目录

第一章　绪论

1.1　什么是密码学

如果我们要求一个从未接触过密码学的人处理一段文字，把这段文字尽可能地加密，让别人无法破解。那么，大多数人在深思熟虑之后，总能提出一些加密算法。这时候，一般人的思路可能会有几个方向。

有的人的方向是把这段文字中的字母之间通过各种复杂的运算进行组合。比如说，把要加密的文字中每两个字母在字母表中的位置进行相加，形成密码。比如说，“cryptography” 中，“cr” 变成了 $3+18=21$, “yp” 变成了 $25+16=41$, “cryptography” 对应的密码就是 “214135252133”。但这样的密码，显然忽略了一点：密码的可解密性。我们之所以要进行加密，是为了安全地传递信息。但接收信息者必须要有解密的方法。但这种加密方式则没有对应的解密方法。所以说，这不是一个合格的加密方式。

这时，足够聪明的人，则考虑到了解密的方法。他们想出了一些类似于古代使用的加密方法。比如说，著名的凯撒密码：把一段由英文字母组成的语句，每个字母都在字母表中往后移 3 个位置。“veni vidi vici” 也就变成了 “yhql ylgl ylfl”. 这种密码的解密方法，也就是每个字母在字母表中向前移动 3 个位置。当问起这种密码体制里最重要的是什么，大部分人都会说是加密算法。如果把加密算法告诉了别人，那这种密码就相当于被破解了。

如果进而问起一个密码体制的组成，大部分人的回答，就是加密算法。也许有些自诩比普通人聪明一点的人，还会加上一个解密算法，也就是，将处理过后的密码变为正常文字的算法。如果用 m 代表要被加密的文字，c 代表被加密出来的密码，E() 代表加密算法，D() 代表解密算法，那么大多数人理解的密码学，本质就是以下这两个式子:

$$c = \mathrm{E}(m)$$
$$m = \mathrm{D}(c)$$

1.1.1　密钥

但是，我们考虑一下实际的情况。根据我们之前的常识，会发现，如果要用密码来传递信息，首先通信的双方必须要在一个安全信道中传递一些额外的信息。比如说，告诉接收者加密的方式，或者更直接地，告诉接收者解密算法。不存在一种加密的方式，能不事先通过安全信道传递信息，而使得只有接收者能够解密。那么，为了使通信更加安全，通信双方对安全信道的使用应该尽可能少。如果 Alice 和 Bob 使用凯撒密码进行通信，而 Alice 事先在安全信道中要对 Bob 说 “我的这种加密方式是把一段由英文字母组成的语句，每个字母都在字母表中往后移 3 个位置。” 这段话是如此之长，而如果考虑是在战场上，由通信员用摩尔斯电码发送，那么花费的这么长的时间显然是不合理的。再者，现代的通信方式一般都是把信息编码成二进制进行传递，那么，相比把一段话编码成二进制，不如找一些数字来代表这个密码。在我们

的这段思想博弈中，一个重要的概念呼之欲出。

拿凯撒密码为例，在其加密算法中，还有一个关键的量：3. 试想，如果通信双方 Alice 和 Bob, Alice 在场的所有人她用的是凯撒密码的加密方式加密她的话，又在安全信道中告诉 Bob“4”, 意味着她使用的凯撒密码里，是将每个字母在字母表的位置上向后移动 4 个位置。那么，即使在场有人能偷听到 Alice 给 Bob 的密码，也无法破解 Alice 想说的是什么，只有掌握了规则“向后移动 4 个位置”的 Bob, 才能正确地破解密码。

如果还是按照之前的说法，一个密码体制包括加密算法和解密算法，那么向后移动 3 个位置的凯撒密码和向后移动 4 个位置的凯撒密码，就成了两个密码体制。但是，这两种加密方式是极其类似的，差别也就只在于一个向后移动 3 个位置，一个向后移动 4 个位置。我们对于这些密码的研究，也许也就十分类似。因此，把这两种加密方式归类为同一种密码体制，似乎是更好的选择。因此，**密钥**就应运而生了。所谓密钥，我们可以粗略地理解成加密算法、解密算法中的参数，也就是我们之前说的 3、4. 通过一个密码体制通信的双方，需要首先在保密信道中确定密钥。而一个密码体制，也就可以由加密算法、解密算法和密钥构成。因此，如果以 k 代表密钥，那么之前的式子就变成了

$$c = \mathrm{E}_k(m)$$
$$m = \mathrm{D}_k(c)$$

这也就是现在通用的密码学。

那么我们思考几个问题：在一次加密通信过程中，加密算法和解密算法中使用的密钥是否必须要相同？密钥是否只能是一个数？可不可以没有密钥？

针对第一个问题，确实存在某些高端的技巧，使得加密算法和解密算法使用的密钥是不同的。事实上，在密码学领域中，根据加密算法和解密算法使用的密钥是否相同，人们将加密方式分为对称加密与非对称加密，分别对应使用同一密钥和使用不同密钥。非对称加密方式也许难以理解，我们也会在充分学习对称加密后再介绍非对称加密。所以，我们接下来讨论的对称加密过程中，请大家记住，加密算法和解密算法使用的是相同的密钥。

