



第 1 章：密码学简介

赵俊舟

junzhou.zhao@xjtu.edu.cn

2025 年 3 月 18 日

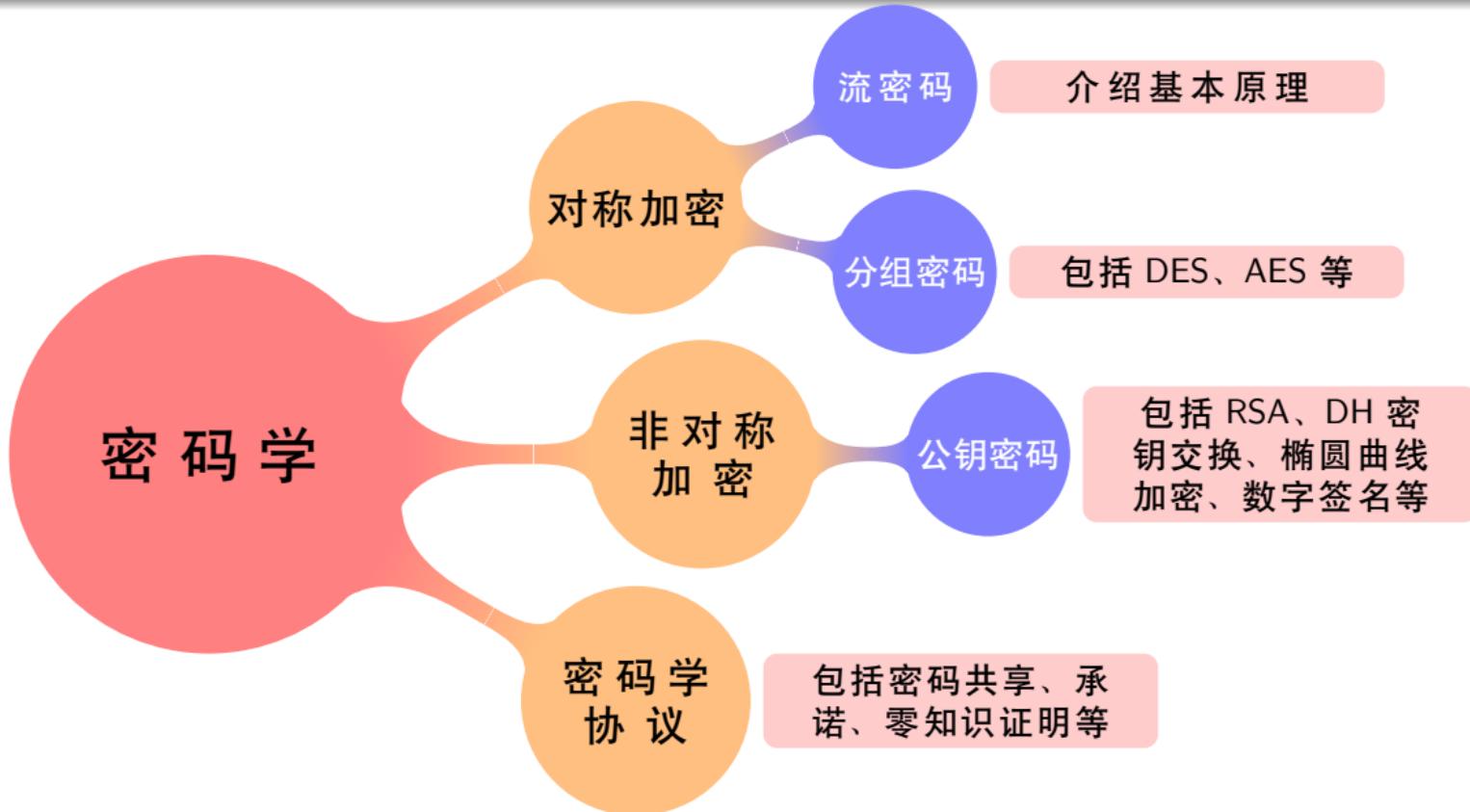
目录

- 1 课程简介
- 2 应用举例
- 3 发展历史
- 4 基本概念
- 5 古典密码

目录

- 1 课程简介
- 2 应用举例
- 3 发展历史
- 4 基本概念
- 5 古典密码

课程内容



教材及参考书

教材：

- 《现代密码学》，杨波，第五版，清华大学出版社

参考书：

- 《密码编码学与网络安全：原理与实践》，电子工业出版社
- 《Introduction to Modern Cryptography》，3rd ed., Jonathan Katz and Yehuda Lindell, CRC Press, 2021
- 《Foundations of Cryptography》，Oded Goldreich, Cambridge University Press, 2004.
- 《The Joy of Cryptography》，Mike Rosulek, 2022.

课程简介

- 学时: 32 学时, 1-8 周
- 成绩: 平时成绩 (考勤、作业等, 10%) + 闭卷考试 (90%)
- <https://junzhouzhao.github.io/courses/crypt>



目录

1 课程简介

2 应用举例

- 加密通信
- 消息认证与数字签名
- 秘密分享
- 安全多方计算
- 零知识证明

3 发展历史

4 基本概念

5 古典密码

目录

1 课程简介

2 应用举例

• 加密通信

- 消息认证与数字签名
- 秘密分享
- 安全多方计算
- 零知识证明

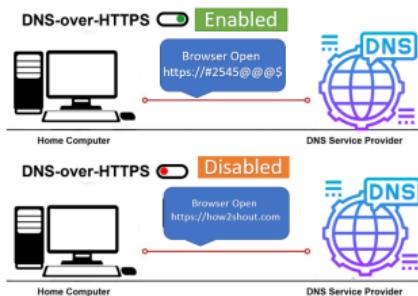
3 发展历史

4 基本概念

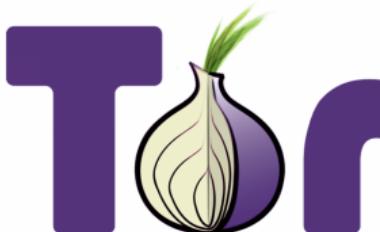
5 古典密码

应用 1：加密通信

- 加密通信协议：HTTPS, DNS over HTTPS 等

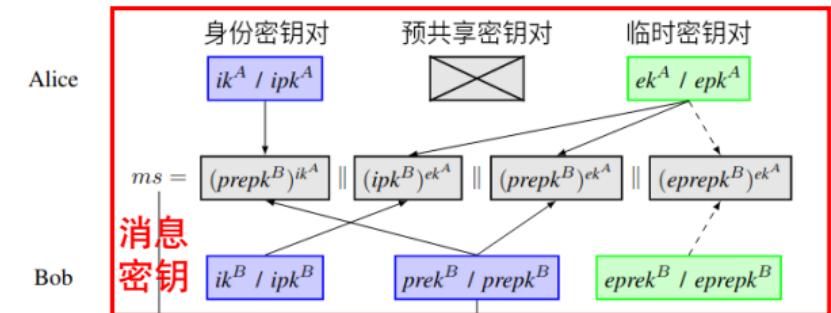


- 加密消息：Signal, WhatsApp
- 匿名网络：Tor, Yandex

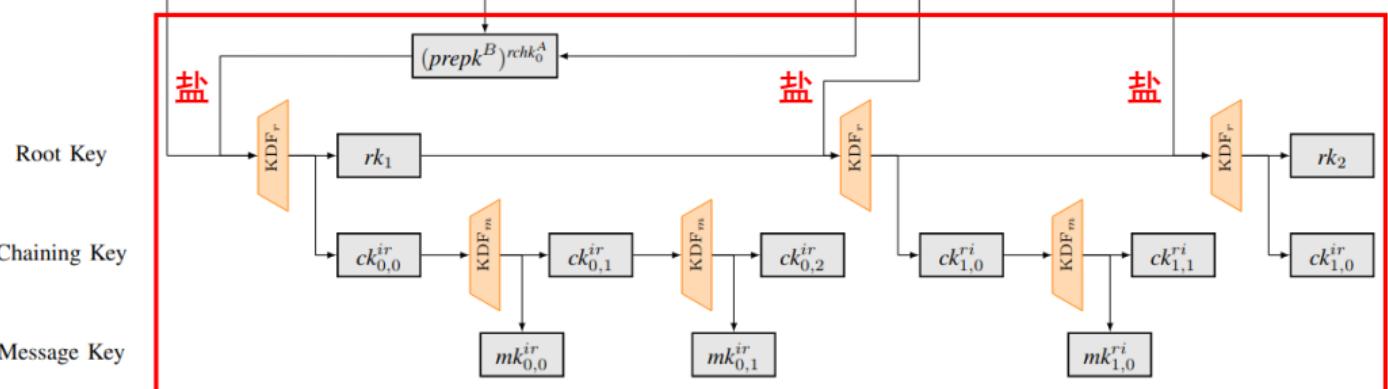
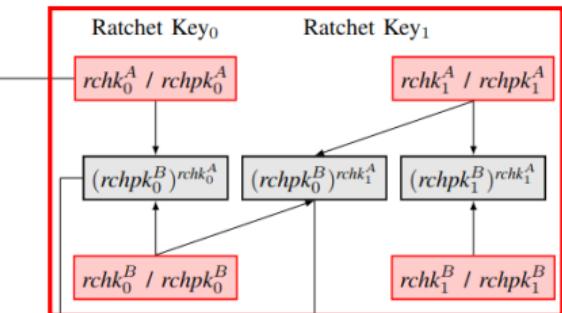


应用 1: Signal 端到端加密核心协议

X3DH 密钥交换



DH 棘轮 (生成盐)



KDF 棘轮 (生成每一轮会话的密钥)

Katriel et al. A formal security analysis of the signal messaging protocol. EuroS&P, 2019.

目录

1 课程简介

2 应用举例

- 加密通信
- 消息认证与数字签名
- 秘密分享
- 安全多方计算
- 零知识证明

3 发展历史

4 基本概念

5 古典密码

应用 2：消息认证与数字签名

 **linuxmint**

Home Download Project About Links Donate

Linux Mint 21 "Vanessa"

Linux Mint 21
Cinnamon Edition

On this page you can download Linux Mint either directly or via torrent as an ISO image.
Make sure to verify your image after downloading it.

Information

-  Size: 2.4GB
-  Installation Guide
-  Release Announcement
-  Release Notes
-  Torrent Download: 64-bit

Integrity & Authenticity

Anyone can produce fake ISO images, it is your responsibility to check you are downloading the official ones.

Download the ISO image, right-click->"Save Link As..." on the sha256sum.txt and sha256sum.txt.gpg buttons to save these files locally, then follow the instructions to verify your downloaded files.

[sha256sum.txt](#)

[sha256sum.txt.gpg](#)

[Verify](#)

12 / 105

应用 2：消息认证与数字签名

PCWorld

NEWS BEST PICKS REVIEWS HOW-TO DEALS Laptops Windows Security MORE

Linux Mint website hacked, ISO downloads replaced with backdoored operating system

If you downloaded Linux Mint on Saturday, February 20th, you may have grabbed a hacked version that includes a backdoor. Here's what you need to know.

