

# 一种基于文件型病毒的粒子群检测方法

朱俚治

(南京航空航天大学 信息中心,江苏 南京 210016)

**摘要:**当今计算机技术的快速发展,使得计算机病毒的智能性日益突出,所以使用单一的传统检测技术在病毒检测过程中,漏检和误检的比例明显上升。为了应对目前病毒体现出的智能性,反病毒技术也必须采用相应的智能技术。文中在查阅了相关资料后,提出一种基于粒子群的病毒检测技术。首先对一个未知属性的程序依照病毒的属性来判断该程序是否是病毒程序。在确定该程序是病毒的前提条件下,再通过粒子群的方法来判断该病毒所属种类。

**关键词:**病毒;粒子群;属性;传染性;智能性

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2014)12-0128-05

doi:10.3969/j.issn.1673-629X.2014.12.030

## A Detection Method for Particle Swarm Based on File Type Virus

ZHU Li-zhi

(Information Center, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract:** With the rapid development of the computer technology, the intelligence of today's computer viruses have become increasingly prominent, so the use of traditional detection technology of single in virus detection process, missed and false detection ratio have increased significantly. In order to cope with the current virus intelligence, anti-virus technology must adopt corresponding intelligent technology. Based on the inspection of the relevant information, put forward a kind of particle swarm based virus detection technology. In this paper, first to an unknown attribute procedures, determine the program whether is the virus program in accordance with the virus attributes. Under the condition of determining the program is the virus, then the particle swarm method is used to determine the category for the virus.

**Key words:** virus; particle swarm; property; contagious; intelligence

## 0 引言

自从第一个病毒出现以来,病毒的更新速度总比反病毒技术来得快,因此为了减少杀毒软件对未知程序的误判,使得检测算法具有一定智能性是十分必要的<sup>[1]</sup>。在当今网络中存在各种病毒,这些病毒对网络用户都有攻击性,如果要防止网络病毒对网络用户的破坏和攻击,必须检测出程序的属性。区分病毒程序和正常程序有多种方法,但这些方法中都有不足之处,因此文中从病毒属性的角度出发,提出一种检测病毒的方法。

病毒程序分析员通过对计算机病毒的分析可知,传染性是计算机病毒的一个重要属性。一个程序是否具有传染性是判断该程序是不是计算机病毒的重要条件<sup>[2]</sup>。如果某种程序具有传染性,那么该程序就是病毒。如果该程序不具备传染性,那么该程序就是非病

毒程序<sup>[2]</sup>。

尽管不同种类病毒的属性存在差异,但新病毒和旧病毒都是病毒,因此它们都具有共同属性。为此文中从程序属性角度提出一种具有一定智能性的病毒检测方法,该方法能够最终判断未知程序是不是病毒。

粒子群算法是粒子寻求最优解的智能算法,算法的智能性体现在搜索过程中粒子之间的相互协作以及粒子不断调整自身的飞行速度和方向,以此来找到最优解。由病毒构成的种群的属性符合自然界的生物种群属性,而粒子群算法正是模仿了生物种群的特性,所以粒子群算法能够应用到检测病毒这一方面。

## 1 病毒技术简介

### 1.1 病毒的定义

病毒的定义如下:计算机病毒是一段附着在其他

程序上的可以自我繁殖的程序代码,复制后生成的新病毒同样具有感染其他程序的功能<sup>[3-4]</sup>。因此,根据定义可以认为计算机病毒是一种有破坏性的程序。

1.2 病毒的传染性

计算机病毒传染模块具有以下两个重要特性:

- (1)寄生性:病毒程序寄生于宿主程序中,宿主程序执行时病毒程序首先被执行,即具有寄生性<sup>[3-4]</sup>。
- (2)传染性:病毒程序能够自我复制到其他宿主程序中,即具有传染性。且在被感染的程序运行时,病毒程序能够进一步感染其他目标程序,从而使病毒得到传播。能够自我复制是病毒最为本质的特性<sup>[3-4]</sup>。