此外，对于对称加密算法，密钥也并不一定是一个数。比如说，加密算法

$$\mathrm{E}_{a,b}(m) = am + b$$

其密钥为 (a, b), 但我们仍认为其使用单一密钥，也就是说，把 (a, b) 看作一个密钥。

此外，可不可能不存在密钥呢？确实有这样的密码体制，但这样却也十分不安全。比如说，加密算法

$$\mathrm{E}(m) = m$$

就是一个无密钥的加密算法。

1.2 密码体制的基本要素

根据之前的讨论，我们就可以得出一个密码体制的基本要素：

- 明文空间 $\mathcal{M}$

  所有可以被加密算法加密的元素组成的集合，加密算法的定义域。明文空间的元素叫做明文 $m \in \mathcal{M}$.

  例如，在凯撒密码中，明文空间就为所有由英文字母组成的字符串。

- 密文空间 $\mathcal{C}$

  所有可以由加密算法输出的元素组成的集合，加密算法的值域。密文空间的元素叫做密文 $c \in \mathcal{C}$.

  我们需要注意到的是，这里的值域，可以理解成二元函数 $\mathrm{E}(k, m)$ 的值域。比如说，对于加密算法

  $$\mathrm{E}_k(m) = m^2 + k^2$$

  其密文空间为 $[0, +\infty)$ 而非 $[k^2, +\infty)$.

- 密钥空间 $\mathcal{K}$

  所有密钥组成的集合。密钥空间的元素叫做密钥 $k \in \mathcal{K}$.

  在非对称加密中，密钥空间分为加密密钥空间和解密密钥空间。

- 加密算法 $\mathrm{E}_k(m)$

  根据密钥生成的特定算法，将明文转化为密文。

- 解密算法 $\mathrm{D}_k(c)$

  根据密钥生成的特定算法，将密文转化为明文。

  对于对称加密算法，也就是只使用一个密钥的加密体制，解密算法与加密算法满足

  $$\mathrm{D}_k(\mathrm{E}_k(m)) = m$$

在我们讨论密码体制的一些性质时，密钥生成算法有时也是必要的。什么是密钥生成算法呢？回忆之前 Alice 和 Bob 的例子，凭什么 Alice 选择的密钥是 4 而不是 25 呢？这就涉及到了密钥生成算法。在这个例子中，密钥生成算法就是 Alice 自己想到哪个密钥就输出哪个密钥。但是，从严格意义上来讲，密钥生成算法是一种概率算法。所谓概率算法，就是在算法的步骤中涉及到了某些概率。比如说，在某个密钥生成算法中，在 $(0, 1)$ 中等概率随机生成一个数 t, 而生成的密钥 k 满足

$$k = \begin{cases} 1 & t \in (0.5, 1) \\ 0 & t = 0.5 \\ -1 & t \in (0, 0.5) \end{cases}$$

这就是一个典型的概率算法。其特点就是在两次运行中输出的结果不一定相同。

因此，我们称一个加密方案包含三个要素：加密算法 E, 解密算法 D, 密钥生成算法 G. 根据定义，我们可以说，一个密码体制由一个加密方案 $(\mathrm{E}, \mathrm{D}, \mathrm{G})$ 及一个明文空间 $\mathcal{M}$ 完全定义。

因此，Alice 和 Bob 的一次加密通信的过程包括：

1. Alice 根据密钥生成算法 G 生成密钥 $k \in \mathcal{K}$.
2. Alice 通过安全信道将 k 告诉 Bob.
3. Alice 将想要传达的明文 $m \in \mathcal{M}$ 根据加密算法加密成密文 $c = \mathrm{E}_k(m)$ 告诉 Bob.
4. Bob 根据之前 Alice 告诉自己的 k、密文 c 及解密算法得出明文 $m = \mathrm{D}_k(c)$

那么，在一个密码体制中，哪个最重要呢？是不是之前我们说的加密算法呢？这里，就不得不提 Kerckhoffs 原则。用现代的语言来说，Kerckhoffs 原则阐述的是：

