**On the Semantic Patterns of Passwords and their Security Impact**

**共有三步**

1. 将密码截断为多个词义段
2. 根据语料库为每个词义段标注词性
3. 根据WordNet库生成每个词义的抽象树，如蝴蝶->昆虫->动物
4. 对口令进行语义预测攻击，如Ilove\_\_\_
5. **断字**

首先根据字符串特点可以分为字符集和间隙集，如anyonejack 123，其中{anyonejack}为单词集，{123}为间隙集。然后根据单词用法对单词集进行分割，并找到最常用的分割方式{anyone，jack}。得到{anyone，jack，123}

源语料库（source corpora）：用于算法的词典，包括：

COCA中的三个字符的单词，出现次数少于100的去掉；

COCA中的两个字符的单词，出新次数前37

COCA中的一个字符的单词，I，a

姓名集，城市集，Surnames，月份集，国家名集，（学校，公司缩写等，论文未涉及）

参考语料库（reference corpora.）：包括了使用频率，N元语义标注的演讲稿，用来确定最有可能的分段方式（歧义情况，如：anyonejack），包括：

COCA，姓名集，城市集， Surnames姓，月份集，国家名集

主要集合：当代美国英语语料库，据有普遍意义的，且语法标注（一元语法，二元语法，三元语法）了的演讲段落，该段落包含单词出现频率。分两个语料库为了了提高效率。

最常见密码策略：连接，替换，拼写修改，插入等。

定义

单词段(word segment)：在源语料库中可以找到，常是字符集

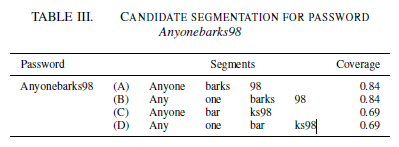
间隙段{gap segment}：在源语料库中找不到，且被单词段或密码边界包围，都可能

a word segment is always alphabetic,

while a gap can include any character (numbers, symbols or letters)

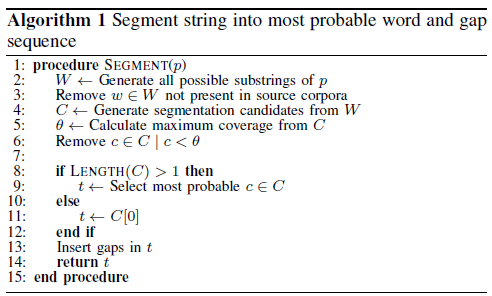
步骤

1. 首先按照间隙段将密码分割为单词段集合P
2. 分别计算每个单词段最可能的分段方式（算法2），并将间隙段重新插入分段结果（算法1）



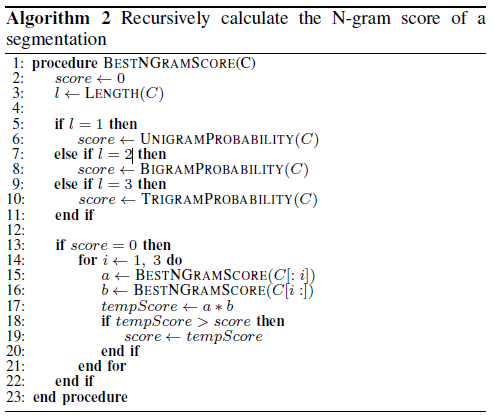
不考虑间隙段的影响，则只需考虑最高的AB两种，coverage相等，则考虑使用频率，用到了相关集，计算使用频率。

分段算法



1. 先找到密码p所有可能的子段，生成集合W
2. 去掉W中不在源集中的数据
3. 从W中生成字段候选项集C
4. 保留C中分段的最大可能性的候选项（调用了算法2）
5. 将C中最大可能性候选项插入间隙值返回

计算一段字符串的N元语义的可能性算法



1. 根据字符串长度计算其的得分
2. 如果字符串长度k＞3，则分别将该字符串分割为a1[0:1] a2[1:k]或者a3[0:3] a4[4:k],再分别回到1进行计算，直到得到最高得分的分割方法

基于参考语料库，算法2中的N-GramProperty计算方式定义如下：

KN：语料库K中所有的N元语义（N元语义：由N个字母组成的单词）

F(KN)：在语料库K中N元语义的总频率

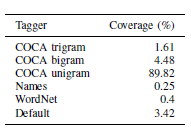
：N元语义在N元语料库中的比例

算法2，输入love，怎么办？？？

1. **标注库**

帮助我们对词进行词性分析，NLTK的nltk.pos\_tags([words])可以帮我们实现

论文按照如下优先级对词性进行标注



Default可以自己定义，论文定义为NN（名词）

根据上下文可以处理一词多词性的问题

**三、语义分类**

经过上述两步处理，我们可以得到一个2元数组：S = [(s1, t1), ..., (sn, tn)]**这一步的计算将输入S，输出K [(s1, t1, c1), ..., (sn, tn, , cn)]**

si：密码P的子段

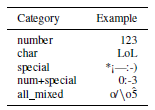
ti：词性标记，如果子段为间隙段，则该值为NULL

Ci：类别（category.）

1. WordNet：NLTk里的一个库，可以计算近义词，反义词，一词多义，等等

我们只对名词和动词标记

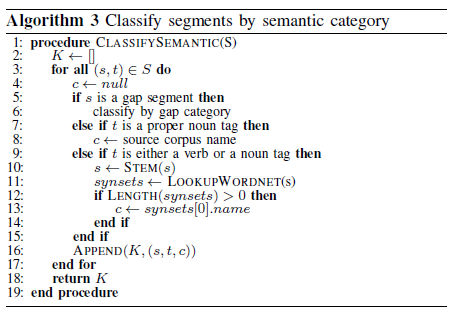
（1）间隙集，我们按照下表标记



（2）专有名词：利用源语料库，标记为月（month），名（names），姓（surname），国家（country），城市（city），如上顺序避免歧义。

（3）名词/动词：获取词根（stem），调用wordNet 得到同义词集。如果该词根有多个词义，则根据使用频率排序，选择第一个同义词集作为语义标记。

语义类别标记算法



1. 泛化（Generalization:）

找到某个词的上层语义（hyperonymy），形成语义树，如，蝴蝶和蜻蜓的上层语义为昆虫，动物，然后通过剪枝算法（tree cut model）M确定要抽象到哪个上层语义。

泛化语义树的中间节点表示一个类别，子节点标识的是一个实例，叶子节点的频率与观察结果相关，并最终累计到中间节点。

剪枝算法规定如下两个矩阵：

Lpar（M）：剪枝大小，与抽象程度成反比

Ldat（M）：从数据到剪枝点的距离，与抽象程度成正比

评估标准：Lmod（M） = Lpar（M） + Ldat（M）



K：截面节点数，|S|：样本大小



s 样本中的同义词集

P(s)：在语义树中该抽象类的概率：



C：一个节点下的叶子节点（同义词集）数量



f（C）：类C在样本中的实例总数

找到使Lmod（M）最小的M

3）调整权重



论文中的例子W=1000、5000、10000