2 粒子群的几个重要基本概念

种群:种群是群体的概念,最常见的种群有鸟类、鱼群。

粒子:粒子是种群中的个体,鸟群和鱼群中的个体就是种群中的粒子。

适应度函数:适应度函数能够为种群的粒子的搜索方向和寻找最优解提供依据,加快种群的收敛速度<sup>[5-6]</sup>。

目标函数:目标函数的目的是使得粒子在搜索空间寻找最优解,该函数的期望值就是粒子的最优解<sup>[5-6]</sup>。

粒子间的协作:如果粒子群中某个粒子找到了最优解,那么其他粒子可以通过粒子之间的相互学习来调整粒子飞行的速度和方向,使得粒子找到最优解。粒子在搜索空间中不会重复搜索同一个位置,这将加快种群收敛速度<sup>[6-7]</sup>。

3 病毒检测函数和算法

3.1 病毒程序与非病毒程序的检测

程序属性判定函数:

$$y = f(x) = \frac{\text{未知属性程序的原子程序属性值}}{\text{文本病毒传染功能模块的属性值}}$$

3.2 程序属性判定

计算机病毒在结构上一般分为三个功能模块:感染机制、触发机制和有效载荷。这里定义的感染机制就包含了病毒自我复制部分<sup>[3,8]</sup>。

计算机病毒传染模块负责传染,而病毒的复制和寄生是传染过程中的两个独立子过程,因此复制和寄生是传染模块的两个子属性<sup>[8-10]</sup>。

提取病毒传染模块中的两个模块:寄生程序和复制程序。计算出寄生程序和复制程序的属性值:寄生程序的属性值用符号记为  $j$ ;复制程序的属性值用符号记为  $t$ 。

实现程序最基本功能的程序称为原子程序。这些

程序实现整个程序的最基本功能。某个未知程序中有  $n$  个原子程序,  $n$  个原子程序具有的属性个数为  $n_1, n_2, \dots, n_n$ , 其属性值分别为  $n'_1, n'_2, \dots, n'_n$ 。

1)未知属性程序的原子程序属性与文本病毒寄生程序属性进行比较。

属性判定函数:

$$y_n = f(x) = \frac{n'_n}{j}; \text{ 令 } x = n'_n$$
  
$$y_1 = \frac{n'_1}{j}, y_2 = \frac{n'_2}{j}, \dots, y_n = \frac{n'_n}{j}$$

讨论:

(1)当原子程序属性值  $n'_n$  都大于寄生程序属性值  $j$ , 这时  $n'_n$  值越小, 则  $y_n = f(x)$  的值越接近于 1, 因此这时  $y_n = f(x)$  取最小值。

(2)当原子程序属性值  $n'_n$  部分大于寄生程序的属性值  $j$ , 部分小于寄生程序的属性值  $j$ , 则需要讨论:

①当  $y_n = f(x) = \frac{n'_n}{j} > 1$ , 这时  $n'_n$  值越小, 则  $y_n = f(x)$  的值越接近于 1, 因此这时  $y_n = f(x)$  取最小值。

②当  $y_n = f(x) = \frac{n'_n}{j} < 1$ , 这时  $n'_n$  值越大, 则  $y_n = f(x)$  的值越接近于 1, 因此这时  $y_n = f(x)$  取最大值。

(3)当原子程序属性值  $n'_n$  都小于寄生程序的属性值  $j$ , 这时  $n'_n$  值越大, 则  $y_n = f(x)$  的值越接近于 1, 因此这时  $y_n = f(x)$  取最大值。

2)未知属性程序的原子程序属性与文本病毒复制属性进行比较。

属性判定函数:

$$y_n = g(x) = \frac{n'_n}{t}; \text{ 令 } x = n'_n$$
  
$$y_1 = \frac{n'_1}{t}, y_2 = \frac{n'_2}{t}, \dots, y_n = \frac{n'_n}{t}$$

讨论:

(1)如果原子程序的属性值  $n'_n$  都大于复制程序属性值  $t$ , 这时  $n'_n$  的值越小,  $y_n = g(x)$  的值越接近于 1, 因此这时  $y_n = g(x)$  取最小值。

(2)如果原子程序属性值  $n'_n$  部分大于复制程序的属性值  $t$ , 部分小于复制程序的属性值  $t$ , 则需要讨论:

①当  $y_n = g(x) = \frac{n'_n}{t} > 1$ , 这时  $n'_n$  的值越小, 则  $y_n = g(x)$  的值越接近于 1, 因此这时  $y_n = g(x)$  取最小值。

②当  $y_n = g(x) = \frac{n'_n}{t} < 1$ , 这时  $n'_n$  的值越大, 则  $y_n = g(x)$  的值越接近于 1, 因此这时  $y_n = g(x)$  取最大值。

