软件供应链安全技术

预览版

荣亮（著）

For them

I owe much inspiration to both my parents. My mother Jannie constantly demonstrated me that computer graphics will never improve nature. She also converted one of my first METAPOST graphics into a patchwork that will remind me forever that handcraft is more vivid than computer artwork. My father Hein has spent a great deal of his life teaching math, and I m sure he would have loved METAPOST. I inherited his love for books. I therefore dedicate this document to them.

Colofon

This manual is typeset with CONTEXT MKIV. No special tricks are used and everything you see in here, is available for CONTEXT users. The text is typeset in Palatino and Computer Modern Typewriter. We used LUATEX as TEX processing engine. Since this document is meant to be printed in color, some examples will look sub–optimal when printed in black and white.

Graphics

The artist impression of one of Hasselts canals at page ?? is made by Johan Jonker. The CDROM production process graphic at page ?? is a scan of a graphic made by Hester de Weert.

Copyright

Publisher

publisher: Boekplan, NL
isbn-ean: 978-94-90688-02-8
website: www.boekplan.nl

Info

internet: www.pragma-ade.com
support: ntg-context@ntg.nl
context: www.contextgarden.net

序言

TBD

荣亮

2019年April月于苏州

目录

1 环游密码世界 5

1.1 密码学中的基本概念 5
1.2 对称加密算法和非对称加密算法 6
1.3 其他密码技术 7
1.4 信息安全所面临的威胁及对策 8
1.5 密码与信息安全常识 9
1.6 本章小结 10

2 密码的历史典故 11

2.1 中国古代人民怎么加密？ 11
2.2 凯撒密码 12
2.3 简单替换密码 13
2.4 复式替换密码：Enigma 20
2.5 本章小结 25

3 对称加密算法 27

3.1 DES 27
3.2 3DES 27
3.3 AES 27
3.4 本章小结 27

4 分组密码的模式 29

4.1 什么是模式 29
4.2 ECB模式 29
4.3 CBC模式 29
4.4 CFB模式 29
4.5 OFB模式 29
4.6 CTR模式 29
4.7 模式之间的比较 29
4.8 本章小结 29

5 非对称加密算法 31

5.1 公钥密码 31
5.2 基本运算 31
5.3 RSA 31
5.4 ECDSA 31
5.5 本章小结 31

第1章　环游密码世界

随着物联网和智能家居的兴起，网络信息安全已经渗透到我们日常生活的方方面面。密码技术作为信息安全的基石，为网络通信提供了安全和可靠的技术手段。本章，我们先整体了解一下密码世界，看看各种密码技术如何为我们的信息安全保驾护航。

1.1　密码学中的基本概念

信息在人与人、人与机器、机器与机器之间交互的过程中存在被第三方（人或计算机）窃取的风险，密码技术提供了信息在通信双方交互的过程中免遭第三方窃取并破解，以及确保通信任何一方不被欺骗的一系列算法。

首先，我们来了解一下与密码技术有关的角色：

- *发送者* (sender)：消息的发送方。
- *接收者* (receiver)：消息的接收方。
- *窃听者* (eavesdropper)：监听在消息传送的通道上，窃取消息的恶意攻击方。
- *破译者* (cryptanalyst)：为研究密码强度而工作的密码破译人员或密码学研究者，要注意和窃听者的本质区别。

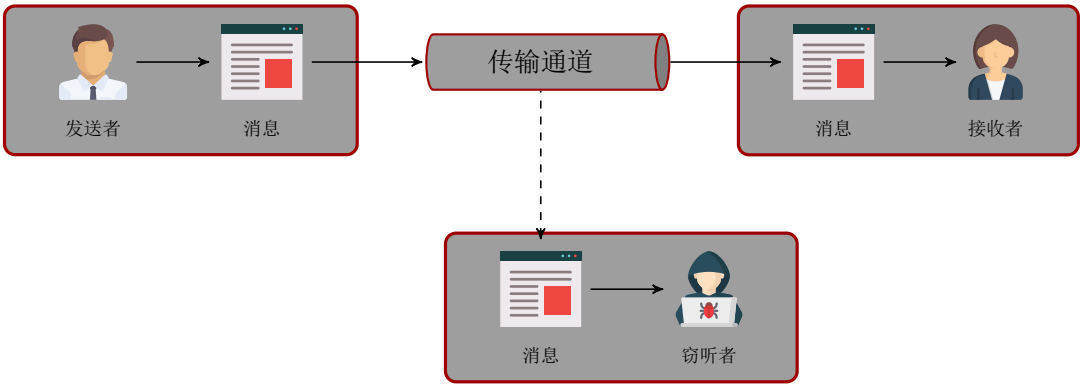


图1.1　消息的发送、接收和窃听

如图 1.1所示，如果发送者不对要发送出去的消息进行任何处理，很容易被窃听者窃取并获知消息的内容。为确保消息的机密性 (confidentiality)，发送者在发送消息之前需要对其进行加密 (encryption)。消息可以是任何类型的数据，例如，邮件、文档和交易等。通常，我们把加密前的消息称之为*明文* (plaintext)，加密之后的消息称之为*密文* (ciphertext)。接收者收到密文后将其恢复回明文的过程称为解密 (decryption)。

因为加密和解密需要相应的密钥才能完成，当窃听者窃取到加密后的密文之后，因为没有密钥，也就无法还原出原始的明文。这就相当于，我们在把重要的机密文件传给接收人之前，先把机密文件锁在保险柜里面，然后把保险柜交给物流公司帮忙交付给接收人，接收人收到之后用保险柜的钥匙打开取出该机密文件。在物流运输的过程中，保险柜有可能面临被丢失的风险，如果有人偷盗了保险柜，因为没有钥匙也无法取出里面的文件。整个过程可以用图 1.2来描述。

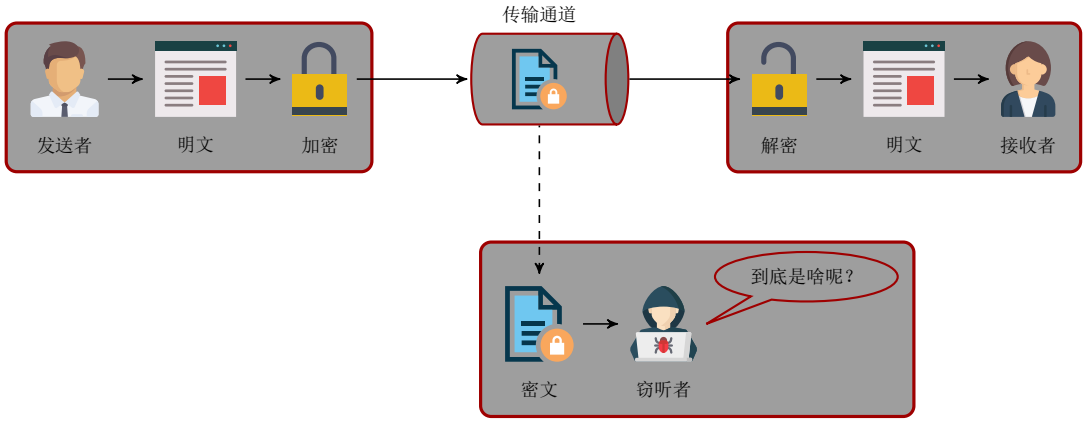


图1.2　消息加密和解密的过程

1.2　对称加密算法和非对称加密算法

在上面的例子中，如果保险箱在运输的过程中被坏人偷盗，坏人有可能使用物理破坏的方法撬开保险箱，取出里面的机密文件。那么，显而易见，保险箱越坚固，坏人就越难破坏保险箱拿到里面的机密文件。所以，保险箱的坚固程度就决定了里面机密文件的安全性到底有多高。

在网络通信领域，网络传输通道是极不可靠的，信息加密后的密文在传输的过程中，存在被窃听者窃取的风险，加密算法也要确保即使密文被窃取也不能在现实的时间内被破解，这也正是加密算法的魅力所在。

从原理上，加密算法被分成两大类，即*对称加密算法* (symmetric encryption algorithm) 和*非对称加密算法* (asymmetric encryption algorithm)。它们最主要的区别在于密钥 (key) 的使用方式不同。区别于现实生活中的“钥匙”，密码算法中的密钥是一串很长的看起来非常杂乱无章的字符序列。

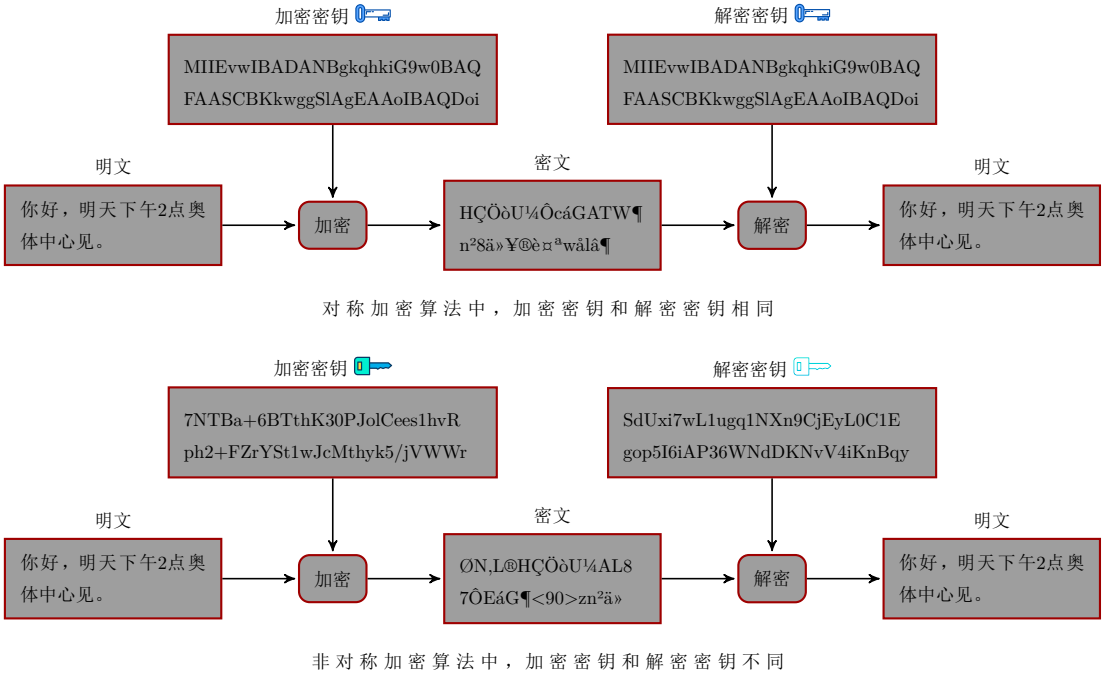


图1.3　对称加密和非对称加密

对称加密算法在加密和解密时使用了相同的密钥，因为加密方和解密方使用了相同的密钥，任何人拿到了密钥就能解密加密后的消息从而获取到原始的数据。因此，对称加密算法的密钥必须在加密方和解密方两者同时妥善保管，不能泄露给任何未经授权的第三方。

相反，非对称加密算法则在加密和解密时使用了不同的密钥。而且，在非对称加密算法下，加密密钥通常被公开，但解密密钥需要由接收者私自妥善保管。据此特点，非对称加密算法又常常被称为*公钥加密算法* (public key encryption)。

