CRYPTO总结

原创

醉等佳人归 于 2021-08-31 11:46:08 发布 418 收藏 9

分类专栏： 网络安全学习 文章标签： python

本文链接：https://blog.csdn.net/qq_43647628/article/details/119848999

网络安全学习 专栏收录该内容

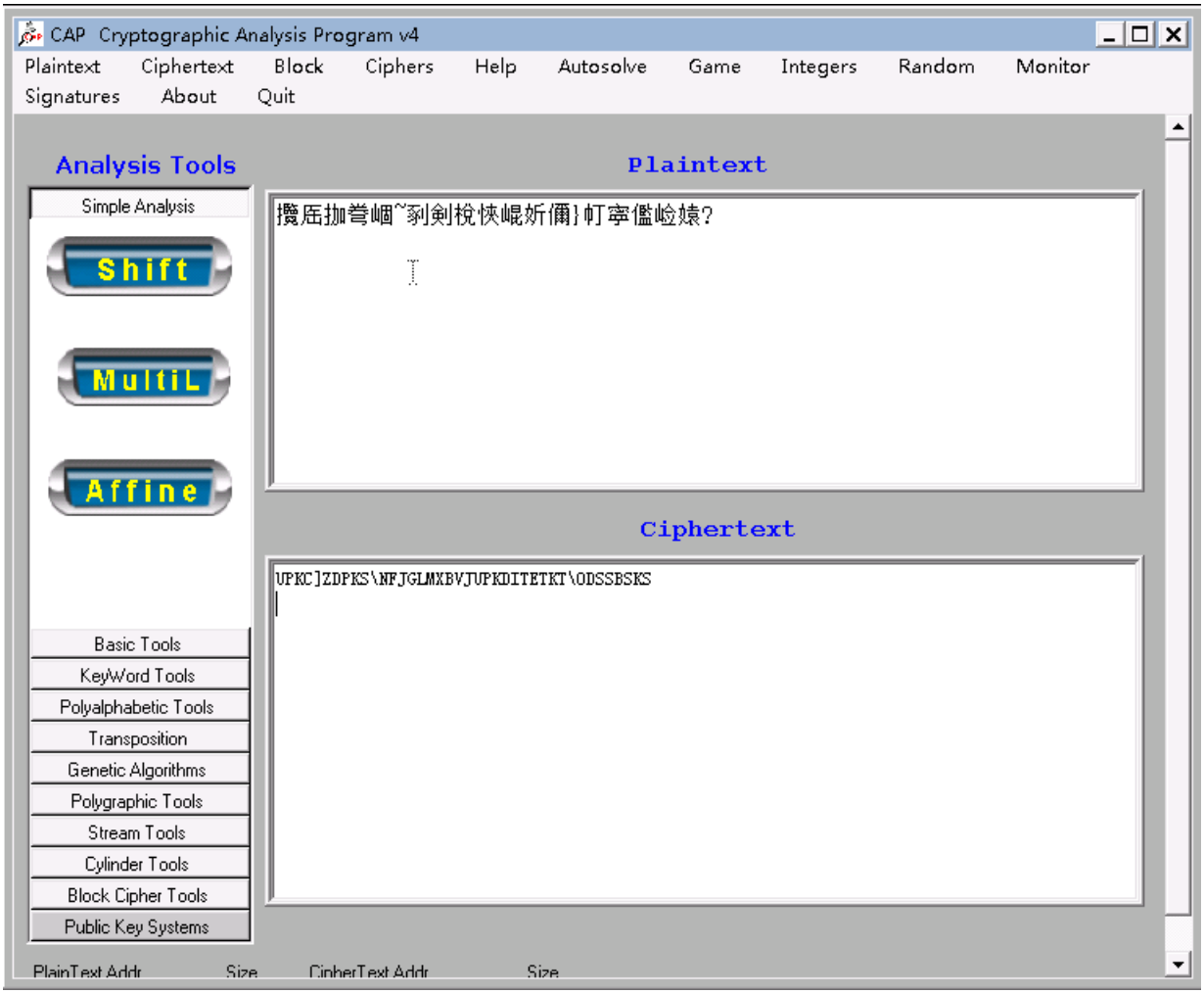
40 篇文章 0 订阅

订阅专栏

文章目录

加解密软件

CAP(Cryptographic Analysis Program)是由DR. Richard Spillman专门为教学而研制的密码制作与分析工具，已经在美国的很多高校得到了广泛地使用，受到了密码学习者的普遍欢迎。



CrypTool

CT1（CrypTool）是一个免费软件程序，使您可以应用和分析加密机制。它具有Windows应用程序的典型外观。CT1包含全面的在线帮助，可以在不深入了解密码学的情况下理解。

可用的算法包括经典和现代密码系统：

经典方法：Caesar密码，ADFGVX密码，双列转置（置换），Enigma加密算法等。

现代方法：RSA和AES算法，混合加密，基于晶格简化和椭圆曲线的算法等。

1.古典密码学

1.1.摩斯电码

摩尔斯密码是替换密码的一种，通过特定的密码置换表对相应的字符进行转换就是摩尔斯密码的编码原理，由于转换方式太过于简单以及编码之后特征太明显，到了今天我们已经很难从影视作品或是简单的ctf竞赛题目之外来找到摩尔斯密码的应用了，但不可否认他在密码学的发展过程中所起到的作用，通过学习摩尔斯密码我们也可以初步体会到替换密码的魅力。
以下就是摩尔斯密码的替换表。
26个字母的摩斯密码表

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
A	.-	B	-...	C	-.-.	D	-..
E	.	F	..-.	G	--.	H	....
I	..	J	.---	K	-.-	L	.-..
M	--	N	-.	O	---	P	.--.
Q	--.-	R	.-.	S	...	T	-
U	..-	V	...-	W	.--	X	-..-
Y	-.--	Z	--..				