(3)如果原子程序的属性值  $n'_n$  都小于复制程序的属性值  $t$ , 这时  $n'_n$  的值越大, 则  $y_n = g(x)$  值越接近于

1, 因此这时  $y_n = g(x)$  取最大值。

结论: 如果有  $y_n = f(x) - g(x) \approx 0$ , 则  $f(x) \approx 1$  并且  $g(x) \approx 1$ 。

根据以上的分析和讨论, 得出的结论是某种未知属性程序的两种属性值十分接近于病毒传染性功能中的两个模块属性值: 寄生性属性值和复制性属性值。

### 3.3 未知程序属性的判定

病毒的传染功能中有两个子程序: 寄生程序和复制程序。

一个未知程序由某些子程序组成,  $n$  个子程序有  $n$  个属性。如果未知属性程序中的某些子程序属性值十分接近于病毒的寄生性程序和复制性程序的属性值, 则任何一个实体与自身的属性存在以下联系。由于属性跟属性值有着严格的对应关系, 有什么类的属性就有什么类的属性值<sup>[6]</sup>。世界上不存在没有值的属性, 也不存在不指向任何属性的属性值<sup>[11]</sup>。

由于属性值能够反映实体的属性, 因此如果某个程序具有的原子程序的属性值十分接近病毒传染模块中的寄生程序和复制程序这两个程序的属性值, 那么可以推断这个未知属性的程序为病毒程序。

## 4 粒子群算法在检测病毒上的应用

### 4.1 粒子群算法简介

粒子群算法是粒子寻找最优解的算法。由于粒子在寻找最优解时能够相互协作, 因此当种群中的某个粒子找到了最优解, 那么与其他粒子通过相互协作同样能够使其他粒子也找到最优解<sup>[12-13]</sup>。

为了使粒子在搜索空间中较快地找到最优解, 算法通过适应函数来控制 and 调整种群中每个粒子的自身飞行的方向和速度。如果粒子在搜索空间中找到了十分符合目标函数的期望值, 则此时粒子也就找到了最优解<sup>[12-14]</sup>。

### 4.2 粒子群在病毒上的应用

由于具有病毒属性的粒子群与通常的粒子群十分相似, 因此这里可以将粒子群的算法在检测病毒方面上进行应用。文中将所有病毒属性的粒子设为一个种群, 那么这些具有病毒属性的未知程序就是种群中的粒子。

在具有病毒属性的粒子群中, 每个粒子都具有自身固有的属性。尽管不同属性的粒子存在差异, 但这些差异是很小的。病毒的重要特性是传染性, 不同病毒的传染模块属性是不同的, 所以在文中将所有病毒传染模块的属性提取出来, 把这些病毒的传染模块组成一个粒子群。

现在有若干种未知种类的病毒, 将这些未知种类病毒的传染模块取出组成求解粒子群, 而这些粒子都

要在搜索空间中找到自己的最优解。为了使未知病毒种群中的粒子找到自己的最优解, 此外还需构建一个搜索空间, 使得每一个粒子在搜索空间中都能找到最优解。为了加快粒子群的收敛速度, 将已知病毒的种类按照一定的规模组成若干个搜索空间。使得求解空间中的粒子在搜索空间中都能较快地找到最优值。

基于粒子具有上述的特点, 为了将粒子群技术应用在病毒检测的技术上, 在这里需要建立两个函数: 目标函数和适应函数。这两个函数具体形式如下:

$$(1) \text{ 目标函数: } y = f(x) = \frac{n}{m} =$$

某种未知文件型病毒传染性模块属性值  
某种已知文件型病毒传染性模块属性值  $\approx 1$ 。

目标函数的作用是描述粒子寻找最优解的衡量标准。

$$(2) \text{ 适应函数: } y = h(x) = \lim_{n \rightarrow m} (1 -$$

某种未知文件型病毒传染性模块属性值  
某种已知文件型病毒传染性模块属性值)  $= 0$ 。

适应函数的作用是调整粒子飞行的方向, 使得粒子的飞行逐步接近最优解。

### 4.3 求解空间粒子群和搜索空间粒子群的划分

(1) 求解空间粒子种群的划分。

从求解空间中取出一个粒子, 求解空间中剩余粒子个数为  $n_1, n_2, \dots, n_n$ ; 其属性值分别为  $n'_1, n'_2, \dots, n'_n$ 。