非对称加密算法是在1976年，由狄菲（Whitfield Diffie）与赫尔曼（Martin Hellman）两位学者以单向函数与单向暗门函数为基础，提出了“非对称密码体制即公开密钥密码体制”的概念，开创了密码学研究的新方向。现代计算机和互联网中的安全体系，很大程度上都依赖于公钥加密算法。

1.3 其他密码技术

加密算法为消息提供了机密性，但信息安全远不止于此，还有更多的问题需要解决。例如，如何保证数据的一致性，确保数据没有被恶意篡改过；如何对信息的来源进行判断，能对伪造来源的信息进行甄别。本节，我们来初步了解一下密码学工具箱中，除加密算法以外的的其他几种密码技术。

1.3.1 单向散列函数

有时候，接收者希望能够验证消息在传递的过程中，没有被篡改过，即入侵者不会用假消息冒充合法消息而达到某些非法的目的。

我们经常会发现，在互联网上下载免费软件的时候，有安全意识的软件发布者会在发布软件的同时发布该软件的散列值 (hash)。散列值就是用*单向散列函数* (one-way hash function) 计算出来的。这样，下载该软件的人可以自行计算所下载文件的散列值与发布者所发布的散列值进行比较。如果两个散列值一致，就说明下载的软件与发布者所发布的软件是相同的。软件发布者通过发布散列值的方法，可以防止有人在软件里植入一些恶意程序来侵害下载该软件的人的计算机系统。

单向散列函数所保证的并不是机密性，而是完整性 (integrity)。散列值通常又称为哈希值、校验和 (checksum)、指纹 (fingerprint) 或消息摘要 (message digest)。

1.3.2 消息认证码

为了确认消息是否来源于所期望的对象，可以使用*消息认证码* (message authentication code) 技术。通过消息认证码，不但能够确认消息是否被篡改，而且能够确认消息是否来自于所期望的通信对象。也就是说，消息认证码不仅能够保证完整性，还能够提供认证机制。

1.3.3 数字签名

我们先来看一个例子：供应商给采购方发来邮件，内容是“该商品的采购价格是10万元”。由于这封邮件涉及到数额巨大的交易，如果你是采购人员，肯定会特别小心，一定要核实该邮件确实来自你联系的供应商。仅仅靠邮件发送者的Email地址是不足以判断这封邮件的实际来源，因为邮件的发送者很容易被伪装 (spoofing)。

另一方面，还有这样一种可能，这封邮件确实是来自于采购方所联系的供应商。但是，供应商后来又反悔想提高采购价格，于是便谎称 “我当时根本就没发送过那封邮件”。像这样事后否认自己做过某件事情的行为，称为抵赖 (repudiation)。现代商战中，大量充斥着这种案例。

当然，还有一种风险，就是供应商发给采购方的邮件在传输过程中，被别有用心的人篡改，将采购费改成了20万元。数字签名是一项能够同时防止伪装、抵赖和篡改等威胁的密码技术。当供应商对邮件的内容加上数字签名之后再通过邮件一起发送，采购方则可以通过对*数字签名* (digital signature) 进行验证 (verify) 来检测出邮件是否被伪装和篡改，还能够防止供应商事后抵赖。

1.3.4 伪随机数生成器

伪随机数生成器 (Pseudo Random Number Generator, PRNG) 用于在系统需要随机数的时候，通过一系列种子值计算出来的伪随机数。因为生成一个真正意义上的“随机数”对于计算机来说是不可能的，伪随机数也只是尽可能地接近其应具有的随机性，但是因为有“种子值”，所以伪随机数在一定程度上是可控可预测的。随机数在密码技术中承担了重要的职责，例如在访问HTTPS加密站点时进行的TLS通信，会生成一个仅用于当前通信的临时密钥（即会话密钥），这个密钥就是基于伪随机数生成器产生的。如果生成的随机数的算法不够好，窃听者就有可能推测出密钥，从而带来通信机密性下降的风险。

1.4 信息安全所面临的威胁及对策

回顾一下，我们前面初步介绍了六种密码技术：

- 对称加密算法
- 非对称加密算法（公钥加密算法）
- 单向散列函数
- 消息认证码
- 数字签名
- 伪随机数生成器

我们同时讨论了每种技术所解决的具体问题，这里把前面的内容再梳理一遍，用图 1.4所示的思维导视图总结了信息安全所面临的潜在威胁以及针对各种安全威胁所能采用的密码技术及对策。我们没有把伪随机数生成器画在图里面，是因为它通常渗透在其他五种密码技术中使用，发挥了非常重要的作用。我们把这六种密码技术统称为*密码学家的工具箱*。

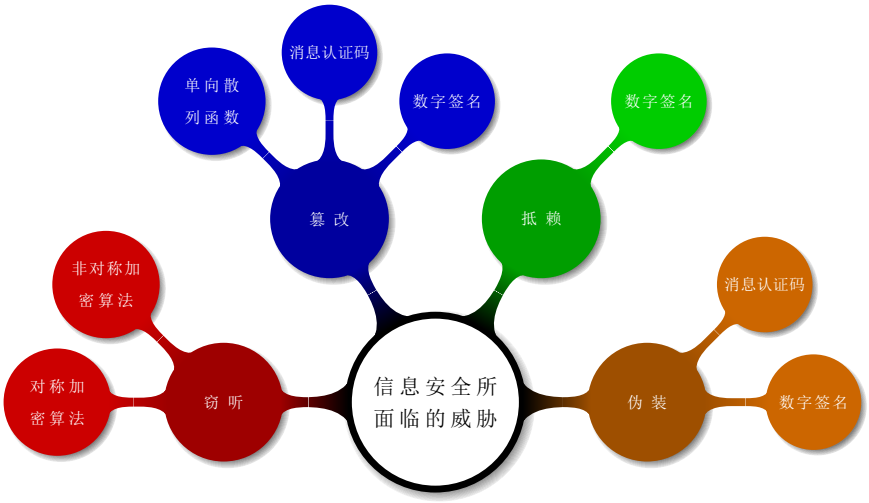


图1.4 信息安全所面临的威胁及其相应的密码技术对策思维导视图

从图 1.4中，我们可以看到，有些密码技术可以用来解决信息安全中的多种威胁，例如，数字签名可以防止篡改、伪装和抵赖，但不提供保密。对于某些面临的威胁，也可能存在多种应对的密码技术，例如为了防止窃听导致信息被泄露，可以使用对称加密算法或非对称加密算法。但是每种密码技术都有着各自的特点，适用于不同的场景。后面章节，我们会对这些密码技术进行深入的探讨，逐个揭开它们的神秘面纱。

1.5 密码与信息安全常识

随着信息技术的飞速发展，计算机的计算能力和存储能力正在以惊人的速度不断提升，我们所熟知的摩尔定律到目前为止仍然成立。大数据和物联网应用正在渗透到社会组织的每一个细胞，几乎对所有行业产生颠覆性和革命性的影响。产业的发展环境逐步成熟，网络基础设施支撑能力大幅提升，网络通信的数据量正在呈现指数级爆炸式增长。在人们的生活越来越依赖互联网的时代，信息安全在网络通信中发挥的作用尤为重要，密码技术为保障信息安全提供了全方位的技术支持。本小节从最佳实践的角度阐述我们应该怎么合理地利用密码技术来保障信息的安全。

1.5.1 任何时候不要尝试发明新的加密算法

刚接触密码技术的软件开发人员，经常会出现这样的想法：我自己设计一个不对外公开的密码算法不就可以保障信息的机密性了吗？这种想法是绝对错误的。加密系统的保密性只应建立在对密钥的保密上，不应该取决于加密算法的保密，这是密码学中的金科玉律。任何时候，我们都不要尝试自己去发明新的加密算法，因为对加密算法的保密是困难的。对手可以用窃取、购买的方法来取得算法、加密器件或者程序。如果得到的是加密器件或者程序，可以对它们进行反向工程而最终获得加密算法。如果只是密钥失密，那么失密的只是和此密钥有关的情报，日后通讯的保密性可以通过更换密钥来补救；但如果是加密算法失密，而整个系统的保密性又建立在算法的秘密性上，那么所有由此算法加密的信息就会全部暴露。

1.5.2 不要使用低强度的密码

很多人对密码的使用有这么一个误区：就算密码强度再低也比不用密码更安全吧。其实，这种想法是非常危险的。与其使用低强度的密码，还不如从一开始就不使用密码。这主要源于用户容易通过“密码”这个词获得一种“错误的安全感”。“信息被加密了”这一事实并不能和信息安全划上句号。攻击者使用暴力穷举（brute-force）等攻击方法就可能破解低强度的密码。

1.5.3 信息安全也是一门社会性课题

有了密码技术，信息安全就能完全得到保证吗？答案是否定的。密码技术只是信息安全的一部分，在信息安全的背景下，社会工程学（social engineering）攻击是一种操纵相关人员泄露出机密信息的攻击方法，建立在使人决断产生认知偏差的基础上，有时候这些偏差被称为“人类硬件漏洞”。犯罪分子利用社会工程学的手法进行诱骗，使受害者不会意识到被利用来攻击网络。当人们没有意识到他们拥有的信息的价值的时候，并不会特意地保护他们所得知的信息，社会工程学正是利用了这一点。

本书不会详细讨论社会工程学攻击，但是为了让大家提高安全意识，防患于未然，特列举以下一些流行的社会工程学攻击：

- 伪装：犯罪分子通过伪装成各种角色来骗取访问权限。例如，伪装成一个看门人、雇员或者客户来获取物理访问权限；冒充贵宾、高层经理或者其他有权或进入计算机系统并察看文件的人。
- 偷窥：通过偷窥方式在他人输入密码时收集他的密码。甚至寻找在垃圾箱中记录密码的纸、电脑打印的文件、快递信息等，往往也可以找到有用的信息。
- 钓鱼：钓鱼涉及虚假邮件、聊天记录或网站设计，模拟与捕捉真正目标系统的敏感数据。比如伪造一条上来自银行或其他金融机构的需要“验证”您登陆信息的消息，来冒充一条合法的登陆页面来骗取你的登录密码。
- 引诱：攻击者可能使用能勾起你欲望的东西引诱你去点击，可能是一场音乐会或一部电影的下载链接，也有可能是你“中奖” 需要兑换礼品的链接，或者是商品大力打折的促销链接。一旦点击了这些链接，你的计算机设备或网络就会感染恶意软件以便于犯罪分子进入你的系统。

上面提到的这些攻击手段，都与密码的强度毫无关系。信息安全是一个复杂的系统性工程，其安全程度往往取决于系统中最薄弱的环节。通常，最薄弱的环节不是密码，而是人类自己。“道高一尺魔高一丈”，信息安全上的漏洞和人性上脆弱的环节也不断被不法分子发掘，我们唯有不断的增强自己的安全意识和时刻保持清醒才能更好地防患于未然。

1.6 本章小结

本章，我们初步了解了密码世界里常用的密码技术，并介绍了使用哪种密码技术来应对信息安全中存在的威胁。我们后在后续章节中更详细地介绍每种密码技术的细节，并为应用开发者介绍怎么在工程中使用各种密码技术。

