

CTF竞赛Crypto类题目 解题实训

主讲人：梅国浚

01

古典密码学

02

对称加密体制

03

非对称密码体制

04

Hash函数介绍

05

Base64编码

单表代换密码

明文密文，一对一

例子：

移位密码

埃特巴什码

基于密钥的单表代换密码

仿射密码

多表代换密码

明文密文，一对多

例子：

*Playfair*密码

*Polybius*密码（棋盘密码）

维吉尼亚密码

*Nihilist*密码

*Hill*密码

其他密码

大部分具有明显特征，易识别

例子：

摩尔斯电码、培根密码、猪圈密码、栅栏密码

单表代换密码：移位密码

概念：

通过把字母移动一定的位数来实现加密和解密。

例如：

$ABC \rightarrow CDE$

区分：

移位密码：移位针对每一个字符

凯撒密码：移位只针对英文字符

单表代换密码：基于密钥的单表代换密码

明文: $m = \text{CASEAR CIPHER IS A SHIFT SUBSTITUTION}$

密文: $C = \text{PHONHM PBKRNM BO H ORBEQ OSAOQBQSQBJI}$

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

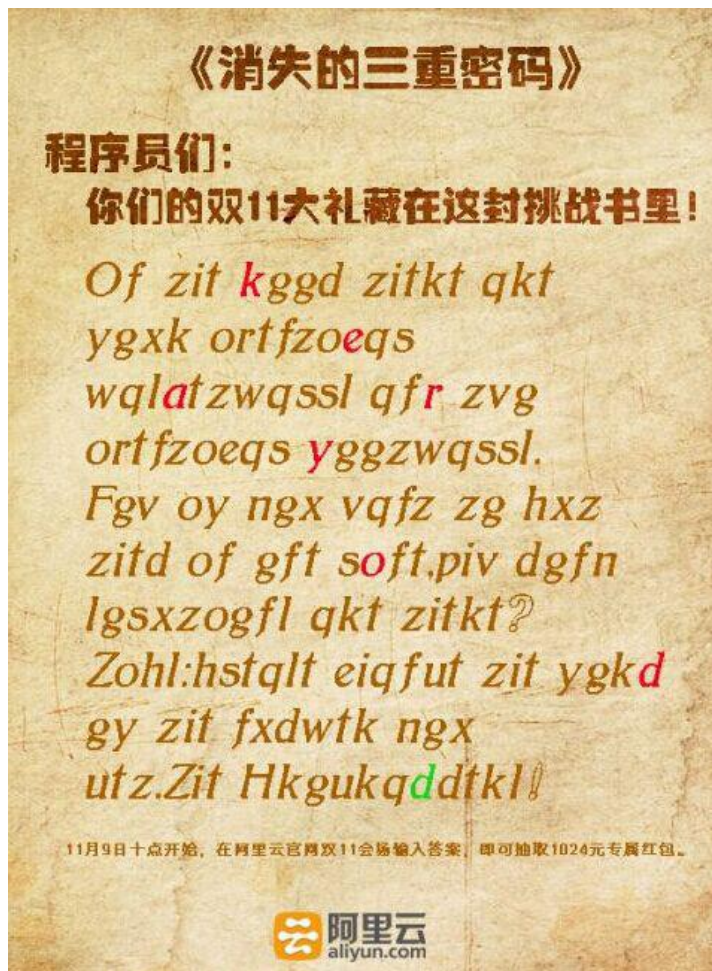
破解思路：字母、词组出现频率

字母: $E > T > A > O > I$

词组: $the > ing > and > her > ere$

在线网站: <http://quipqiup.com/index.php>

单表代换密码：基于密钥的单换密码



提示：keyboard



多表代换密码：维吉尼亚密码

维吉尼亚密码引入了“密钥”的概念，即根据密钥来决定用哪一行的密表来进行替换，以此来对抗字频统计。

加密过程：

明文：come greatwall

密钥：crypto

1. 填充明文。

明文	c	o	m	e	g	r	c	a	t	w	a	l	l
密钥	c	r	y	p	t	o	c	r	y	p	t	o	c

多表代换密码：维吉尼亚密码

2. 查表得到密文：*e f k t z f e r r l t z n*

明文	c	o	m	e	g	r	c	a	t	w	a	l	l