将求解空间中的某个粒子的属性值与求解空间中所有粒子的属性值进行比较。

属性判定函数:

$$g(x) = \left| 1 - \frac{\text{某个求解空间中的粒子属性值}}{\text{求解空间中所有粒子的属性值}} \right|; \text{ 令}$$

$x = j$

如果粒子  $j$  与求解空间中的粒子属性的比较有以下结论:

$$g(x) = \left| 1 - \frac{j}{n_1} \right| \approx 0, g(x) = \left| 1 - \frac{j}{n_2} \right| \approx 0, \dots,$$

$$g(x) = \left| 1 - \frac{j}{n_n} \right| \approx 0$$

则将求解空间中这  $n$  个粒子组成一个子种群, 并且将这  $n$  个粒子从求解空间种群分离出去, 组成一个独立的求解空间  $A_1$ 。

再在剩余求解空间中依次取出粒子重复以上的过程(1), 直到将求解空间种群中的粒子划分为若干子种群  $A_1, A_2, \dots, A_n$ 。

(2) 搜索空间粒子种群的划分。

从搜索空间中取出一个粒子, 求搜索空间中剩余粒子个数为  $n_1, n_2, \dots, n_n$ ; 其属性值分别为  $n'_1, n'_2, \dots, n'_n$ 。

将搜索空间中的某个粒子的属性值与搜索空间中所有粒子的属性值进行比较。

属性判定函数:

$$T(x) = \left| 1 - \frac{\text{某个搜索空间中的粒子属性值}}{\text{搜索空间中所有粒子的属性值}} \right| ; \text{令}$$
$$x = j$$

如果粒子  $j$  与搜索空间中的粒子属性的比较有以下结论:

$$T(x) = \left| 1 - \frac{j}{n_1} \right| \approx 0, T(x) = \left| 1 - \frac{j}{n_2} \right| \approx 0, \cdots,$$
$$T(x) = \left| 1 - \frac{j}{n_n} \right| \approx 0$$

则将搜索空间中这  $n$  个粒子组成一个子种群,并且将这  $n$  个粒子从搜索空间种群分离出去,组成一个独立的搜索空间  $B_1$ 。

再在剩余搜索空间中依次取出粒子重复以上的过程(2),直到将搜索空间种群中的粒子划分为若干独立的搜索空间种群  $B_1, B_2, \cdots, B_n$ 。

4.4 粒子群技术在检测病毒上的实现

在子种群  $A_1, A_2, \cdots, A_n$  中依次选出一个种群  $A_n$ , 在种群  $A_n$  中提取粒子  $a$ , 粒子  $a$  的属性值为  $a'$ 。

在子种群  $B_1, B_2, \cdots, B_n$  中依次选出一个种群  $B_n$ , 种群  $B_n$  中有粒子数  $b_1, b_2, \cdots, b_n$ , 粒子群的属性值分别为  $b'_1, b'_2, \cdots, b'_n$ 。

(1) 目标函数:  $y = f(x) = \frac{n}{m} =$

某种未知文件型病毒传染性模块属性值  
某种已知文件型病毒传染性模块属性值  $\approx 1$ , 文中令  $y$

$$= f(x) = \frac{n}{m} = \frac{a'}{b_n}。$$

粒子  $a$  寻找最优解过程如下:

$$y = f(x) = \frac{a'}{b_1}, y = f(x) = \frac{a'}{b_2}, \cdots, y = f(x) = \frac{a'}{b_n}$$

如果存在  $y = f(x) = \frac{a'}{b_n} \approx 1$ , 则粒子  $a$  找到了最优解。

如果粒子群  $A$  中的粒子  $a$  找到了最优解, 那么粒子群  $A$  中的其他粒子根据粒子群中粒子协作和竞争机制, 同样能够很快地找到自身的最优解。根据相同的理由, 种群  $A_1, A_2, \cdots, A_n$ , 都能很快找到各自的最优解。

(2) 适应函数:  $y = h(x) = \lim_{n \rightarrow m} (1 - \frac{\text{某种未知文件型病毒传染性模块属性值}}{\text{某种已知文件型病毒传染性模块属性值}}) = 0$ 。

适应函数的作用是调整粒子飞行的方向, 使得粒子的飞行逐步接近最优解。

①当  $y = h(x) = \lim_{n \rightarrow m} (1 -$

$\frac{\text{某种未知文件型病毒传染性模块属性值}}{\text{某种已知文件型病毒传染性模块属性值}}) > 0$ , 有  $1 - \frac{n}{m} > 0$ 。

