DOI:10.14132/j.cnki.1673-5439.2009.03.006
第29卷 第3期 南京邮电大学学报(自然科学版) Vol.29 No.3
2009年 6月 Journal of Nanjing University of Posts and Telecommunications(Natural Science) Jun. 2009

一种基于对称加密和隐写术的反取证方法

王 灿,秦志光
(电子科技大学 计算机科学与工程学院,四川 成都 610054)

摘 要:提出了一种结合使用对称加密和隐写术的反取证方法。该方法用对称加密和异或运算对文件进行隐秘处理,较好地缓解了传统对称加密中短密钥便于记忆和长密钥更加安全之间的矛盾。在对称密钥泄露的情况下仍能保证隐秘文件的安全。该方法用低开销大幅度提高了破解难度,对隐写的载体文件没有特殊要求,适合在反取证环境中使用。根据此方法,开发了基于Windows平台的反取证原型工具StegEncrypt,该命令行工具可用于使保存或传送的文件难以暴力破解。

关键词:反取证;对称加密;隐写术
中图分类号:TP393.08 文献标识码:A 文章编号:1673-5439(2009)03-0027-05

An Anti-forensic Scheme Based on Symmetrical Encryption and Steganography

WANG Can, QIN Zhi-guang
(School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu 610054, China)

Abstract: In this paper, a anti-forensic scheme based on symmetrical encryption and steganography is proposed. According to the scheme, the secret file is processed by the algorithm of symmetrical encryption and XOR, and it can not be reconstructed by the attacker even the symmetrical key is revealed. The contradiction between convenience of short key and security of long key, which exists in traditional symmetrical encryption, is mitigated by the scheme. The difficulty of cracking is notably augmented with a low cost and there is no special requirement on the carrier files using in steganography. So the scheme is suitable for the application environment of anti-forensic. On the basis of the scheme, a prototype tool, which is called StegEncrypt, is developed in Windows platform. This command-line tool can be used to make it hard to decrypt the files, which are to be saved or transferred by force.

Key words: anti-forensic; symmetrical encryption; steganography

0 引 言

目前,在计算机网络犯罪手段与网络安全防御技术之间的对抗不断升级的形势下,计算机取证(computer forensic)越来越受到重视[1-2]。在计算机取证技术发展的同时,入侵者也在绞尽脑汁对付取证,并由此兴起了一种新的技术——反计算机取证(anti-computer forensic,反取证)技术。知己知彼方能百战不殆,因此通过研究反取证技术以保证计算机取证的科学性和有效性是非常重要的[3]。

入侵者在实施入侵的过程中,会在他所使用的计算机(简称工具机)和目标计算机(简称目标机)中留下大量的电子证据。为了不被发现,高明的入侵者会在行动后删除或破坏这些电子证据,从而尽量隐藏或不在系统中留下具有法律意义的证据,使得取证方法无效。目前,反取证的主要技术有数据摧毁(data wiping)、数据加密(data encryption)、数据隐写(data steganography)、数据混淆(data obfusca-

收稿日期:2009-02-27
通讯作者:秦志光 电话:(028)83202201 E-mail:qinzg@uestc.edu.cn

tion)和针对取证工具缺陷进行攻击等[4-6]。

虽然利用数据摧毁技术擦除所有潜在的证据是最有效的反取证方法,但很多情况下入侵者需要暂时保留一些有用数据文件和工具程序。为了使这些秘密文件不被发现,入侵者常常对文件进行加密或隐写处理。另外入侵者在从目标机传回窃取的敏感数据时,为了防止被发现,也需要对数据进行加密或隐写处理[7]。

但单纯的加密技术或隐写技术都存在一些不足之处[8-9]。例如,加密技术在算法选定后,其安全性依赖于密钥的长度和随机性。由于人脑并不擅长于记忆长的随机密钥,使用者更倾向于使用较短的和有意义的密钥,如生日、姓名、昵称等。但随着计算能力的不断增强,基于字典的密钥猜测对于破解短密钥和非随机密钥越来越有效。另一方面,传统的隐写技术需要大量的开销来隐写相对少的信息,并且其隐秘性是建立在对手不知情的前提下,如果这种手法被对手发现,就会变得毫无价值[10]。而且传统的隐写技术对载体文件有一定的要求,当需要隐写的文件较多时,短时间内很难在计算机上找到足够多的合适载体文件。

为了克服这些弱点,本文提出了一种结合使用两种技术的反取证方法[11-12]。该方法在用户使用短密钥的情况下仍有较强的对抗暴力破解的能力,而且即使对称密钥泄露,对手仍不能解密文件。