密钥	c	r	y	p	t	o	c	r	y	p	t	o	c

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

密钥

多表代换密码：维吉尼亚密码

破解方式：

1. 确定密钥长度（Kasiski测试法，重合指数法）
2. 确定密钥
3. 恢复明文

多表代换密码：Nihilist密码

又称关键字密码：明文+关键字=密文

加解密过程（关键字：helloworld）

1. 构建加解密矩阵

- ① 新建一个 5×5 的矩阵。
- ② 将字符不重复的依次填入矩阵(helowrd)
- ③ 剩下部分按照字母顺序依次填充
- ④ 字母i和j等价

	1	2	3	4	5
1	h	e	l	o	w
2	r	d	a	b	c
3	f	g	i	k	m
4	n	p	q	s	t
5	u	v	x	y	z

2. 加密过程

参照矩阵 M ，进行加密：

如： $a \rightarrow M[2,3] \rightarrow 23$

$t \rightarrow M[4,5] \rightarrow 45$

3. 解密过程

参照矩阵 M ，进行解密：

如： $23 \rightarrow M[2,3] \rightarrow a$

$45 \rightarrow M[4,5] \rightarrow t$

其他古典密码：培根密码

使用两种不同字体作为信息载体，分别代表A和B，结合加密表进行加密与解密

加密过程：

1. 待加密信息转换：

security

s	e	c	u	r	i	t	y
BAABA	AABAA	AAABA	BABAA	BAAAB	ABAAA	BAABB	BBAAA

2. 正体 → A，斜体 → B

3. 将信息嵌入文本，得到密文

To be, or not to be--that is the question: Whether 'tis

A aaaaa	J abaab	S baaba
B aaaab	K ababa	T baabb
C aaaba	L ababb	U babaa
D aaabb	M abbba	V babab
E aabaa	N abbab	W babba
F aabab	O abbba	X babbb
G aabba	P abbbb	Y bbaaa
H aabbb	Q baaaa	Z bbaab
I abaaa	R baaab	

密文样例: 

