第35卷 第10期 计 算 机 工 程 2009年5月
Vol.35 No.10 Computer Engineering May 2009
·安全技术· 文章编号：1000—3428(2009)10—0164—04 文献标识码：A 中图分类号：TP391.41

基于DCT的图像置乱程度评价方法

瞿新南，孙秋艳
(常州信息职业技术学院计算机与软件学院，常州 213164)

摘 要：针对当前出现的各种各样的置乱算法，为更好地衡量这些算法对不同图像的置乱程度，利用DCT系数的良好特性，结合人眼视觉掩蔽特点，提出一种新的描述图像置乱程度的评价方法。实验表明，该方法与人的主观评价相接近，能较好地反映同一图像和不同图像的置乱效果。

关键词：图像置乱；离散余弦变换；骑士巡游变换；置乱度

Image Scrambling Degree Evaluation Method Based on DCT

QU Xin-nan, SUN Qiu-yan
(Dept. of Computer and Software, Changzhou College of Information Technology, Changzhou 213164)

【Abstract】There appears a lot of scrambling methods, but no one is the best. In order to measure the scrambling degree of these algorithms to different images better, this paper proposes a new method for accessing the image scrambling degree, which utilities the favorable performance of Discrete Cosine Transform(DCT) coefficients and the feature of masking of human vision system. Experiments show that the method correlates with subjective assessment and its results can reflect scrambling effect well of the same image and different images.

【Key words】image scrambling; Discrete Cosine Transform(DCT); knight-tour transform; scrambling degree

1 概述

置乱技术作为一种常见的图像加密方法，是一项值得深入研究的课题。已有很多图像置乱的方法，如Arnold变换[1]、骑士巡游变换[2]等，这些变换置乱图像后的直观效果各不相同。目前评价图像置乱效果的方法主要分为2类：主观评价方法和客观评价方法。主观评价方法是由人眼直接对置乱图像进行评价，具有简单直观等特点，但会有很多主观因素影响评价结果。因此，对置乱程度客观评价的研究有着重要的理论和实际意义。文献[1]用概率论知识给出了灰度差分熵置乱度的定义，文献[3]提出了用不动点、像素移动的平均距离、自然序、汉明相关性等方法来进行图像置乱程度的评价。文献[4]提出了基于相邻灰度差的衡量方法，这些评价方法各有特点，但至今没有一种十分理想的置乱程度评价方法。

本文结合现有文献，分析了图像离散余弦变换(Discrete Cosine Transform, DCT)系数的分布特点，在考虑人类视觉系统特性的基础上提出了一种新的置乱度算法。

2 置乱变换及已有评价方法

2.1 基于Arnold变换的置乱算法

Arnold变换(Cat mapping)[1]是Arnold在遍历理论的研究中提出的一类裁剪变换，二维Arnold变换算法简单且具有周期性，是图像加密的经典方法之一。一幅大小为256×256的图像恢复原图的迭代周期为192，迭代次数较低(在原图附近)或较高(接近恢复周期)时，Arnold算法可看出图像的大致内容。经6次变换后，图像可被充分置乱，视觉上呈现为杂乱无序的，类似于噪声的分布，且运算也较为简单。

2.2 基于骑士巡游变换的置乱算法

2.2.1 传统置乱算法

文献[2]描述了基于骑士巡游变换的传统置乱算法的概念、原理和具体加密方法，所用加密矩阵皆为狭义巡游矩阵。目前广义Hamilton路径也在不断的研究实验之中[5]，这大大增加了图像加密的密钥个数。行列平移交换是矩阵最基本的变换，所以本文也对原巡游矩阵做了行列变换，产生一系列的类生矩阵。这些矩阵各有特点且又不遵循一定的规则，没有原矩阵和变换方法就不能再现。用它们来加密图像效果好、安全性高，同时可以建立各巡游矩阵的类生子密钥库。

