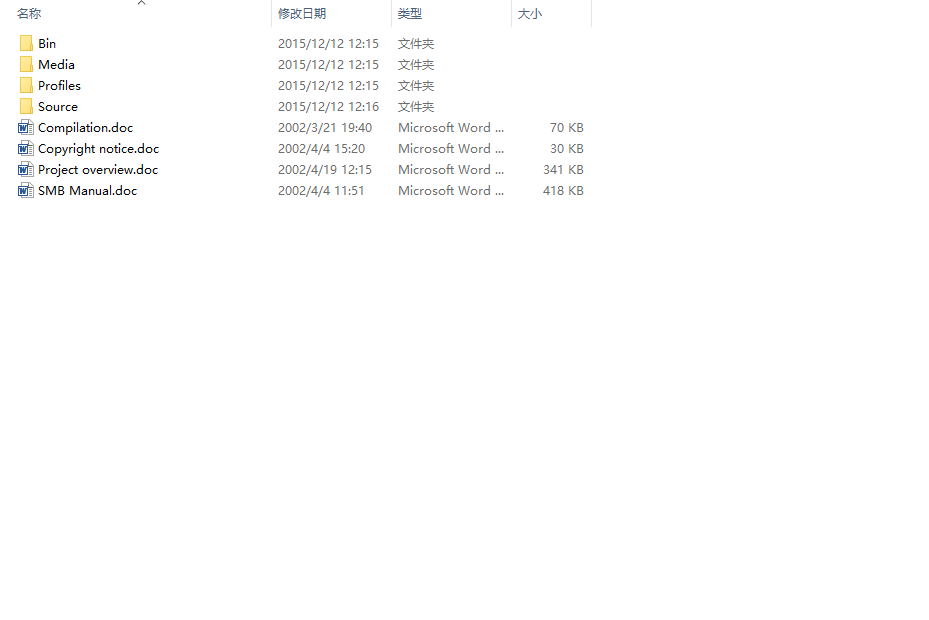
# StirMark的了解与使用

文档制作及演讲：耿东飞   
软件调试：沈鑫楠，杨珍  
资料收集： 王宪章

在对水印系统进行性能评价的过程中，需要对水印系统进行一系列的攻击，以测试其性能，这些攻击是指一个水印系统在实际使用过程中可能会遭受的有意或无意的各种攻击。而Strimark可从多方面来测评水印算法的鲁棒性，模拟多种水印攻击手段来测试水印的鲁棒性。

下载[stirmark benchmark 4.0](http://translate.googleusercontent.com/translate_c?depth=1&hl=zh-CN&rurl=translate.google.com.hk&sl=auto&sp=nmt4&tl=zh-CN&u=http://www.petitcolas.net/fabien/watermarking/stirmark/&xid=17259,15700021,15700124,15700149,15700168,15700173,15700186,15700191,15700201&usg=ALkJrhgz3Tw-rqzGKbgZeWVMHjeLmCAvWA" \t "_blank) :[binary and C/C++ source code](http://www.petitcolas.net/fabien/software/StirMarkBenchmark_4_0_129.zip)并解压缩。

  
目录说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| / | Bin/ | Benchmark/ | the benchmarking tool itself |  |  |
|  |  | Libraries/ | libraries sent by users |  |  |
|  | Profiles/ |  | profiles are text configuration files describing the test to be applied (one profile per application and per 'robustness' level) |  |  |
|  | Media/ | Input/ | Images/ | Set1/ | 输入的图片放这里 |
|  |  |  |  | Set2/  …  MyFolder/ | ... |
|  |  | Output/ | (Same substructure is created for input.) |  | 输出图片的资料夹 |
|  |  |  |  |  |  |

使用步骤  
Step 1.将嵌入浮水印的影像(watermarked image)放：Media/ Input/ Images/ Set1/   
Step 2.到Profiles/下找SMBsettings.ini：改攻击参数。 可不改，用预设。   
Step 3.到Bin\Benchmark\下找StirMark Benchmark.exe ，双击运行。   
Final Step:执行后，所有攻击后的图在：Media/ Output/ Images/ Set1/中。 