提倡安全性不能建立在对算法的保密上。

也就是说，我们如果要证明一个加密体制的安全性，不能指望算法的保密性。我们应默认加解密算法可以被所有人知道（事实上也确实如此）。也就是说，真正值得保密的，是密钥。如果潜在的敌手获得密钥，那么根据公开的解密算法，那么他就可以从窃得的密文中获得明文。

1.3 经典密码

1.3.1 基础概念

我们讨论了密码体制的基本要素之后，就可以介绍一些经典的密码，让大家更好地理解这些术语了。

值得指出的是，这些密码都是古代欧洲人的研究成果，当时并没有如今“数字化”的概念。因此，这些密码，都是针对拉丁字母进行的加密。因此，我们首先要引入一些概念：

函数 $C(m)$ 将拉丁字母 m 映射到它在字母表中的位置上，比如 $C(a)=1, C(z)=26$. 函数 $I(n)$ 将位于 1 和 26 之间的数字映射到字母表中相应位置的拉丁字母上，比如 $I(1)=a, I(26)=z$.

在讨论经典密码时，一些极其基础的数论记号及知识可以让我们更加方便、更加简洁地叙述、理解这些经典密码。

我们用 $\gcd(a,b)$ 表示 a 与 b 的最大公因数。

用 $a \bmod b$ 表示整数 a 除以 b 后的余数（取值范围为 0 到 $b-1$），比如说 $15 \bmod 6 = 3, 12 \bmod 6 = 0, (-4) \bmod 6 = 2$.

若 $a \bmod b = 0$, 即 a 能整除 b, 我们则记为 $a \mid b$. 如 $2 \mid 4, 3 \nmid 4$.

若 $(a-b) \mid c$, 我们则称 a 与 b 模 c 同余，记作 $a \equiv b \pmod{c}$. 如 $16 \equiv 23 \pmod{7}$.

对于整数 a, b, 若存在整数 c 使得 $ac \equiv 1 \pmod{b}$, 则称 c 为 a 在模 b 时的逆，记作 a^{-1}. 并非所有的整数都有逆，如在模 4 的情况下，整数 2 就没有逆。对于整数 a, b, 在模 b 时 a 存在逆的充分必要条件为 $\gcd(a,b)=1$.

1.3.2 单表代换密码

单表代换密码的典型，就是凯撒密码。如果用我们上述的记号来表示凯撒密码的过程，那么如果设明文为字符串 “$m_1 m_2 \cdots m_n$”, 密文为字符串 “$c_1 c_2 \cdots c_n$”, m_i, c_i 均代表一个拉丁字母。凯撒密码的加密算法

$$c_i = \mathrm{E}(m_i) = C((I(m_i)+3) \bmod 26) \tag{1.1}$$

解密算法

$$m_i = \mathrm{D}(c_i) = C((I(c_i)-3) \bmod 26) \tag{1.2}$$

它通过对字母表中每个字母进行固定的代换，得到密码。单表替换密码则是凯撒密码的推广，引入了密钥。从数学意义上，可以作如下定义：

设明文为字符串 “$m_1 m_2 \cdots m_n$”, 密文为字符串 “$c_1 c_2 \cdots c_n$”, m_i, c_i 均代表一个拉丁字母。如果把整数对 (a,b) 作为密钥，其中 $a \neq 0$, 那么其加密算法

$$c_i = \mathrm{E}_{a,b}(m_i) = C((aI(m_i)+b) \bmod 26)$$

解密算法

$$m_i = \mathrm{D}_{a,b}(c_i) = C\left(\left(a^{-1}(I(c_i) - b)\right) \bmod 26\right)$$

其中 a^{-1} 为 a 模 26 的逆。

事实上，如果我们令 $n_i = I(m_i), q_i = I(\mathrm{E}_{a,b}(m_i)), e_i = I(\mathrm{E}_{a,b}(m_i)), d_i = I(\mathrm{D}_{a,b}(\mathrm{E}_{a,b}(m_i)))$，那么

$$q_i \equiv an_i + b \pmod{26}$$

故

$$\begin{aligned} d_i &\equiv a^{-1}(q_i - b) \\ &\equiv a^{-1}(an_i + b - b) \\ &\equiv n_i \pmod{26} \end{aligned}$$

也就是说，

$$I(\mathrm{D}_{a,b}(\mathrm{E}_{a,b}(m_i))) \equiv m_i \pmod{26}$$

故

$$\mathrm{D}_{a,b}(\mathrm{E}_{a,b}(m_i)) = m_i$$

这也就证明了这个加密算法是正确的算法。

让我们不要再纠结于繁琐的数学符号，我们来从直观上看一看这个加密算法。任意取一个密钥，比如说，$a = 3, b = 7$, 就会对应的生成一张加密表和解密表：

表 1.1: $a = 3, b = 7$ 时的加密表

明文	a	b	c	d	e	f	g	h	i	j	k	l	m
密文	h	k	n	q	t	w	z	c	f	i	l	o	r
明文	n	o	p	q	r	s	t	u	v	w	x	y	z
密文	u	x	a	d	g	j	m	p	s	v	y	b	e