一个广义10×10棋盘的骑士巡游路线如矩阵 $\boldsymbol{T}$ 所示[6]，称为巡游矩阵，将其做行列变换产生的类生矩阵之一如 $\boldsymbol{T}1$。

$$\boldsymbol{T}=\begin{bmatrix}1&10&67&46&95&56&29&8&23&50\\12&65&4&69&48&27&6&53&86&25\\45&96&57&2&9&22&51&94&55&30\\68&47&100&11&66&85&24&49&28&7\\3&70&13&64&5&54&87&26&21&52\\58&39&44&97&60&91&78&31&84&93\\37&16&73&42&99&80&33&18&75&82\\14&63&90&71&40&35&20&61&88&77\\43&98&59&38&17&74&83&92&79&32\\72&41&36&15&62&89&76&81&34&19\end{bmatrix},\boldsymbol{T}1=\begin{bmatrix}1&58&67&44&95&60&29&78&23&84\\12&37&4&73&48&99&6&33&86&75\\45&14&57&90&9&40&51&20&55&88\\68&43&100&59&66&17&24&83&28&79\\3&72&13&36&5&62&87&76&21&34\\10&39&46&97&56&91&8&31&50&93\\65&16&69&42&27&80&53&18&25&82\\96&63&2&71&22&35&94&61&30&77\\47&98&11&38&85&74&49&92&7&32\\70&41&64&15&54&89&26&81&52&19\end{bmatrix}$$

$\boldsymbol{T}$ 及其类生矩阵 $\boldsymbol{T}1$ 对原图进行置乱的结果如图1所示。

(a)原图 (b)巡游矩阵置乱效果 (c)类生矩阵置乱效果

图1 巡游矩阵及其类生矩阵的置乱效果

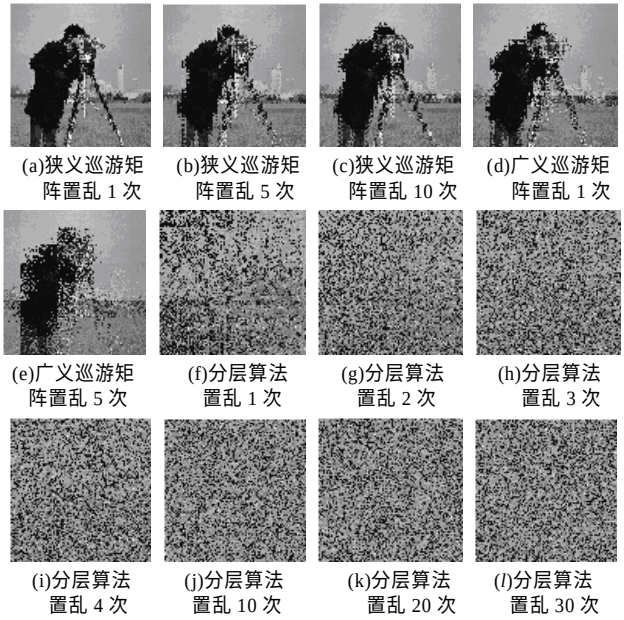
作者简介：瞿新南(1980－)，女，硕士研究生，主研方向：图形与图像处理；孙秋艳，硕士研究生

收稿日期：2008-11-10 **E-mail：**quxn666@163.com

2.2.2　分层置乱算法

分析文献[2]及图 1 可知，不管是狭义巡游矩阵、广义巡游矩阵还是它们的类生矩阵，对图像进行像素位置置乱后，图像只是模糊了细节。为了将图像像素彻底打乱同时提高置乱速度，提出了分块分层置乱算法。先将图像分成较大的块，用巡游矩阵对此大像素块进行块块位置置乱，这样就把图像的像素搬移到相对原位置较远的地方，克服了传统置乱只是在原像素附近进行位置改变的缺点。然后用传统置乱将大像素块置乱的块边界打乱，防止“块效应”的出现。最后将图像分成相对较小的块进行小像素块置乱，以使图像的相邻像素分开得足够远。用此算法加密图像，不仅速度快，而且置乱效果好。

选用大小为 256×256 的图像做置乱处理，结果如图 2 所示。图 2(a)~图 2(c)是传统算法采用狭义巡游矩阵不同次数的置乱结果，图 2(d)、图 2(e)是传统算法采用广义巡游矩阵不同次数的置乱结果，图 2(f)~图 2(*l*)是分层算法采用 2 种矩阵不同次数的置乱结果。