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
0	-----	1	.----	2	..---	3	...--
4	....-	5	.....	6	-....	7	--...
8	---..	9	----.				

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
.	.-.-.-	:	---...	,	--..--	;	-.-.-.
?	..--..	=	-...-	'	.----.	/	-..-.
!	-.-.--	-	-....-	_	..--.-	"	.-..-.
(	-.--.	)	-.--.-	$	...-..-	&	....
@	.--.-.						

1.2.栅栏密码

我们以2栏栅栏密码为例来讲解它的加密和解密过程。

比如明文：THERE IS A CIPHER

两个一组，得到：(TH) (ER) (E) (IS) (A) © (IP) (HE) (R)

先取出第一个字母：TEEI IHR

再取出第二个字母：HR SACPE

连在一起就是：TEEI IHRHR SACPE

还原为所需密码。

而解密的时候，我们先把密文从中间分开，变为两行：

TEEI IHR

HR SACPE

再按上下上下的顺序组合起来：

THERE IS A CIPHER

1.3.凯撒密码

在密码学中，凯撒密码是一种最简单且最广为人知的加密技术。它是一种替换加密的技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例如，当偏移量是3的时候，所有的字母A将被替换成D，B变成E，以此类推。

1.4.ROT13

ROT13是凯撒密码的一种变体，即移位数为13。ROT13是它自己本身的逆反，也就是说，要还原ROT13，应用加密同样的算法即可得，故同样的操作可用再加密与解密。

1.5.维热纳尔加密

1.5.1.简介

1）维热纳尔密码，是一个很著名的多码加密法，它实际上是自动密钥（autokey）加密法的一个简化形式，是基于关键词的加密系统，但不是向单码关键词加法那样使用关键词来定义替换形式，关键词写在明文的上面，并不断重复书写，这样每个明文字母都与一个关键词的字母关联。它是由法国外交家Blaise de Vigenère发明的。

2）公元16世纪晚期，想要获得更高的保密度的人获得了一种设计更加精细的密码表。

法国外交家Blaise de Vigenère发明了一种方法来对同一条信息中的不同字母用不同的密码进行加密。这样，同样的E在一个位置可能被M所取代，而在另一个位置的E则有可能以K的面目出现。这样，就可以防止任何人利用频率分析法解密该条信息。

1.5.1.原理

在维热纳尔（Vigenère）的密码中，发件人和收件人必须使用同一个关键词（或者同一文字章节），这个关键词或文字章节中的字母告诉他们怎么样才能前后改变字母的位置来获得该段信息中的每个字母的正确对应位置。比如如果关键字“BIG”被使用了，发件人将把信息按三个字母的顺序排列。第一个三字母单词的第一个字母将应当向后移动一个位置（因为B是排在A后面的字母），第二个字母需要向后移动8位（I是A后面第8个字母），而第三个字母需要向后移动6位（G是A后面第6个字母）

加密的流程：

文字按下面的顺序来进行加密：

明文：THE BUTCHER THE BAKER AND THE CANDLESTICK MAKER。（屠夫、面包师和蜡烛匠）。

关键密钥：BIG BIGBIGB IGB IGBIG BIG BIG BIGBIGBIGBI GBIGB

密文：UPK CCZDPKS BNF JGLMX BVJ UPK DITETKTBODS SBSKS

1.6.Affine加解密

1.6.1.简介

1）单码加密法的另一种形式称为仿射加密法（affine cipher）。在仿射加密法中，字母表的字母被赋予一个数字，例如a=0，b=1，c=2…z=25。仿射加密法的密钥为0-25直接的数字对。仿射加密法与单码加密法没什么不同，因为明文的每个字母分别只映射到一个密文字母。

1.6.2.原理

加法密码和乘法密码结合就构成仿射密码，仿射密码的加密和解密算法是：

C= Ek(m)=(k1m+k2) mod n

M= Dk©=k3(c- k2) mod n（其中（k3 ×k1）mod26 = 1）

仿射密码具有可逆性的条件是gcd(k1, n)=1。当k1=1时，仿射密码变为加法密码，当k2=0时，仿射密码变为乘法密码。

仿射密码中的密钥空间的大小为nφ(n)，当n为26字母，φ(n)=12，因此仿射密码的密钥空间为12×26 = 312。

加密：

设密钥K= (7, 3), 用仿射密码加密明文hot。

三个字母对应的数值是7、14和19。分别加密如下：

(7×7 + 3) mod 26 = 52 mod 26 =0

(7×14 + 3) mod 26 = 101 mod 26 =23

(7×19 + 3) mod 26 =136 mod 26 =6

三个密文数值为0、23和6，对应的密文是AXG。

解密：

按照上例来解密的，也就是用仿射密码解密密文AXG，密钥k=（7,3）。

三个字母对应的数值是0、23、6。解密如下：

由解密Dk©=k3(c- k2) mod n（其中（k3 ×k1）mod26 = 1）；

可知k3×7=1（mod 26）（其实，就是1/mod26），也就是存在整数t，使7×k3+26t=1。（1）

利用辗转相除法求解k3：[1]

26 = 7 * 3 + 5；（2）（对26作形如：a * b + c，其中 c 就是余数）

7 = 5 * 1 + 2；（3）（作形如： a = c * m + n，其中 a，c 是上一步的， m 是乘数，n 是余数）

5 = 2 * 2 + 1；（一直循环上一步，直到余数 n = 1）

进行回代：

1 = 5 - 2 * 2

= 5 -（7 - 5 * 1） * 2（第一个2用（3）式来代替，也就是2 = 7 - 5 * 1）

= 3 * 5 - 2 * 7

= 3 *（26 - 7 * 3） - 2 * 7（5用（2）式来代替，也就是5 = 26 - 7 * 3）

= -11 * 7 + 3 * 26（直到不用进行代替，也就是得到只有3和26的表达式）

对比（1）式可知：t = 3，k3 = -11；

所以：Dk©=k3(c- k2) mod n <=> Dk©=15(c- 3) mod 26 .