表 1.2: $a = 3, b = 7$ 时的解密表

密文	a	b	c	d	e	f	g	h	i	j	k	l	m
明文	p	y	h	q	z	i	r	a	j	s	b	k	t
密文	n	o	p	q	r	s	t	u	v	w	x	y	z
明文	c	l	u	d	m	v	e	n	w	f	o	x	g

那么我们根据这张表，就可以很快地进行加密和解密的工作了。

我们再回到之前所说的加密体制的基本要素：其明文空间 $\mathcal{M}$ 为由拉丁字母组成的任意长度的字符串组成的集合，密文空间 $\mathcal{C} = \mathcal{M}$. 密钥空间 $\mathcal{K} = \{(a, b) \mid a, b \in \mathbb{Z}, \gcd(a, 26) = 1\}$(这里 $\gcd(a, 26) = 1$ 的条件是因为解密算法中要求 a 模 26 的逆存在。).

1.3.3 多表代换密码

为了进一步提高安全性，古代的人们想到也许一张表并不足够安全，不妨使用多张表。因此，多表代换密码应运而生。

假设一共有 t 张加密表，人们是怎么做的呢？从明文的第一个字符开始，第一个字符使用第一张加密表进行加密，第二个字符使用第二张加密表进行加密，以此类推，第 t 个字符使用第 t 张加密表进行加密。到了第 $t+1$ 个字符，则又回到第一张加密表进行加密。用数学的语言怎么叙述这件事呢？

对于明文字符串 $M=m_1m_2\cdots m_n$, 我们要求其长度满足 $n=tp$. 我们将明文字符串等分成 t 个列向量 $M_1,M_2,\ldots,M_t$, 其中 $M_i=\left(m_{t(i-1)+1},m_{t(i-1)+2},\ldots,m_{t(i-1)+p}\right)^{\mathrm{T}}$. 对密文字符串 C 也作同样的划分 $C_1,C_2,\ldots,C_t$. 取密钥为 (A,B), 其中矩阵 A 为 $p\times p$ 的可逆矩阵，且满足 $\gcd\left(|A|,26\right)=1$, B 为 p 维列向量。

那么多表代换密码的加密算法为

$$C_i=\mathrm{E}_{A,B}\left(M_i\right)=C\left(\left(AI\left(M_i\right)+B\right)\bmod 26\right) \tag{1.3}$$

解密算法为

$$M_i=\mathrm{D}_{A,B}\left(C_i\right)=C\left(\left(A^{-1}I\left(C_i-B\right)\right)\bmod 26\right) \tag{1.4}$$

其中 A^{-1} 满足

$$A^{-1}A\bmod 26=\begin{pmatrix}1&0&\cdots&0\\0&1&\cdots&1\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&1\end{pmatrix}$$

1.3.4 一次一密

上述的单表代换密码和多表代换密码看似十分安全，但是如果用来加密由拉丁字母组成的用语言逻辑形成的一句话时，却有一个致命的弱点。虽然这些密码不会直接暴露明文，但却会暴露明文中各个字母出现的频率。我们知道，在任何一门由字母组成的语言文字中，每个字母出现的频率在语句十分长时是趋向于一个定值的。比如说，在英文中，有一个著名的短语：“ETAOIN SHRDLU”. 这个短语是英文中出现频率最高的 12 个字母，从高到低排列。根据这些频率，也是一个可以用于攻破这两种密码的方法。那么，有没有什么方法能使密文不显示明文中每个字母出现的频率呢？一次一密的方法就是答案。

为了加密某个长度为 n 的字符串，我们取另一个长度为 n 的字符串作为密钥。所得的密文就是明文每个字符在字母表中的位置与密钥每个字符在字母表中的位置相加。由密文得到明文也就是密文中每个字符在字母表中的位置与密钥每个字符在字母表中的位置相减。

这种方式之所以称为一次一密，是因为同一串密钥只能使用一次。试想如果有人窃得了用同一串密钥加密的两个密文 C_1,C_2, 将这两个字符串中的每个字符按其在字母表中的位置相减，那么如果出现 0, 那么对应位置就就有可能是出现频率比较高的几个字母。当然，更严谨的论证可以在之后看到。

1.4 加密系统的安全性

为了从数学上定义加密系统的安全性，我们必须引入一些和概率有关的定义。