第2章　密码的历史典故

从密码学发展历程来看，可分为古典密码和现代密码两类。古典密码有着悠久的历史，是以字符为基本加密单元的密码。而现代密码则以信息块为基本的加密单元。古典密码和现代密码的分水岭大致就是在计算机问世的时候。本章将重点回顾古典密码的发展历史并分享一些和密码相关的有趣典故和历史事件。

2.1　中国古代人民怎么加密？

早期加密算法主要使用在军事中，中国历史上最早关于加密算法的记载出自于周朝兵书《六韬·龙韬》中的《阴符》和《阴书》。其中《阴符》记载了：

> 太公曰："主与将，有阴符，凡八等。有大胜克敌之符，长一尺。破军擒将之符，长九寸。降城得邑之符，长八寸。却敌报远之符，长七寸。警众坚守之符，长六寸。请粮益兵之符，长五寸。败军亡将之符，长四寸。失利亡士之符，长三寸。诸奉使行符，稽留，若符事闻，泄告者，皆诛之。八符者，主将秘闻，所以阴通言语，不泄中外相知之术。敌虽圣智，莫之能识。"

简单来说，阴符是以八等长度的符来表达不同的消息和指令，属于密码学中的替代法（图 2.1），在应用中是把信息转变成敌人看不懂的符号，但知情者知道这些符号代表的含义。

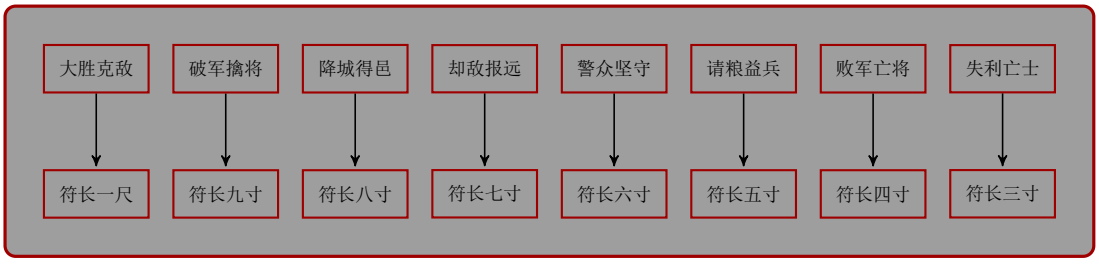


图2.1　阴符所蕴含的加密原理：替换法

阴符只能表述最关键的八种信号，无法表达丰富的含义和传递更具体的消息。所以，《阴书》又作了补充：

> 武王问太公曰："引兵深入诸侯之地，主将欲合兵，行无穷之变，图不测之利，其事烦多，符不能明；相去辽远，言语不通。为之奈何？"太公曰："诸有阴事大虑，当用书，不用符。主以书遗将，将以书问主。书皆一合而再离，三发而一知。再离者，分书为三部。三发而一知者，言三人，人操一分，相参而不相知情也。此谓阴书。敌虽圣智，莫之能识。"

阴书作为阴符的补充，所有密谋大计，都应当用阴书，而不用阴符。国君用阴书向主将传达指示，主将用阴书向国君请示问题，这种阴书都是一合而再离（把一封书信分为三个部分）、三发而一知（派三个人送信，每人负责其中的一部分）。阴书运用了文字拆分法直接把一份文字拆成三份（图 2.2），由三种渠道发送到目标方手中。敌人只有同时截获三份内容才可能破解阴书上写的内容。

无论是阴符，还是阴书，都有着一定的局限性。一是有可能被对方截获而难以达到传递消息的目的，二是有可能被对方破译内容并被对方将计就计加以利用。因此，并不是"敌虽圣智，莫之能识"。张献忠袭取襄阳就说明了这一点。

崇祯十三年七月，张献忠率领起义军突破明军防线，进入四川，杨嗣昌亦率明军十万尾随追击。面对强敌，张献忠挥师东进，于次年二月进入湖北兴山、当阳。在东进途中，起义军活捉了由襄阳（今湖北襄樊市）回四川的杨嗣昌的军使。张献忠从其口中得知杨嗣昌大营所在地襄阳城防空虚，决定奔袭襄阳。他杀掉使者，搜出所携带的兵符，挑选了二十八名起义军战士，换上明军的衣

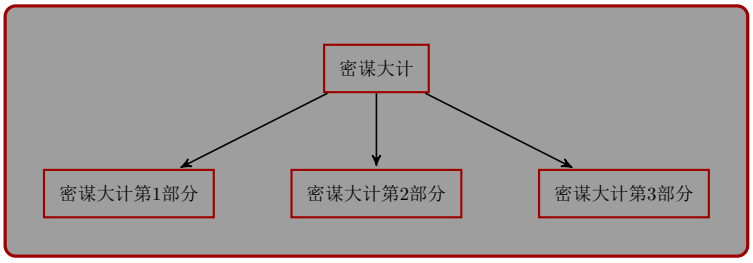


图2.2 阴书所蕴含的加密原理：文字分拆法

服，持兵符先行。张献忠自己则亲率二千精骑，随后跟进，一昼夜急行三百里，直扑襄阳。伪装成明军的起义军士兵到达襄阳时正是夜间，他们自称是督师杨嗣昌派来调运军械的，并出示兵符。守城明军用小筐吊上兵符，细心查验，完全吻合，才命开门放入。城门刚打开，二十八名起义军战士一涌而入，挥刀砍杀守门明军，占领城门。张献忠率领的后续部队恰好赶到，顺利入城。一时杀声震天，明军惊慌失措，被迫投降。起义军杀死襄王朱翊铭，降俘明军数千人，占领襄阳，杨嗣昌闻讯呕血而死。此战表明，无论是阴符还是阴书，都不是万无一失的。

2.2 凯撒密码

我们把视角切换到世界历史的长河中，看看古罗马时期凯撒大帝是怎么使用加密算法对军事信息进行加密的。根据罗马早期纪传体作者盖乌斯·苏维托尼乌斯的记载，恺撒大帝的加密策略很简单，就是把字母按照字母表顺序向后移动几位，但是偏移量（offset）只有他和将军知道，如果移动后超过了字母表中的最后一个字母（对于英文字母表而言就是Z），就回到字母表的第一个字母重新开始下一轮。以英文字母表为例，在偏移量为3的情况下，A将会替换为D，B将会被替换为E，W会被替换为Z，X会被替换为A；明文HELLO会被转换为密文KHOOR。这种加密方法又被称为移位加密。

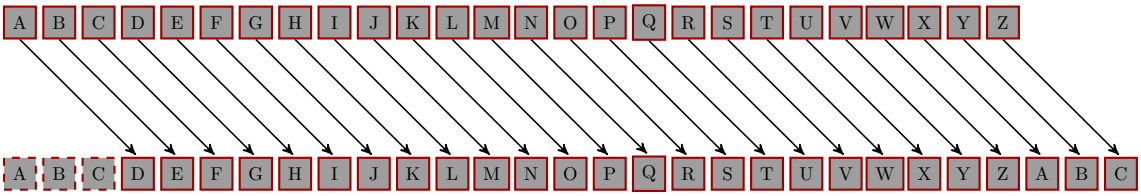


图2.3 凯撒密码的加密原理：替换法

凯撒密码的解密方法也很一目了然，只需要将密文中的每个字母向相反方向平移规定的偏移量便可解密出明文。恺撒密码的加密、解密算法还能够通过同余的数学方法进行计算。首先将字母表中的字母按顺序用数字代替，$A=0$，$B=1$，…，$Z=25$。此时偏移量为k的加密算法的数学公式即为：

$$E_k(x) = (x + k) \bmod 26$$

解密算法的数学公式可以表示为：

$$D_k(x) = (x + 26 - k) \bmod 26$$

从加密和解密数学公式可以看出，当偏移量$k=13$时（字母表内所有字母数量的一半），凯撒密码加密和解密算法的公式完全相同，这是一种特殊的凯撒密码的变种算法，被称为ROT13。ROT13在英文网络论坛常常用作隐藏八卦、妙句、谜题解答以及某些脏话的工具，目的是逃过版主或管理员的匆匆一瞥。因为ROT13的加密和解密计算公式完全相同，很明显，文字经过两次 ROT13加密之后，会恢复成原来的文字。

凯撒密码中的偏移量就相当于加密算法中的密钥。这个偏移量必须由发送者和接收者事先约定好。那么，当接收者以外的人窃取到用凯撒密码加密后的密文之后，是不是就无法破解这个密文了呢？或者换句话说，凯撒密码能够被破解吗？

破解密码的复杂度很大程度上取决于密钥空间（keyspace）的大小，所谓密钥空间是指密钥的取值范围到底有多大。凯撒密码中的密钥是偏移量k，其取值范围为0至25的整数，共26种可能的取值，密钥空间非常有限。攻击者往往可以采用暴力破解（brute-force attack）的方法就可以轻而易举地破解凯撒密码。

假设发送者和接收者之间约定的偏移量为3，那么明文CRYPTOGRAPHY加密后的密文则为FUBSWRJUDSKB。当第三方窃听到密文之后，由于凯撒密码的密钥空间只有26种可能的取值，窃听者可以使用穷举搜索（exhausitive search）的方法对每种可能的密钥取值尝试一遍：

```
k =  0: FUBSWRJUDSKB => FUBSWRJUDSKB
k =  1: FUBSWRJUDSKB => ETARVQITCRJA
k =  2: FUBSWRJUDSKB => DSZQUPHSBQIZ
k =  3: FUBSWRJUDSKB => CRYPTOGRAPHY
k =  4: FUBSWRJUDSKB => BQXOSNFQZOGX
k =  5: FUBSWRJUDSKB => APWNRMEPYNFW
k =  6: FUBSWRJUDSKB => ZOVMQLDOXMEV
k =  7: FUBSWRJUDSKB => YNULPKCNWLDU
k =  8: FUBSWRJUDSKB => XMTKOJBMVKCT
k =  9: FUBSWRJUDSKB => WLSJNIALUJBS
k = 10: FUBSWRJUDSKB => VKRIMHZKTIAR
k = 11: FUBSWRJUDSKB => UJQHLGYJSHZQ
k = 12: FUBSWRJUDSKB => TIPGKFXIRGYP
k = 13: FUBSWRJUDSKB => SHOFJEWHQFXO
k = 14: FUBSWRJUDSKB => RGNEIDVGPEWN
k = 15: FUBSWRJUDSKB => QFMDHCUFODVM
k = 16: FUBSWRJUDSKB => PELCGBTENCUL
k = 17: FUBSWRJUDSKB => ODKBFASDMBTK
k = 18: FUBSWRJUDSKB => NCJAEZRCLASJ
k = 19: FUBSWRJUDSKB => MBIZDYQBKZRI
k = 20: FUBSWRJUDSKB => LAHYCXPAJYQH
k = 21: FUBSWRJUDSKB => KZGXBWOZIXPG
k = 22: FUBSWRJUDSKB => JYFWAVNYHWOF
k = 23: FUBSWRJUDSKB => IXEVZUMXGVNE
k = 24: FUBSWRJUDSKB => HWDUYTLWFUMD
k = 25: FUBSWRJUDSKB => GVCTXSKVETLC
```

纵览所有尝试的破解，就会发现只有当$k = 3$的时候，密文FUBSWRJUDSKB才可以解密出有意义的字符序列 CRYPTOGRAPHY，即“密码学”的英文单词。因此，凯撒密码是一种极其不安全的加密方法，可以被攻击者在很快的时间内破解，无法保护重要的秘密。