对于第一位 A：

-11 (0 - 3) mod 26 =（ -11 * -3 ）mod 26 = 7;

对于第二位 X：

-11 (23 - 3) mod 26 = (-11 * 20) mod 26 = （ -220 ） mod 26 = (26 * -9) + 14 = 14;

用计算器求 （-220） mod 26 ，不同的计算器会有不同的结果，百度的计算器求得就是 14 ，直接百度搜索：（-220） mod 26 就可以了，不能直接在计算器上输入 -220mod26 ，那样会得出负数。其实，可以这样算，算出（-11）mod 26 =15，再计算（15 * 20）mod26 = 14)

对于第三位 G ：

-11 （ 6 - 3 ） mod 26 = （ -11 * 3 ）mod 26 = （ -33 ）mod 26 = 19;（计算方法如上）

三个明文值为 7,14,19，对应的明文是HOT，也就是hot。

1.7.Playfair加密算法

1.7.1.简介

1）Playfair密码（英文：Playfair cipher 或 Playfair square）是一种替换密码，1854年由查尔斯·惠斯通（Charles Wheatstone）的英国人发明。经莱昂·普莱费尔提倡在英国军地和政府使用。

它有一些不太明显的特征：密文的字母数一定是偶数；任意两个同组的字母都不会相同，如果出现这种字符必是乱码和虚码。

它使用方便而且可以让频度分析法变成瞎子，在1854到1855年的克里米亚战争和1899年的布尔战争中有广泛应用。但在1915年的一战中被破译了。

编写分三步：1.编制密码表 2.整理明文 3.编写密文 构成部分：1.密钥 2.明文3.密文4.注明的某个字母代替的另一个字母。

2）Playfair密码的算法是依据一个5*5的正方形组成的密码表来编写，密码表里排列有25个字母。如果一种语言字母超过25个，可以去掉使用频率最少的一个。如，法语一般去掉w或k，德语则是把i和j合起来当成一个字母看待。英语中z使用最少，可以去掉它。

1.7.2.原理

加密的流程：

1）编制密码表

在这个5*5的密码表中，共有5行5列字母。第一列（或第一行）是密钥,其余按照字母顺序。密钥是一个单词或词组，若有重复字母，可将后面重复的字母去掉。当然也要把使用频率最少的字母去掉。如：密钥是Live and learn,去掉后则为liveandr。如果密钥过长可占用第二列或行。

如密钥crazy dog，可编制成：

C	O	H	M	T
R	G	I	N	U
A	B	J	P	V
Y	E	K	Q	W
D	F	L	S	X

2）整理明文

第二步整理明文。将明文每两个字母组成一对。如果成对后有两个相同字母紧挨或最后一个字母是单个的，就插入一个字母X（或者Q）。

如，communist，应成为co,mx,mu,ni,st。

3）编写密文

最后编写密文。对明文加密规则如下：

1 若p1 p2在同一行，对应密文c1 c2分别是紧靠p1 p2 右端的字母。其中第一列被看做是最后一列的右方。如，按照前表，ct对应oc

2 若p1 p2在同一列，对应密文c1 c2分别是紧靠p1 p2 下方的字母。其中第一行被看做是最后一行的下方。

3 若p1 p2不在同一行，不在同一列，则c1 c2是由p1 p2确定的矩形的其他两角的字母（至于横向替换还是纵向替换要事先约好，或自行尝试）。如，按照前表，wh对应tk或kt。

如，依照上表，明文where there is life,there is hope.

可先整理为wh er et he re is li fe th er ei sh op ex

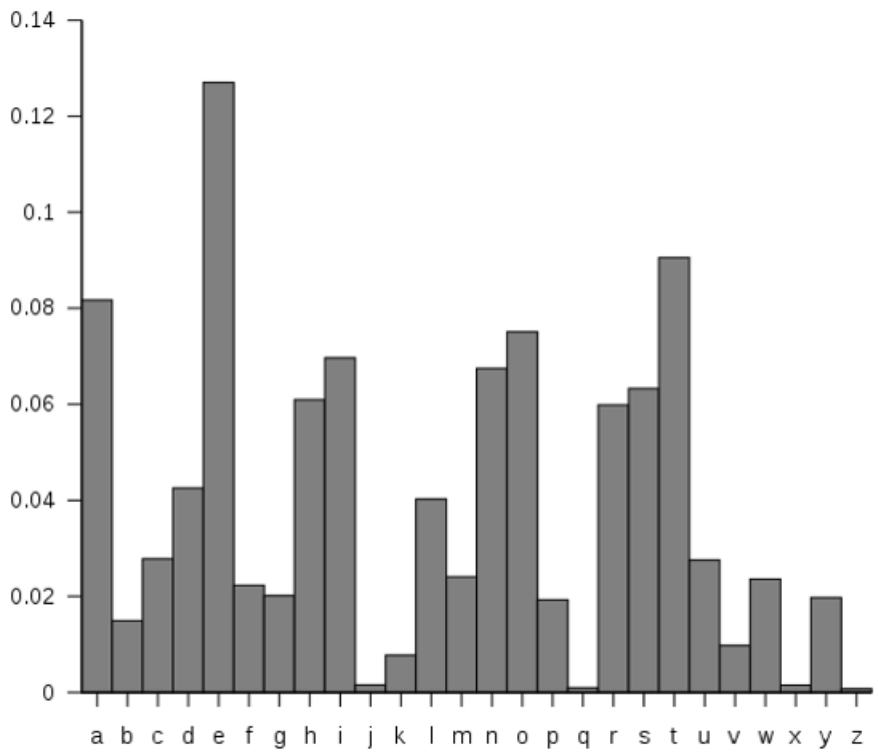
然后密文为：kt yg wo ok gy nl hj of cm yg kg lm mb wf