By **Nick Mediati**

PCWorld | FEB 21, 2016 10:02 AM PST

If you downloaded Linux Mint on Saturday, February 20th, you may have unknowingly downloaded a hacked version of the operating system.

According to a [blog post on the Linux Mint site](#), hackers broke into the Linux Mint website at some point on Saturday and made changes in order to direct users toward downloading "a modified Linux Mint ISO, with a backdoor in it." Using the hacked version could allow hackers to steal your private information. According to Linux Mint, the hack only affects those who downloaded the Linux Mint 17.3 Cinnamon edition from the Linux Mint website on Saturday.



应用 2：消息认证与数字签名

文件内容：三个 ISO 文件的消息摘要及消息摘要的数字签名。

```
→ linuxmint ls -l
total 2839692
-rw-rw-r-- 1 jzzhao jzzhao 2907832320 Nov  6 20:04 linuxmint-22-cinnamon-64bit.iso
-rw-rw-r-- 1 jzzhao jzzhao         286 Nov  6 20:15 sha256sum.txt
-rw-rw-r-- 1 jzzhao jzzhao        833 Nov  6 20:09 sha256sum.txt.gpg
→ linuxmint cat sha256sum.txt
7a04b54830004e945c1eda6ed6ec8c57ff4b249de4b331bd021a849694f29b8f *linuxmint-22-cinnamon-64bit.iso
78a2438346cf69a1779b0ac3fc05499f8dc7202959d597dd724a07475bc6930 *linuxmint-22-mate-64bit.iso
55e917b99206187564029476f421b98f5a8a0b6e54c49ff6a4cb39dcfeb4bd80 *linuxmint-22-xfce-64bit.iso
→ linuxmint cat sha256sum.txt.gpg
-----BEGIN PGP SIGNATURE-----
iQIzBAABCgAdFiEEJ96xVkJG88719KRMA+Ea6JbrgkFAmaenwYACgkQMA+Ea6Jb
rgmbVRAAnrlVmEBRWce/lwImwzjEj3FIDwJ70A4h9+gnBVJmXysjY0/ubZTBkFGE
M/5w0gQfyA06m89zRhcyCo6nG4MHXdJ15AenuTBER5V/i4cN/VCN0x0ktCyK0G7o
pXLiVkp5LqY2acdiwjajQIYBuLDuaTuMJSFgYA/dhuTy6u2U3Av1due2Q0rass5y
8Wkn00snS4yQiRLQYgHji1Kg7CBG5GyyZFLt+vYVDbw4U5vQ8Dxv1gDxnNbT6j
uNySskMicQuqVkadif2jykWTTfBfsl7l44AZBsJdrT9j0rcaNFiRjea00x0FF9
+CZDFdcriKcAhyw+dVLeJeUFzmFha7o0czSop2KOLQHCC866ikYX9SvCy5kYD4et
0+bN0XA/W/PaVtjhRxIMgZfWYjzD4EYNyAwunyYbT/mq2WF1KwC34V3mu1343my
AbxmdzaWIfPpADIMqf2cJrm3FaDictUfvQ21g9GZR2nKV9U+0YEy7+LQZLLf3Vn
e0Kns0+LD5DB0jirD12mgGZX05abIhoH7SYFppuIyWxe1Sc2Yp8h1RMhE6uo4PBA
fRThx4T3v5q+GmN7lmtvCpxkuVFr0FxHUpHuMDxNRo2IwJ0TyJQu+Ult6oRoEaHY
iX2NkgmIt5LQ+UdyT0ol6KqsfxlU/0yrkJjctC6Nu8hJdwrCAmQ=
=hr4l
-----END PGP SIGNATURE-----
```

应用 2：消息认证与数字签名

文件完整性验证：确定所下载的文件是否被篡改过

```
→ linuxmint sha256sum -c sha256sum.txt
linuxmint-22-cinnamon-64bit.iso: OK ←
sha256sum: linuxmint-22-mate-64bit.iso: No such file or directory
linuxmint-22-mate-64bit.iso: FAILED open or read
sha256sum: linuxmint-22-xfce-64bit.iso: No such file or directory
linuxmint-22-xfce-64bit.iso: FAILED open or read
sha256sum: WARNING: 2 listed files could not be read
```

身份验证：确定所下载的文件是否来自官方

```
→ linuxmint gpg --verify sha256sum.txt.gpg sha256sum.txt
gpg: Signature made Tue 23 Jul 2024 02:03:50 AM CST
gpg:                               using RSA key 27DEB15644C6B3CF3BD7D291300F846BA25BAE09
gpg: Good signature from "Linux Mint ISO Signing Key <root@linuxmint.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                               There is no indication that the signature belongs to the owner.
Primary key fingerprint: 27DE B156 44C6 B3CF 3BD7  D291 300F 846B A25B AE09
```

应用 2：消息认证与数字签名

篡改消息摘要：等效于篡改 ISO 文件内容

```
→ linuxmint vi sha256sum.txt
→ linuxmint cat sha256sum.txt
8a04b54830004e945c1eda6ed6ec8c57ff4b249de4b331bd021a849694f29b8f *linuxmint-22-cinnamon-64bit.iso
78a2438346cfe69a1779b0ac3fc05499f8dc7202959d597dd724a07475bc6930 *linuxmint-22-mate-64bit.iso
55e917b99206187564029476f421b98f5a8a0b6e54c49ff6a4cb39dcfeb4bd80 *linuxmint-22-xfce-64bit.iso
```

导致完整性验证失败：

```
→ linuxmint sha256sum -c sha256sum.txt
linuxmint-22-cinnamon-64bit.iso: FAILED
sha256sum: linuxmint-22-mate-64bit.iso: No such file or directory
linuxmint-22-mate-64bit.iso: FAILED open or read
sha256sum: linuxmint-22-xfce-64bit.iso: No such file or directory
linuxmint-22-xfce-64bit.iso: FAILED open or read
sha256sum: WARNING: 2 listed files could not be read
sha256sum: WARNING: 1 computed checksum did NOT match
```

身份验证同样失败：

```
→ linuxmint gpg --verify sha256sum.txt.gpg sha256sum.txt
gpg: Signature made Tue 23 Jul 2024 02:03:50 AM CST
gpg:                               using RSA key 27DEB15644C6B3CF3BD7D291300F846BA25BAE09
gpg: BAD signature from "Linux Mint ISO Signing Key <root@linuxmint.com>" [unknown]
```

目录

1 课程简介

2 应用举例

- 加密通信
- 消息认证与数字签名
- 秘密分享**
- 安全多方计算
- 零知识证明

3 发展历史

4 基本概念

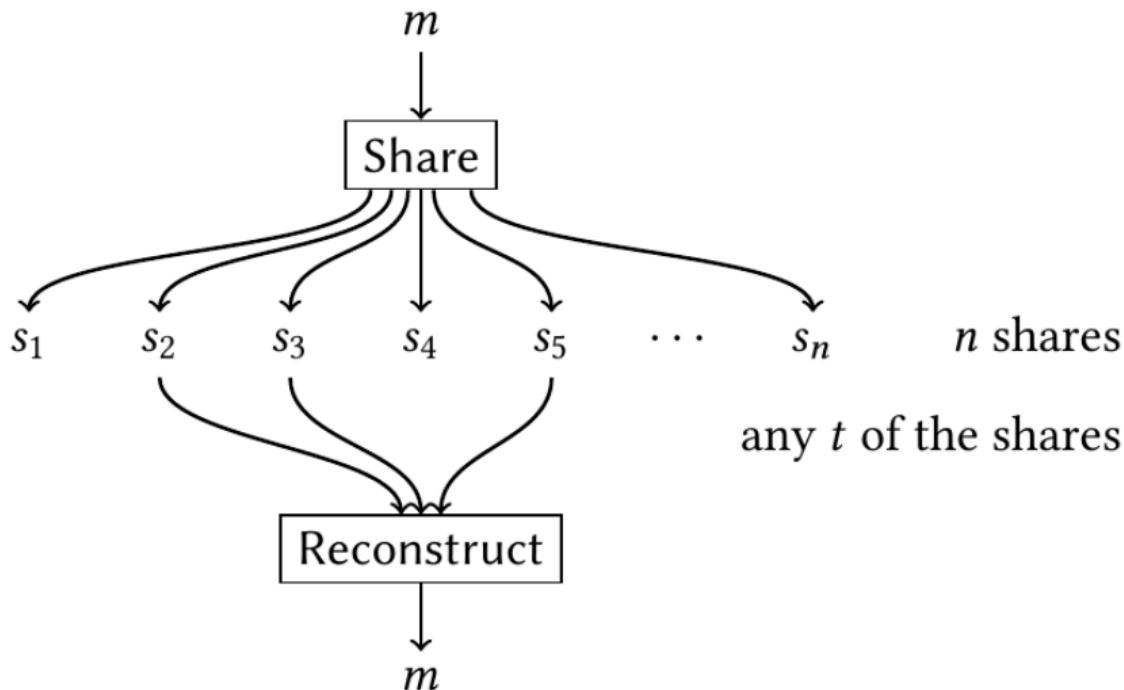
5 古典密码

应用 3：秘密共享 (Secret Sharing)



俄罗斯的核武器系统密码保存在总统手中，国防部也保存一份。当最高层作出发动核打击决定时，总统和国防部长分别通过不同的通讯网，将两组不同的密码传送到总参作战部电脑控制中心，经过运算形成一组有 12 位数字的第三套预发密码，再由特种通讯系统通过特殊频率传递给核潜艇指挥官和导弹发射基地，指挥官再按照程序输入密码，完成操作。

应用 3: (t, n) -门限秘密共享



消息 m 被拆分为 n 份交给 n 个人，只有当其中至少 t 个人同意时，才能共同恢复原始消息 m 。

目录

1 课程简介

2 应用举例

- 加密通信
- 消息认证与数字签名
- 秘密分享
- **安全多方计算**
- 零知识证明

3 发展历史

4 基本概念

5 古典密码

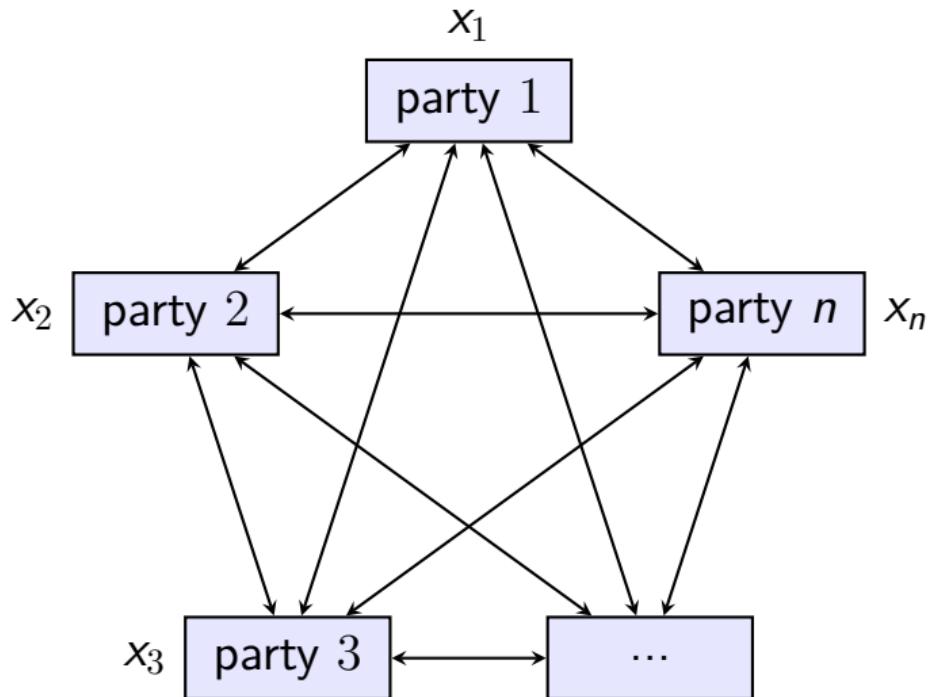
应用 4：百万富翁问题与安全多方计算

两个富翁在不泄漏个人财产具体数额的前提下，如何比较谁更富有¹？



¹Yao A C. Protocols for secure computations[C]// Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, 1982.