2.3 简单替换密码

2.3.1 什么是简单替换密码

凯撒密码通过将明文中的每个字符按照在字符表中的顺序平移固定数量的字符数来生成密文。由于字符偏移量的取值空间极其有限，致使凯撒密码能被轻而易举地破解。我们也提到了密钥空间这个概念，凯撒密码就是因为过小的密钥空间可以被攻击者使用暴力破解的方法在非常快的时间内被破解。你可能意识到凯撒密码这种通过平移字符来实现字符替换的方法过于公式化，如果把这种映射用随机化的方式打乱，是不是就完美了呢？这就是我们接下来要讨论的简单替换密码。

简单替换密码将字母表中的26个字母，分别与其他字母建立一一映射的关系，这种映射关系不像凯撒密码那样通过平移字符这种线性化的方法，而是用一个映射表来描述明文字符和密文字符之间的映射关系，这种映射表也称为字符替换表。为了更直观地展示字符之间的映射关系，我们把明文

中的字符都用小写字母表示，密文中的字符都用大写字母表示。图 2.4 就是一个简单的字符替换表。

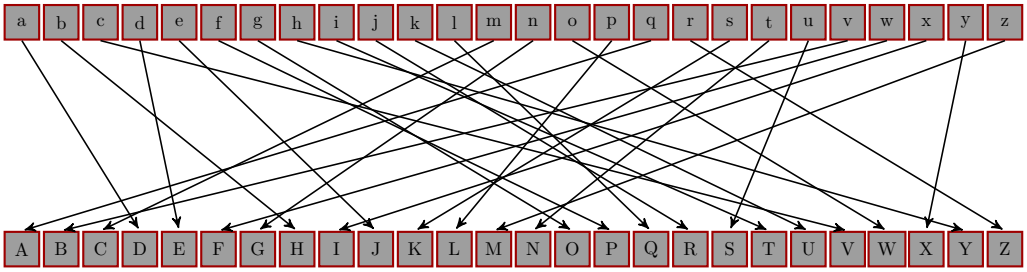


图2.4 简单替换密码的映射表

显然，图 2.4表示的字符替换关系不像凯撒密码那么有规律，明文字符和密文字符之间的映射看起来是无章可循的。可以说，凯撒密码是简单字符替换密码的一个特例。为了更好地展示明文字符和密文字符之间的替换关系，我们对图 2.4稍作转换，图 2.5，但仍然保持字符之间原来的映射关系。

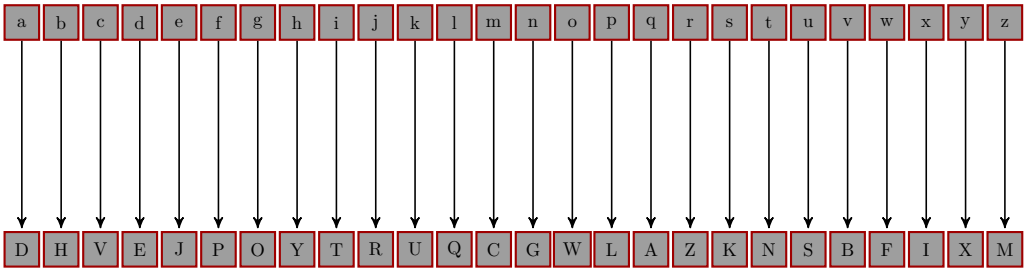


图2.5 变换后的简单替换密码的映射表

凯撒密码可以用暴力破解来破译，但简单替换密码则不然。前面，我们提到，密码算法被破译的困难程度取决于密钥空间的大小。我们来看看简单替换密码的密钥空间。明文字母中的a可以对应A，B，...，Z这26个字母中的任意一个（26种），b可以对应除了a所对应的字母以外的剩余25个字母中的任意一个（25种）。以此类推，我们可以计算出简单替换密码的密钥空间大小是：

$$26 \times 25 \times 24 \times \cdots \times 1 = 403291461126605635584000000$$

这个数字约等于400×10^{24}，密钥的数量如此巨大，用暴力破解进行穷举搜索就非常困难了。我们假设以当前（2018年11月）排名第一的Summit超级计算机在峰值性能下每秒约200亿次浮点计算的速度来遍历密钥的话，要遍历完所有的密钥也需要花费超过6千万年的时间。这还是用我们当前最顶级的超级计算机的峰值计算速度来遍历的，普通的家用计算机将要花费几百亿年的时间才能遍历完所有的密钥。由此可见，简单替换密码的密钥空间是足够大的。

2.3.2 用频率分析的方法破解简单替换密码

超大的密钥空间让破译简单替换密码看起来变得不可能，但密码破译工作者发现用频率分析的密码破译方法，使破译简单替换密码成为可能了。

在任何一种书面语言中，不同的字母或字母组合出现的频率各不相同。而且，对于以这种语言写的任意一段文本，都具有大致相同的特征字母分布。比如，在英语中，字母e出现的频率很高，而x出现的较少。类似地，字母组合st、ng、th以及qu等双字母组合出现的频率非常高，nz、qj组合则极少。表 2.1是人们从大量的英文文章中统计出的字母频率。

字母	频率	字母	频率
e	11.1607%	m	3.0129%
a	8.4966%	h	3.0034%
r	7.5809%	g	2.4705%
i	7.5448%	b	2.0720%
o	7.1635%	f	1.8121%
t	6.9509%	y	1.7779%
n	6.6544%	w	1.2899%
s	5.7351%	k	1.1016%
l	5.4893%	v	1.0074%
c	4.5388%	x	0.2902%
u	3.6308%	z	0.2722%
d	3.3844%	j	0.1965%
p	3.1671%	q	0.1962%

表2.1 英文字母出现的频率表

简单替换密码的密钥空间如此巨大，但它的弱点也是显而易见的，就是明文中相同的字母在转换为密文后总是被同一个字母所替换。我们参考这个英文字母频率表来实际尝试破译一段密文。现在，假设我们得到下面一段经过简单替换密码加密过后的密文，其明文都是小写英文字母。

```
EAOQASBAEEAOQECPENFECQMRQFENABDCQECQSENFBAOZQSNBECQUNBLEAFRKKQSECQFZNBVFPBLPSSADFAKAR
ESPVQARFKASERBQASEAEPWQPSUFPVPNBFEPFQPAKESAROZQFPBLOIAGGAFNBVQBLECQUEALNQEAFZQQGBAUAS
QPBLOIPFZQQGEAFPIDQQBLECQCQPSEPTCQPBLECQECARFPBLBPERSPZFCATWFECPEKZQFCNFCQNSEAENFPTAB
FRUUPENABLQHAREZIEAOQDNFCLEALNQEAFZQQGEAFZQQGGQSTCPBTQEALSQPUPIECQSQFECQSROKASNBECPEF
ZQQGAKLQPECDCPELSQPUFUPITAUQDCQBDQCPHQFCRKKZQLAKKECNFUASEPZTANZURFEVNHQRFGPRFQECQSQFE
CQSQFGQTEECPEUPWQFTPZPUNEIAKFAZABVZNKQKASDCADARZLOQPSECQDCNGFPBLFTASBFAKENUQECAGGSQFF
ASFDSABVECQGSARLUPBFTABERUQZIECQGPBVFAKLNFGSNXLZAHQECQZPDFLQZPIECQNBFAZQBTQAKAKKNTQPB
LECQFGRSBFECPEGPENQBEUQSNEAKECRBDASECIEPWQFDCQBCQCNUFQZKUNVCECNFMRNQERFUPWQDNECPOPSQO
ALWNBDCADARZLKPSLQZFOQPSEAVSRBEPBLFDQPERBLQSPDQPSIZNKQOREECPEECQLSQPLAKFAUQECNBVPKEQS
LQPECECQRBLNFTAHQSQLTARBESIKSAUDCAFQOARSBBAESPHQZZQSSQERSBFGRXXZQFECQDNZZPBLUPWQFRFSP
ECQSOQPSECAFQNZZFDQCPHQECPBKZIEAAECQSFECPEDQWBADBAEAKECRFTABFTNQBTQLAQFUPWQTADPSLFAKR
FPZZPBLECRFECQBPENHQCRQAKSQFAZRENABNFFNTWZNQLAQSDNECECQGPZQTPFEAKECARVCEPBLQBEQSGSNFQ
FAKVSQPEGNETCPBLUAUQBEDNECECNFSQVPSLECQNSTRSSQBEFERSBPDSIPBLZAFQECQBPUQAKPTENAB
```

首先，我们统计这段密文中各个字母出现的次数和频率，结果如表 2.2所示。

根据密码研究工作者总结出来的字母频率表 2.1，字母e的出现频率远高于其他字母。经统计后，我们发现密文中字母Q的出现频率最高。我们先暂且假设字母Q就是由e变换而来的，这样，我们把密文中的Q替换回e，就得到下面的字母序列。

```
EAOeASBAEEAOeECPENFECeMReFENABDCeECeSENFBAOZeSNBECeUNBLEAFRKKeSECeFZNBVFPBLPSSADFAKAR
ESPVeARFKASERBeASEAEPWePSUFPVPNBFEPFePAKESAROZeFPBLOIAGGAFNBVeBLECeUEALNeEAFZeeGBAUAS
ePBLOIPFZeeGEAFPIDeeBLECeCePSEPTCePBLECeECARFPBLBPERSPZFCATWFECPEKZeFCNFCeNSEAENFPTAB
FRUUPENABLeHAREZIEAOeDNFCLEALNeEAFZeeGEAFZeeGGeSTCPBTeEALSePUPIECeSeFECeSROKASNBECPEF
ZeeGAKLePECDCPELSePUFUPITAUeDCeBDeCPHeFCRKKZeLAKKECNFUASEPZTANZURFEVNHeRFGPRFeECeSeFE
CeSeFGeTEECPEUPWeFTPZPUNEIAKFAZABVZNKeKASDCADARZLOePSECeDCNGFPBLFTASBFAKENUeECAGGSeFF
ASFDSABVECeGSARLUPBFTABERUeZIECeGPBVFAKLNFGSNXLZAHeECeZPDFLeZPIECeNBFAZeBTeAKAKKNTePB
LECeFGRSBFECPEGPENeBEUeSNEAKECRBDASECIEPWeFDCeBCeCNUFeZKUNVCECNFMRNeERFUPWeDNECPOPSeO
ALWNBDCADARZLKPSLeZFOePSEAVSRBEPBLFDePERBLeSPDePSIZNKeOREECPEECeLSePLAKFAUeECNBVPKEeS
LePECECeRBLNFTAHeSeLTARBESIKSAUDCAFeOARSBBAESPHeZZeSSeERSBFGRXXZeFECeDNZZPBLUPWeFRFSP
```

字母	次数	频率	字母	次数	频率
Q	137	12.47%	K	34	3.09%
E	117	10.65%	D	28	2.55%
A	93	8.46%	U	28	2.55%
P	84	7.64%	T	24	2.18%
F	82	7.46%	G	22	2.00%
C	75	6.82%	O	15	1.36%
S	68	6.19%	I	14	1.27%
B	65	5.91%	V	14	1.27%
N	53	4.82%	W	10	0.91%
L	42	3.82%	H	8	0.73%
Z	41	3.73%	X	3	0.27%
R	40	3.64%	M	2	0.18%