将密文变成大写，然后几个字母一组排列。

如5个一组就是KTYGW OOKGY NLHJO FCMYG KGLMM BWF

1.8.频率破解法

字母频率，就是指各个字母在文本材料中出现的频率。统计表明，在英语语料中各个字母的频率分布是有规律的，比如最常见的字母是e。英语中各个字母大致的频率分布如下图所示：

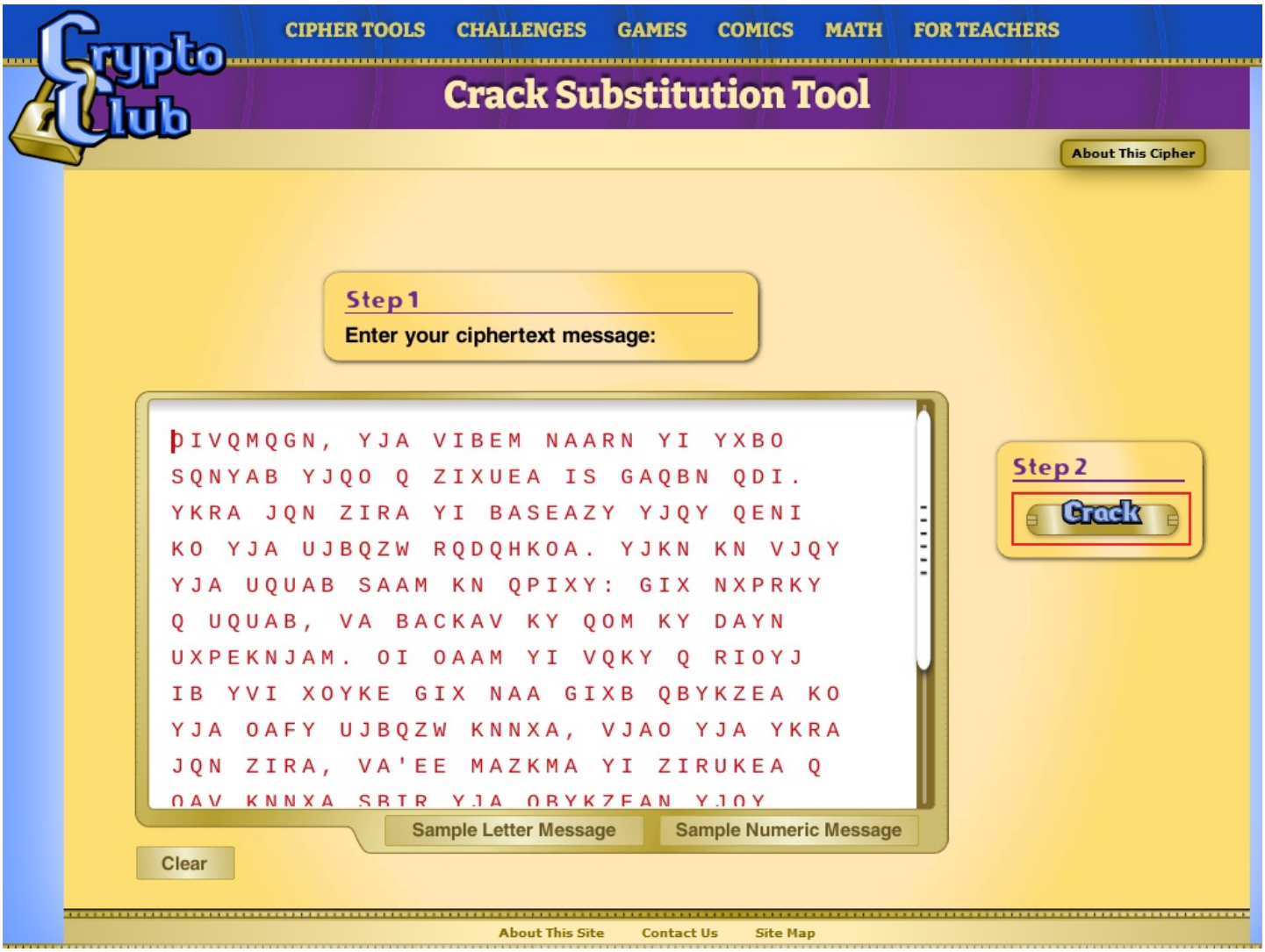


打开secret.txt文件，得到的内容如下：oivqmqgn, yja vibem naarn yi yxbo sqnyab yjqo q zixuea is gaqbn qdi. ykra jqn zira yi baseazy yjqy qeni ko yja ujbqzw rqdqhkoa. yjkn kn vjqy yja uquab saam kn qpixy: gix nxprky q uquab, va backav ky qom ky dayn uxpeknjam. oi oaam yi vqky q rioyj ib yvi xoyke gix naa gixb qbykzea ko yja oafy ujbqzw knnxa, vjao yja ykra jqn zira, va’ee mazkma yi zirukea q oav knnxa sbir yja qbykzean yjqy jqca paao nxprkyyam. yjqy’n pqnkzqeeg ky. qom dbqp gix seqd jaba, zbguyiiiniziieqrkbkdjy?

