读取主密钥存储区下的 Preferred 文件，获取当前系统使用的主密钥文件及其创建时间，如果创建时间与当前系统时间相差超过了 90 天，则重新生成一个主密钥文件。

为了防止攻击者对同一个加密主密钥进行长期的攻击，微软引入了主密钥的更新机制，更新时间微软设置为 90 天。即若 Preferred 文件中指示的主密钥创建时间与系统当前时间相差 90 天以上，将生成一个新的主密钥，新的主密钥将以同样的加密方式保护用户数据。这种主密钥更新策略有效防止了攻击者破解唯一的主密钥后即可访问用户所有的受保护数据。因为主密钥会更新，因而 DPAPI 必须提供一种机制能够解密历史主密钥加密下的数据块。其实，DPAPI 不删除任何过期的主密钥，所有主密钥文件保存在用户的配置文件目录下，且全都受到用户登录密码的保护，并且每一个加密块都存储着当时加密它的主密钥全局唯一标识符（GUID）。当需要解密加密块时，DPAPI 从加密块中提取 GUID，找到对应主密钥文件进行相应的数据解密。

2.2 解密主密钥

若 Preferred 文件指示的主密钥没有过期，DPAPI 将解密对应的主密钥文件，获取 64 字节的主密钥。

主密钥受到用户登录密码的保护。DPAPI 首先使用 SHA–1 安全散列函数作用于用户登录密码，然后将此密码散列和 16 字节的干扰值以及迭代次数提供给基于口令的密钥派生函数 PBKDF2，用于派生一个会话密钥；然后用此会话密钥作为对称加密算法的加密密钥，对主密钥进行加密，将加密后的主密钥存储在用户的配置文件目录下。

为了防止主密钥被篡改，主密钥将被计算 HMAC 加密散列。DPAPI 将使用 SHA 版 HMAC 加密散列算法并以密码散列作为加密密钥作用于 16 字节干扰值，进而派生对应加密散列值。该加密散列值再次作为 HMAC 的密钥计算主密钥的加密散列值，计算后的加密散列值同加密后的主密钥一起存于主密钥文件中。

由于主密钥受到用户登录密码的保护，而用户登录密码又是可修改的。因此，DPAPI 必须提供一种机制，使得在用户修改登录密码后仍然可以正常解密主密钥。其实，DPAPI 对密码修改模块进行了 Hook 操作。当用户修改密码时，所有主密钥都将根据新的密码重新加密。另外，用户配置文件目录下有个历史凭据文件 CREDHIST，当用户修改密码时，旧密码的 SHA–1 散列值会用新的密码进行加密，然后将加密后的结果存放在文件的底部。因此，如果当前系统登录密码无法解密主密钥，DPAPI 将使用当前密码解密历史凭据文件，获取上一次历史密码散列值，然后用这个历史密码散列值解密主密钥。如果解密又失败了，历史密码散列值将再次用于解密历史凭据文件，获取更旧的历史密码散列值。如此下去，直到成功解密主密钥为止。

2.3 使用主密钥加密数据

主密钥并不直接作为加密密钥来保护数据。DPAPI 首先将主密钥、16 字节干扰值以及应用程序提供的额外熵参数 3 者组合派生出一个会话密钥，然后用这个会话密钥对数据进行加密。但这个会话密钥永远不会被保存，DPAPI 选择存储用于派生会话密钥的 16 字节干扰值。这些干扰值是用来产生加密数据的关键。当 DPAPI 需要解密加密块时便从加密块中提取这 16 字节的干扰值，并以同加密相同的方式派生出会话密钥，然后用该会话密钥对数据进行解密。