01

古典密码学

02

对称加密体制

03

非对称密码体制

04

Hash函数介绍

05

Base64编码

对称密码体制：

加密、解密密钥相同或者很容易从其中一个推出另一个

代表算法：DES、AES、RC4、A5

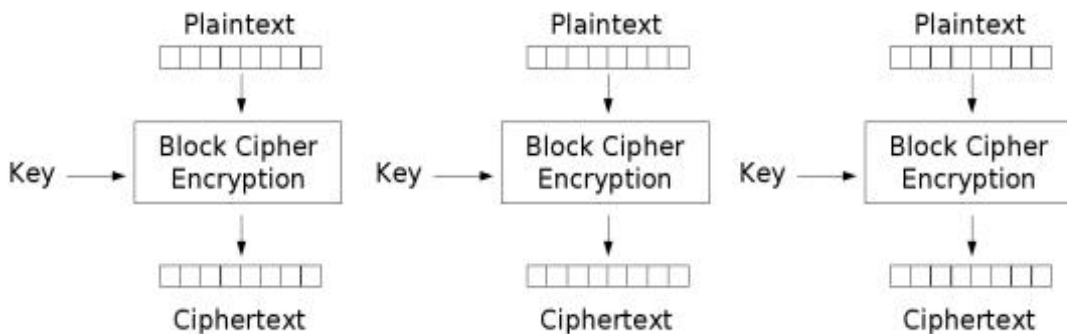
非对称密码体制：（公钥密码体制）

加密密钥与解密密钥没有直接关系

代表算法：RSA、ElGamal公钥密码体制、椭圆曲线公钥密码体制

分组密码：

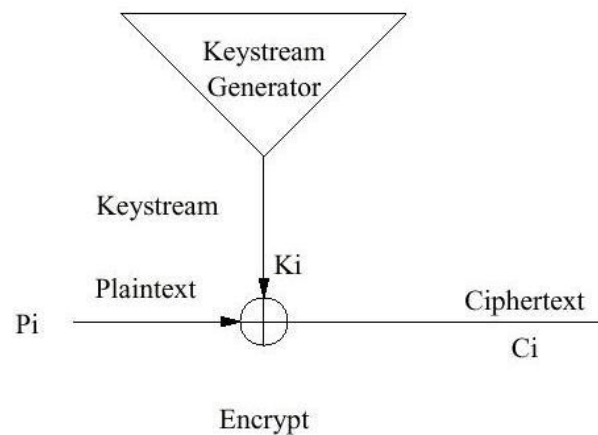
又称块密码，将明文消息的二进制序列划分成固定大小的块，每块分别在密钥控制下变换成等长的二进制密文序列。



分组密码加密示意图

序列密码：

又称流密码，将明文消息的二进制序列逐位加密，产生密文。

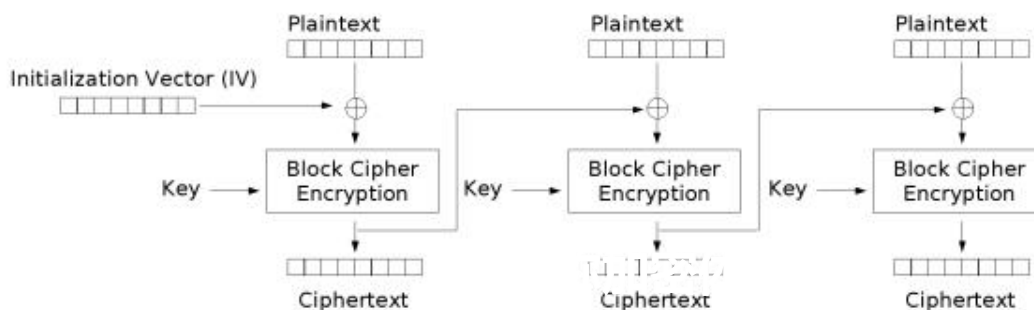


序列密码加密示意图

分组密码的工作模式：

- ◆ 前一个分组的加密结果会影响到下一个分组的加密结果：

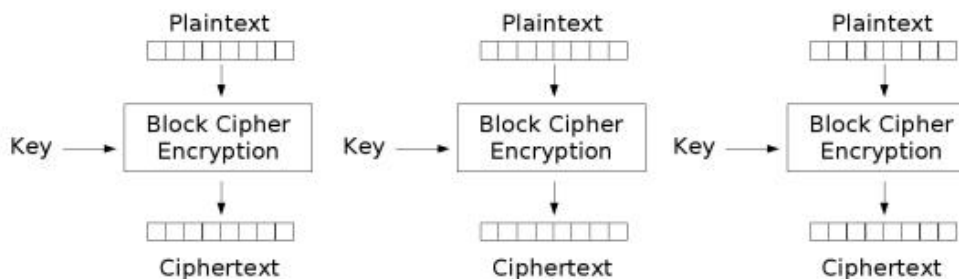
如：CBC模式，CFB模式，OFB模式



Cipher Block Chaining (CBC) mode encryption

- ◆ 前一个分组的加密结果和下一个分组独立：

如：CTR模式，ECB模式



Electronic Codebook (ECB) mode encryption

课堂练习：

We encrypted a flag with AES-ECB encryption using a secret key, and got the hash:

E220eb994c8fc16388dbd60a969d4953f042fc0bce25
dbef573cf5

22636a1ba3fafa1a7c21ff824a5824c5dc4a376e75

However, we lost our plaintext flag and also lost our key and we can't

seem to decrypt the hash back :

(. Luckily we encrypted a bunch of other flags with the same key. Can you recover the lost flag using this?

