

# 对于MD5的彩虹表攻击方法的设计和实现

答辩学生：张瑜


指导教师：谢敏

# 目录

- ① 课题介绍
- ② 彩虹表原理
- ③ 设计过程及参数分析
- ④ 攻击数据及比较
- ⑤ 使用数据库的改进
- ⑥ 工作总结及结论
- ⑦ 研究展望

- 课题介绍

哈希函数：MD5

任意长度的输入            128比特的消息摘要

用途：文件校验、数字签名

标准文件：RFC1321

- 课题介绍

时间—空间折中 (Time-Memory Trade-Off)

穷搜索:  $N$ 次计算

穷存储:  $N$ 个单位的字节

折中

*“A cryptanalytic time-memory trade off”*

1980年    Martin E Hellman

- 课题介绍

时间—空间折中 (Time-Memory Trade-Off)

已知原文长度和MD5值, 要找出原文  
定义哈希函数<sub>H</sub> 和归约函数<sub>R</sub>

$$S_i \xrightarrow{H(S_i)} h_i \xrightarrow{R(h_i)} S_{i+1}$$

$$S_{1,1} \xrightarrow{f(S_{1,1})} S_{1,2} \xrightarrow{f} \dots \xrightarrow{f} S_{1,t}$$

▪  
▪  
▪

$$S_{m,1} \xrightarrow{f(S_{m,1})} S_{m,2} \xrightarrow{f} \dots \xrightarrow{f} S_{m,t}$$



经典表

● 课题介绍

经典表的破解过程

$S_{1,1} \xrightarrow{f(S_{1,1})} S_{1,2} \xrightarrow{f} \dots \xrightarrow{f} S_{1,t}$

▪  
▪  
▪

$S_{m,1} \xrightarrow{f(S_{m,1})} S_{m,2} \xrightarrow{f} \dots \xrightarrow{f} S_{m,t}$

经典表

已知哈希值<sup>h</sup>

第一轮:

$S_1 = R(h)$

查表并判断

第二轮

$S_2 = f(R(h))$

查表并判断

▪  
▪  
▪

第( t-1 )轮

$S_{t-1} = f \dots f(R(h))$

查表并判断

查表：在表格中找到终点与  $S_i$  相匹配的链

判断：恢复该链并判断原文是否在链中

- 彩虹表原理

2003年 Philippe Oechslin

*“Making a faster cryptanalytic time-memory trade off”*

经典表的最大问题在于当两条链的某处发生碰撞后，这两条链的后续部分都会相同，这造成了信息的冗余，直接导致成功率的下降。

$$S_{1,1} \xrightarrow{f_1(S_{1,1})} S_{1,2} \xrightarrow{f_2} \dots \xrightarrow{f_{t-1}} S_{1,t}$$

▪  
▪  
▪

$$S_{m,1} \xrightarrow{f_1(S_{m,1})} S_{m,2} \xrightarrow{f_2} \dots \xrightarrow{f_{t-1}} S_{m,t}$$



彩虹表



$$P_{table} = 1 - \prod_{i=1}^t \left(1 - \frac{m_i}{N}\right)$$

- 设计过程及参数分析

### 阶段一：表的生成

1. 随机选取开始字符串；
2. 开始产生一定长度的链；
3. 将链的起始字符串和终点字符串放在内存中；
4. 继续产生下一条链，碰撞检测后并存入内存；
5. 排序后将表格写入文件。

### 阶段二：解的查找

1. 读取表格文件，即加载到内存中；
2. 对哈希值 $h$ 使用归约函数 $R$ 和哈希函数 $H$ ，得到新的字符串及其哈希值；
3. 对比表格中的终点字符串，若有相匹配的，则重新生成该条链，并判断解是否在该链中，否则返回步骤2；
4. 输出结果。



- 设计过程及参数分析

## 归约函数设计

---

### Function 1: 归约函数reduce

---

**Input:** 哈希值hash, 归约种子seed, 步骤step, 字符串长length, 字符集合charSet

**Output:** 字符串string

```
1 begin
2   step = step + seed
3   for index in [0, length ) do
4       string[index] = charSet[ ( step ^ hash ) % count]
5       step = step +1
6       index = index +1
7       *hash ++
8   end
9   return  string
10 end
```