(a)狭义巡游矩阵置乱 1 次　(b)狭义巡游矩阵置乱 5 次　(c)狭义巡游矩阵置乱 10 次　(d)广义巡游矩阵置乱 1 次

(e)广义巡游矩阵置乱 5 次　(f)分层算法置乱 1 次　(g)分层算法置乱 2 次　(h)分层算法置乱 3 次

(i)分层算法置乱 4 次　(j)分层算法置乱 10 次　(k)分层算法置乱 20 次　(*l*)分层算法置乱 30 次

图 2　图像的骑士巡游置乱

由图 2 观察得出，传统算法多次迭代的效果并不显著，置乱 10 次时原图的内容仍清晰可辨。但分层置乱算法置乱 3 次就可将原图充分置乱，主观上 3 次、4 次、10 次、20 次和 30 次的置乱效果相似，所以需要客观标准来反映其差别。

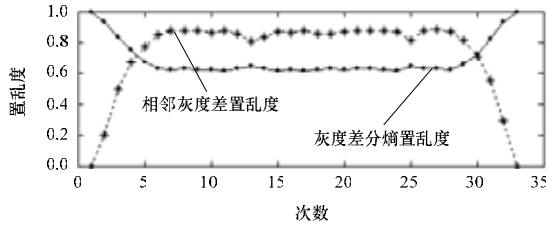
2.3　已有评价方法

文献[1]对灰度差分熵置乱度定义进行了改进。文献[3]中的置乱距离计算时间太长，不适合实际应用。文献[4]中定义的置乱度实质是一样的，都是取相邻灰度差的和。

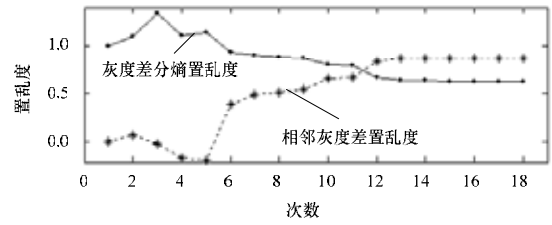
将一幅大小为 256×256 的图像进行一个周期的 Arnold 置乱，对不同次数的骑士巡游置乱、平移和旋转。用灰度差分熵置乱度、相邻灰度差置乱度计算其置乱度曲线，如图 3 所示。

由图 3 分析，2 种置乱度对于同一图像不同处理效果的描述，基本上和主观评价一致。相对来说，图 3(a)中相邻灰度差置乱度的取值范围大一些，更有利于描述不同次数置乱效果的差别；图 3(b)中的前 5 个点对应的是图像的几何变换。

对 21 幅不同的图像用 Arnold 变换置乱 6 次，对应的置乱度曲线如图 4 所示。



(a)Arnold 一个周期的置乱



(b)骑士巡游置乱和几何变换

图 3　2 种定义的同一图像置乱度曲线

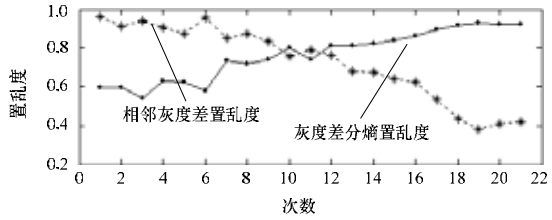


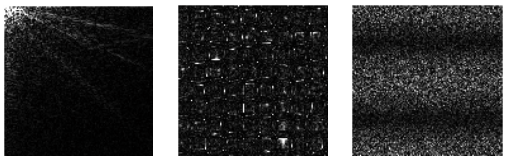
图 4　2 种定义的不同图像置乱度曲线

对不同图像具有同等置乱效果的各密图，理想客观评价应是在某一数值附近波动的近似水平的曲线。由图 4 可知，灰度差分熵置乱度、相邻灰度差置乱度都不具有此特征，这正是 2 种置乱度的不足之处。

3　基于 DCT 的置乱程度评价

DCT 是一种与傅里叶变换紧密相关的正交数学变换，是一种准最佳变换，变换公式见文献[7]，其变换前后的信号熵和能量保持不变。

对图像做二维 DCT 变换，得到 1 个 DC 直流分量，其余为 AC 交流分量，DC 系数代表图像的平均亮度，低频 AC 系数中含有丰富的图像纹理和边缘信息，而高频 AC 系数则包含图像的一些细节。如图 5(a)所示，图像的绝大部分能量在变换后集中在左上角的直流和低频系数中，反映了图像的主要内容，右下角的大多数高频系数趋向于 0。图像充分置乱后近似高斯分布，不存在纹理和边缘，对其做 DCT 变换，图 5(b)是 Arnold 算法置乱原图 6 次的对应频域图。图 5(c)是骑士巡游分层算法置乱原图 3 次的对应频域图。观察知直流分量的大小和位置都保持不变，但 AC 系数不再集中分布，而是随着置乱程度的不同对应不同的频域图。置乱程度越小，能量越集中，反之则越分散，基于此本文提出一种新的置乱度计算公式。