1 方 法

1.1 方法的原理

本文的方法主要分3个步骤对秘密文件进行隐秘处理,其原理如图1所示。

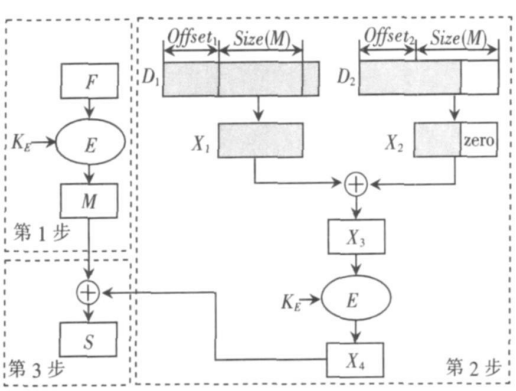


图1 方法原理图

(1)对称加密。

假设需要隐秘处理的原文件为 F,使用选定的对称加密算法 E和对称密钥 K_E 对其进行加密,得到加密后的文件为 M,即 $M=E(F, K_E)$。

对称加密算法 E可以由使用者根据需求任意选择,如常用的 DES、3DES和 AES等。使用对称加密是因为相对非对称加密来说其运算量较小,加密速度快,更适合反取证应用环境。

(2)生成异或算子。

① 从计算机系统中任选出2个普通文件 D_1 和 D_2,并分别从2个文件中复制出与 M大小一样的数据块 X_1 和 X_2;

② 把数据块 X_1 和 X_2 进行异或运算得到 X_3,即 $X_3=X_1\oplus X_2$;

③ 使用与加密 F的相同对称加密算法 E和密钥 K_E 对 X_3 进行加密,得到 X_4,即 $X_4=E(X_3, K_E)$,称 X_4 为异或算子。

这里的 D_1 和 D_2 是任意选取的2个内容不易发生变化的普通文件,即不包含任何秘密信息的公开文件,它们可以是图片文件、mp3文件、可执行文件、系统 dll文件等等,但究竟使用的是哪2个文件则是保密的,只能使用者自己知道。例如可以选择 Windows系统中普遍存在的 winmine.exe(扫雷游戏)和 freecell.exe(空当接龙游戏),或者选择 WINWORD.EXE和 EXCEL.EXE等等。这样使用者很容易记忆,但对手要在大量的普通文件中确定对方使用的是哪2个文件却很困难。

为了进一步增加安全性,在从 D_1 和 D_2 中取数据块时还引入了两个文件偏移量 $Offset_1$ 和 $Offset_2$,即从 D_i 文件头偏移 $Offset_i$ 的地方开始复制数据($i=1, 2$),直到复制的数据块大小与 M大小相同,即 $Size(X_i)=Size(M)$。如果复制到文件末尾时,$Size(X_i)<Size(M)$,则在后面补“0”,直到 X_i 与 M大小相等为止。$Offset_i$ 可以由使用者任意指定或者根据某一规则生成,其中后者更方便记忆和使用。本文中用以下规则生成 $Offset_i$:

① 选定一个控制密钥 K_c,例如使用自己喜欢的电影角色的名字“Mickle Scofield”;

② 用 SHA-1哈希函数对“Mickle”和“Scofield”分别计算哈希值 $Hash_1$ 和 $Hash_2$,即 $Hash_1=SHA\text{-}1(Mickle)=90358491245320428 26(dec)$, $Hash_2=SHA\text{-}1(Scofield)=41668730823549184 12(dec)$;

③ 计算 $Offset_1=Hash_1 \bmod Size(D_1)$ 得到文件偏移量。假设 $Size(D_1)=1\ MByte(1048576\ bits)$, $Size(D_2)=2\ MByte(2097152\ bits)$,则 $Offset_1=90358491245320428 26 \bmod 1048576=113738$, $Offset_2$

=4166873082354918412 mod2097152=451596。

使用者只需要记住自己指定的 K_C(电影角色的名字),就可以根据规则很容易地计算出文件偏移量,而对手要尝试猜测这个偏移量是很困难的。

(3)生成隐秘文件。

① 将异或算子 X_4与加密后的文件 M进行异或运算,得到最终的隐秘文件 S,即 $S=M\oplus X_4$;

② 将文件 F的指针移动到文件起始位置,再将文件 S的数据写入,即把文件 S写入到硬盘上原来保存文件 F的位置上。

由于本方法所使用的对称加密算法和异或运算都不会改变文件的大小,因此文件 S和文件 F的大小相等,这样原来的文件 F就完全被新的隐秘文件 S覆盖。当需要使用文件 F时,可以用隐秘文件 S来恢复出原文件 F,其步骤如下:

① 用前面生成异或算子的方法,由文件 D_1和 D_2计算出异或算子 X_4;

② 将 X_4与 S进行异或运算得到 M,即 $S+X_4=M\oplus X_4\oplus X_4=M\oplus(X_4\oplus X_4)=M$;

③ 用解密算法 D和密钥 K_E对 M进行解密,得到原文件 F,即 $F=D(M, K_E)$。

由于使用者知道 D_1、D_2、生成偏移量的规则和控制密钥 K_C,因此计算 X_4是容易的。有了 X_4就可以计算出 M,而解密算法 D又是公开的,使用者在掌握 K_E的情况下由 M计算 F是容易的。所以,使用者可以很容易地从 S恢复出 F。这种方法既对原文件 F的信息进行了加密,同时又对 F的信息进行了隐写,这也是将文件 S称为隐秘文件的原因。首先,"秘"体现在使用对称加密算法 E对 F的信息进行了加密,对手在不知道 K_E的情况下要解密出 F的信息具有一定的难度。其次,"隐"体现在对手要恢复出 F,必须要计算出 X_4,而生成 X_4所用的 D_1和 D_2究竟是哪 2个文件,生成偏移量的规则和控制密钥 K_C都是保密的。因此对手如要进行暴力破解,将面临从所有的计算机文件中去确定这 2个文件和偏移量的问题,其计算量可想而知。这就相当于把 F的信息隐写到了系统文件的海量信息中去。

1.2　方法的效率和安全性

该方法可以用于在工具机及目标机上隐秘保存文件,或用于提高被传送文件的隐秘性,防止对手使用取证工具对文件进行恢复和分析。鉴于特殊的使用环境,方法的效率和安全性是最重要的特性,该方法有以下两方面优势。

(1)用很低的开销大幅度提高了破解运算量。

相对非对称加密,对称加密算法虽然具有运算速度快的优点,但在当前不断提高的计算能力和并行计算的挑战下,其安全性越来越弱,尤其是在使用短密钥的情况下更是如此。本文的方法在对称加密的基础上增加的运算只有异或运算,而计算机处理异或运算的速度是非常快的,因此该方法只在对称加密的基础上增加了很少的开销,但却大幅度提高了暴力破解的运算量。为简化问题的说明,作以下假设:

① 使用 DES128加密算法[13],并已知对称密钥(128 bits)是由使用者提供的 6位英文小写字母经过编码和哈希运算生成的;

② 计算机中有 100个普通文件可供选择,每个文件大小都是 1 kByte(1 024 bits)。

如仅使用 DES128加密,对手使用暴力破解,所有的可能性有 $O=26^6\approx2^{28}$种,这个运算量对目前的计算能力而言并不大。而使用本文提出的方法,从 100个文件中选出 2个,有 $C(100, 2)=4\,950$种可能,每个文件的偏移量有 $2^{10}=1\,024$种可能,因此总的可能性有 $O=26^6\times C(100, 2)\times2^{10}\times2^{10}\approx2^{60}$种,即提高了 2^{32}倍。而且实际情况是计算机系统中可供选择的文件远不止 100个,很多文件的大小也不止 1 kByte,随着文件数量和大小的增加,可能性数量会以指数速度增长,安全性也随之大幅度提高。

(2)保存上的安全性。

现代的取证工具可以从普通删除甚至格式化的硬盘中恢复出文件,为避免对手从硬盘上恢复出原文件 F,本方法在保存隐秘文件 S时先将文件 F的指针移动到起始位置再写入,即把 S写入到了硬盘上原来保存 F的位置。由于 F和 S大小相等,保存 S后完全覆盖了原来 F的位置,使得对手很难恢复 F。另一方面,所使用的 K_E和 K_C只在处理过程中存在于内存,并不保存在硬盘上,文件处理完成后,随着相应内存的释放就消失了。

(3)对称密钥 K_E泄漏不会导致方法失败。

传统的对称加密算法的安全性完全依赖于密钥 K_E,一旦密钥泄露,对手就可以恢复出所有用该密钥加密的文件。但在本文的方法中,要恢复出原文件 F,必须知道对称密钥 K_E、选取的 2个文件、偏移量的生成规则和控制密钥 K_C四者缺一不可。因此即使对称密钥 K_E泄露了,对手也不能恢复出原文件 F。