3 DPAPI 离线解密方法

由以上对 DPAPI 加密过程的分析容易得出 DPAPI 离线解密过程，如图 2 所示。

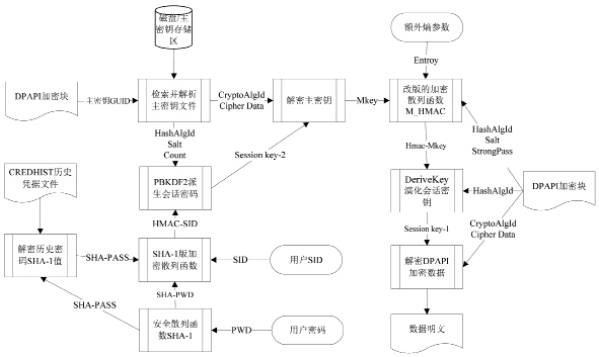

图2 DPAPI离线解密过程

DPAPI 离线解密过程大致可以分为以下几点。

3.1 定位主密钥

由对 DPAPI 加密过程的分析可知主密钥文件会定期更新，Preferred 文件中存储着最后生成的主密钥的 GUID，然而任意给定的一个 DPAPI 加密块并不一定是用最新的主密钥进行的加密。为了快速定位 DPAPI 加密块对应的主密钥文件，DPAPI 加密块中存储当时加密它的主密钥 GUID。DPAPI 加密块大致结构如图 3 所示。

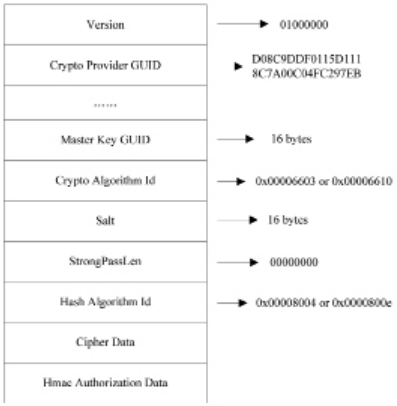

图3 DPAPI加密块结构

从 DPAPI 加密块结构可知其除了存储主密钥 GUID 外，还存储解密时所需的其他关键数据信息，包括所使用的对称加密算法标识、安全散列算法标识、干扰值等。其中第一个字段表示版本号，为固定值 0x00000001；第二个字段表示加密服务提供程序的 GUID，也为固定值 D08C9DDF0115D1118C7A00C04FC297EB。由于标准的 DPAPI 加密块总是以单个完整的文件存在，所以标准的 DPAPI 加密块有可能包含在文件中作为文件的一部分存在，甚至有些文件可以包含多个标准的 DPAPI 加密块。由于 DPAPI 加密块的前两个字段总是为固定值，因此可以以此为特征在文件中检索和提取标准的 DPAPI 加密块。

3.2 解密主密钥

Windows 的主密钥文件和历史凭据文件有其固定的路径结构，不同操作系统默认的主密钥的存储路径不同，如表 1 所示。

表1 主密钥路径

操作系统	主密钥文件的存储路径
Windows XP	{System}\Documents and Settings\{UserName}\Application Data\Microsoft\Protect\{SID}\
Windows 2003	
Windows Vista	{System}\Users\{UserName}\AppData\Roaming\Microsoft\Microsoft\Protect\{SID}\
Windows 7	
Windows 8	

其中，{System} 表示操作系统所在分区；{UserName} 是 Windows 系统账户名，每个系统账户都有其对应的若干个主密钥；{SID} 是 Windows 系统为区分不同账户而为它们分配的全局唯一标识符，它们的格式是固定的。用户名主要是为了方便系统用户记忆而设置的，Windows 系统内部并不用其来区分不同的用户。和主密钥对应的历史凭据文件则位于主密钥的上一层目录。