表2.2 密文中各英文字母出现的次数和频率

```
ECeSOePSECAFeNZZFDeCPHeECPBKZIEAAECeSFECPEDeWBADBAEAKECRFTABFTNeBTeLAeFUPWeTADPSLFAKR
FPZZPBLECRFECeBPENHeCReAKSeFAZRENABNFFNTWZNeLAeSDNECECeGPZeTPFEAKECARVCEPBLeBEeSGSNFe
FAKVSePEGNETCPBLUAUeBEDNECECNFSeVPSLECeNSTRSSeBEFERSBPDSIPBLZAFeECeBPUeAKPTENAB
```

英文文章中，以字母e结尾的单词，the的出现频率极高，对上面的这段字符序列，进一步统计发现ECe出现了27次，远远高于其他以e结尾的包含3个字母的字符串的出现次数。我们进一步假定t被替换成了E，h被替换成了C，于是，我们继续将上面字符序列中E和C分别替换回t和h，得到：

```
tAOeASBAttAOethPtNFtheMReFtNABDhetheStNFBAOZeSNBtheUNBLtAFRKKeStheFZNBVFPBLPSSADFAKAR
tSPVeARFKAStRBeAStAtPWePSUFPVPNBFtPFePAKtSAROZeFPBLOIAGGAFNBVeBLtheUtALNetAFZeeGBAUAS
ePBLOIPFZeeGtAFPIDeeBLthehePStPThePBLthethARFPBLBPtRSPZFhATWFthPtKZeFhNFheNStAtNFPTAB
FRUUPtNABLeHARtZItAOeDNFhLtALNetAFZeeGtAFZeeGGeSThPBTetALSePUPItheSeFtheSROKASNBthPtF
ZeeGAKLePthDhPtLSePUFUPITAUeDheBDehPHeFhRKKZeLAKKthNFUAStPZTANZURFtVNHeRFGPRFetheSeFt
heSeFGeTtthPtUPWeFTPZPUNtIAKFAZABVZNKeKASDhADARZLOePStheDhNGFPBLFTASBFAKtNUethAGGSeFF
ASFDSABVtheGSARLUPBFTABtRUeZItheGPBVFAKLNFGSNXLZAHetheZPDFLeZPItheNBFAZeBTeAKAKKNTePB
LtheFGRSBFthPtGPtNeBtUeSNtAKthRBDASthItPWeFDheBhehNUFeZKUNVhthNFMRNetRFUPWeDNthPOPSeO
ALWNBDhADARZLKPSLeZFOePStAVSRBtPBLFDePtRBLeSPDePSIZNKeORtthPttheLSePLAKFAUethNBVPKteS
LePththeRBLNFTAHeSeLTARBtSIKSAUDhAFeOARSBBAtSPHeZZeSSetRSBFGRXXZeFtheDNZZPBLUPWeFRFSP
theSOePSthAFeNZZFDehPHethPBKZItAAtheSFthPtDeWBADBAtAKthRFTABFTNeBTeLAeFUPWeTADPSLFAKR
FPZZPBLthRFtheBPtNHehReAKSeFAZRtNABNFFNTWZNeLAeSDNththeGPZeTPFtAKthARVhtPBLeBteSGSNFe
FAKVSePtGNtThPBLUAUeBtDNththNFSeVPSLtheNSTRSSeBtFtRSBPDSIPBLZAFetheBPUeAKPTtNAB
```

进一步分析，我们发现thPt也多次出现，英文中单词that出现的频率也是特别高的。同时，我们发现P在这段密文中出现的频率也是极高的，我们几乎可以不假思索地猜测a被替换成了P。把P替换回a，我们得到：

```
tAOeASBAttAOethatNFtheMReFtNABDhetheStNFBAOZeSNBtheUNBLtAFRKKeStheFZNBVFaBLaSSADFAKAR
tSaVeARFKAStRBeAStAtaWeaSUFaVaNBFtaFeaAKtSAROZeFaBLOIAGGAFNBVeBLtheUtALNetAFZeeGBAUAS
eaBLOIaFZeeGtAFaIDeeBLtheheaStaTheaBLthethARFaBLBatRSaZFhATWFthatKZeFhNFheNStAtNFaTAB
FRUUatNABLeHARtZItAOeDNFhLtALNetAFZeeGtAFZeeGGeSThaBTetALSeaUaItheSeFtheSROKASNBthatF
ZeeGAKLeathDhatLSeaUFUaITAUeDheBDehaHeFhRKKZeLAKKthNFUAStaZTANZURFtVNHeRFGaRFetheSeFt
heSeFGeTtthatUaWeFTaZaUNtIAKFAZABVZNKeKASDhADARZLOeaStheDhNGFaBLFTASBFAKtNUethAGGSeFF
ASFDSABVtheGSARLUaBFTABtRUeZItheGaBVFAKLNFGSNXLZAHetheZaDFLeZaItheNBFAZeBTeAKAKKNTeaB
LtheFGRSBFthatGatNeBtUeSNtAKthRBDASthItaWeFDheBhehNUFeZKUNVhthNFMRNetRFUaWeDNthaOaSeO
ALWNBDhADARZLKaSLeZFOeaStAVSRBtaBLFDeatRBLeSaDeaSIZNKeORtthattheLSeaLAKFAUethNBVaKteS
```

LeaththeRBLNFTAHeSeLTARBtSIKSAUDhAFeOARSBBAtSaHeZZeSSetRSBFGRXXZeFtheDNZZaBLUaWeFRFSa
theSOeaSthAFeNZZFDehaHethaBKZItAAtheSFthatDeWBADBAtAKthRFTABFTNeBTeLAeFUaWeTADaSLFAKR
FaZZaBLthRFtheBatNHehReAKSeFAZRtNABNFFNTWZNeLAeSDNththeGaZeTaFtAKthARVhtaBLeBteSGSNFe
FAKVSeatGNtThaBLUAUeBtDNththNFSeVaSLtheNSTRSSeBtFtRSBaDSIaBLZAFetheBaUeAKaTtNAB

继续猜测，theSe会不会是there呢，Leath会不会是death呢，于是，我们用r和d分别替换回S和L，得到：

tAOeArBAttAOethatNFtheMReFtNABDhethertNFBAOZerNBtheUNBdtAFRKKertheFZNBVFaBdarrADFAKAR
traVeARFKArtRBeArtAtaWearUFaVaNBFtaFeaAKtrAROZeFaBdOIAGGAFNBVeBdtheUtAdNetAFZeeGBAUAr
eaBdOIaFZeeGtAFaIDeeBdtheheartaTheaBdthethARFaBdBatRraZFhATWFthatKZeFhNFheNrtAtNFaTAB
FRUUatNABdeHARtZItAOeDNFhdtAdNetAFZeeGtAFZeeGGerThaBTetAdreaUaIthereFtherROKArNBthatF
ZeeGAKdeathDhatdreaUFUaITAUeDheBDehaHeFhRKKZedAKKthNFUArtaZTANZURFtVNHeRFGaRFethereFt
hereFGeTtthatUaWeFTaZaUNtIAKFAZABVZNKeKArDhADARZdOeartheDhNGFaBdFTArBFAKtNUethAGGreFF
ArFDrABVtheGrARdUaBFTABtRUeZItheGaBVFAKdNFGrNXdZAHetheZaDFdeZaItheNBFAZeBTeAKAKKNTeaB
dtheFGRrBFthatGatNeBtUerNtAKthRBDArthItaWeFDheBhehNUFeZKUNVhthNFMRNetRFUaWeDNthaOareO
AdWNBDhADARZdKardeZFOeartAVrRBtaBdFDeatRBderaDearIZNKeORtthatthedreadAKFAUethNBVaKter
deaththeRBdNFTAHeredTARBtrIKrAUDhAFeOARrBBAtraHeZZerretRrBFGRXXZeFtheDNZZaBdUaWeFRFra
therOearthAFeNZZFDehaHethaBKZItAAtherFthatDeWBADBAtAKthRFTABFTNeBTedAeFUaWeTADardFAKR
FaZZaBdthRFtheBatNHehReAKreFAZRtNABNFFNTWZNedAerDNththeGaZeTaFtAKthARVhtaBdeBterGrNFe
FAKVreatGNtThaBdUAUeBtDNththNFreVardtheNrTRrreBtFtRrBaDrIaBdZAFetheBaUeAKaTtNAB

结合Dhether和Dhat，我们推测D是由w替换过来的。进一步， DNth极有可能就是with，以此类推，DNZZ可能是will，NF可能是is。用w、i和l分别替换D、N和Z，得到：

tAOeArBAttAOethatiFtheMReFtiABwhethertiFBAOleriBtheUiBdtAFRKKertheFliBVFaBdarrAwFAKAR
traVeARFKArtRBeArtAtaWearUFaVaiBFtaFeaAKtrAROleFaBdOIAGGAFiBVeBdtheUtAdietAFleeGBAUAr
eaBdOIaFleeGtAFaIweeBdtheheartaTheaBdthethARFaBdBatRralFhATWFthatKleFhiFheirtAtiFaTAB
FRUUatiABdeHARtlItAOewiFhdtAdietAFleeGtAFleeGGerThaBTetAdreaUaIthereFtherROKAriBthatF
leeGAKdeathwhatdreaUFUaITAUewheBwehaHeFhRKKledAKKthiFUArtalTAilURFtViHeRFGaRFethereFt
hereFGeTtthatUaWeFTalaUitIAKFAlABVliKeKArwhAwARldOearthewhiGFaBdFTArBFAKtiUethAGGreFF
ArFwrABVtheGrARdUaBFTABtRUelItheGaBVFAKdiFGriXdlAHethelawFdelaItheiBFAleBTeAKAKKiTeaB
dtheFGRrBFthatGatieBtUeritAKthRBwArthItaWeFwheBhehiUFelKUiVhthiFMRietRFUaWewithaOareO
AdWiBwhAwARldKardelFOeartAVrRBtaBdFweatRBderawearIliKeORtthatthedreadAKFAUethiBVaKter
deaththeRBdiFTAHeredTARBtrIKrAUwhAFeOARrBBAtraHellerretRrBFGRXXleFthewillaBdUaWeFRFra
therOearthAFeillFwehaHethaBKlItAAtherFthatweWBAwBAtAKthRFTABFTieBTedAeFUaWeTAwardFAKR
FallaBdthRFtheBatiHehReAKreFAlRtiABiFFiTWliedAerwiththeGaleTaFtAKthARVhtaBdeBterGriFe
FAKVreatGitThaBdUAUeBtwiththiFreVardtheirTRrreBtFtRrBawrIaBdlAFetheBaUeAKaTtiAB

靠近句尾的地方出现了withthisreVard，这个可能是with this regard，也可能是with this reward。但是我们可以立即排除后者，因为我们在上一步已经推测了D是由w替换过来的，所以，我们推测V是由g替换过来的。现在，我们把已经推测出来的字母放到表 2.3中，如表 2.3所示。