如果  $1 - \frac{n}{m} > 0$  时, 有以下讨论:

由  $1 - \frac{n}{m} > 0$ , 可得  $n < m$ , 但目标函数中  $y = f(x)$

$$= \frac{n}{m} \approx 1, \text{需要 } n \approx m。$$

为了使  $n$  无限接近于  $m$ , 达到  $n \approx m$  的目的, 因此在搜索空间中需要寻找属性值比粒子  $m$  属性值大的粒子, 这样才能找到粒子的最优解。

②当  $y = h(x) = \lim_{n \rightarrow m} (1 - \frac{\text{某种未知文件型病毒传染性模块属性值}}{\text{某种已知文件型病毒传染性模块属性值}}) < 0$ , 有  $1 - \frac{n}{m} < 0$ 。

如果  $1 - \frac{n}{m} < 0$  时, 有以下讨论:

由  $1 - \frac{n}{m} < 0$ , 可得  $n > m$ , 但目标函数中  $y = f(x)$

$$= \frac{n}{m} \approx 1, \text{需要 } n \approx m。$$

为了使  $n$  无限接近于  $m$ , 达到  $n \approx m$  的目的, 因此在搜索空间中需要寻找属性值比粒子  $m$  属性值小的粒子, 这样才能找到粒子的最优解。

③当  $y = h(x) = \lim_{n \rightarrow m} (1 - \frac{\text{某种未知文件型病毒传染性模块属性值}}{\text{某种已知文件型病毒传染性模块属性值}}) = 0$ , 有  $1 - \frac{n}{m} = 0$ 。

如果  $1 - \frac{n}{m} = 0$ , 则此时粒子在搜索空间中找到了最优解。

结论: 根据以上的推理和分析, 可以知道由粒子群得出的最优解, 就可以判定某种未知种类文本程序属于哪一种病毒程序。

4.5 粒子群技术检测病毒的算法

- (1) 初始化搜索空间中的粒子群。
- (2) 初始化求解空间中的粒子群。
- (3) 将未知种类病毒组成的种群, 划分成若干个子种群, 并且将子种群中的粒子数目控制在 30 以内。
- (4) 同样将搜索空间中的粒子种群划分为若干个子种群, 并且将子种群中的粒子数目控制在 30 以内。
- (5) 某个病毒组成的种群中的粒子在搜索空间中寻找最优解。

(6)使用适应函数对粒子飞行的方向和速度不断进行调整。

(7)如果粒子群中某些粒子在第一次搜索中未找到最优解,将这些粒子提取出来组成一个新粒子群。

(8)将新组成的粒子群在下一个搜索空间中继续寻找最优解。

## 5 结束语

由于智能病毒检测算法在当今检测病毒方面的重要性,因此在文中提出了一种基于粒子群的病毒检测方法,该方法具有一定的智能性。文中首先介绍了未知程序和病毒属性以及粒子群的基本概念之后,通过比较未知程序的属性和病毒的特有属性—传染性之间的关系,来判断该未知的程序是否属于病毒程序。在这基础上将判定为病毒的未知程序,再使用粒子群智能算法对该病毒程序进行种类的判定。最后给出了粒子群检测病毒的算法。

### 参考文献:

- [1] 高海兵,周 驰,高 亮. 广义粒子群优化模型[J]. 计算机学报,2005,28(12):1980-1987.
- [2] 刘春卉. 属性值与属性特征语义语法差异考察—兼谈相关歧义结构[J]. 汉语学习,2008(3):43-47.
- [3] 王倍昌. 走进计算机病毒[M]. 北京:人民邮电出版社,2010.
- [4] 秦志光,张凤荔,刘 峤. 计算机病毒原理与防范技术

[M]. 北京:科学出版社,2012.