主密钥文件共包含 5 个数据单元，分别为主密钥头部单元、用户主密钥单元、本地加密密钥单元、历史凭据标识单元和域密钥备份单元。结构如图 4 所示。

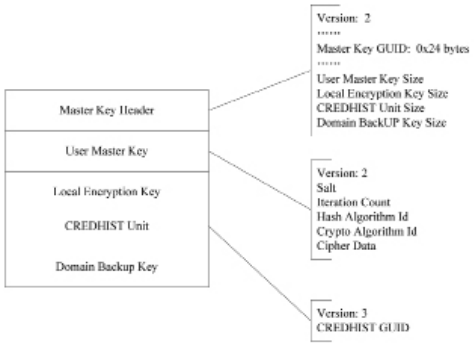

图4 主密钥文件结构

其中，主密钥头部单元包含一个主密钥 GUID，该标识与 DPAPI 中指示的加密密钥唯一标识相对应。另外，主密钥头部单元还包含了指示其他各单元占用字节数的字段。历史凭据单元也包含一个全局唯一标识符（CREDHIST GUID），用来指示跟主密钥文件对应的历史凭据文件。对于域密钥备份单元，只有域环境下的用户才会有，此单元的数据经过了域管理员的公钥加密处理。对于单机用户，无论什么系统都没有域密钥备份单元，这篇文章主要解析用户主密钥单元。用户主密钥单元包含一个经过加密的二进制加密块，其中的加密数据就是主密钥。

用户主密钥单元中包含 PBKDF2 采用的迭代次数、安全散列算法类型和对称加密算法类型等字段，不同的操作系统有着不同的值，如表 2 所示。

表2 用户主密钥单元参数

操作系统	PBKDF2 迭代次数	安全散列算法类型	加密算法类型
Windows XP	4000	SHA-1	DES-3
Windows 2003	4000	SHA-1	DES-3
Windows Vista	24000	SHA-1	DES-3
Windows 7	5600	SHA-512	AES-256
Windows 8	8000	SHA-512	AES-256

由 DPAPI 的加密过程分析可知，要使 DPAPI 正常运作，必须保存用户所有的历史登录密码的 SHA-1 值。用户所有的历史登录密码的 SHA-1 值存储在一个名为 CREDHIST 的历史凭据文件中，其结构如图 5 所示。

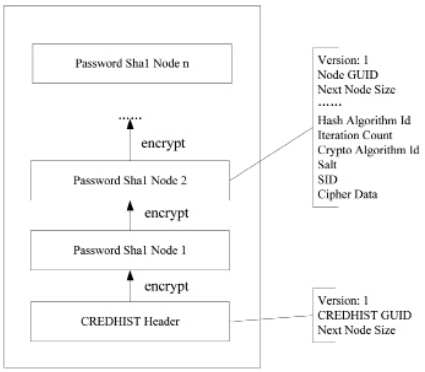

图5 历史凭据文件结构

CREDHIST 以链表的形式存储用户历史登录密码的 SHA-1 值，每个用户的登录密码散列值作为链表的一个节点，整个链表在文件中以反向形式存储，即表头在最后，倒数第二个记录是第一个节点，依次类推。而每个链表节点都具有相同的数据格式，并且每个节点的 SHA-1 值被前一个节点的 SHA-1 值加密处理，加密的方式与主密钥的加密方式一样。

图 2 的解密主密钥过程中涉及一个基于口令的密钥派生函数（PBKDF2）的调用，微软的 PBKDF2 跟工业标准 (PKCD#5)[8] 不太一样，其内部做了部分改动。工业标准的 PBKDF2 函数循环内的伪随机函数的输入是上一次循环伪随机函数的输出，而微软 PBKDF2 函数伪随机函数的输入则是上两次相邻伪随机函数输出的异或。

3.3 解密DPAPI加密块

从图 2 的 DPAPI 离线解密过程中可以知道，在解密 DPAPI 加密块时先需要从主密钥派生一个会话密钥，派生过程中调用了改版的加密散列函数（M_HMAC()）和一个演化密钥函数（DeriveKey()）。M_HMAC() 输入包括 DPAPI 提供的额外熵参数。M_HMAC() 具体描述如图 6 所示。

令
H()表示对应的散列函数
K 表示密钥，这里对应主密钥的 SHA-1 值
M 表示需要认证的数据，这里对应 Salt
E 表示额外熵参数
S 表示强密码
// 表示两个字符串的连接
⊕表示异或
Opad 表示外部填充块(0x5c5c5c…5c5c，与对应散列算法输出相同长度的 16 进制常量值)
Ipad 表示内部填充块(0x363636…3636，与对应散列算法输出相同长度的 16 进制常量值)
则
M_HMAC(K, m, e, s) 算法的数学定义为：
M_HMAC(K, m, e, s) = H((K ⊕ opad) // H((K ⊕ ipad) //m) // e // s)