## 测试参数说明表

|  |  |  |
| --- | --- | --- |
| [stirmark benchmark 4.0](http://translate.googleusercontent.com/translate_c?depth=1&hl=zh-CN&rurl=translate.google.com.hk&sl=auto&sp=nmt4&tl=zh-CN&u=http://www.petitcolas.net/fabien/watermarking/stirmark/&xid=17259,15700021,15700124,15700149,15700168,15700173,15700186,15700191,15700201&usg=ALkJrhgz3Tw-rqzGKbgZeWVMHjeLmCAvWA) | example(预设) | 说明 |
| Test1=Test\_PSNR |  | Strength of the watermark embedding |
| Test2=Test\_EmbedTime |  | ;Number of embeddings with random key per media ; This is used to compute the average embeding and extraction time |
| Test3=Test\_AddNoise | list=5 |  |
| Test4=Test\_JPEG | list=5 | JPEG\_5：等级为5的JPEG压缩。(等级低，压缩比高)。 |
| Test5=Test\_MedianCut | list=3 | 中位数滤波攻击。MEDIAN\_3：采用3X3的遮罩。 |
| Test6=Test\_ConvFilter1(Gaussian filtering)Convolution | filter1 = 3 3 9 1 2 1 2 4 2 1 2 1 | 和高斯函数做convolution。 也就是该点像素以高斯函数做为附近像素值的加权平均之权重，加权后的均值取代原该点像素。所以有平滑、模糊的效果。  3的遮罩为[1 2 1;2 4 2;1 2 1]。 |
| Test7=Test\_SelfSimilarities | filter2 = 3 3 9 0 1 0 1 5 1 0 1 0 | 3的遮罩为[0 1 0;1 5 1;0 1 0]。 |
| Test8=Test\_RemoveLines |  | 使用RGB色系，RGB通道，交换攻击 |
| Test9=Test\_Cropping |  |  |
| Test10=Test\_Rescale | list=50 | 取中间部分。50：表示取下中间长宽50%做为测试 |
| Test11=Test\_Rotation | list=10 | 缩放百分比：10% |
| Test12=Test\_RotationCrop | list=0.1 | 逆时针转0.1度 |
| Test13=Test\_RotationScale |  |  |
| Test14=Test\_Affine |  |  |
| Test15=Test\_SmallRandomDistortions |  | Affine |
| Test16=Test\_LatestSmallRandomDistortions |  |  |

## 实验截图

### 设置参数SMBsettings.ini：

[Tests]

Test1=Test\_PSNR

Test2=Test\_EmbedTime

Test3=Test\_AddNoise

Test4=Test\_JPEG

Test5=Test\_MedianCut

Test6=Test\_ConvFilter

Test7=Test\_SelfSimilarities

Test8=Test\_RemoveLines

Test9=Test\_Cropping

Test10=Test\_Rescale

Test11=Test\_Rotation

Test12=Test\_RotationCrop

Test13=Test\_RotationScale

Test14=Test\_Affine

Test15=Test\_SmallRandomDistortions

Test16=Test\_LatestSmallRandomDistortions

[Test\_Rotation]

; angles in degrees

list=-2 -1 -0.75 -0.5 -0.25 0.25 0.5 0.75 1 2 5 10 15 30 45 90

[Test\_RotationCrop]

; angles in degrees

list=-2 -1 -0.75 -0.5 -0.25 0.25 0.5 0.75 1 2

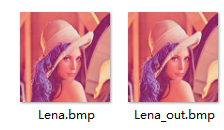
[Test\_RotationScale]