(a)原图　(b)Arnold 变换　(c)骑士巡游变换

图 5　原图及其置乱图的 DCT 系数图

定义　假定图像 $A=[a(i,j)]_{m\times n}$ 置乱后的图像为 $B=[b(i,j)]_{m\times n}$，A, B 对应的 DCT 系数图分别为 D_A, D_B，则

定义图像 A 的归一化置乱程度为

$$d=\frac{\sum_{i=2}^{8}\sum_{j=2}^{8}|D_A(i,j)|-\sum_{i=2}^{8}\sum_{j=2}^{8}|D_B(i,j)|}{\sum_{i=2}^{8}\sum_{j=2}^{8}|D_A(i,j)|+\sum_{i=2}^{8}\sum_{j=2}^{8}|D_B(i,j)|} \tag{1}$$

因为图像在传输的过程中要进行压缩处理，高频分量一般都被置为 0，所以只能选用图 5(a)左上角的明亮部分参与公式计算。又因为 DC 系数的值较大且在置乱前后保持不变，第 1 行第 1 列的 AC 系数代表图像的强边缘，置乱前后变化太大，以至掩盖了图像纹理和弱边缘的改变，所以式(1)中去掉了 DC 系数和第 1 行第 1 列的低频以及大部分的高频 AC 系数。容易证明以下性质：

性质 1 $-1<d<1$。

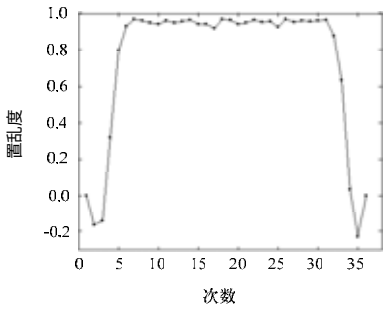
性质 2 当图像没有置乱时，$d=0$。

4 实验结果与分析

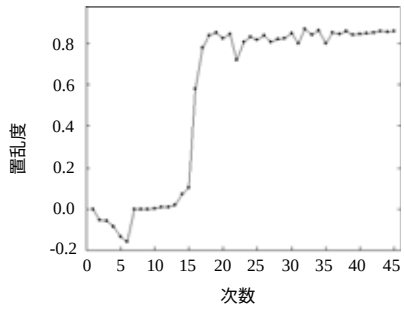
结合本文提出的置乱算法和评价标准，对 2 组图像进行了测试实验。

4.1 同一图像的不同变换

对图 2 中的原图像进行一个周期内的不同次数的 Arnold 置乱，增加平移、旋转、亮度等变换及其更多次数的骑士巡游置乱。用式(1)计算的置乱度曲线如图 6 所示，表 1 给出了图 2 中各子图对应的置乱度具体值。



(a)Arnold 不同次数的置乱



(b)骑士巡游不同次数的置乱及平移、旋转

图 6　置乱次数与置乱程度的关系

表 1　骑士巡游各置乱变换的置乱度

图 2 的子图	置乱度
(a)	0.002 8
(b)	0.009 1
(c)	0.009 9
(d)	0.017 8
(e)	0.072 3
(f)	0.581 8
(g)	0.782 1
(h)	0.841 7
(i)	0.855 1
(j)	0.819 0
(k)	0.803 0
(*l*)	0.860 2

由表 1、图 3 和图 6 可知，式(1)和已有相邻灰度差置乱度对应同一图像不同处理的置乱度曲线一致，若图像没有做任何变化或旋转 90°的倍数时置乱度为 0。若进行了平移、旋转、拉伸或置乱程度很小的变换，则置乱度小于 0 或者接近于 0；未充分置乱时置乱效果越好，置乱度相对越大；当图像充分置乱时，人眼无法分辨哪一幅密图的置乱效果相对较好，置乱度能反映其不同，可据此判断。