到此为止，排在前五的高频字母只剩下A还没有推测出来，那我们对照字母频率表 2.1，发现高频字母中只有o和n还没有被反推出来。我们大胆地假设，A就是由o或者n替换过来的，然后，我们通过whA很快排除n，所以，我们推测A是由o替换过来的，继续还原字母序列，我们得到：

toOeorBottoOethatiFtheMReFtioBwhethertiFBoOleriBtheUiBdtoFRKKertheFliBgFaBdarrowFoKoR
trageoRFKortRBeortotaWearUFagaiBFtaFeaoKtroROleFaBdOIoGGoFiBgeBdtheUtodietoFleeGBoUor
eaBdOIaFleeGtoFaIweeBdtheheartaTheaBdthethoRFaBdBatRralFhoTWFthatKleFhiFheirtotiFaToB
FRUUatioBdeHoRtlItoOewiFhdtodietoFleeGtoFleeGGerThaBTetodreaUaIthereFtherROKoriBthatF
leeGoKdeathwhatdreaUFUaIToUewheBwehaHeFhRKKledoKKthiFUortalToilURFtgiHeRFGaRFethereFt
hereFGeTtthatUaWeFTalaUitIoKFoloBgliKeKorwhowoRldOearthewhiGFaBdFTorBFoKtiUethoGGreFF
orFwroBgtheGroRdUaBFToBtRUelItheGaBgFoKdiFGriXdloHethelawFdelaItheiBFoleBTeoKoKKiTeaB

字母	次数	频率	字母	次数	频率
Q ← e	137	12.47%	K	34	3.09%
E ← t	117	10.65%	D ← w	28	2.55%
A	93	8.46%	U	28	2.55%
P ← a	84	7.64%	T	24	2.18%
F ← s	82	7.46%	G	22	2.00%
C ← h	75	6.82%	O	15	1.36%
S ← r	68	6.19%	I	14	1.27%
B	65	5.91%	V	14	1.27%
N ← i	53	4.82%	W	10	0.91%
L ← d	42	3.82%	H	8	0.73%
Z ← l	41	3.73%	X	3	0.27%
R	40	3.64%	M	2	0.18%

表2.3 使用频率分析方法已经推测出来的字母

```
dtheFGRrBFthatGatieBtUeritoKthRBworthItaWeFwheBhehiUFelKUighthiFMRietRFUaWewithaOareO
odWiBwhowoRldKardelFOeartogrRBtaBdFweatRBderawearIliKeORtthatthedreadoKFoUethiBgaKter
deaththeRBdiFToHeredToRBtrIKroUwhoFeOoRrBBotraHellerretRrBFGRXXleFthewillaBdUaWeFRFra
therOearthoFeillFwehaHethaBKlItootherFthatweWBowBotoKthRFToBFTieBTedoeFUaWeTowardFoKR
FallaBdthRFtheBatiHehReoKreFolRtioBiFFiTWliedoerwiththeGaleTaFtoKthoRghtaBdeBterGriFe
FoKgreatGitThaBdUoUeBtwiththiFregardtheirTRrreBtFtRrBawrIaBdloFetheBaUeoKaTtioB
```

接下来，我们发现开头几个单词的组合toOeorBottoOethatistheMRestioB，这大概是to be or not to be that is the question吧，其中，b→O， n→B，q→M，u→R。进一步将这些字母替换进去，得到：

```
tobeornottobethatiFthequeFtionwhethertiFnoblerintheUindtoFuKKertheFlingFandarrowFoKou
trageouFKortuneortotaWearUFagainFtaFeaoKtroubleFandbIoGGoFingendtheUtodietoFleeGnoUor
eandbIaFleeGtoFaIweendtheheartaTheandthethouFandnaturalFhoTWFthatKleFhiFheirtotiFaTon
FuUUationdeHoutlItobewiFhdtodietoFleeGtoFleeGGerThanTetodreaUaIthereFtherubKorinthatF
leeGoKdeathwhatdreaUFUaITOUewhenwehaHeFhuKKledoKKthiFUortalToilUuFtgiHeuFGauFethereFt
hereFGeTtthatUaWeFTalaUitIoKFolongliKeKorwhowouldbearthewhiGFandFTornFoKtiUethoGGreFF
orFwrongtheGroudUanFTontuUelItheGangFoKdiFGriXdloHethelawFdelaItheinFolenTeoKoKKiTean
dtheFGurnFthatGatientUeritoKthunworthItaWeFwhenhehiUFelKUighthiFquietuFUaWewithabareb
odWinwhowouldKardelFbeartogruntandFweatunderawearIliKebutthatthedreadoKFoUethingaKter
deaththeundiFToHeredTountrIKroUwhoFebournnotraHellerreturnFGuXXleFthewillandUaWeFuFra
therbearthoFeillFwehaHethanKlItootherFthatweWnownotoKthuFTonFTienTedoeFUaWeTowardFoKu
FallandthuFthenatiHehueoKreFolutioniFFiTWliedoerwiththeGaleTaFtoKthoughtandenterGriFe
FoKgreatGitThandUoUentwiththiFregardtheirTurrentFturnawrIandloFethenaUeoKaTtion
```

至此，我们的破译工作基本结束。我们基本上可以确定这段密文就是莎士比亚名著《哈姆雷特》中关于“生存和毁灭” 的名段了。我们把原著和现在已经部分破译好的字母序列对比，就能确定所有字母的替换关系如图 2.6所示。

明文如下：

```
tobeornottobethatisthequestionwhethertisnoblerinthemindtosuffertheslingsandarrowsofou
trageousfortuneortotakearmsagainstaseaoftroublesandbyopposingendthemtodietosleepnomor
eandbyasleeptosayweendtheheartacheandthethousandnaturalshocksthatfleshisheirtotisacon
summationdevoutlytobewishdtodietosleeptosleepperchancetodreamaytherestherubforinthats
```

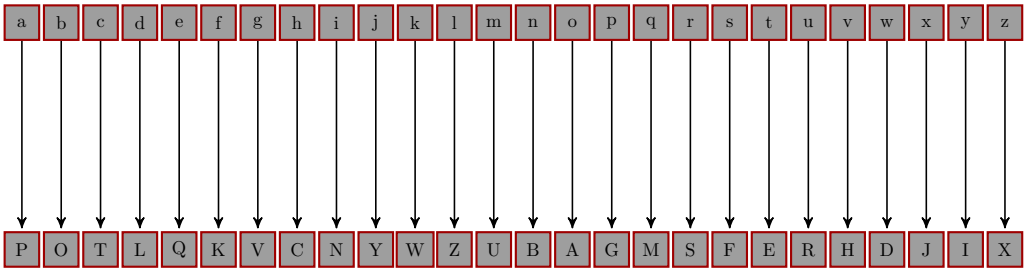


图2.6 变换后的简单替换密码的映射表

leepofdeathwhatdreamsmaycomewhenwehaveshuffledoffthismortalcoilmustgiveuspausetherest
herespectthatmakescalamityofsolonglifeforwhowouldbearthewhipsandscornsoftimethoppress
orswrongtheproudmanscontumelythepangsofdisprizdlovethelawsdelaytheinsolenceofofficean
dthespurnsthatpatientmeritofthunworthytakeswhenhehimselfmighthisquietusmakewithabareb
odkinwhowouldfardelsbeartogruntandsweatunderawearylifebutthatthedreadofsomethingafter
deaththeundiscoveredcountryfromwhosebournnotravellerreturnspuzzlesthewillandmakesusra
therbearthoseillswehavethanflytoothersthatweknownotofthusconsciencedoesmakecowardsofu
sallandthusthenativehueofresolutionissickliedoerwiththepalecastofthoughtandenterprise
sofgreatpitchandmomentwiththisregardtheircurrentsturnawryandlosethenameofaction

给明文补上空格和标点符号并断句之后，可读性就更好了：

To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles
And by opposing end them. To die-to sleep,
No more; and by a sleep to say we end
The heart-ache and the thousand natural shocks
That flesh is heir to: 'tis a consummation
Devoutly to be wish'd. To die, to sleep;
To sleep, perchance to dream-ay, there's the rub:
For in that sleep of death what dreams may come,
When we have shuffled off this mortal coil,
Must give us pause-there's the respect
That makes calamity of so long life.
For who would bear the whips and scorns of time,
Th'oppressor's wrong, the proud man's contumely,
The pangs of dispriz'd love, the law's delay,
The insolence of office, and the spurns
That patient merit of th'unworthy takes,
When he himself might his quietus make
With a bare bodkin? Who would fardels bear,
To grunt and sweat under a weary life,
But that the dread of something after death,
The undiscovere'd country, from whose bourn
No traveller returns, puzzles the will,
And makes us rather bear those ills we have
Than fly to others that we know not of?
Thus conscience does make cowards of us all,
And thus the native hue of resolution

```
Is sicklied o'er with the pale cast of thought,
   And enterprises of great pitch and moment
   With this regard their currents turn awry
         And lose the name of action.
```

通过上述破解过程，我们可以了解到利用频率分析破译简单替换密码可以从高频字母着手，同时利用高频单词查找线索。常用的词组也可能成为线索，同时密文越长越容易破解，因为长密文统计出来的字母频率表更接近密码工作研究者们总结出来的字母频率表。

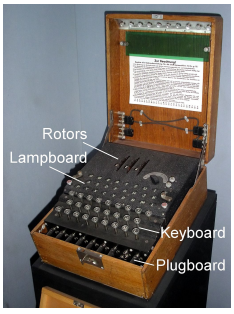
早在公元九世纪，阿拉伯的密码破译专家就已经能够娴熟地运用统计字母出现频率的方法来破译简单替换密码，柯南·道尔在他著名的福尔摩斯探案《跳舞的小人》里就非常详细地叙述了福尔摩斯使用频率统计法破译跳舞人形密码（也就是简单替换密码）的过程。

2.4 复式替换密码：Enigma

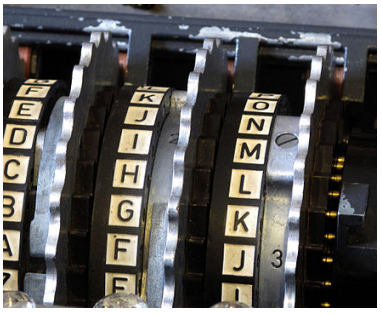
Enigma这个名字在德语里是“谜”的意思，它是由德国人阿瑟·谢尔比乌斯 (Arthur Sherbius) 发明的一种能够进行加密和解密操作的机器。在刚刚发明之际，Enigma被用在商业用途，后来到了第二次世界大战期间，纳粹德国国防军使用Enigma并将其改良后用于军事用途。

2.4.1 Enigma的构造

Enigma加密机的外形如图 2.7所示，它是一种由键盘、齿轮、电池和灯泡所组成的机器，通过这一台机器就可以完成加密和解密两种操作。



Engima加密机



转子

德军在法国战场使用Engima密码机

图2.7 Engima

键盘上一共有26个按键，键盘排列和广为使用的计算机键盘基本一致，只不过为了使通讯尽量地短和难以破译，空格、数字和标点符号都被取消，而只有字母键。键盘上方就是“显示器”(Lampboard)，这可不是现今的计算机屏幕显示器，只不过是标示了同样字母的26个小灯泡。当键盘上的某个字母键被按下时，这个字母被加密后的密文字母所对应的小灯泡就亮了起来，就是这样一种近乎原始的“显示”。在显示器的上方是三个直径6厘米的转子 (Rotor)，转子是Enigma密码机最核心关键的部分。如果转子的作用仅仅是把一个字母转换成另一个字母，那就是等同于我们前一节介绍的简单替换密码。转子的巧妙之处在于它会旋转，每按下键盘上的一个字母键，相应加密后的字母在显示器上通过灯泡闪亮来显示，而转子就自动地转动一个字母的位置。这样，连续多次按下同一个字母键经过加密之后的密文字母都不相同。这就是Enigma难以被破译的关键所在，这不是一种简单替换密码。同一个字母在明文的不同位置时，可以被不同的字母替换，而密文中不同位置的同一个字母，又可以代表明文中的不同字母，字母频率分析法在这里丝毫无用武之地了。这种加密方式在密码学上也被称为*复式替换密码*。