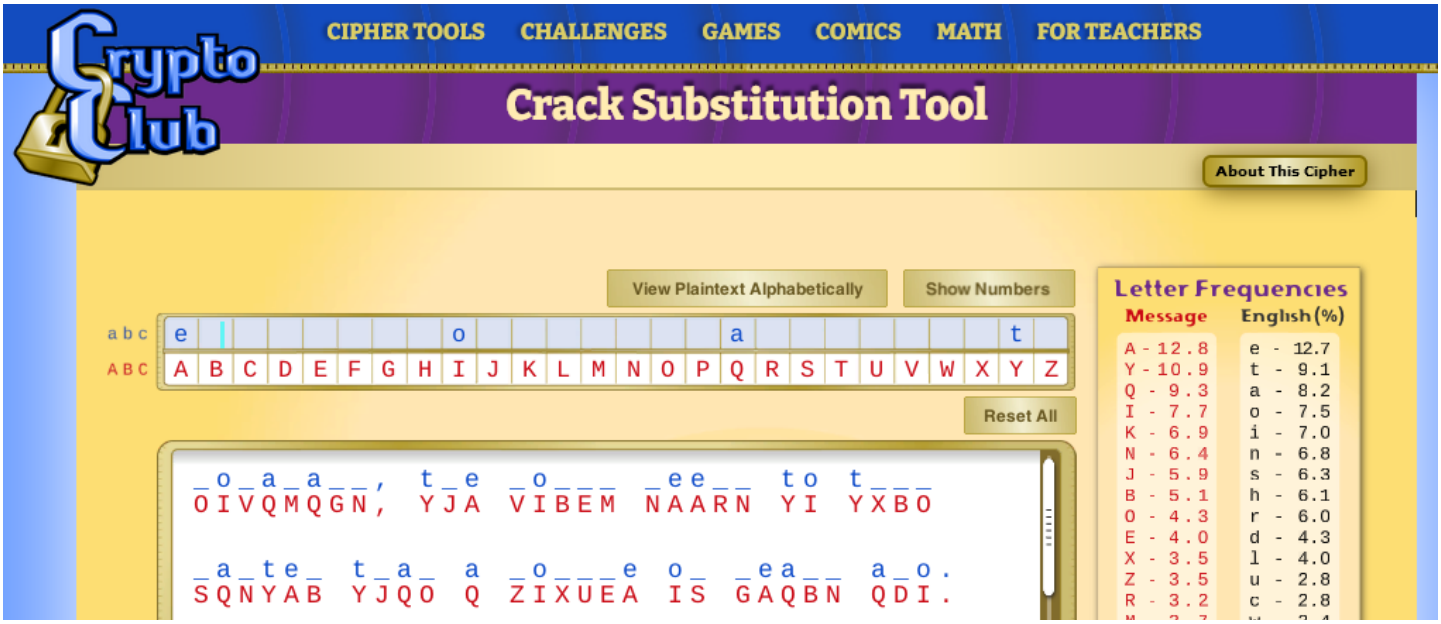
这里可以尝试对这段文本进行凯撒密码变换，但是尝试1-13这些偏移都看不到任何有意义的结果，因为这里不再使用简单的凯撒变换了，这里使用的是任意的单表代替。在凯撒密码中，所有的字符经过变换后，他们的偏移量都是一样的，比如a经过变换后得到d，那么b经过变换后就是e；而在任意的单表代替中，每个字符都唯一映射到另一个字符，字符串映射之间是没有规律的。

经过使用暴力方式对此类加密进行破解非常麻烦，但是可以使用语言的一些规律对其发起攻击。首先把字母使用的相对频率统计出来，与英文字母的使用频率分布进行比较，可以猜测出一部分映射，然后配合对英语中的构词规律的分析，就可以猜测出其他的映射，按照这个思路基本就能完成密码分析过程。

页面 https://cryptoclub.org/#vAllTools 提供了一个方便的操作界面供我们对此类问题进行分析。打开该页面然后填入密文，点击 Crack，如下图所示：



接下来就可以从频率上对密文进行分析了。根据右侧的频率统计，我们可以尝试对前面四个字母进行替换，在左侧的矩形文本框中，在密文对应字符的上面可以填写解密后的明文，因此，这里得到的密文到明文的映射为：A-e, Y-t, Q-a, I-o。



使用实验步骤二中提供的页面，可以完成对密文的破解，但是需要耗费一定的时间和精力，使用已有的成熟的解密脚本，我们可以快速完成破解过程。

页面 https://github.com/alexbers/substitution_cipher_solver 提供了一个Substitution Cipher Solver工具，可以快速完成对单表替换类密码的分析。从地址 https://github.com/alexbers/substitution_cipher_solver/archive/master.zip 可以下载到这个工具

2.现代密码

2.1.序列密码（流密码）

2.1.1.简介

序列密码的概念

序列密码也称为流密码（Stream Cipher），它是对称密码算法的一种。序列密码具有实现简单、便于硬件实施、加解密处理速度快、没有或只有有限的错误传播等特点，因此在实际应用中，特别是专用或机密机构中保持着优势，典型的应用领域包括无线通信、外交通信。 1949年Shannon证明了只有一次一密的密码体制是绝对安全的，这给序列密码技术的研究以强大的支持，序列密码方案的发展是模仿一次一密系统的尝试，或者说“一次一密”的密码方案是序列密码的雏形。如果序列密码所使用的是真正随机方式的、与消息流长度相同的密钥流，则此时的序列密码就是一次一密的密码体制。若能以一种方式产生一随机序列（密钥流），这一序列由密钥所确定，则利用这样的序列就可以进行加密，即将密钥、明文表示成连续的符号或二进制，对应地进行加密,加解密时一次处理明文中的一个或几个比特。