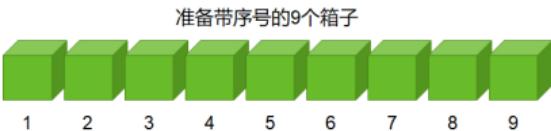
应用 4：安全多方计算的一般形式



$$(y_1, \dots, y_n) = F(x_1, \dots, x_n)$$

应用 4：百万富翁问题的一种通俗解法

- 假定富翁 Alice 和 Bob 的财富值 x, y 取值范围为 $\{1, \dots, 9\}$ (以 $x = 3, y = 7$ 为例)。
- 第一步：Alice 准备带序号 $1 \sim 9$ 的 9 个箱子。



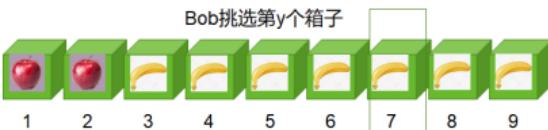
- 第二步：Alice 在箱子内分别放入苹果和香蕉。如果箱子序号小于自己的财富值 x ，则放入苹果；否则放入香蕉。



- 然后，Alice 将带序号的 9 个箱子密封后交给 Bob。

应用 4：百万富翁问题的一种通俗解法

- 第三步：Bob 收到带序号的箱子后，挑选序号与自己财富值 y 相等的箱子，然后撕掉序号，扔掉其他箱子。



- 第四步：Bob 当着 Alice 的面打开选中的箱子，如果是香蕉，则 $y \geq x$ ；如果是苹果则 $x > y$ 。



- 虽然 Alice 和 Bob 都看到箱子里放的是香蕉（MPC 计算结果），但由于 Alice 不知道箱子序号，所以 Alice 不知道 Bob 选的是第几个箱子，即 Alice 不知道 Bob 的财富 y ；
- Bob 不知道序号 1 ~ 6 的 6 个箱子里从第几个箱子开始由苹果变成香蕉，即 Bob 不知道 Alice 的财富 x 。

目录

1 课程简介

2 应用举例

- 加密通信
- 消息认证与数字签名
- 秘密分享
- 安全多方计算
- 零知识证明

3 发展历史

4 基本概念

5 古典密码

应用 5：零知识证明——寻找瓦利

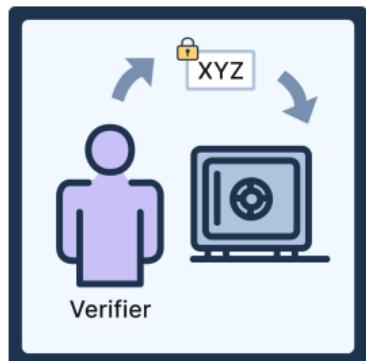


瓦利

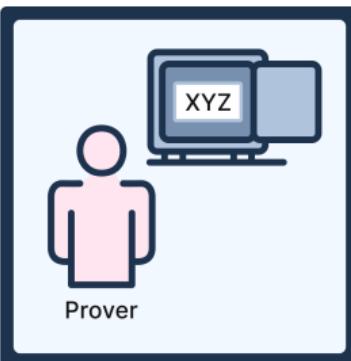


应用 5：零知识证明——成员证明问题

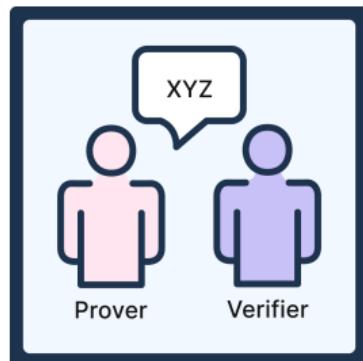
- **问题**：你遇到一个不认识的人，但她声称也是你所在团队的成员，怎样做才可以信任她？
- 幸运的是，你的团队有一个保险箱，只有你的团队成员知道保险箱密码，可以打开保险箱。



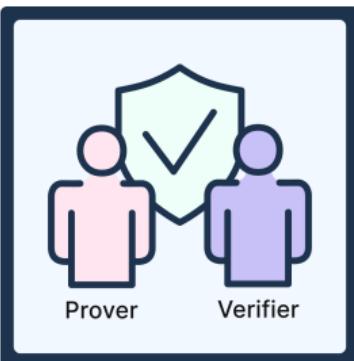
1. 验证者写一条秘密信息并放入锁定的保险箱中



2. 符合要求的证明者有密码，能打开保险箱



3. 证明者记下保险箱中的秘密信息并交给验证者



4. 验证者确信证明者真的知道密码，因此可以信任

应用 5：零知识证明——核裁军谈判

- **问题**：一个国家的核武库属于国家最高机密，两个国家进行谈判后决定削减核弹头数量分别到一个约定的具体数目。
- 几年后，两个国家都声称核弹头数量已经削减到了约定数量。如何使一个国家相信另一个国家确实削减到了约定数量？

ARTICLE

doi:10.1038/nature13457



A zero-knowledge protocol for nuclear warhead verification

Alexander Glaser¹, Boaz Barak² & Robert J. Goldston³

ARTICLE

Received 27 Feb 2016 | Accepted 12 Aug 2016 | Published 20 Sep 2016

DOI: 10.1038/ncomms12890

OPEN

A physical zero-knowledge object-comparison system for nuclear warhead verification

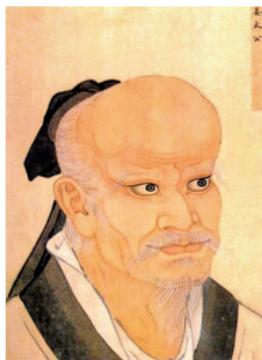
Sébastien Philippe¹, Robert J. Goldston², Alexander Glaser¹ & Francesco d'Errico^{3,4}

目录

- 1 课程简介
- 2 应用举例
- 3 发展历史
- 4 基本概念
- 5 古典密码

中国古代加密技术：姜子牙阴符

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。



(? - 1015BC 或
1036BC)



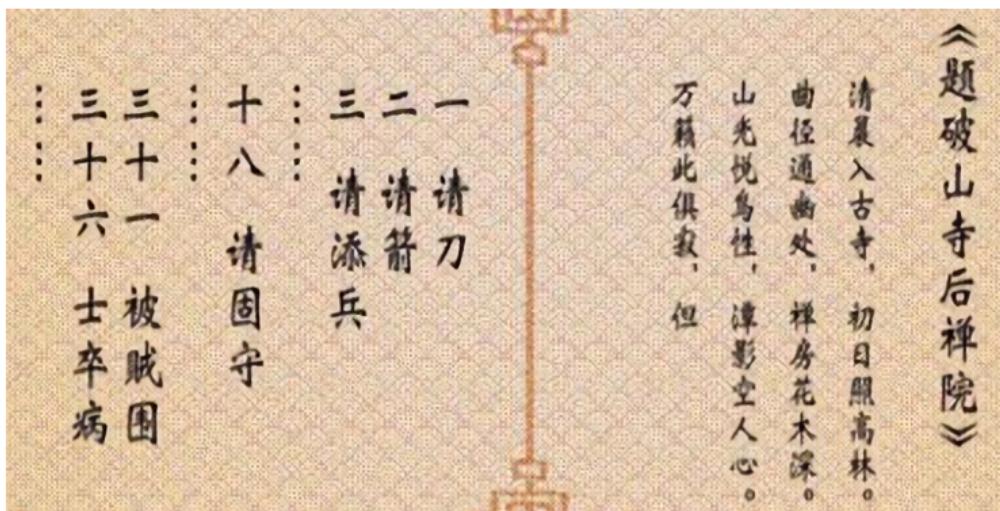
中国古代加密技术：牙璋、虎符

一份在君王手里，一份在将领手里，两两相对才能调动军队。



中国古代加密技术：五言律诗秘钥加密法

- 约定一首 40 字的五言律诗保密，文字不得重复；
- 如果需要补充兵力，前方将领从密码本中查出“请添兵”的编号，是第三，则将律诗中第三个字写到文书中，发给后方；
- 后方从律诗中找到该字的位置，从而得到编号，得知情报。



中国古代加密技术：戚继光声韵加密法，反切法

柳边求气低，波他争日时。莺蒙语出喜，打掌与君知。

春花香，秋山开，嘉宾欢歌须金杯，孤灯光辉烧银缸。之东郊，过西桥，鸡声催初天，奇梅歪遮沟。

- **加密**：前一首诗歌的前 15 个字作为声母，依次编号为 1 – 15；后一首诗歌的 36 字为韵母，按顺序编号为 1 – 36；然后再将当时字音的八种声调，也按顺序编号为 1 – 8，就编写出完整的“反切码”体系。
- **解密**：如果密码的编号是“5-25-2”，5 是声母“低”字，25 是韵母“西”字，2 是声调的二声。据此，“5-25-2”就可以读为“敌”字。

Auguste Kerckhoffs 与柯克霍夫准则

- 奥古斯特 · 柯克霍夫 (Auguste Kerckhoffs, 1835 – 1903), 荷兰语言学家与密码学家。
- 人们尝试发送加密信息已有 2000 多年历史, 但是在 1900 年以前, 只有两个想法对现代密码学产生了重大影响, 其一为**柯克霍夫准则**。
- 柯克霍夫准则体现在所有现代密码学中。
- 香农后来提出了类似观点: “*The enemy knows the system*”, 称为**香农准则**。



柯克霍夫

柯克霍夫准则

A cryptographic system should be secure even if everything about the system, except the key, is public knowledge.