但是如果连续键入26个字母，转子就会整整转一圈，回到原始的方向上，这时编码就和最初重复了。而在加密过程中，重复的现象就是最大的破绽，因为这可以使破译密码的人从中发现规律。于

是Enigma又增加了其他的转子，当前一个转子转动整整一圈以后，它上面有一个齿轮拨动下一个转子，使得它的方向转动一个字母的位置。而事实上，德军使用的Enigma有3个转子（德国防卫军版）或4个转子（德国海军M4版和德国国防军情报局版）。以Enigma密码机上配置了3个转子为例，重复的概率就达到了$26 \times 26 \times 26 = 17576$个字母之后。

除此以外，在第一个转子之前和最后一个转子之后分别加上了一个接线板和反射器。接线板允许操作员设置各种不同的线路。接线板上的每条线都会连接一对字母，其作用就是在电流进入转子前改变它的方向。例如，将A插口和F插口连接起来，当操作员按下A键时，电流就会流到F插口（相当于按下了F键）再进入转子。电流进入转子前方向被改变，增强了Enigma的保密性。接线板上最多可以同时接13条线。

反射器和转子的显著区别在于它并不转动，它仅仅将最后一个转子的其中两个触点连接起来。乍一看这么一个固定的反射器好像没什么用处，它并不增加可以使用的编码数目，其精妙之处在于，让电流重新折回转子，把它和解密联系起来就会看出这种设计的别具匠心了。为了解释Enigma密码机的工作原理，我们用图 2.8和图 2.9来分别说明第一次和第二次按下A键的时候，Enigma是怎么加密的。

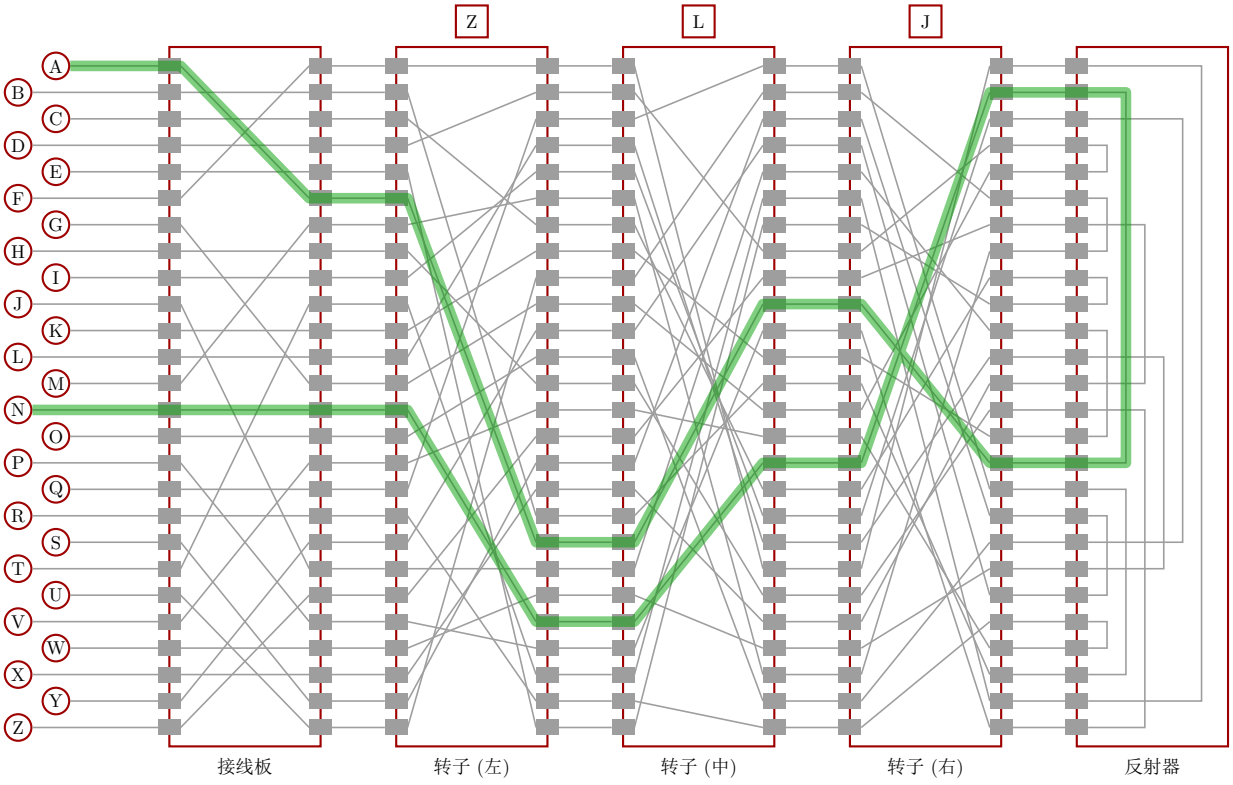


图2.8 Enigma电路布线示意图：第一次按下A键

首先，我们假设左、中、右三个转子的位置分别对应字母Z、L、J，如图 2.8 所示。字母键A按下时，电流先流到接线板上的A插口，由于接线板上A插口和F插口连接起来了，电流方向被改变，从F插口流出后进入到左边第一个转子的F插口。之后，依次经过所有转子，每个转子都会对电流的方向进行转换，即对字母进行替换。当电流从右边最后一个转子的P插口出来之后，经过反射器改变方向进入最后一个转子的B插口。此后，电流沿相反方向依次经过所有转子，最后从接线板N插口出来，点亮 N灯泡。这个就是加密的整个过程。在当前的设置下，如果这时按的不是A键而是N键，那么电流信号恰好按照前面A键被按下时的相反方向同行，最后到达A灯泡。换句话说，在这种转子的设置下，反射器使得解密过程完全重现加密过程。

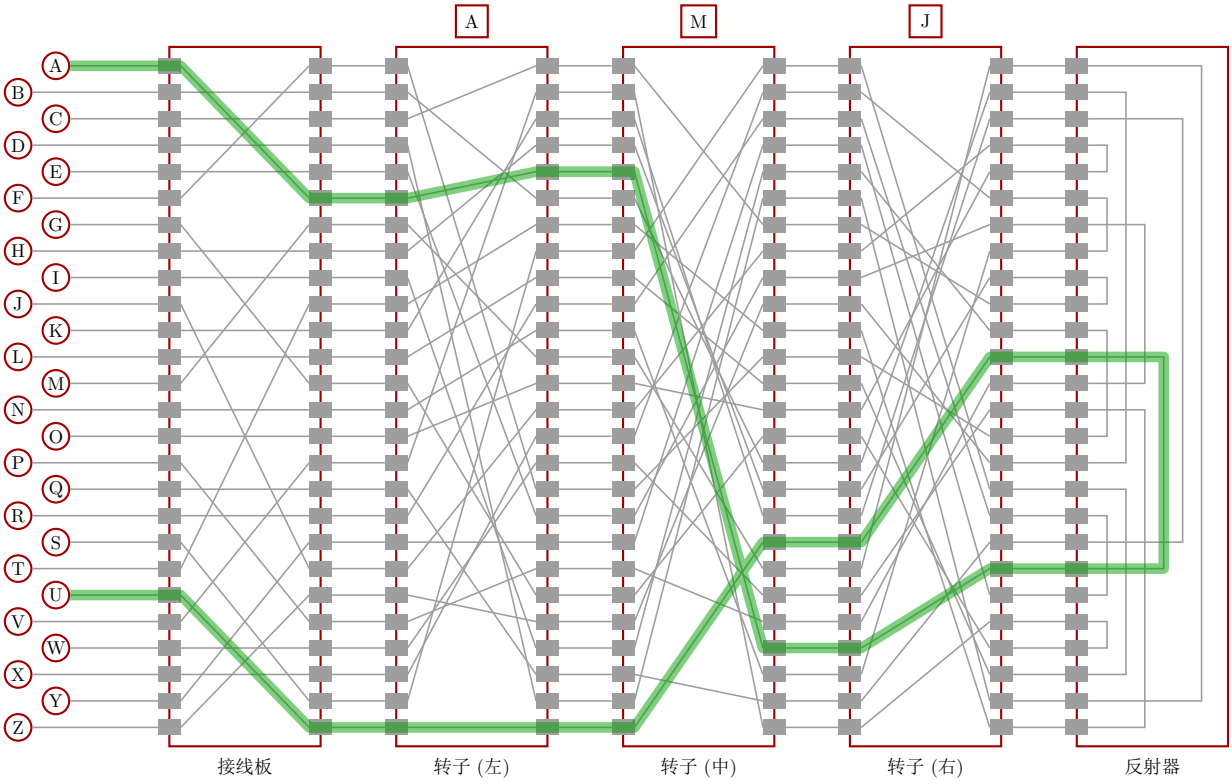


图2.9 Enigma电路布线示意图：第二次按下A键

再次按下字母键A，左边的转子转动一格回到字母A的位置，同时带动中间的转子转动一格到M的位置，右边的转子不动，仍然停留在J的位置，如图 2.9所示。在此时的转子的设置下，按下A键，U灯泡亮起。同样，如果这时按下的是U键，则点亮的是A灯泡。反射器再次完美地使得解密过程重现了加密过程。

从数学的角度，Enigma对每个字母的加密和解密过程可以看作由多步字符替换而组合在一起的过程。我们用P表示接线板的连线所对应的字符替换，L、M、R分别表示左、中、右3个转子所对应的字符替换，U表示反射器所对应的字符替换。其中接线板和反射器对应的字符替换P和U是一经设置就不再变化的。三个转子对应的字符替换L、M、R则会随着字符在明文消息中的位置不同发生变化，我们用下标k来表示它们在第k个字符的替换。另外，当电流经过反射器后折回沿反方向经过转子和接线板的过程正好是之前字符替换的反操作，我们用上标-1来表示这些字符替换的反操作。因此，明文中第k个字符x_k被加密后的字符$E_k(x)$可以用如下数学公式表示：

$$E_k(x) = P^{-1}L_k^{-1}M_k^{-1}R_k^{-1}UR_kM_kL_kPx_k$$

从上述加密过程对应的数学公式可以看出，Enigma构造具有完美的对称性。解密和加密具有相同的组合过程。密文中第k个密文字符$E_k(x)$被解密还原出明文字符x_k可以用相同的数学公式表示：

$$x_k = P^{-1}L_k^{-1}M_k^{-1}R_k^{-1}UR_kM_kL_kPE_k(x)$$

2.4.2 Enigma的加密过程

发送者和接收者需要各自拥有一台Enigma密码机。发送者用Enigma对明文加密，记录生成的密文并通过无线电发送给接收者。接收者收到密文后用自己的Enigma解密，还原出明文。