图6 M_HMAC()描述

DeriveKey() 对应于 CryptoAPI 的 CryptDeriveKey(), 具体描述如图 7 所示。

令
H()表示对应的散列函数
F-t()表示取前 t 个字节数据
n 表示需要演化出的密钥长度
d 表示被演化的数据(对应 M_HMAC()输出)，并通过填充 0 补齐 64 字节长
//表示两个字符串的连接
⊕表示异或
Opad 表示外部填充块(0x5c5c5c…5c5c，64 字节长的 16 进制常量值)
Ipad 表示内部填充块(0x363636…3636，64 字节长的 16 进制常量值)
则
DeriveKey(d, n)算法的数学定义为：
DeriveKey(d, n) = F-n(H(d ⊕ ipad) // H(d ⊕ opad))

图7 DeriveKey()描述

4 DPAPI 离线解密的应用

在 DPAPI 离线解密方法的基础开发了 DPAPI 加密块的离线解密工具 (DPAPI decryption tool，DDT)。下面以获取远程桌面密码为例讨论 DDT 在取证中的应用。

Windows XP-SP1 自带的远程桌面程序为 mstsc.exe（5.x 版本，更新的操作系统为 6.x 版本，登录密码的加密方式和存储位置已改变）。当选择保存远程桌面登录密码时，默认会在“我的文档”下生成一个隐藏的 Default.rdp 文件，其以“键 - 值”对的形式存储了登录时的各种信息，如图 8（a）所示，其中 password 键对应的值是登录密码经过 DPAPI 加密处理所生成数据块的 16 进制编码。将 Defualt.rdp 输入到 DDT 解密后，结果如图 8（b）所示。

(a) (b)

图8 解密前后对比图

5 结束语

电子取证涉及的技术非常复杂，一些用户相关敏感数据往往都经过加密存储，使得取证过程变得十分艰难。在 Windows 操作系统中，DPAPI 作为具有加密功能的最主要接口之一，保护着大量的用户私密数据。本文从 DPAPI 加密原理出发，给出了 DPAPI 离线解密方法，开发了对应的离线解密工具，为办案人员快速获取重要用户信息提供了便利。然而，文章给出的 DPAPI 离线解密方法只能适用于单机用户，无法解密域管理员控制下的 DPAPI 加密数据。域环境控制下的 DPAPI 加密机制相对于单击用户有很大不同，是未来研究工作的重点。（责编 马珂）

参考文献

[1]Michael Howard,David LeBlan 著 . 程永敢 , 翁海燕 , 朱涛江等译 . Writing Secure Code[M]. 北京 : 机械工业出版社，2005.

[2] 张航 , 吴灏 , 许蓉 .IE8.0 登录信息保护机制的缺陷分析与利用 [J]. 计算机工程 ,2013,39(01):313-317.

[3]E.Bursztein,I.Fontarensky,M.Martin,J.Picod. Beyond files recovery, OWADE cloud-based forensic[C].BlackHat USA,2011.

[4]J.Picode,E.BurszTein.Reversing DPAPI and Stealing Windows Secrets Offline[C].BlackHat DC,2010.

[5] 庄继辉 , 吴灏 .Windows 证书保护机制的缺陷分析 [J]. 计算机工程 ,2011,37(23):144-146.

[6]Burzstein J P.Recovering Windows Secrets and EFS Certicates Offine[C].Proc. of the 4th USENIXConference on Offensive Technologies.Berkeley,USA:USENIX Association,2010.

[7] 张松散 , 陶荣 , 于国华 . 安全散列算法 SHA-1 的研究 [J]. 计算机安全 ,2010,(10):3-5.

[8]B.Kaliski. Password-Based Cryptography Specification Version 2.0[EB/OL].http://tools.ietf.org/html/rfc2898,2000-09.