William Friedman

- 威廉·弗里德曼 (William Friedman, 1891 – 1969), 美国陆军密码专家。
- 1918 年发表著作《The Index of Coincidence and its Applications in Cryptography》，被认为是现代密码学最重要的著作之一。
- 1930 年代，他领导了陆军的一个研究部门 Signals Intelligence Service (SIS)，其中一部分服务一直延续到五十年代。
- 三十年代晚期，在他的指导下，Frank Rowlett 破解了日本人的 PURPLE 加密机（紫密），截获了日本的大量外交和军事的秘密。



弗里德曼

Claude Shannon 与分组密码设计的准则

- 克劳德 · 香农 (Claude Shannon, 1916 – 2001), 美国数学家、信息论创始人。
- 1948 年, 香农发表《The Communication Theory of Secrecy System》, 成为现代密码学理论基础。
- 1949 年, 香农发表论文《保密系统的通信理论》, 首次将密码学研究置于坚实的数学基础上。
- 证明了一次一密 (one-time pad) 的理论安全。
- 提出分组密码设计应遵循的准则: 扩散和混淆。
- 证明了消息冗余使得破译者统计分析成功的理论值 (唯一解距离)。



香农

John Nash 与密码学安全性的一般性准则

- 约翰·纳什 (John Nash, 1928 – 2015), 美国数学家, 博弈论创建者。
- 1955 年, 纳什在一封给 NSA 的信中提出了计算安全的思想。
- 遗憾的是, 纳什的信一直处于机密状态, 直到 2012 年才公开。如果纳什的想法能提早公开, 那么势必会加速现代密码学的发展。



约翰·纳什

计算安全的思想

It doesn't really matter whether attacks are impossible, only whether attacks are computational infeasible.

纳什的信件

We see immediately that in principle the enemy needs very little information to begin to break down the process. Essentially, as soon as n bits of ~~enciphered~~ message have been transmitted the key is about determined. This is no security, for a practical key should not be too long. But this does not consider how ~~easy~~^{or difficult} it is for the enemy to make the computation determining the key. If this computation

, although possible in principle, were sufficiently long at best then the process could still be secure in a ~~practical~~ ~~sense~~.

沉寂期

1949 – 1967，密码学研究处于沉寂时期。

Horst Feistel 与数据加密标准 DES

- Horst Feistel (1915 – 1990), 德裔美国密码学家。
- Horst Feistel 在 IBM 工作期间于 1971 年发明分组加密算法 Lucifer 密码, 提出 Feistel 密码结构。
- Feistel 密码结构激发了 70 年代对数据加密标准 DES 的研发高潮。
- 1976 年 – 1977 年, 美国国家标准局正式公布实施数据加密标准 DES。



Horst Feistel

Whitfield Diffie, Matin Hellman 与公钥密码学

- 1975 年, W. Diffie 和 M. Hellman 发表论文《New Directions in Cryptography》, 提出公开密钥思想, 揭开现代密码学研究的序幕。
- 该开创性研究获得 2015 年图灵奖。



W. Diffie

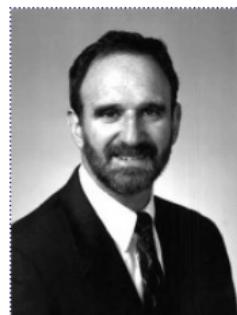
Stanford | News

Home Find Stories For Journalists Contact

Stanford Report, March 1, 2016

Stanford cryptography pioneers Whitfield Diffie and Martin Hellman win ACM 2015 A.M. Turing Award

The groundbreaking algorithm from Whitfield Diffie and Martin Hellman enabled a secure Internet and sparked a clash with the NSA that foreshadowed current privacy battles between government agencies and Silicon Valley companies.



M. Hellman

Whitfield Diffie, Matin Hellman 与公钥密码学

644

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976

New Directions in Cryptography

Invited Paper

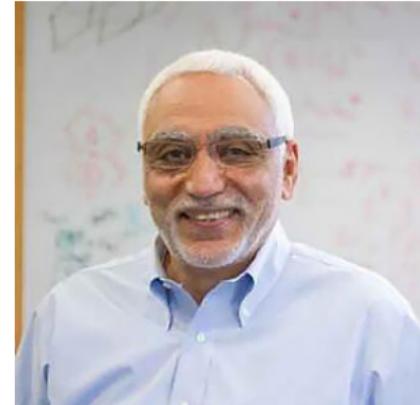
WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation

R. Rivest, A. Shamir, L. Adleman 及 A. El Gamal

- 1977–1978, Ronald Rivest, Adi Shamir, Len Adleman 第一次提出公开密钥密码系统的实现方法 RSA。
- 1981, 成立 International Association for Cryptology Research。
- 1985, Abbas El Gamal 提出概率密码系统 ElGamal 方法。
- 2000, Advanced Encryption Standard (AES)



姚期智

- 1946 年 12 月 24 日出生于中国上海，祖籍湖北孝感，幼年随父母移居中国台湾，中科院院士。
- 2000 年图灵奖获得者，是唯一获得该奖的华人学者（截至 2023 年）。
- **贡献 1**：建立理论计算机科学的重要次领域：通讯复杂性和伪随机数生成计算理论；
- **贡献 2**：奠定现代密码学基础，在基于复杂性的密码学和安全形式化方法方面有根本性贡献；
- **贡献 3**：解决线路复杂性、计算几何、数据结构及量子计算等领域的开放性问题并建立全新典范。



姚期智

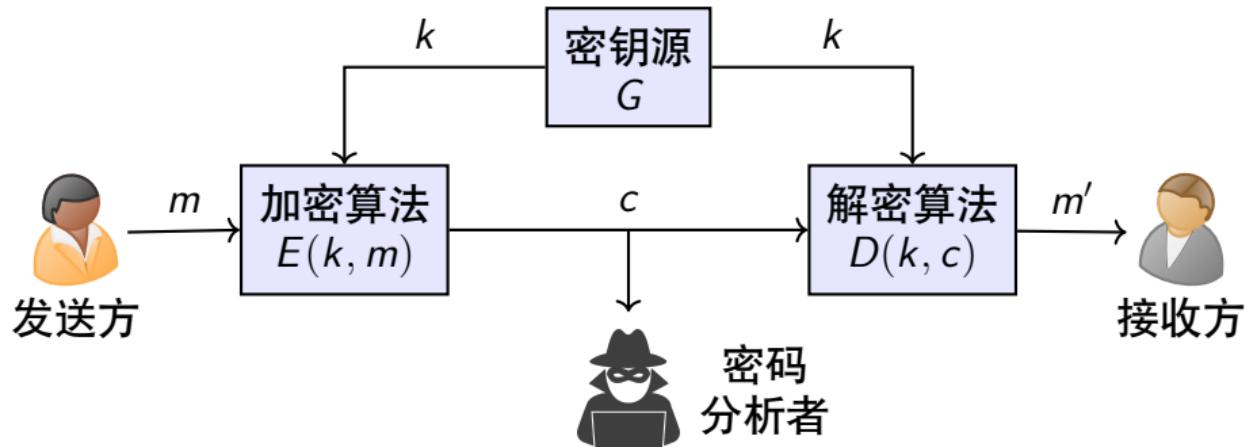
目录

- 1 课程简介
- 2 应用举例
- 3 发展历史
- 4 基本概念
- 5 古典密码

密码学基本术语

- **Cryptology**: 保密学, 源自希腊语;
- **Cryptography**: 密码编码学, 研究如何将明文转换为密文;
- **Cryptanalysis**: 密码分析学, 研究如何破译密文得到明文或获得密钥;
- **Cipher**: 加密方法;
- **Encipher, encryption**: 将明文转换成密文的过程;
- **Decipher, decryption**: 将密文还原成明文的过程;
- **Plaintext (cleartext)**: 原始的可读数据, 称为消息或明文;
- **Ciphertext (cryptogram)**: 加密后得到的密文;
- **Key**: 密钥, 对加密与解密过程进行控制的参数

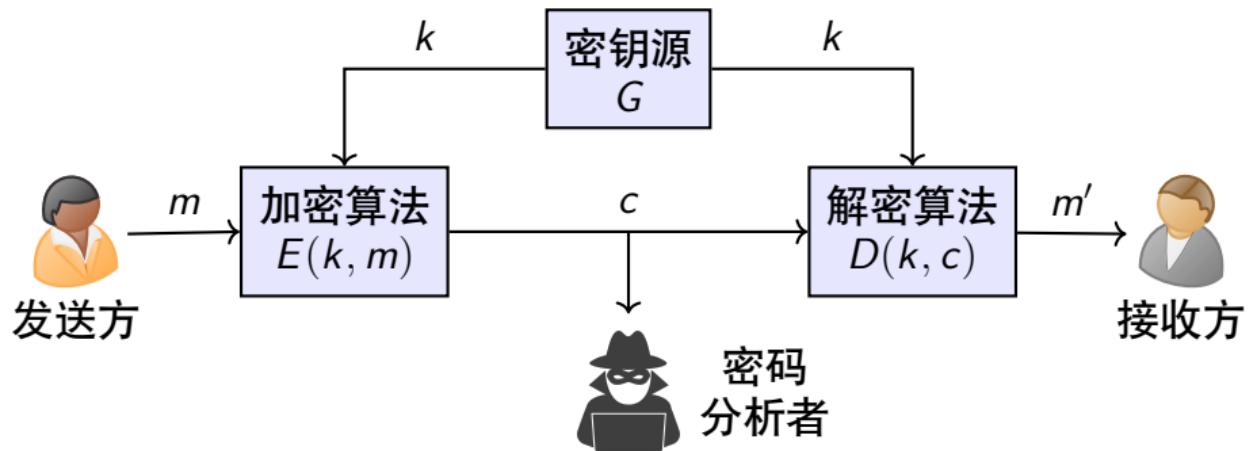
保密通信系统的一般模型



由定义在空间 $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ 上的运算 (G, E, D) 构成，其中

- \mathcal{K} 为密钥空间， \mathcal{M} 为明文空间， \mathcal{C} 为密文空间
- $G: \{0, 1\}^* \mapsto \mathcal{K}$ 为密钥生成函数或密钥源
- $E: \mathcal{K} \times \mathcal{M} \mapsto \mathcal{C}$ 为加密运算，并且 $c = E(k, m)$
- $D: \mathcal{K} \times \mathcal{C} \mapsto \mathcal{M}$ 为解密运算，并且 $m = D(k, c)$

保密通信系统的要求



- **正确性**: $D(k, E(k, m)) = m$
- **保密性**: 由密文或明密文推测密钥和明文，在计算上不可行。
- **Kerckhoffs 准则**: 系统的安全性不依赖于对加解密算法的保密，而是密钥。
- **计算效率**: 加解密算法的计算效率应足够高，便于系统实现。

理论安全、计算安全与实际安全

- **理论安全**要求密码分析者不能由密文获取关于明文的**任何**信息。根据香农定理，要实现理论安全，要求密钥长度不能短于明文长度，因此理论安全不切实际。
- **计算安全**考虑密码分析者的**实际运算能力**，如果一个运行时间最多为 t 的敌手最多只能以概率 ϵ 成功破解加密体制，则称该加密体制计算安全。
 - 例如，一个敌手使用目前最先进的计算机运行时间不超过 200 年，破解密码体制的概率不超过 2^{-60} 。
- **实际安全**将一个密码体制的安全构建在一个**数学难题**之上，此时密码体制的安全性等于该数学问题的困难程度。这时候，只要证明了该问题困难程度符合安全需求，那么可以认为密码体制是实际安全的。