课堂练习解答:

选择明文攻击: 已知密文、一些用相同密钥加密得到的明密文对

AES-128算法:

密钥长度: 128bit

明文分块长度: 128bit = 16byte

密文分块长度: 128bit = 16byte

e220eb994c8fc16388dbd60a969d4953
abctf{looks_like

f042fc0bce25dbef573cf522636a1ba3
_you_can_break_a

fafa1a7c21ff824a5824c5dc4a376e75

es

01

古典密码学

02

对称加密体制

03

非对称密码体制

04

Hash函数介绍

05

Base64编码



非对称密码体制

RSA算法 ----- 基于大整数因子分解的 困难性

一、密钥生成

- 1、选取两个大素数 p 和 q
- 2、计算乘积，其中 $\varphi(n)$ 为 n 的欧拉函数：

$$n = p \times q$$

$$\varphi(n) = (p - 1)(q - 1)$$

- 3、随机选取整数 $e(1 < e < \varphi(n))$ 作为公钥，要求满足：

e 和 $\varphi(n)$ 互质

- 4、计算私钥 d ， d 满足：

$$d \times e \equiv 1 \pmod{\varphi(n)}$$

即： $d \equiv e^{-1} \pmod{\varphi(n)}$

则：公钥为 (e, n) ，私钥为 d

二、加密过程

- 1、把消息 M 分组为长度为 L ($L < \log_2^n$) 的消息分组：

$$M = m_1 m_2 \dots m_t$$

- 2、使用公钥加密明文：

$$c_i \equiv m_i^e \pmod{n} \quad (1 \leq i \leq t)$$

- 3、得到密文： $C = c_1 c_2 \dots c_t$

三、解密过程

- 1、把密文按长度为 L 分组得：

$$C = c_1 c_2 \dots c_t$$

- 2、用私钥 d 解密密文：

$$m_i \equiv c_i^d \pmod{n} \quad (1 \leq i \leq t)$$

- 3、得到明文消息：

$$M = m_1 m_2 \dots m_t$$

RSA算法攻击 ----- 共模攻击

攻击条件：

1. 当两个用户公钥(n,e)中模数n相同
2. 使用不同的e，加密同一个明文m

攻击原理：

设两个用户的公开钥分别为 e_1 和 e_2 ，且两者互素。明文消息是m，密文分别为：

$$c_1 \equiv m^{e_1} \bmod n$$

$$c_2 \equiv m^{e_2} \bmod n$$

当攻击者截获 c_1 和 c_2 后，就可如下恢复出明文m。用扩展欧几里得算法求出 $re_1 + se_2 = 1$ 的两个整数r和s，由此可得：

$$c_1^r c_2^s \equiv m^{re_1} m^{se_2} \bmod n$$

$$\equiv m^{(re_1 + se_2)} \bmod n$$

$$\equiv m \bmod n$$



非对称密码体制

RSA算法攻击 ----- 小公钥指数攻击

攻击条件：e特别小，比如e=3

攻击原理：考虑到加密关系满足：

$$C \equiv M^3 \pmod{n}$$

则：

$$M^3 = C + k * n, k \in N$$

$$M = \sqrt[3]{C + k * n}, k \in N$$

攻击方法：

对密文一次加一个N，然后开三次根号直到判断结果为整数为止。

编程工具：

- ◆ Python gmpy库：gmpy.root(a,b)，返回为一个元组(x,y)，其中x(mpz型变量)为a开b次方的值，y是判断x是否为整数的变量(1代表整数，0代表小数)
- ◆ Python gmpy2库：gmpy2.iroot(a,b)