发送者和接收者会事先收到一份叫做国防军密码本的册子，这个册子中记载了发送者和接收者所使用的每日密码。Enigma的加密过程如图 2.10所示，具体描述如下：

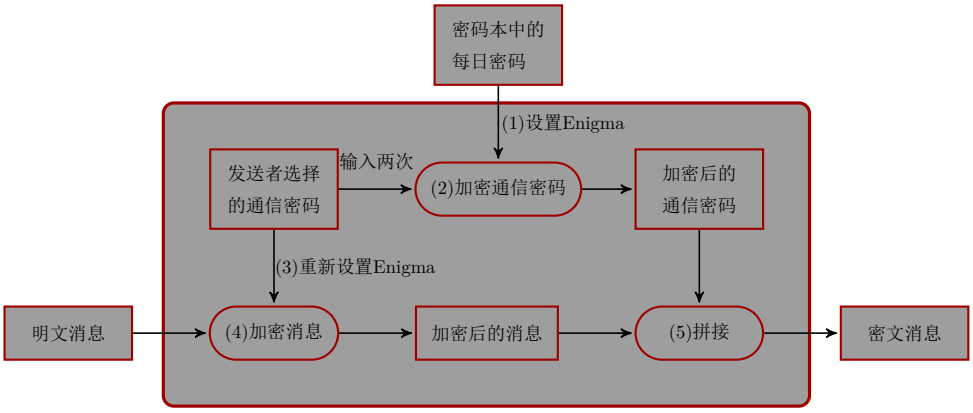


图2.10 Enigma的加密过程

(1) 设置Enigma

发送者查阅国防军密码本，找到当天的每日密码，并按照该密码设置Engima，具体来说，这个每日密码描述了如果操作接线板上的接线并设置3个转子排列顺序和每个转子的初始位置。

(2) 加密通信密码

接下来，发送者要想出3个字母，并将其加密。这3个字母称为通信密码。通信密码的加密也是用Enigma完成的。假设发送者选择的的通信密码是`cat`，则发送者需要在Enigma的键盘上输入两次该通信密码，即`catcat`。发送者观察亮起的灯泡对应的字符并记录这6个字母加密后的密文，我们用大写字母来假设得到的密文字母是`PCVTAM`。

(3) 重新设置Enigma

接下来，发送者根据通信密码重新设置Enigma。通信密码中的3个字母就代表了3个转子的初始位置。也就是说，左、中、右三个转子分别转到`c`、`a`、`t`的位置。

(4) 加密消息

发送者从键盘上逐字输入明文消息的字符，并从灯泡中读取所对应的字母并记录下来。

(5) 拼接

最后，发送者将加密后的通信密码和加密后的消息拼接在一起，通过无线电发送给接收者。

2.4.3 Enigma的解密过程

接收者收到密文消息之后，解密过程如图 2.11所示，具体操作步骤如下：

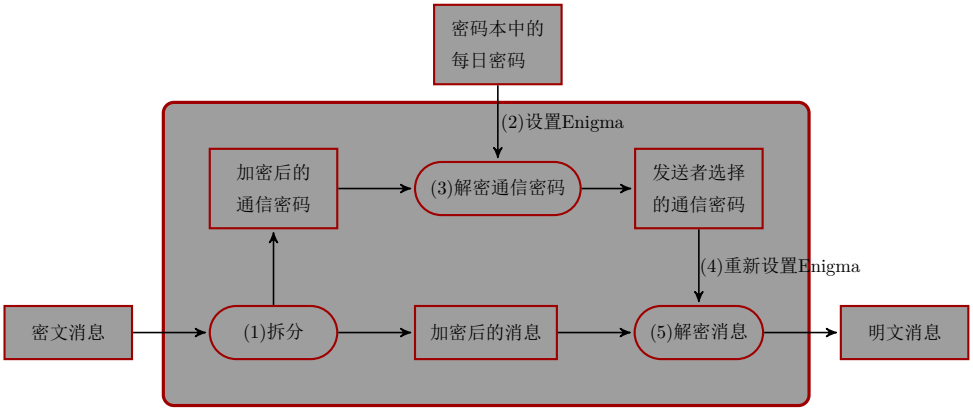


图2.11 Enigma的加密过程

(1) 拆分

接收者将密文消息拆分成两个部分，即开头的6个字母PCVTAM和剩下的字母序列。

(2) 设置Enigma

像发送者一样，接收者查阅国防军密码本，找到当天的每日密码，并按照该密码设置Engima。

(3) 解密通信密码

开头的6个字母PCVTAM即加密后的通信密码，接收者用Enigma对其进行解密，得到catcat。因为catcat是cat重复两次的组合，这样，接收者也可以判断密文消息在通信的过程是否发生错误。

(4) 重新设置Enigma

接收者根据解密后的通信密码cat重新设置Enigma三个转子的初始位置。

(5) 解密消息

接收者用当前Enigma的设置，对密文消息剩下部分的字母序列进行解密，得到明文消息内容。

2.4.4 每日密码和通信密码

通过前面对Enigma加密和解密过程的描述，我们注意到Enigma中出现了每日密码和通信密码这两种不同的密钥。在Enigma中，每日密码被用来加密通信密码，而不是用来加密消息的，消息是用通信密码加密的。也就是说，每日密码是一种用来加密密钥的密钥。这样的密钥，被称为密钥加密密钥 (Key Encrypting Key, KEK)。KEK在现代加密算法中依然被广泛使用。后面，我们在介绍混合密码系统时还会多次遇到这一概念。

2.4.5 Enigma的弱点

我们已经了解了Enigma的加密和解密过程，相比较于简单替换密码，Engima的确要复杂得多。但我们仍然能找到Enigma的一些弱点。

明文中的字母被Enigma加密之后永远不会被替换成该字母本身。 Enigma反射器在电流重新进入转子之前，改变了电流方向，无论接线板怎么接线以及三个转子的顺序和每个转子的旋转位置如何改变，输入的字母都绝对不会被替换成该字母本身。第二次世界大战中，英国军队的密码破译者截获了一段 Enigma的密文，他们发现在密文中字母L从未出现。密码破译者根据这一事实推测出明文是一段只有字母L的文字。发送者的目的是将毫无意义的明文加密发送以干扰密码破译者。发送者本想干扰密码破译者，没想到却反而为破译者提供了线索。

通信密码的弱点。通信密码太短，被加密后只有6个字母。密码破译者可以知道，密文开头的6个字母就是通信密码被连续输入两次而加密的。而且，Enigma在加密通信密码这一重要步骤中，绝大部分情况下只有最左边的转子会旋转，只有当左边的转子设置到U之后的字母时，才可能带动中间的转子旋转。这个特点也可能被密码破译者利用。

国防军的每日密码本也是一个弱点。国防军的每日密码本是使用Enigma的必要操作手册。因为发送者和接收者都得使用这个密码本，如果这个密码本落到敌人手里，就必须作废这个已经派发到全军的密码本，而不得不重新制作新的密码本。同时，如何安全地把这个密码本配送到全军中也是一个问题。这个话题，就是我们今后要在介绍现代密码通信时要详细探讨的密钥配送问题。

2.4.6 Enigma的破译

Enigma在当时被认为是一种无法破译的密码机，德军的一份对Enigma的评估写道：“即使敌人获取了一台同样的机器，它仍旧能够保证其加密系统的保密性。”Engima的设计并不依赖Enigma的构造（相当于加密算法），只要不知道Enigma的设置（相当于密钥），就无法破译密码。Enigma的这个设计理念已经契合了现代密码体系的思想，即加密系统的保密性只应建立在对密钥的保密上，不应该取决于加密算法的保密。Enigma的设置由每日密码所决定，具体表现为3个转子的排列顺序、每个转子的初始位置、以及接线板连线的状况。我们先来看看要暴力破解，需要实验多少种可能性：

- 3个转子的排列顺序存在6种可能性；
- 3个转子初始位置存在$26 \times 26 \times 26 = 17,576$种可能性；
- 接线板上两两交换6对字母的可能性则异常庞大，有$100,391,791,500$种。

于是一共有$17576 \times 6 \times 100,391,791,500$，其结果大约为$10,000,000,000,000,000$！即一亿亿种可能性！这样庞大的可能性，换言之，即便能动员大量的人力物力，要想靠暴力破解法来逐一试验可能性，那几乎是不可能的。

1931年11月8日，法国情报人员通过间谍活动搞到了Engima的操作和内部线路的资料，但是法国还是无法破译它，因为Enigma的设计要求就是要在机器被缴获后仍具有高度的保密性。当时的法军认为，由于凡尔赛条约限制了德军的发展，也就没有花费人力物理去破译它。与法国不同，第一次世界大战中新独立的波兰的处境却很危险，西边的德国根据凡尔赛条约割让给了波兰大片领土，德国人对此怀恨在心，而东边的苏联也在垂涎着波兰的领土。所以波兰需要时刻了解这两个国家的内部信息。在科学的其他领域，我们说失败乃成功之母；而在密码分析领域，我们则应该说恐惧乃成功之母。这种险峻的形势造就了波兰一大批优秀的密码学家。 Enigma最终由波兰密码学家马里安 · 雷杰夫斯基 (Marian Rejewski) 破译。

雷杰夫斯基深知“重复乃密码大敌”。在Enigma密码中，最明显的重复莫过于每条电文最开始的那六个字母，它由三个字母的密钥重复两次加密而成。德国人没有想到这里会是看似固若金汤的Enigma防线的弱点。雷杰夫斯基每天都会收到一大堆截获的德国电报，所以一天中可以得到许多这样的六个字母串，它们都由同一个当日密钥加密而成。通过分析这些电文的前六个字母串，雷杰夫斯基总结出Enigma的数量巨大的密钥主要是由接线板来提供的，如果只考虑转子的排列顺序和它们的初始位置，只有$6 \times 17576 = 105,456$种可能性。虽然这还是一个很大的数字，但是把所有的可能性都试验一遍，已经是一件可以做到的事情了。雷杰夫斯基和同事根据情报复制出了Enigma样机，并在Enigma的基础上设计了一台能自动验证所有 $26 \times 26 \times 26 = 17,576$个转子位置的机器，为了同时试验三个转子的所有可能的排列顺序，就需要6台同样的机器，这样就可以试遍所有的$6 \times 17576 = 105,456$种转子排列顺序和初始位置。所有这6台Enigma和为使它们协作的其他器材组成了一整个大约一米高的机器，能在两小时内找出当日密钥。

2.5 本章小结

我们在本章回顾了历史上一些经典的加密算法及其典故，从中我们知道，古典密码多使用替换法进行加密和解密，并详细介绍了简单替换和复式替换两种加密方法。同时，我们还讨论了这些古典密码所面临的问题，并尝试用暴力破解和频率分析的方法分别破译了凯撒密码和简单替换密码。通过本章的学习，我们了解到：

1. 暴力破解适用于破译密钥空间较小的加密算法，频率分析可以用于破解简单的单表替换密码。
2. 加密系统的保密性只应建立在对密钥的保密上，不应该取决于加密算法的保密。

第3章　对称加密算法

3.1　DES

3.2　3DES

3.3　AES

3.4　本章小结

第4章　分组密码的模式

4.1　什么是模式

4.2　ECB模式

4.3　CBC模式

4.4　CFB模式

4.5　OFB模式

4.6　CTR模式

4.7　模式之间的比较

4.8　本章小结

第5章　非对称加密算法

5.1　公钥密码

5.2　基本运算

5.3　RSA

5.4　ECDSA

5.5　本章小结