(4)即使猜出了某次的偏移量也无法恢复出控制密钥 K_C。

如前所述,在由控制密钥 K_C 计算偏移量时,使用了哈希函数和模运算,而这两种运算都具有单向性。也就是说,由 K_C 计算 $Hash_1$ 是容易的,由 $Hash_1$ 计算 $Offset_1$ 也是容易的;但如果要由 $Offset_1$ 计算 $Hash_1$,根据模运算的特点,有很多的可能解,而由 $Hash_1$ 反向计算 K_C 更是一个计算上不可行的困难问题。因此即便对称密钥 K_E 泄露了,对手也知道了 D_1 和 D_2 是哪2个文件,并且使用穷举法猜测出了偏移量 $Offset_1$ 和 $Offset_2$,他也不能计算出 K_C。因此他只能恢复出那些用 D_1 和 D_2 进行隐秘处理的文件,而那些选用另外2个文件进行隐秘处理的文件仍然是安全的,因为即便使用相同的 K_C,$Offset_1$ 也会随着文件 D_1 大小的不同而不同。

2 实 现

根据上述方法,我们用 VC++开发了一种适用于 Windows平台的反取证原型工具 StegEncrypt[11]。该工具是一个命令行工具,主要用于在工具机或目标机上保存重要文件而不被对手轻易解读,以及对传送的敏感文件进行隐秘处理,防止被对手的防御系统(防火墙、IDS等)分析其内容而发现入侵行为。

StegEncrypt由1个主程序模块和6个子模块组成,如图2所示。

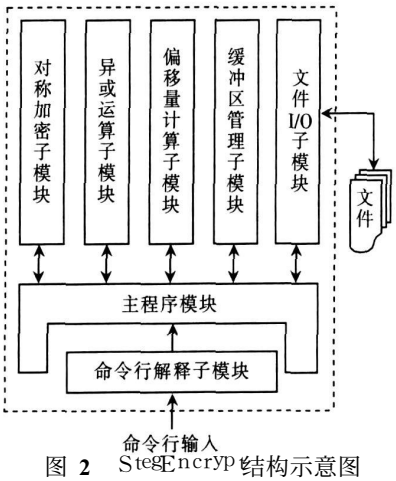


图2 StegEncrypt结构示意图

(1)主程序模块调用各子模块提供的接口来完成对文件的隐秘处理和保存;

(2)命令行解释子模块负责解释使用者的命令行输入和接收输入的参数。命令行输入的格式为:StegEncrypt<-e/-d sourcefile><-f filename1><-f2 filename2><-k1 K_E><-k2 K_C>,其中 sourcefile是要进行隐秘(或解密)处理的 F(或S)的文件名,filename1和 filename2分别是 D_1 和 D_2 的文件名,K_E 是对称密钥,K_C 是控制密钥;

(3)对称加密子模块用 DES128算法和 K_E 对相关数据进行加密或解密;

(4)异或运算子模块负责对相关数据进行异或运算;

(5)偏移量计算子模块用 SHA-1函数和 Mod运算对控制密钥 K_C 进行处理,生成偏移量 $Offset_1$ 和 $Offset_2$;

(6)文件 I/O模块负责把指定的文件数据读入缓冲区或把缓冲区的数据写回到指定文件位置;

(7)缓冲区管理子模块负责缓冲区的动态分配和释放,这些缓冲区用于存放读出的文件数据和计算的中间数据。

StegEncrypt的工作流程如图3所示。

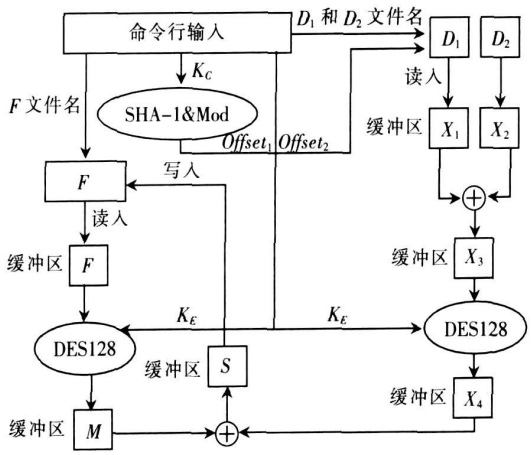


图3 StegEncrypt工作流程图

3 结 论

对文件进行加密或将其隐写到图片等载体文件中,使其不容易被对手发现或者理解是反取证技术的常用手段。但加密时是使用长的随机密钥以提高安全性,还是使用短的有意义密钥以方便记忆和使用是一对矛盾;另一方面,载体文件大小决定了可隐写信息的多少,当需要隐写的文件较多时,在短时间内找到足够多的合适载体文件并不容易。为克服这些弱点,本文提出了一种结合加密和隐写技术对文件进行隐秘处理的方法,该方法在对称加密的基础上,用异或算子对加密后的文件进行了二次处理。异或算子的值取决于使用者选取的2个普通文件、文件偏移量的生成规则和控制密钥。该方法的优点主要有:(1)原来对手一旦使用基于字典的猜测等手段获取了密钥就可以解密文件,而使用本文方法后,对手要解密文件需要掌握更多的知识(对称密