---

## • 设计过程及参数分析

### 最优参数的分析:

自变量: 链长<sup>*t*</sup>、链数<sup>*m*</sup>、表格数<sup>*l*</sup>

因变量: 生成时间<sup>*T<sub>g</sub>*</sup>、占用空间<sup>*M*</sup>、破解时间<sup>*T*</sup>、成功率<sup>*P*</sup>

折中系数: <sup>*TM<sup>2</sup>*</sup>

$$T_g = mtl \quad (1)$$

$$M = ml \quad (2)$$

$$T = \frac{1}{2}(t^2 + t)l + \frac{m l t^3}{6N} \quad (3)$$

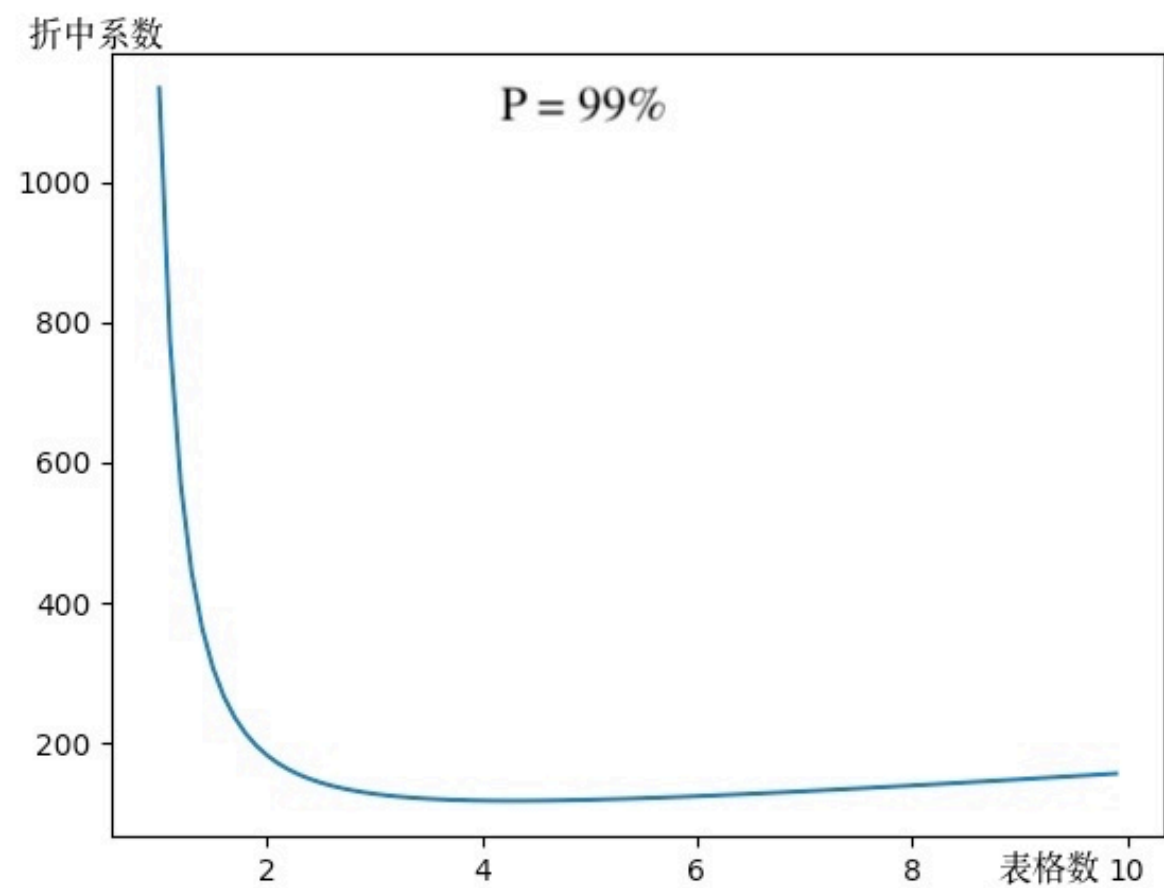
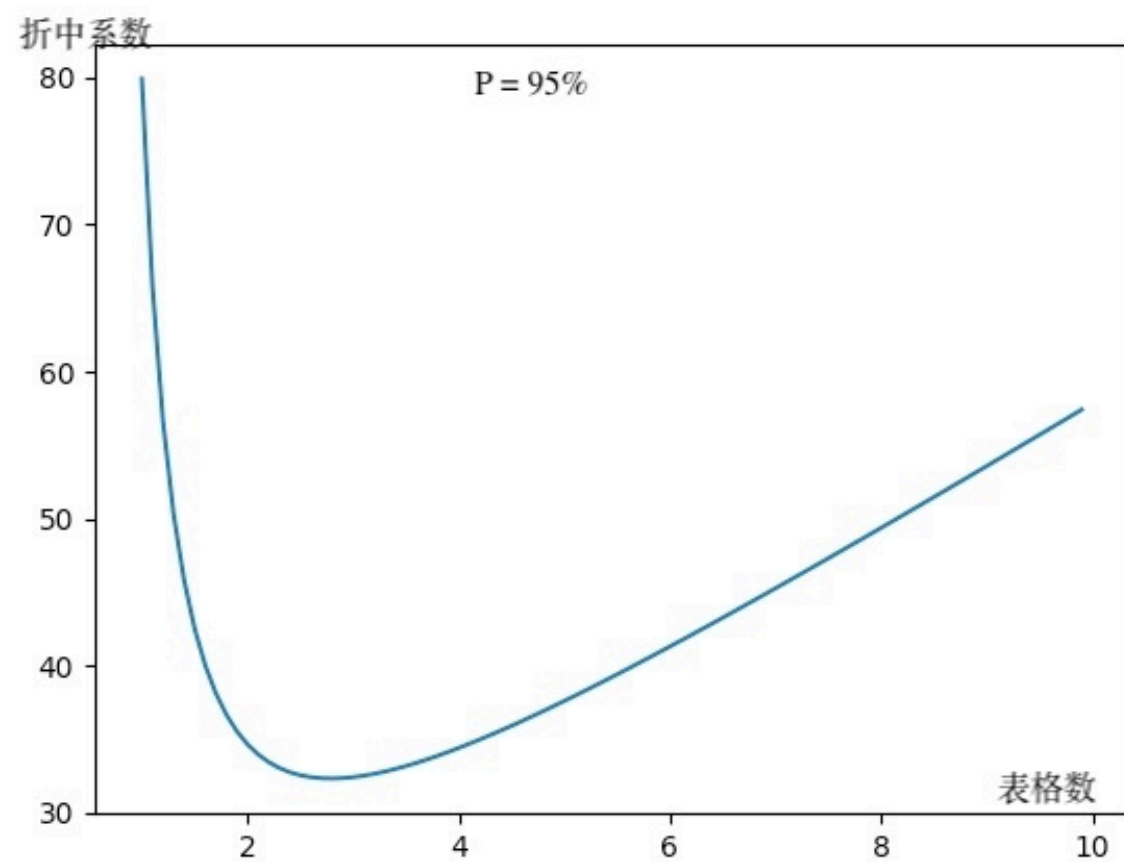
$$P = 1 - (1 - P_{table})^l \quad (4)$$