RC4的概念

RC4是由罗纳德·李维斯特在1987年开发出来的，虽然它的官方名是“Rivest Cipher 4”，但是首字母缩写RC也可以理解为"Ron's Code"。

RC4开始时是商业密码，没有公开发表出来，但是在1994年9月份的时候，它被人匿名公开在了Cypherpunks 邮件列表上，很快它就被发到了sci.crypt 新闻组上，随后从这传播到了互联网的许多站点。随之贴出的代码后来被证明是真实的，因为它的输出跟取得了RC4版权的私有软件的输出是完全相同的。由于算法已经公开，RC4也就不再是商业秘密了，只是它的名字“RC4”仍然是一个注册商标。RC4经常被称作是“ARCFOUR”或者"ARC4"（意思是称为RC4），这样来避免商标使用的问题。

RC4已经成为一些常用的协议和标准的一部分，如1997年的WEP和2003/2004年无线卡的WPA; 和1995年的SSL，以及后来1999年的TLS。让它如此广泛分布和使用的主要因素是它不可思议的简单和速度，不管是软件还是硬件，实现起来都十分容易。

雪崩效应

在密码学中，雪崩效应（avalanche effect）指加密算法（尤其是块密码和加密散列函数）的一种理想属性。雪崩效应是指当输入发生最微小的改变（例如，反转一个二进制位）时，也会导致输出的不可区分性改变（输出中每个二进制位有50%的概率发生反转）。

2.1.2.简介

相关涉及变量：

1）密钥流：RC4算法的关键是依据明文和密钥生成相应的密钥流，密钥流的长度和明文的长度是相应的。也就是说明文的长度是500字节，那么密钥流也是500字节。当然，加密生成的密文也是500字节。由于密文第i字节=明文第i字节^密钥流第i字节；

2）状态向量S：长度为256。S[0],S[1]…S[255]。每一个单元都是一个字节。算法执行的不论什么时候。S都包含0-255的8比特数的排列组合，仅仅只是值的位置发生了变换；

3）暂时向量T：长度也为256，每一个单元也是一个字节。

假设密钥的长度是256字节。就直接把密钥的值赋给T，否则，轮转地将密钥的每一个字节赋给T。

4）密钥K：长度为1-256字节。注意密钥的长度keylen与明文长度、密钥流的长度没有必定关系。通常密钥的长度取为16字节（128比特）。

具体实现算法：

1、初始化S和T
for i=0 to 255 do
S[i] =i;
T[i]=K[imodkeylen];

2、初始排列S
for i=0 to 255 do
j= (j+S[i]+T[i])mod256;
swap(S[i],S[j]);

3、产生密钥流
for r=0 to len do //r为明文长度，r字节
i=(i+1) mod 256;
j=(j+S[i])mod 256;
swap(S[i],S[j]);
t=(S[i]+S[j])mod 256;
k[r]=S[t];

最后将明文与秘钥流按位进行异或操作

解密操作：直接将密文与密钥流进行再次的异或操作

2.1.3.穷举破解

安全研究人员称，现在世界上近三分之一的HTTPS加密连接可被破解，并且效率极高。这种针对RC4加密的破解技术，同样也可以用来破解WiFi数据包。

研究人员很早前就发现可以利用RC4中的统计偏差，导致可对加密信息中的一些伪随机字节能进行猜测。在2013年，科学家利用这个漏洞设计了一次攻击实验：他们在2000小时内猜出一个基础身份认证cookie中包含的字符。后来技术改进后，研究人员只需约75小时猜解就能得到94%的准确率。（引用：‘https://www.freebuf.com/news/72622.html’）

破解长度大的RC4密文是一个浩大的工作，类似于128bits秘钥长度加密的密文，在我们个人计算机上是难以得到破解的，但我们可以结合CrypTool，对之前我们所加密过的密文进行破解分析，站在攻击者的角度，体会破解RC4密文关于时间复杂性的递增性。

2.2.IDEA加密算法

2.2.1.简介

1）IDEA是International Data Encryption Algorithm的缩写，即国际数据加密算法，它的原型是1990年由瑞士联邦技术学院X.J.Lai和Massey提出的PES。1992年，Lai和Massey对PES进行了改进和强化，产生了IDEA。这是一个非常成功的分组密码，并且广泛的应用在安全电子邮件PGP中。

2. IDEA加密算法是一个分组长度为64位的分组密码算法，密钥长度为128位,同一个算法即可勇于加密，也可用于解密。发展IDEA也是因为感到DES具有密钥太短等缺点。IDEA的密钥为128位，这么长的密钥在今后若干年内应该是安全的。这是基于“相异代数群上的混合运算”设计思想，算法运用硬件与软件实现都很容易，而且比DES算法在实现上快的多。IDEA自问世以来，已经经历了大量的详细审查，对密码分析具有很强的抵抗能力，在多种商业产品中被使用。

3）目前IDEA在工程中已有大量应用实例，PGP(Pretty Good Privacy）就使用IDEA作为其分组加密算法；安全套接字层SSL（Secure Socket Layer）也将IDEA包含在其加密算法库SSLRef中；IDEA算法专利的所有者Ascom公司也推出了一系列基于IDEA算法的安全产品，包括：基于IDEA的Exchange安全插件、IDEA加密芯片、IDEA加密软件包等。IDEA算法的应用和研究正在不断走向成熟。

4）对称密码算法（Symmetric cipher）：加密密钥和解密密钥相同，或实质上等同，即从一个易于推出另一个。又称传统密码算法（Conventional cipher)、秘密密钥算法或单密钥算法。

5）分组密码（Block cipher）：将明文分成固定长度的组，用同一密钥和算法对每一块加密，输出也是固定长度的密文。——DES、IDEA、RC2、RC4、RC5

分组密码是将明文消息编码表示后的数字（简称明文数字）序列，划分成长度为n的组（可看成长度为n的矢量），每组分别在密钥的控制下变换成等长的输出数字（简称密文数字）序列。