4.2 不同图像的同等程度置乱

选用了 43 幅纹理亮度各不相同的图像用 2 种置乱算法对其充分置乱，Arnold 置乱 6 次，骑士巡游变换 3 次。置乱效果与图 2(h)相似，主观上置乱后的各幅图像除了明亮程度不同外，并无太大的差别。取无明显纹理的图像 1 幅，纹理简单的图像 1 幅，纹理复杂的图像 1 幅和纹理正常的图像 9 幅作为代表图像，如图 7 所示，表 2 给出了其充分置乱时的置乱度具体值。对应的置乱度曲线如图 8 所示。

(a)图像 1　(b)图像 2　(c)图像 3　(d)图像 4

(e)图像 5　(f)图像 6　(g)图像 7　(h)图像 8

(i)图像 9　(j)图像 10　(k)图像 11　(*l*)图像 12

图 7　部分代表性的图像序列

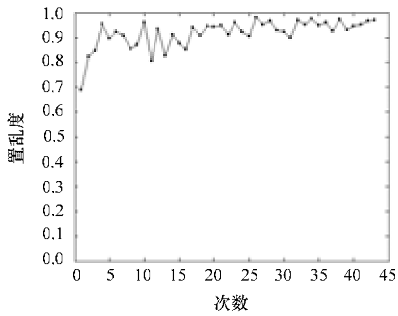
表 2　各图像充分置乱时的置乱度

图 7 的子图	Arnold 置乱	骑士巡游置乱
(a)	0.689 3	0.052 8
(b)	0.855 6	0.722 6
(c)	0.851 4	0.757 1
(d)	0.981 7	0.815 0
(e)	0.965 0	0.838 2
(f)	0.928 1	0.775 1
(g)	0.974 0	0.840 0
(h)	0.948 4	0.742 2
(i)	0.972 6	0.837 3
(j)	0.950 6	0.809 8
(k)	0.967 3	0.841 7
(*l*)	0.968 0	0.840 6

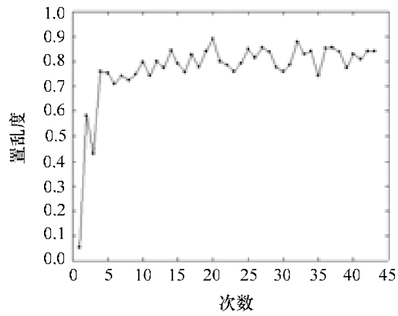
由图 7、图 8 和表 2 观察，得到以下结论：

结论 1 原图结构简单，无纹理边缘时，如图 7 中的第 1 幅图，置乱前后没有明显的改变，此时置乱度也相对较小，见图 8 曲线中的前几个点。

结论 2 原图有正常的纹理结构时，不管明亮程度如何，充分置乱后，各图的置乱度基本在某一数值附近上下起伏，且浮动的幅度不大。也就是说置乱度只和图像的置乱效果有关系，效果相似则置乱度相近，并不随着原图像的不同而有很大的改变。见图 8 曲线中的平缓部分，很好地克服了 2 种置乱度的缺点。



(a)Arnold 对不同图像的充分置乱



(b)骑士巡游不同图像的充分置乱

图 8　不同图像同等程度置乱时的置乱度

结论 3 只从像素的均匀分布来看，Arnold 变换比骑士巡游变换的置乱效果要好一些，对应的置乱度也相对大一些，但考虑到 Arnold 算法具有周期性，破解更容易，所以骑士巡游算法的安全性更高。这是由算法本身决定的，和置乱度没有关系。

5　结束语

由于骑士巡游变换具有非常大的密钥空间，因此密图安全性高，可以有效抵制预测攻击，但只能进行图像的细节置乱。本文提出的分层算法克服了此不足，使图像可充分置乱。从分析 DCT 系数的分布特点出发，给出了置乱度的定义。实验表明，该置乱度公式计算量小，所测数据和人的主观评价一致，归一化处理使得置乱度的定义更加规范和科学，且不同的图像对同样变换的结果相近，具有普遍适用性，是一种有效的客观评价方法。

参考文献

[1] 商艳红，李　南. 基于纹理特征的数字图像置乱效果分析[J]. 武汉大学学报: 理学版, 2004, 50(S1): 213-216.

[2] 柏　森，曹长修，曹龙汉. 基于骑士巡游变换的图像细节隐藏技术[J]. 中国图象图形学报, 2001, 6(11): 96-100.