密码体制的分类

- **对称密码体制**：加解密密钥相同，加密能力和解密能力是结合在一起的，开放性差；
- **非对称密码体制**：加解密密钥不同，从一个密钥导出另一个密钥是计算上不可行的，加密能力和解密能力是分开的，开放性好。
- **流密码**：也称序列密码，明文以比特流或字节流的形式进行加解密；
- **分组密码**：明文按照定长进行分组，然后对分组整体进行加解密。
- **确定型密码**：当明文和密钥确定后，密文也就唯一地确定了；
- **概率型密码**：当明文和密钥确定后，密文产生不确定。

密码攻击类型 (Threat Model)

对密码分析者攻击能力的假设：除了知道加解密算法，还知道下面信息：

- **唯密文攻击**：知道密文；
- **已知明文攻击**：除了知道密文外，还知道一些明密文对；
- **选择明文攻击**：知道密文，且可选择一些明密文对用于密码分析；
- **选择密文攻击**：知道密文，且可选择一些密文及其对应明文对用于密码分析；
- **选择文本攻击**：同时可选择明文或选择密文。

从上往下，密码分析者的攻击能力逐渐增强。

目录

1 课程简介

2 应用举例

3 发展历史

4 基本概念

5 古典密码

- 代换密码
- 置换密码
- 转轮密码机

目录

1 课程简介

2 应用举例

3 发展历史

4 基本概念

5 古典密码

- 代换密码

- 置换密码

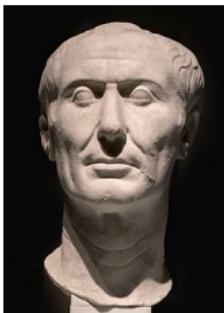
- 转轮密码机

代换密码 (Substitution Cipher)

- 代换密码 (也叫替换密码) 将一个明文字母替换成其他字母，实现加密，通过逆替换实现解密，是最简单的密码。
- 常见的代换密码包括：
 - 凯撒密码或移位密码
 - Playfair 密码
 - Hill 密码
 - Vigenère 密码
 - 一次一密 (One-Time Pad)
- 分为单表代换、多表代换等。

凯撒密码 (Caesar Cipher)

- 已知最早的代换密码是由古罗马时期的 Julius Caesar 发明的凯撒密码 (Caesar Cipher)，用于加密作战命令。

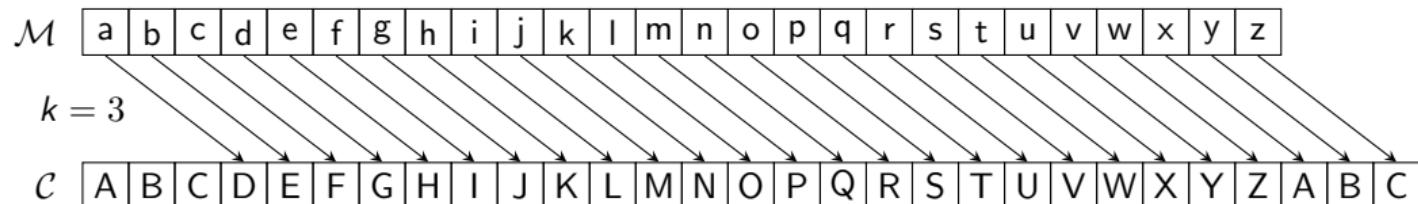


(公元前 100 年 – 公元前 44 年)

There are also letters of his to Cicero, as well as to his intimates on private affairs, and in the latter, if he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out...

The Lives of the Caesars, the Deified Julius (110CE)

凯撒密码



- 将每个明文字母向前循环移 3 位，得到密文字母；对密文进行相反操作得到明文。
- 加密**： $c = (m + 3) \bmod 26$
- 解密**： $m = (c - 3) \bmod 26$

例 (凯撒密码加密)

利用凯撒密码加密 “begin the attack now”，忽略空格，得到密文
EHJLQWKHDWWDFNQRZ

移位密码

- 凯撒密码其实不存在密钥，任何知道凯撒密码算法的人都可以轻易破解密码。
- 移位密码对凯撒密码进行改进，引入密钥 $k \in \{0, \dots, 25\}$ ，表示循环移位的位数。

$$\text{加密: } c = E(k, m) \triangleq (m + k) \bmod 26$$

$$\text{解密: } m = D(k, c) \triangleq (c - k) \bmod 26$$

- 移位密码的安全性如何？

例 (破解移位密码)

尝试破解密文 EHJLQWKHDWWDFNQRZ

移位密码的安全性

k	尝试解密后
0	ehjlqwkhdwwdfnqrz
1	dgikpvjgcvvcempqy
2	cfhjouifbuubdlopx
3	begintheattacknow
4	adfhamsgdzsszbjmnv
5	zceglrfcyrryailmu
6	ybdfkqebxqqxzhklt
7	xacejpdawppwygjks
8	wzbdiooczvoovxfijr
	...

- 移位密码的密钥空间大小为 26
- 容易对密钥空间进行**穷举攻击** (brute-force attack)

密码安全原则一：密钥空间应足够大
任何安全的密码体制都应该具有足够大的密钥空间，使穷举攻击不可行。

- 多大的密钥空间才算**足够大**？

对不同密钥空间进行穷举攻击的开销

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/ μ s	Time Required at 10^6 Decryptions/ μ s
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu$ s = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu$ s = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu$ s = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu$ s = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu$ s = 6.4×10^{12} years	6.4×10^6 years

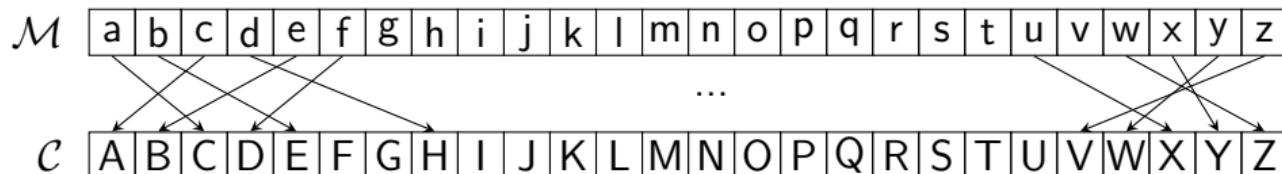
对不同密钥空间进行穷举攻击的开销

Amazon EC2 集群执行不同数量 CPU 时钟需要的花费
(根据 2018 年定价)

<i>clock cycles</i>	<i>approx cost</i>	<i>reference</i>
2^{50}	\$3.50	<i>cup of coffee</i>
2^{55}	\$100	<i>decent tickets to a Portland Trailblazers game</i>
2^{65}	\$130,000	<i>median home price in Oshkosh, WI</i>
2^{75}	\$130 million	<i>budget of one of the Harry Potter movies</i>
2^{85}	\$140 billion	<i>GDP of Hungary</i>
2^{92}	\$20 trillion	<i>GDP of the United States</i>
2^{99}	\$2 quadrillion	<i>all of human economic activity since 300,000 BC⁴</i>
2^{128}	<i>really a lot</i>	<i>a billion human civilizations' worth of effort</i>

单表代换密码

- 为增强移位密码的安全性，设计代换表时可以不仅仅是依次替换，而是允许任意替换，称为**单表代换密码**。
- 单表代换中每个明文字母可以映射到任意一个密文字母，密钥是 26 个字母的任意置换，共有 $26! > 4 \times 10^{26}$ 种可能密钥。



单表代换密码举例

例

使用代换表：

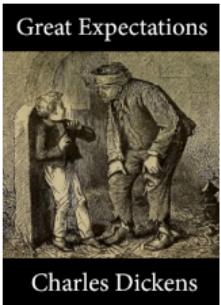
Plaintext: abcdefghijklmnopqrstuvwxyz
Ciphertext: DVQFIBJWPESCXHTMYAUOLRGZN

对消息 “if we wish to replace letters” 进行加密：

Plaintext: ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSUUUFYA

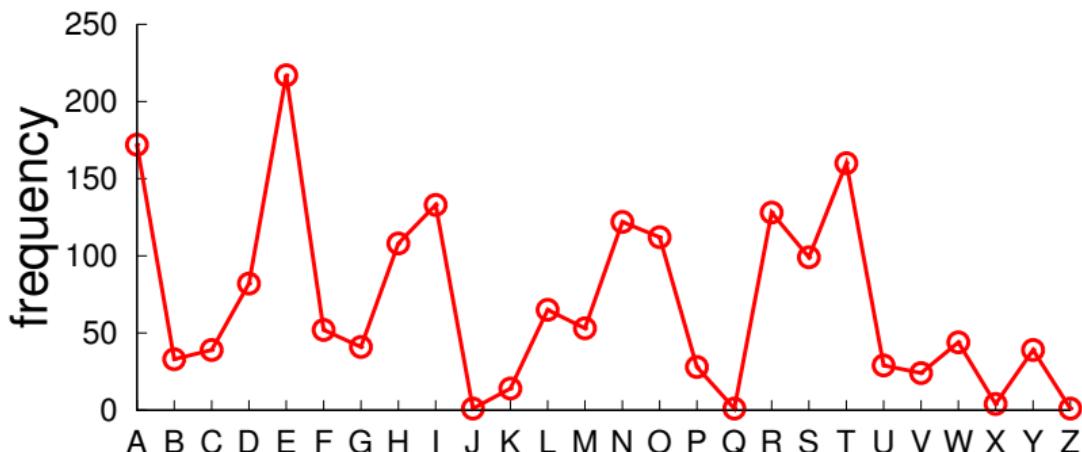
- 单表代换密码有 $26! > 4 \times 10^{26}$ 种可能密钥，似乎已经足够大了，单表代换密码是否真的安全？

《远大前程》: 查尔斯 · 狄更斯著长篇小说

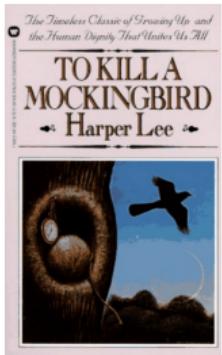


My father's family name being Pirrip, and my Christian name Philip, my infant tongue could make of both names nothing longer or more explicit than Pip. So, I called myself Pip, and came to be called Pip.

I give Pirrip as my father's family name, on the authority of his tombstone and my sister,—Mrs. Joe Gargery, who married the blacksmith. As I never saw my father or my mother, and never saw any likeness of either of them (for their days were long before the days of photographs), my first fancies regarding what they were like were unreasonably derived from their tombstones. The shape of the letters on my father's, gave me an odd idea that he was a square, stout, dark man, with curly black hair. From the character and turn of the inscription, "Also Georgiana Wife of the Above," I drew a childish conclusion that my mother was freckled and sickly. To five little stone lozenges, each about a foot and a half long, which were arranged in a neat row beside their grave, and were sacred to the memory of five little brothers of mine,—who gave up trying to get a living, exceedingly early in that universal struggle,—I am indebted for a belief I religiously entertained that they had all been born on their backs with their hands in their trousers-pockets, and had never taken them out in this state of existence. Ours was the marsh country, down by the river, within, as the river wound, twenty miles of the sea. wilderness beyond the churchyard, intersected with dikes and mounds and gates, with scattered cattle feeding on it, was the marshes; and that the low leaden line beyond was the river; and that the distant savage lair from which the wind was rushing was the sea; and that the small bundle of shivers growing afraid of it all and beginning to cry, was Pip.