2.2.2.原理

任务一：IDEA加解密算法的原理
64-位数据分组被分成4个16-位子分组：xl，X2，x3，x4。这4个子分组成为算法的第一轮的输入，总共有8轮。在每一轮中，这4个子分组相列相异或，相加，相乘，且与6个16-位子密钥相异或，相加，相乘。在轮与轮间，第二和第：个子分组交换。最后在输出变换中4个子分组与4个子密钥进行运算。

在每一轮中，执行的顺序如下：（以下表述中的相加指的是两个数mod 2^256 相加，例如：（a + b) mod p，其结果是a+b算术和除以p的余数，也就是说，（a+b) = kp +r，则 (a+b) mod p =r,又例如对于下列表述中的“（2)X2和第二个子密钥相加”就是指用X2与第二个子密钥的和除以2^16(即65536)后的余数。对于以下表述中的相乘，指的是：（a × b) mod p，其结果是 a × b算术乘法除以p的余数，又例如对于下列表述中的“（1)X1和第一个子密钥相乘。”就是指用X1和第一个子密钥相乘后的积除于(2^16+1)(即65537)后的余数。异或指的是不进位加法。）

(1)X1和第一个子密钥相乘。

(2)X2和第二个子密钥相加。

(3)X3和第三个子密钥相加。

(4)X4和第四个子密钥相乘。

(5)将第（1)步和第（3)步的结果相异或。·

(6)将第（2)步和第（4)步的结果相异或。

(7)将第（5)步的结果与第五个子密钥相乘。

(8)将第（6)步和第（7)步的结果相加。

(9)将第（8)步的结果与第六个子密钥相乘。

(10)将第（7)步和第（9)步的结果相加。

(11)将第（1)步和第（9)步的结果相异或。

(12)将第（3)步和第（9)步的结果相异或。

(13)将第（2)步和第（10)步的结果相异或。

(14)将第（4)步和第（10)步的结果相异或。

每一轮的输出是第（11)、（12)、（13)和（14) 步的结果形成的4个子分组。将中间两个分组分组交换（最后一轮除外）后，即为下一轮的输入。

经过8轮运算之后，有一个最终的输出变换：

(1) X1和第一个子密钥相乘。

(2) X2和第二个子密钥相加。

(3) X3和第三个子密钥相加。

(4) X4和第四个子密钥相乘。

最后，这4个子分组重新连接到一起产生密文。

产生子密钥也很容易。这个算法用了52个子密钥（8轮中的每一轮需要6个，其他4个用与输出变换）。首先，将128-位密钥分成8个16-位子密钥。这些是算法的第一批8个子密钥（第一轮六个，第二轮的头两个）。然后，密钥向左环移25位后再分成8个子密钥。开始4个用在第二轮，后面4个用在第三轮。密钥再次向左环移25位产生另外8个子密钥，如此进行D算法结束。

解密过程基本上一样，只是子密钥需要求逆且有些微小差别，解密子密钥要么是加密子密钥的加法逆要么是乘法逆。（对IDEA而言，对于模256十1乘，全0子分组用256=-l来表示，因此0的乘法逆是0)。计算子密钥要花点时间，但对每一个解密密钥，只需做一次。

2.3.DES算法

2.3.1.简介

DES全称为Data Encryption Standard，即数据加密标准，是一种使用密钥加密的块算法，1977年被美国联邦政府的国家标准局确定为联邦资料处理标准（FIPS），并授权在非密级政府通信中使用，随后该算法在国际上广泛流传开来。需要注意的是，在某些文献中，作为算法的DES称为数据加密算法（DataEncryption Algorithm,DEA），已与作为标准的DES区分开来。

DES算法就是一个把64位的明文输入块变为64位密文输出块的算法,它所使用的密钥也是64位（其实只使用到了56位，其余8位位奇偶校验位）

DES算法的入口参数有三个：Key、Data、Mode。

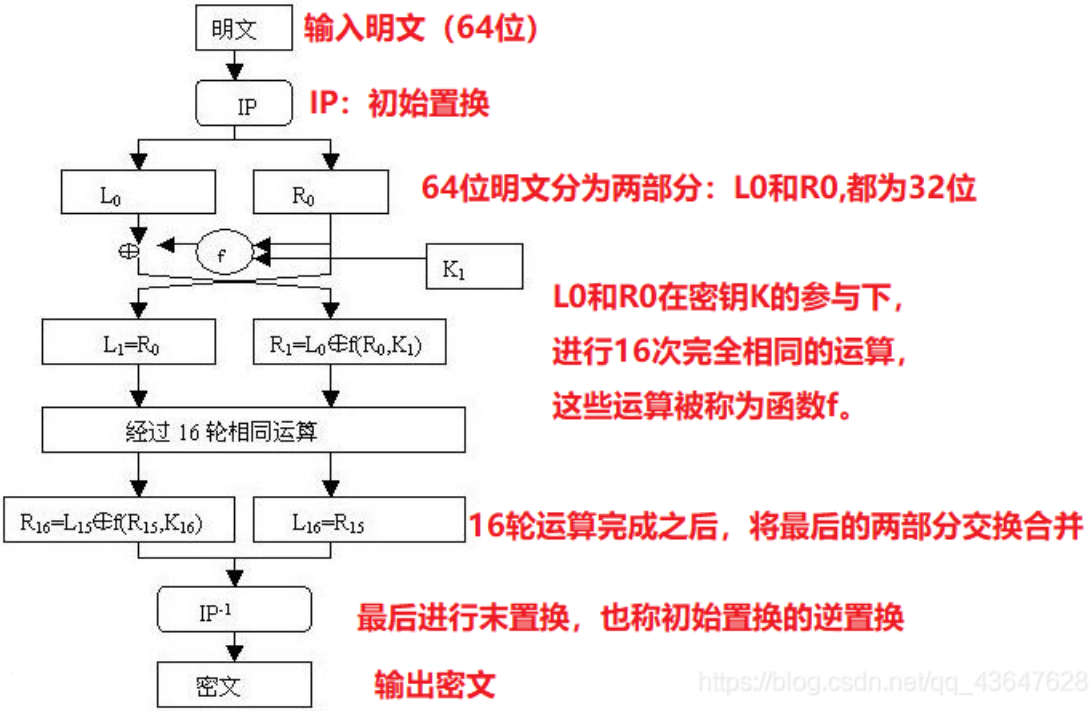
Key（密钥）：为7个字节共56位，是DES算法的工作密钥（若说密钥为64位，其指的也是56位的秘钥加上8位奇偶校验位，奇偶校验位为第8,16,24,32,40,48,56,64位）