[3] 刘向东，焉德军. 基于排序变换的混沌图像置乱算法[J]. 中国图象图形学报, 2005, 10(5): 657-660.

[4] 向德生，熊岳山. 基于约瑟夫遍历的数字图像置乱算法[J]. 计算机工程与应用, 2005, 41(10): 44-46.

[5] 宁宣熙，宁安琪. 广义象棋盘中的马步哈密顿圈问题及其实证研究[J]. 南京航空航天大学学报, 2004, 36(3): 383-387.

[6] Chia G L, Ong S H. Generalized knight's Tours on Rectangular Chessboards[J]. Discrete Applied Mathematics, 2005, 150(1-3): 80-98.

[7] 鲁业频，李凤亭，朱仁义，等. 基于 DCT 编码的新进展[J]. 中国图象图形学报, 2004, 9(1): 1-10.

编辑　顾逸斐

(上接第 163 页)

3.4　编译运行

将 libipt_incl.c 拷贝到<iptables/extensions/>下，内核态的 ipt_incl.c 文件拷贝到</linux/net/ipv4/Netfilter/>中，然后更改 Iptables 的 makefile 文件，再更改内核的 Konfig 和 Makefile，重新编译启用内核，再次编译安装 Iptables，使该模块集成到 Iptables 中。

这个模块在内核 linux-2.6.16.18，Iptables-1.3.7 下编译通过。通过测试，能够达到预期的效果。

4　算法的说明及性能分析

首先讨论全局数组 LIMIT_IP。如果从用户态接收到一个 C 类网络号，那么该数组的长度 $LEN=254\times2+X$，X 为一个小整数，它使得数组的长度为一个质数，比如取 523。经过反复试验，数组长度为质数时，映射 IP 时冲突的几率很小。乘以 2 是为了使数组中有空档，当 IP 为非限制 IP 时能尽快地跳出该模块。TCP 连接限制算法中采取查找死连接是为了防止网卡漏抓 FIN 或 RST。连接生存时间 T 的取值对于 TCP 和 UDP 连接应不同，因为 TCP 产生死连接的几率较少，取值可以大一些，此处取 60 s。而 UDP 在连接数达到最大之后，按照查找死连接来更新连接，取值就应该小一些，此处取 30 s，以满足正常的网络访问。

该模块的内核态部分对包进行实时处理，因此必须考虑时间复杂度。下面对该算法的时间复杂度进行详细分析。该算法的时间复杂度分为 2 个部分，即开始时的 hash 部分和查找连接部分，如果源地址不在 LIMIT_IP 数组中，则需要 2 次 hash 调用，其时间复杂度为 $2\times o(\text{hash})$。如果当前连接已经达到限制数，且有死连接存在，在查找连接数的同时，记下是否是死连接，则查找连接数的时间复杂度为 $2\times o$(限制的连接数)，即该模块的时间复杂度大约为 $2\times o(\text{hash})+2\times o$(限制的连接数)。其中，$o(\text{hash})$与 LIMIT_IP 数组的长度有关，最好取一个质数，这样可大幅降低冲突率。$o(\text{hash})$最差为 $o(254)$ (C 类 IP)，这几乎是不可能达到的。在实际试验中，$o(\text{hash})$在绝大多数情况下等于 $o(1)$，极个别的 IP 会有冲突，但一般都小于 $o(5)$，连接数一般不会超过 20，或者更少。即查找连接部分的时间复杂度小于 $2\times o(20)$，所以在绝大多数情况下，该算法的时间复杂度不超过 $2\times(o(5)+o(20))$，这个时间复杂度是可以接受的。

5　结束语

综上所述，该连接控制模块可以对单个 IP、IP 段、网络号进行 Per-IP 连接数限制，用户可以方便地在终端提交要限制的 IP 和连接数，有效地限制了客户端的并发连接数。

参考文献

[1] 李江涛，姜永玲. P2P 流量识别与管理技术[J]. 电信科学, 2005, 21(3): 57-61.

[2] 杨沙洲. Linux Netfilter 实现机制和扩展技术[EB/OL]. (2003-03-01). http://www.ibm.com/developerworks/cn/linux/l-ntflt/index.html.

[3] Bouliane N. Writing Your Own Netfilter Match[EB/OL]. (2005-02-21). http://linuxfocus.org.

编辑　顾逸斐