《杀死一只知更鸟》：哈珀·李著长篇小说

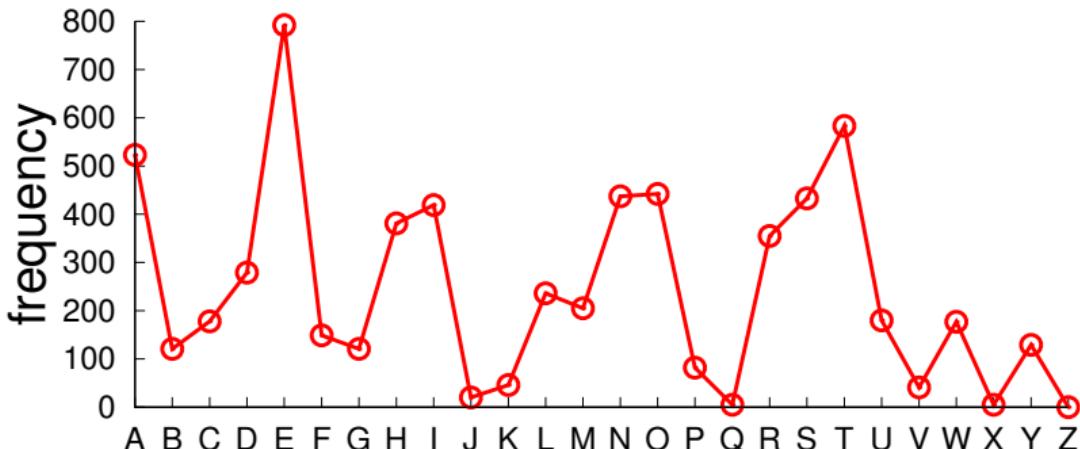


When he was nearly thirteen, my brother Jem got his arm badly broken at the elbow. When it healed, and Jem's fears of never being able to play football were assuaged, he was seldom self-conscious about his injury. His left arm was somewhat shorter than his right; when he stood or walked, the back of his hand was at right angles to his body, his thumb parallel to his thigh. He couldn't have cared less, so long as he could pass and punt.

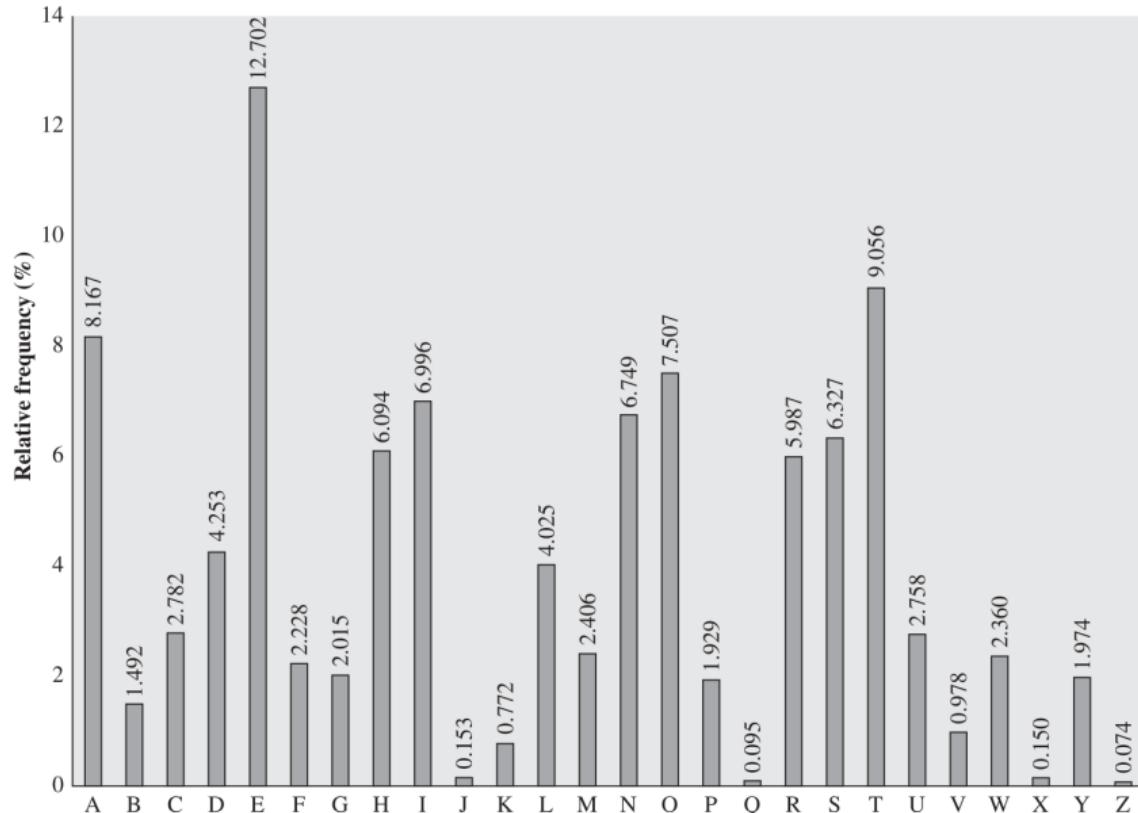
When enough years had gone by to enable us to look back on them, we sometimes discussed the events leading to his accident. I maintain that the Ewells started it all, but Jem, who was four years my senior, said it started long before that. He said it began the summer Dill came to us, when Dill first gave us the idea of making Boo Radley come out.

I said if he wanted to take a broad view of the thing, it really began with Andrew Jackson. If General Jackson hadn't run the Creeks up the creek, Simon Finch would never have paddled up the Alabama, and where would we be if he hadn't? We were far too old to settle an argument with a fist-fight, so we consulted Atticus. Our father said we were both right.

Maycomb was an old town, but it was a tired old town when I first knew it. In rainy weather the streets turned to red slop; grass grew on the sidewalks, the courthouse sagged in the square. Somehow, it was hotter then: a black dog suffered on a summer's day; bony mules hitched to Hoover carts flicked flies in the sweltering shade of the live oaks on the square. Men's stiff collars wilted by nine in the morning. Ladies bathed before noon, after their three-o'clock naps, and by nightfall were like soft teacakes with frostings of sweat and sweet talcum.



英文字母的相对使用频率



利用语言的统计特性进行密码分析

- 单表代换密码的密钥空间看似足够大，可以抵御穷举攻击，其实不然，这是因为语言往往具有统计特性。
- 人类的语言是有冗余性的，字母使用的频率并不一样：英文字母 E 是使用最频繁，然后是 T, R, N, I, O, A, S 等；有些字母使用得很少，如 Z, J, K, Q, X；双字母也有统计特性，例如 TH 等。
- 这样可以得到英文字母使用频率分布表，最早由阿拉伯科学家在公元九世纪发现。
- 单表代换不能掩盖字母出现的频率，只要统计密文中字母出现的频率，与已知的统计值做比较就可以分析出明密文字母的对应关系。

单表代换密码攻击举例

例

- 给定密文：

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSX
AIZVUEPHZHMDZSHZOWSFPAPPDTSPVQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- 统计相关字母出现的次数，可以猜测 P 和 Z 是 e 和 t，ZW 是 th，这样 ZWP 就是 the。
- 这样反复试验并不断修正错误，最后可得：

it was disclosed yesterday that several informal
but direct contacts have been made with political
representatives of the viet cong in moscow

Playfair 密码

- 两种常用方法用于减少明文结构在密文中的残留度：
 - 多字母代换密码：对明文中的多个字母一起加密；
 - 多表代换密码
- Playfair 密码是最著名的多字母代换密码，由英国科学家 Charles Wheatstone 在 1854 年发明的，以 Lyon Playfair 的名字命名。
- 首次应用于克里米亚战争（1854），最后一次应用是在一战。于 1915 年被德国破解，其变种被德国和英国应用于二战。
- Playfair 密码把明文中的双字母作为一个单元转换成密文的双字母。

Playfair 密码

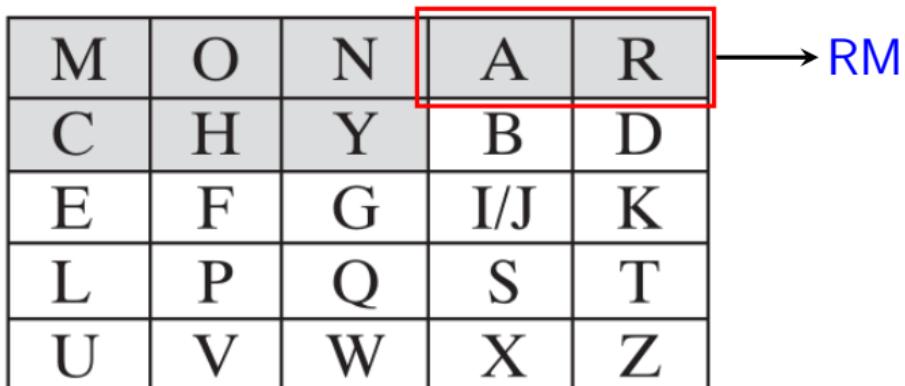
- Playfair 算法基于一个由密钥词构成的 5×5 密钥矩阵。
- 先在 5×5 密钥矩阵中填上密钥词，去掉重复字母。
- 再将剩余的字母按字母表的顺序从左至右、从上至下填在矩阵剩下的格子中，I 和 J 当作一个字母。

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

使用密钥词 MONARCHY

Playfair 密码的代换规则

- 将明文拆为字母对，如果字母对的两个字母是相同的，则在其中插入一个填充字母，如 ‘x’， “balloon” 变成 “ba lx lo on”。
- 对明文的每个字母对，利用密钥矩阵进行代换。
- 代换规则 1：**落在同一行的明文字母对中的字母由其右边的字母来代换，每行中最右的字母用该行最左边的第一个字母来代换，如 “ar” 加密成 “RM”。



A diagram illustrating the Playfair cipher key matrix. It is a 5x5 grid of letters. The letters are arranged as follows:

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

The letters A and R are highlighted with a red box. An arrow points from the highlighted row to the text "RM", indicating the result of the encryption for the pair "ar".