Data（数据）：为8个字节64位，是要被加密或被解密的数据

Mode（模式）： 为DES的工作方式,有两种:加密或解密。

算法特点：分组比较短、密钥太短、密码生命周期短、运算速度较慢。

2.3.2.原理

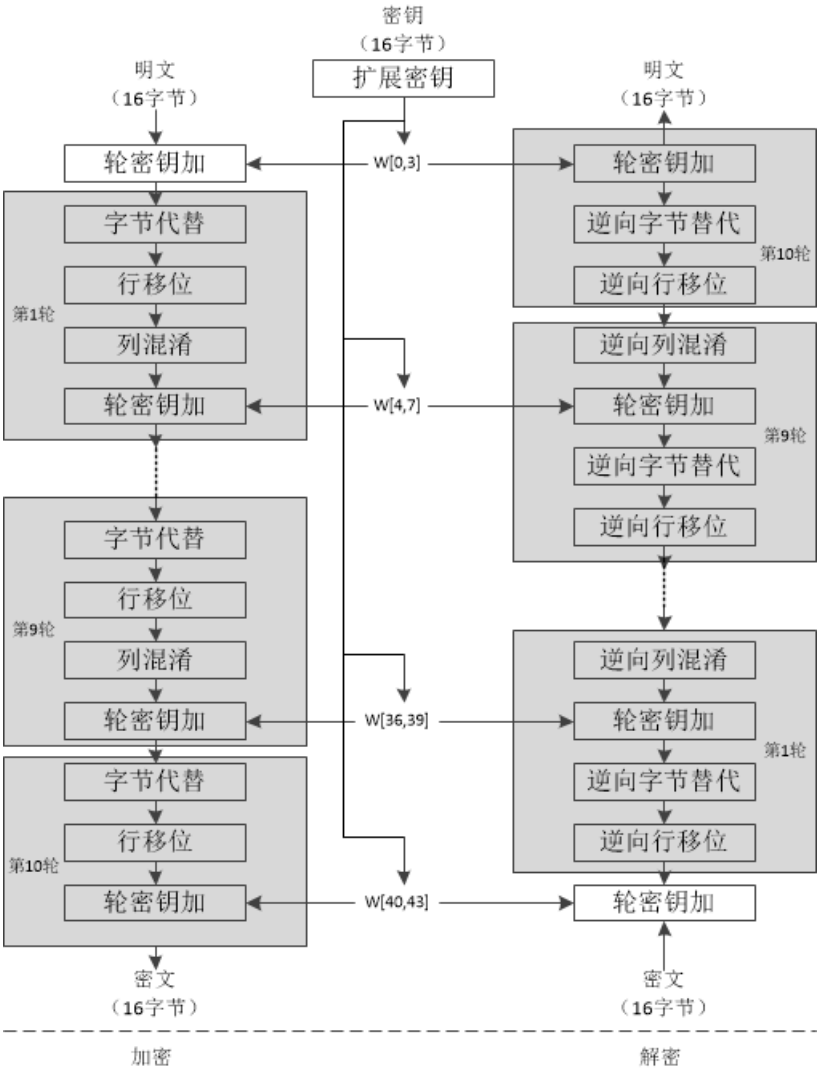


2.4.AES加密

2.4.1.简介

高级加密标准（Advanced Encryption Standard 缩写：AES），在密码学中又称Rijndael加密法，是美国联邦政府采用的一种区块加密标准。这个标准用来替代原先的DES，已经被多方分析且广为全世界所使用。经过五年的甄选流程，高级加密标准由美国国家标准与技术研究院（NIST）于2001年11月26日发布于FIPS PUB 197，并在2002年5月26日成为有效的标准。2006年，高级加密标准已然成为对称密钥加密中最流行的算法之一。

2.4.2.原理



AES的加密过程是在一个4×4的字节矩阵上运作，其初值就是一个明文区块（矩阵中一个元素大小就是明文区块中的一个Byte）。加密时，各轮AES加密循环（除最后一轮外）均包含4个步骤：

字节替代（SubBytes）—透过一个非线性的替换函数，用查找表的方式把每个字节替换成对应的字节。

行移位（ShiftRows）—将矩阵中的每个横列进行循环式移位。

列混淆（MixColumns）—为了充分混合矩阵中各个直行的操作。这个步骤使用线性转换来混合每内联的四个字节。最后一个加密循环中省略MixColumns步骤，而以另一个AddRoundKey取代。

轮密钥加（AddRoundKey）—矩阵中的每一个字节都与该次回合密钥（round key）做XOR运算；每个子密钥由密钥生成方案产生。

AES的解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。加解密中每轮的密钥分别由初始密钥扩展得到。算法中16字节的明文、密文和轮密钥都以一个4x4的矩阵表示。

3.密码学攻击