; angles in degrees

list=-2 -1 -0.75 -0.5 -0.25 0.25 0.5 0.75 1 2

### 测试图片

原图片与对其进行DCT水印得到含有隐藏信息的图像，本测验使用后者Lena\_out.bmp：

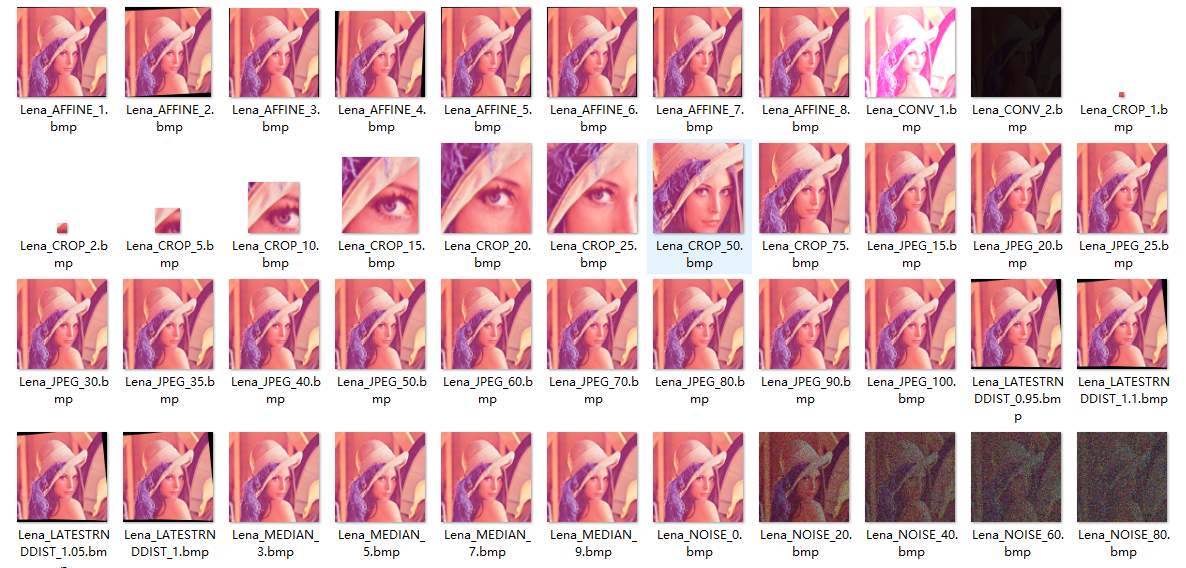


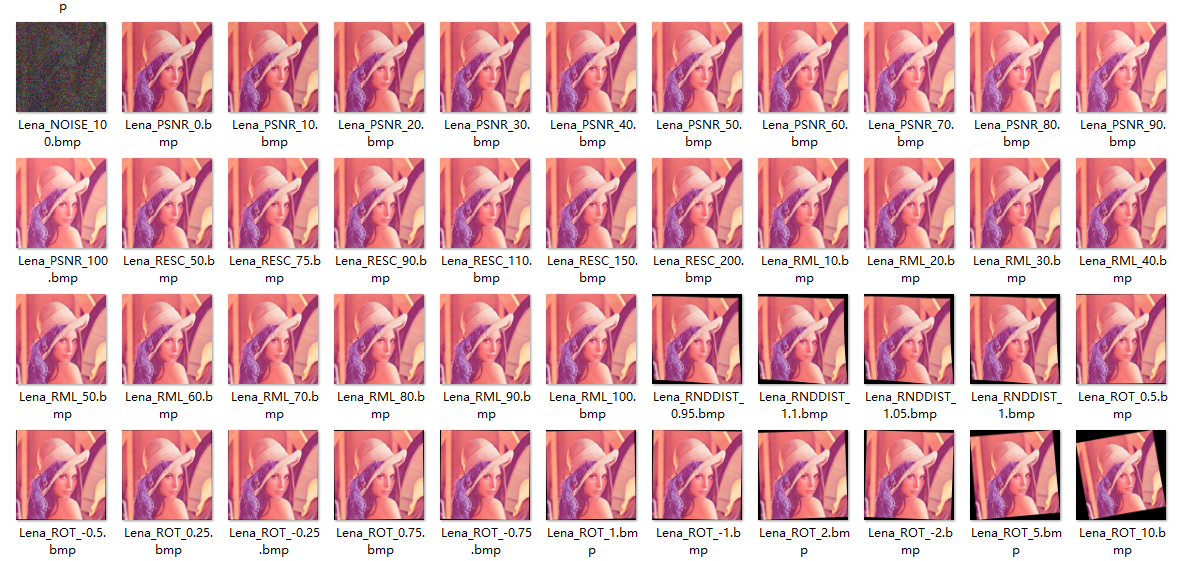
### 运行过程

### 

### 运行结果

### 攻击结果





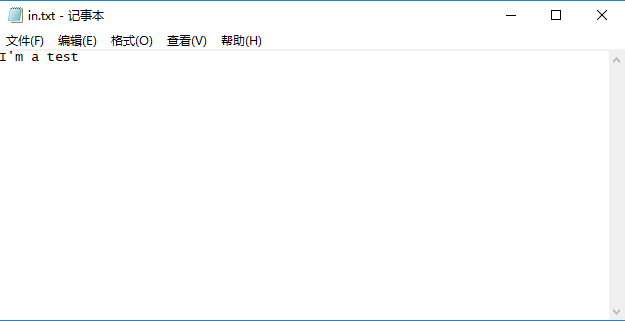
### 提取测试

用提取算法提取信息，看信息是否能正确提取。以旋转攻击为例：发现提取信息后是乱码，说明该算法不能抵抗旋转攻击。

旋转90°攻击得到的图片：

### 

原图片提取结果：



攻击图片提取结果：