Playfair 密码的代换规则

- 代换规则 2：落在同一列的明文字母对中的字母由其下面的字母来代换，每列中最下面的一个字母用该列最上面的第一个字母来代换，如“mu”加密成“CM”。
- 代换规则 3：其他的每组明文字母对中的字母按如下方式代换：该字母所在行为密文所在行，另一字母所在列为密文所在列，如“hs”变换成“BP”，“ea”代换为“IM”或“JM”。

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

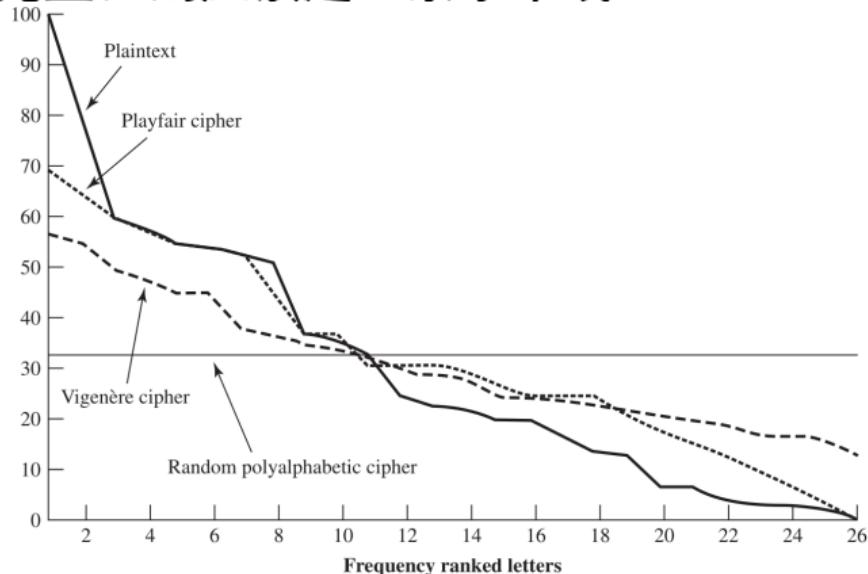
M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair 密码的安全性

- Playfair 密码的安全性比单字母的单表代换密码高。
- 因为有 26 个字母，所以共有 $26 \times 26 = 676$ 个字母对，对字母对进行判断要比对单个字母困难得多。
- 单字母的相对频率比字母对的相对频率有更好的统计规律，利用频率分析字母对就更困难，需要 676 维的频率表。
- Playfair 密码的密文仍然完好地保留了明文语言的大部分结构特征，它仍然是相对容易攻破的，几百个字母的密文就足够分析出规律了。

字母出现的相对频率

- “明文” 曲线画出 7 万个字母的频率分布，对文中出现的每个字母计数，结果除以字母 e 的出现次数，按降序排序。
- 加密后的曲线体现了加密后字母频率分布被掩盖的程度，如果完全被掩盖，则应该是一条水平线。



Hill 密码

- 1929 年美国数学家 Lester Hill 发明 Hill 密码。为每个字母指定一个数值，将 m 个连续明文替换成 m 个密文，由 m 个线性方程决定，例如 $m = 3$ 时

$$c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$$

$$c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$$

$$c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$$

- 用矩阵表示为

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \bmod 26$$

- 加密**: $C = E(K, P) = KP \bmod 26$
- 解密**: $P = D(K, C) = K^{-1}C \bmod 26 = P$

Hill 密码加密举例

例 (Hill 密码加密)

- 明文为 paymoremoney, 加密密钥为

$$K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

- 明文前三个字母用向量 $[15, 0, 24]^\top$ 表示, 则

$$C = K \cdot \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} \bmod 26 = \begin{bmatrix} 375 \\ 819 \\ 486 \end{bmatrix} \bmod 26 = \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} = \text{LNS}$$

- 照此转换剩下字母, 可得密文 LNSHDLEWMTRW。

Hill 密码解密举例

例 (Hill 密码解密)

- 解密需要用到矩阵 K 的逆 K^{-1}

- 由 $K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$ 可以得到 $K^{-1} = \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix}$

- $KK^{-1} = I$ 可以验证如下

$$\begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \cdot \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix} = \begin{bmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{bmatrix} \bmod 26 = I$$

多表代换密码

- **多表代换密码**：使用多个代换表对明文消息进行多重单表代换加密。
- 因为需要猜测更多的字母表，并且频率分布特性也变得平坦了，所以使得密码破译更加困难。
- 多表代换的特点：
 - 采用相关的单表代换规则；
 - 密钥决定给定变换的具体规则。
- 最简单的多表代换密码为**维吉尼亚密码**（Vigenère Cipher）。

Vigenère 密码

- Vigenère 密码的代换规则集由 26 个凯撒密码的代换表组成，每个代换表对明文字母移位 $0 \sim 25$ 次。
- 密钥词中的密钥字母用来代换明文字母 a ，故移位 3 次的凯撒密码由密钥字母 d 代表。
- 加密一条消息需要与消息一样长的密钥，通过重复密钥实现。
- 加密：给定密钥字母 x 和明文字母 y ，密文字母是位于 x 行和 y 列的那个字母。
- 解密：密钥字母决定行，行里密文字母所在列的顶部字母就是明文字母。

Vigenère 密码举例

例 (使用密钥词 deceptive)

key: **deceptive**deceptive

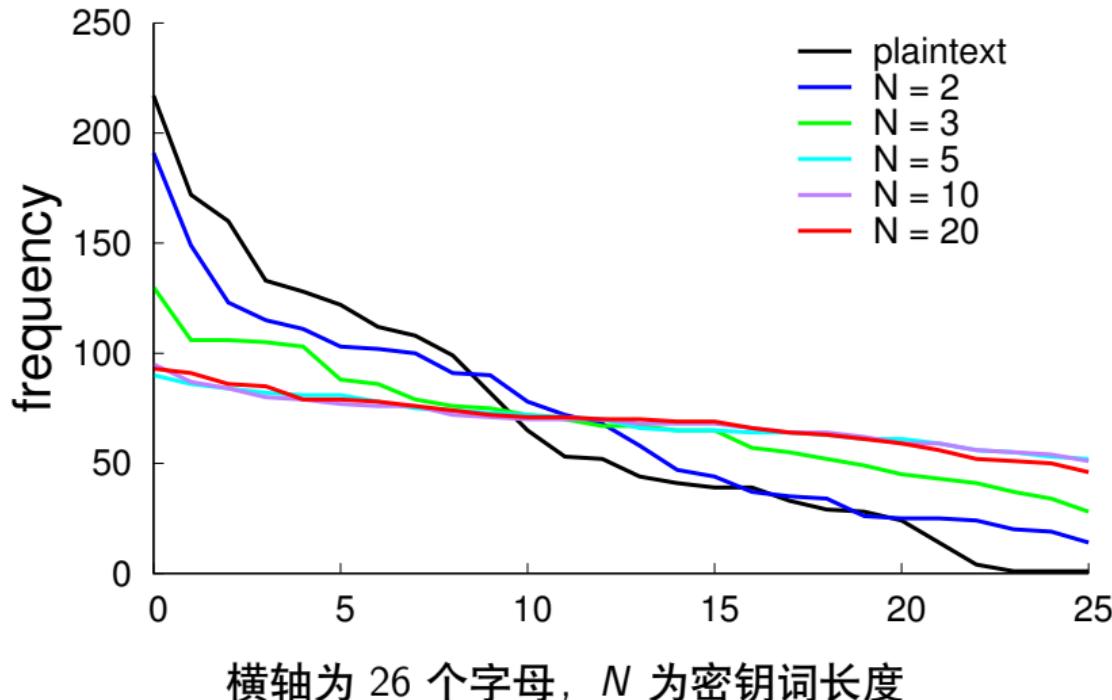
plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

		Plaintext																									
		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
Key	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	W	X	Y	Z	A	
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	V	W	W	X	Y	Z	A	
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	
l	L	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	
m	M	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	
n	N	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	
o	O	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	
p	P	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
q	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
r	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
s	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
t	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
u	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
v	V	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	
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	

Vigenère 密码的安全性

- 每一个明文字母可以有多个密文字字母对应，这样字母使用的频率特性减弱了，但是没有完全消失。



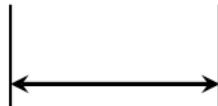
Vigenère 密码的安全性

- 破译的关键是判定密钥词的长度，可以通过发现重复序列来判断。

key: deceptivedeceptivedeceptive

plaintext: wearediscoverededsavemyself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ



相距整数个密钥词长度

最终措施是选择与明文毫无统计关系且和它一样长的密钥。

一次一密 (One-Time Pad)

- 一次一密 (One-Time Pad, 简称 OTP):
 - 每个消息使用与之等长且随机的密钥来加密。
 - 一个密钥只对一个消息加解密，之后弃之不用。
- OTP 满足理论安全，是不可攻破的。
- 1882 年 Frank Miller 首次描述了 OTP，之后又被其他人再次发明。Gilbert Vernam 在 1919 年申请了基于异或运算的 OTP 的专利。
- 在 1900 年以前的密码学研究中，只有两个想法对现代密码学有用，其中之一为柯克霍夫准则，另外一个则为一次一密。

一次一密 (One-Time Pad)

- OTP 运算基于二进制数据而非字母。

- **加密**:

$$c = m \oplus k$$

其中 \oplus 表示按位进行异或运算。

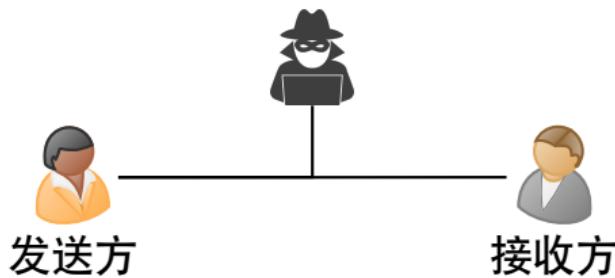
- **解密**:

$$m = c \oplus k$$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

一次一密的安全性

- 假设 Alice 和 Bob 使用 OTP 进行通信，从攻击者的视角观察到一个密文等同于得到下面算法的一个输出结果：



OTP($m \in \{0, 1\}^\lambda$):

$$k \leftarrow_R \{0, 1\}^\lambda$$

$$c = k \oplus m$$

return c

- 一个好的加密算法不应由密文得到关于明文的**任何信息**。
- 由于密钥 k 是每次随机产生的，因此 OTP 输出的密文在攻击者看来是**随机的**。

一次一密的安全性

- 例如给定两个不同输入 $m_1 = 010$ 和 $m_2 = 111$ ，输出分布为

Pr	k	$m_1 = 010$		$m_2 = 111$	
		$output$	$c = k \oplus 010$	$output$	$c = k \oplus 111$
$\frac{1}{8}$	000		010		111
$\frac{1}{8}$	001		011		110
$\frac{1}{8}$	010		000		101
$\frac{1}{8}$	011		001		100
$\frac{1}{8}$	100		110		011
$\frac{1}{8}$	101		111		010
$\frac{1}{8}$	110		100		001
$\frac{1}{8}$	111		101		000

- OTP 得到 $\{0, 1\}^3$ 中每种可能输出的概率都为 $1/8$ 。
- 密码分析者不能由密文区分 m_1 和 m_2 ，因此满足理论安全。

一次一密的安全性

定理 (OTP 满足理论安全)

OTP 对于 $\forall m \in \{0, 1\}^\lambda$ 产生的输出都是 $\{0, 1\}^\lambda$ 上的均匀分布。

证明.