- [5] 朱丽莉,杨志鹏,袁 华. 粒子群优化算法分析及研究进展[J]. 计算机工程与应用,2007,43(5):24-27.
- [6] Salman A, Ahmad I, Al-Madani S. Particle swarm optimization for task assignment problem[J]. Microprocessors and Microsystems,2002,26:363-371.
- [7] Higashi N, Iba H. Particle swarm optimization with Gaussian mutation[C]//Proceedings of the 2003 IEEE swarm intelligence symposium. [s. l.]:IEEE Press,2003:72-79.
- [8] 刘 俭,唐朝京,张森强. 一种计算机病毒的检测方法[J]. 计算机工程,2004,30(6):127-129.
- [9] 郑 晶,王春生. 新一代病毒检测技术研究[J]. 网络安全技术与应用,2009(6):23-25
- [10] 韩兰胜. 计算机病毒原理与防治技术[M]. 武汉:华中科技大学出版社,2010.
- [11] 郭剑毅,李 真,余正涛,等. 领域本体概念实例、属性和属性值的抽取及关系预测[J]. 南京大学学报(自然科学版),2012,48(4):383-389.
- [12] 潘 峰,李位星,高 琪. 粒子群优化算法与多目标优化[M]. 北京:北京理工大学出版社,2013.
- [13] Bergh F, Engelbrecht A P. Training product unit networks using cooperative particle swarm optimizers[C]//Proceedings of international joint conference on neural networks. Washington:[s. n.],2001:126-131.
- [14] Harmer P K, Williams P D, Gansch G H, et al. An artificial immune system architecture for computer security applications[J]. IEEE Transactions on Evolutionary Computation,2002,6(3):252-280.

(上接第 127 页)

- [3] Tsunoo Y, Tsujihara E, Shigeri M, et al. Impossible differential cryptanalysis of CLEFIA[J]. LNCS,2008,5086:398-411.
- [4] Li Yanjun, Wu Wenling, Zhang Lei. Improved integral attacks on reduced-round on CLEFIA block cipher[J]. LNCS,2012,7115:28-39.
- [5] Information technology—security techniques—lightweight cryptography—part 2: block ciphers[S]. Switzerland:ISO/IEC,2012.
- [6] Aoki K, Ichikawa T, Kanda M, et al. Camellia: a 128-bit block cipher suitable for multiple platforms—design and analysis[C]//Proc of SAC 2001. Waterloo, Ontario, Canada:[s. n.],2000:39-56.
- [7] Axel Y P B. Lightweight cryptography: cryptographic engineering for a pervasive world[D]. Bochum: Ruhr-University Bochum,2009.
- [8] Serf P. The degrees of completeness, of avalanche effect, and of strict avalanche criterion for mars, rc6, rijndael, serpent, and two fish with reduced number of rounds[EB/OL]. 2000-02-03. <http://www.cosic.esat.kuleuven.ac.be/nessie/reports/phase1/sagwp3-003.pdf>.
- [9] Biham E, Shamir A. Differential cryptanalysis of DES-like

cryptosystems[J]. Journal of Cryptology,1991,4(1):3-72.

- [10] Matsui M. Linear cryptanalysis of DES cipher[J]. LNCS,1994,765:386-397.
- [11] Kanda M, Takashima Y, Matsumoto T, et al. A strategy for constructing fast round functions with practical security against differential and linear cryptanalysis[C]//Proc of SAC 1999. Kingston, Ontario, Canada:[s. n.],1999:264-279.
- [12] Kanda M. Practical security evaluation against differential and linear cryptanalysis for feistel ciphers with SPN round function[C]//Proc of SAC 2001. [s. l.]:[s. n.],2001:324-338.
- [13] Hong S, Lee S, Lim J, et al. Provable security against differential and linear cryptanalysis for the SPN structure[J]. LNCS,2001,1978:273-283.
- [14] Daemen J, Rijmen V. The design of Rijndael: AES—the advanced encryption standard (information security and cryptography)[M]. Berlin:Springer-Verlag,2002.
- [15] Biham E, Biryukov A, Shamir A. Cryptanalysis of skipjack reduced to 31 rounds using impossible differential[J]. Journal of Cryptology,2005,18(4):291-311.
- [16] Kim J, Hong S, Sung J, et al. Impossible differential cryptanalysis for block cipher structures[J]. LNCS,2003,2904:82-96.

# 一种基于文件型病毒的粒子群检测方法

作者：[朱偲治, ZHU Li-zhi](#)  
作者单位：[南京航空航天大学 信息中心, 江苏 南京, 210016](#)  
刊名：[计算机技术与发展](#)   
英文刊名：[Computer Technology and Development](#)  
年, 卷(期): 2014(12)

引用本文格式: [朱偲治, ZHU Li-zhi](#) [一种基于文件型病毒的粒子群检测方法](#)[期刊论文]-[计算机技术与发展](#)  
2014(12)