$$P_{table} = 1 - \prod_{i=1}^t (1 - \frac{m_i}{N}) \quad \text{其中} \quad \frac{m_i}{N} = \frac{1}{\frac{N}{m} - \frac{i-1}{2}} \quad (5)$$

$$TM^2 = (\frac{1}{2} + \frac{2((1-P)^{-\frac{1}{2l}} - 1)}{6}) 2((1-P)^{-\frac{1}{2l}} - 1)^2 l^3 N^2 \quad (6)$$

$$mt = 2((1-P)^{-\frac{1}{2l}} - 1)N \quad (7)$$

- 设计过程及参数分析



固定成功率后的折中系数曲线

- 攻击数据及比较

明文空间	链长	链数	生成时间	平均破解用时	占用空间	单表成功率
36^4	800	1000	1s	0.16s	21KB	36.5%
	1000	1600	2s	0.2s	34KB	59.9%
36^5	2000	14000	21.25s	1.09s	308KB	32.7%
	1000	56000	63.58s	0.22s	1.2MB	65.0%
36^6	1000	500000	340.69s	0.27s	11.5MB	20.0%
	1500	750000	893.90s	0.51s	17.3MB	42.2%

未使用最优参数的攻击结果

\*成功率基准：1000次破解进行3轮

- 攻击数据及比较

明文空间	链长	链数	生成时间	平均破解用时	占用空间	单表成功率
36 <sup>4</sup>	800	3275	6.18s	0.11s	69KB	76.0%
	1000	2620	6.30s	0.17s	55KB	74.0%
36 <sup>5</sup>	800	117909	239.8s	0.115s	2.6MB	75.0%
	1000	94327	242.3s	0.184s	2.1MB	74.5%
36 <sup>6</sup>	1560	2176782	2h32m34s	0.27s	50.1MB	75.0%
	2000	1697890	2h31m50s	0.695s	39.1MB	76.2%

使用最优参数的攻击结果，固定成功率为99.0%

\*成功率基准：1000次破解进行3轮

- 攻击数据及比较

1. 生成时间正比于表格大小
2. 占用存储空间正比于表格链数
3. 破解用时正比于表格大小，但主要受链长影响
4. 单表成功率要高于预测的成功率

- 攻击数据及比较

$$T = \frac{1}{2}(t^2 + t)l + \frac{m l t^3}{6N} \quad (3)$$

$$P = 1 - (1 - P_{table})^l \quad (4)$$

$$P_{table} = 1 - \prod_{i=1}^t (1 - \frac{m_i}{N}) \quad (5)$$

$m_i$

表示彩虹表每一列的字符串中不同字符串的个数

$$m_1 = m$$

$$m_i = N(1 - (1 - \frac{1}{N})^{m_{i-1}})$$

- 攻击数据及比较

明文空间	链长	链数	生成时间	平均破解用时	占用空间
$36^4$	800	3275	0.6s	0.14s	51KB
$36^5$	800	117909	22.8s	0.14s	1.79MB
$36^6$	1560	2176782	20m29.5s	0.52s	33.2MB

相同机器下软件RainbowCrack的攻击数据



- 攻击数据及比较

明文空间	链长	链数	表格数	平均破解用时	占用空间	成功率
2 <sup>27</sup>	4666	35000000	5	13.26s/表格	286.7MB	99.9%
2 <sup>27</sup>	800	38223872	1	12.9s/表格	未给出	78.8%

Philippe Oechslin的攻击数据

- 使用数据库的改进

原先实现方式的缺陷：

生成和破解过程中表格文件都会写入到内存中，当表格过大时无法提供足够的内存。



最大列数：2000  
最大行数：2<sup>64</sup>  
最大容量：140TB

- 使用数据库的改进

明文空间	链长	链数	生成时间	平均破解用时	占用空间	生成时占用内存
$36^4$	800	3725	16s	0.3s	127KB	596K~616K
$36^5$	800	117909	9m36s	3.5s	4.5MB	710K~910K
$36^6$	1560	2176782	5h6m9s	8.75s	102MB	740K~1.6M

使用数据库后的攻击结果

- 使用数据库的改进



在移动设备上彩虹表破解

- 工作总结及结论

## 工作总结

- 学习了MD5算法，并了解哈希函数在计算机安全中的应用
- 学习了时间—空间折中思想和彩虹表构造方法
- 实现了针对MD5的彩虹表攻击
- 整理了攻击数据并与其他人的数据进行了对比
- 使用SQLite减少了内存的消耗，并在移动设备上进行了测试

- 工作总结及结论

## 结论

- 彩虹表攻击方法是一种有效的密码分析方法
- 折中的考量即参数的选择要依破解环境改变
- 使用数据库存储彩虹表是一种很好的方式

- 研究展望

- 计算阶段的加速

1. 使用GPU加速计算
2. 使用分布式计算平台分摊计算成本

- 算法上的优化

1. 减少错误警报
2. 根据使用场景进行合适的折中