钥、使用的是哪 2个普通文件和文件的偏移量),因此对称密钥即使泄露,被加密文件仍然是安全的;(2)在增加很少开销的情况下,极大地增加了可能性空间,因此即使使用短密钥,仍然有较强的对抗暴力破解的能力;(3)使用的保存方法完全覆盖了原文件的存储空间,使其可以较好地对抗取证工具所用的数据恢复技术;(4)对选用的 2个普通文件没有特殊要求,提高了其适用性;(5)使用的对称加密和异或运算的计算速度都很快,适合反取证应用环境的要求。根据该方法,我们开发出了原型工具 StegEncrypt。

该方法目前的不足之处在于将处理后的隐秘文件写入原文件时,会改变原文件的时间属性,而这可能会引起对手的注意并成为证据。在下一步的研究中,我们将对该方法和 StegEncrypt工具进行改进,使其可以首先提取出原文件的时间属性(包括创建时间、最后访问时间和修改时间),然后再对其进行读写操作,最后再将原文件的时间属性恢复到读写操作之前的状态。

参考文献:

[1] 丁丽萍,王永吉.计算机取证的相关法律技术问题研究[J].软件学报,2005,16(2):260-275.
DING Liping, WANG Yongji. Study on relevant law and technology issues about computer forensics[J]. Journal of Software, 2005, 16(2): 260-275.

[2] 王玲,钱华林.计算机取证技术及其发展趋势[J].软件学报,2003,14(9):1635-1644.
WANG Ling, QIAN Hualin. Computer forensics and its future trend[J]. Journal of Software, 2003, 14(9): 1635-1644.

[3] BERNATO S. The rise of anti-forensics[EB/OL]. http://www.csconline.com/article/print/221208.

[4] 张有东,王建东.反计算机取证技术研究[J].河海大学学报:自然科学版,2007,35(1):104-107.
ZHANG Youdong, WANG Jiandong. Research on computer anti-forensics[J]. Journal of Hohai University: Natural Sciences, 2007, 35(1): 104-107.

[5] 殷联甫.计算机反取证技术研究[J].计算机系统应用,2005(10):46-49.
YIN Lianpu. Research on computer anti-forensics[J]. Applications of the Computer Systems, 2005(10): 46-49.

[6] SARTIN B. Anti-forensics: distorting the evidence[J]. Computer Fraud & Security, 2006(5): 4-6.

[7] BERGHEL H. Hiding data, forensics, and anti-forensics[J]. Communications of The ACM, 2007, 50(4): 15-20.

[8] FORTE D, Richard power A. tour through the realm of anti-forensics[J]. Computer Fraud & Security, 2007(6): 18-20.

[9] FOGIE S. Anti-forensics[EB/OL]. http://www.informit.com.

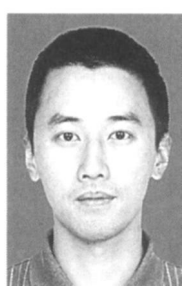
[10] 陈祖义,龚俭,徐晓琴.计算机取证的工具体系[J].计算机工程,2005,31(5):162-164.
CHEN Zuyi, GONG Jian, XU Xiaoqin. Tool set of computer forensics[J]. Computer Engineering, 2005, 31(5): 162-164.

[11] LEE S S, CHANG K Y, LEE D, et al. A new anti-forensic tool based on a simple data encryption scheme[C]// Future generation communication and networking. Piscataway: IEEE, 2007.

[12] HILLEY S. Anti-forensics with a small army of exploits[J]. Digital Investigation, 2007(4): 13-15.

[13] WILLIAM S.密码编码学与网络安全:原理与实践[M].第 2版.北京:电子工业出版社,2001:52-57.
WILLIAM S. Cryptography and Network Security: Principles and Practice[M]. 2ed. Beijing: Publishing House of Electronics Industry, 2001: 52-57.

作者简介:



王　灿(1977-),男,四川成都人。电子科技大学计算机科学与工程学院博士研究生。主要研究方向为信息安全。

秦志光(1956-),男,四川隆昌人。电子科技大学计算机科学与工程学院教授,博士生导师。主要研究方向为信息安全。