非对称密码体制

RSA衍生算法 ----- Rabin 算法

一、密钥生成

1、选取两个大素数 p 和 q

2、计算乘积： $n = p \times q$

Rabin算法的特征就在于 $e=2$

二、加密

1、选取明文空间 $P = \{0, \dots, n - 1\}$

2、明文 $m \in P$

3、则密文为：

$$c = m^2 \bmod n$$

三、解密

1、计算出 m_p 和 m_q ：

$$m_p = \sqrt{c} \bmod p$$

$$m_q = \sqrt{c} \bmod q$$

2、用扩展欧几里得计算出 y_p 和 y_q ：

$$y_p \times p + y_q \times q = 1$$

3、解出四个明文（其中一个为正确答案）

$$a = (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \bmod n$$

$$b = n - a$$

$$c = (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \bmod n$$

$$d = n - c$$

PS：如果 $p \equiv q \equiv 3 \pmod{4}$ ，则：

$$m_p = c^{\frac{1}{4}(p+1)} \bmod p$$

$$m_q = c^{\frac{1}{4}(q+1)} \bmod q$$

01

古典密码学

02

对称加密体制

03

非对称密码体制

04

Hash函数介绍

05

Base64编码

关于Hash函数

- ◆ 又称为：数字指纹、消息摘要、散列值
- ◆ 将任意长度的输入变换得到固定长度的输出
- ◆ 用于确保消息的完整性
- ◆ 单向性：对于任意消息 x ，计算 $H(x)$ 容易，相反则很难实现
- ◆ 破解方式：暴力破解

Hash算法分类种类

算法类型	输出Hash值长度
MD5	128bit/256bit
SHA1	160bit
SHA256	256bit
SHA512	512bit

Hashcat工具介绍

目前最好的基于CPU和GPU破解HASH软件

使用：Hash类型+字典文件+（支持格式的文件）

通过-m参数指定HASH类型

#	Name	Category
900	MD4	Raw Hash
0	MD5	Raw Hash
5100	Half MD5	Raw Hash
100	SHA1	Raw Hash
10800	SHA-384	Raw Hash
1400	SHA-256	Raw Hash
1700	SHA-512	Raw Hash
5000	SHA-3(Keccak)	Raw Hash
10100	SipHash	Raw Hash
6000	RipeMD160	Raw Hash
6100	Whirlpool	Raw Hash

常规使用说明：

- [Basic Examples] -		
Attack-Mode	Hash-Type	Example command
Wordlist	\$P\$	hashcat -a 0 -m 400 example400.hash example.dict
Wordlist + Rules	MD5	hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force	MD5	hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a
Combinator	MD5	hashcat -a 1 -m 0 example0.hash example.dict example.dict

01

古典密码学

02

对称加密体制

03

非对称密码体制

04

Hash函数介绍

05

Base64编码

编码原理：

1. 按照字符串长度，每3个字符分为一组，将字符ASCII码转换为8bit二进制，得到一组 $3 \times 8 = 24\text{bit}$ 的数据。当余下字符不足以构成24bit时，差额部分用0填充
2. 将这24bit分为4个6bit的字节
3. 将每个6bit转换为10进制数，对照Base64码表，得到对应编码后的字符

举例：

字符	L	U	C	
ASCII 字节	76	117	99	
8bit 字节	01001100	01110101	01100011	
6bit 字节	010011	000111	010101	100011
十进制	19	7	21	35
对应编码	T	H	V	j

Base64编码

字符	L	U	C	Y	Y			
ASCII 字节	76	117	99	121	121			
8bit 字节	01001100	01110101	01100011	01111001	01111001	00000000		
6bit 字节	010011	000111	010101	100011	011110	010111	100100	000000
十进制	19	7	21	35	30	23	36	异常
对应编码	T	H	V	j	e	X	k	=

若Base64编码后有1个'=',
则
一次可以隐藏2bit信息

字符	L	U	C	Y				
ASCII 字节	76	117	99	121				
8bit 字节	01001100	01110101	01100011	01111001	00000000	00000000		
6bit 字节	010011	000111	010101	100011	011110	010000	000000	000000
十进制	19	7	21	35	30	16	异常	异常
对应编码	T	H	V	j	e	Q	=	=

若Base64编码后有2个'=',
则
一次可以隐藏4bit信息

Thanks for watching

谢谢