给定 $m, c \in \{0, 1\}^\lambda$, 考虑 $\text{OTP}(m)$ 产生输出 c 的概率。注意到

$$c = k \oplus m \Leftrightarrow k = m \oplus c$$

因此

$$P(\text{OTP}(m) = c) = P(k = m \oplus c)$$

- 也就是说, 只有当 $k = m \oplus c$ 时, $\text{OTP}(m)$ 的输出才为 c 。
- 由于 k 是 $\{0, 1\}^\lambda$ 上的均匀分布, 因此这个概率是 $1/2^\lambda$ 。
- 所以对于所有 m 和 c , $\text{OTP}(m)$ 的输出为 c 的概率都是 $1/2^\lambda$, 即服从均匀分布。

一次一密的局限性

- 产生大规模随机密钥有实际困难。
- 密钥的分配和保护无法保证。

目录

1 课程简介

2 应用举例

3 发展历史

4 基本概念

5 古典密码

● 代换密码

● 置换密码

● 转轮密码机

置换密码 (Transposition Ciphers)

- 置换，亦称 transposition 或者 permutation。
- 置换密码通过改变明文字母的相对位置实现加密，并不替换明文字母，即明文内容形式不变。
- 通过重新安排明文字母的位置来隐藏明文内容信息，而不是用其他字母来代换明文字母。
- 这种方法是很容易破译的，因为密文拥有与明文一样的字母频率统计特性。

置换密码

- 一维变换：矩阵转置

C	A	N	Y
O	U	U	N
D	E	R	S
T	A	N	D

明文：can you understand
密文：CODTAUEANURNYNNSD

- 二维变换：图形转置

D			
T	A	N	
N	D	E	R
C	A	N	Y
	O	U	U

明文：can you understand
密文：DNSUARUTEODYNNAC

栅栏技术 (Rail Fence cipher)

按照对角线的顺序写出明文，按行的顺序读出作为密文。

例 (栅栏技术)

例如，加密 meet me after the toga party:

m	e	m	a	t	r	h	t	g	p	r	y
e	t	e	f	e	t	e	o	a	a	t	

可以得到密文 MEMATRHTGPRYETEFETEOAAT

行置换密码 (Row Transposition Ciphers)

- 一个更复杂的方案是把消息一行一行地写成矩形块，然后按列读出，但是把列的次序打乱，列的次序就是算法的密钥。
- 可以采用多步置换来得到相对较高的安全性。

例

Key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p

o s t p o n e

d u n t i l t

w o a m x y z

Ciphertext: TTNAAPMTSUOAODWCOIXKNLYPETZ

乘积密码 (Product Ciphers)

- 单纯的代换密码或者置换密码是不安全的，因为语言具有统计特性。
- 可以考虑连续使用若干个这样的密码使其难以破解，但是
 - 两次代换只能生成更复杂的代换，即多次代换等价于一次代换；
 - 两次置换只能生成更复杂的置换，即多次置换等价于一次置换。
- 如果在一次代换之后进行一次置换，可以生成一种新的更难破解的密码，这就是乘积密码。

💡 乘积密码是从古典密码通往现代密码的桥梁。

目录

1 课程简介

2 应用举例

3 发展历史

4 基本概念

5 古典密码

- 代换密码

- 置换密码

- 转轮密码机

转轮密码机 (Rotor Machines)

- 在现代密码系统出现之前，转轮密码机是最为广泛使用的多重加密器，尤其是在第二次世界大战中。
- 其中德国 Enigma 密码机是密码学界划时代的丰碑。
- 1918 年，德国发明家亚瑟·谢尔比乌斯 (Arthur Scherbius) 发明了一种能够自动编码的机器。谢尔比乌斯给自己所发明的电气编码机械取名“Enigma”，意为“哑谜”。
- Enigma 是一种用于加密与解密文件的密码机。确切地说，Enigma 是一系列相似的转子机械的统称，包括了一系列不同的型号。
- 不同型号的 Enigma 密码机大都包含四个共同部分：键盘、转子、显示器和接线板。

Enigma 密码机

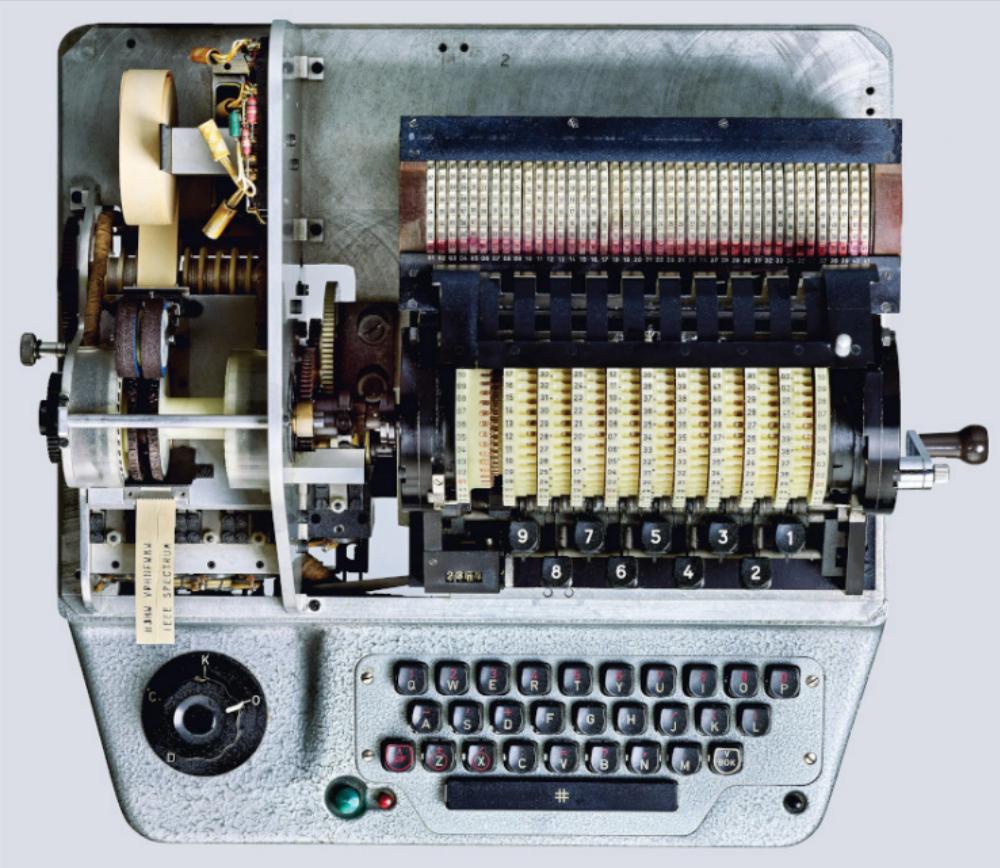


亚瑟·谢尔比乌斯



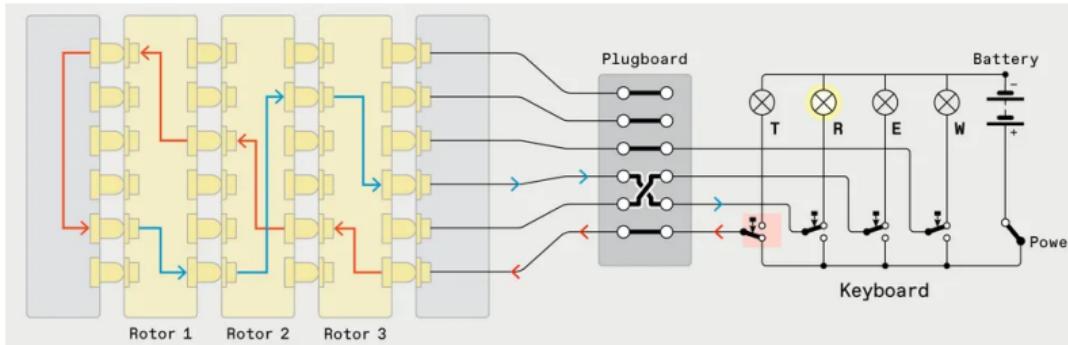
Enigma 密码机

Enigma 密码机其他型号 (HX-63)



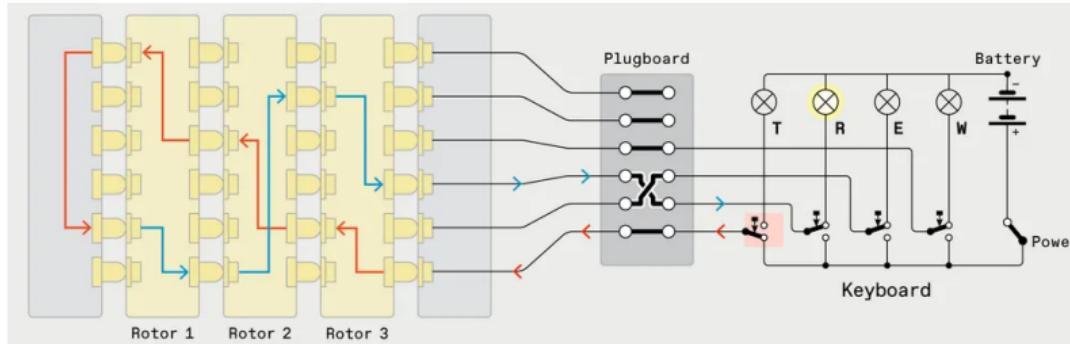
Enigma 密码机原理

- 转轮密码机实现了一个非常复杂、变化多端的多表代换密码。
 - 每个转轮有 26 个输入和 26 个输出，每个输入仅与一个输出相连，一个转轮就定义了一个单表代换。
 - 每按下一个键，转轮旋转一个位置，内部连线相应改变，改变下一次代替。
 - 一个转轮转 26 个位置后回到初始状态，因此可以形成 26 种单表代换。



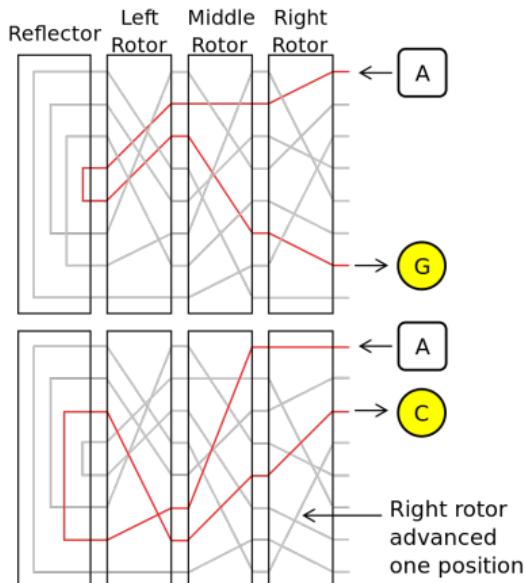
Enigma 密码机原理

- 为使代换更复杂，可以把多个转轮连接起来。Enigma 密码机有 3 个转轮，从提供的 5 个转轮中随机选择。
- 三个转轮以不同的速度移动，3 个转轮的机器的周期是 26^3 。
- 为进一步阻止密码分析，有些转轮机在每个转轮上还有不同的起始位置号。
- 尾部有一块连接板，可以连接 6 ~ 10 个插板，用来改变字母对之间的映射关系。



Enigma 密码机原理

- 最左边的转轮与反射板相接触，反射板使最左边转轮的不同端子随机相连，作用是使电路闭合。
- 由于反射板的存在，一个明文字母不会映射为它自己。

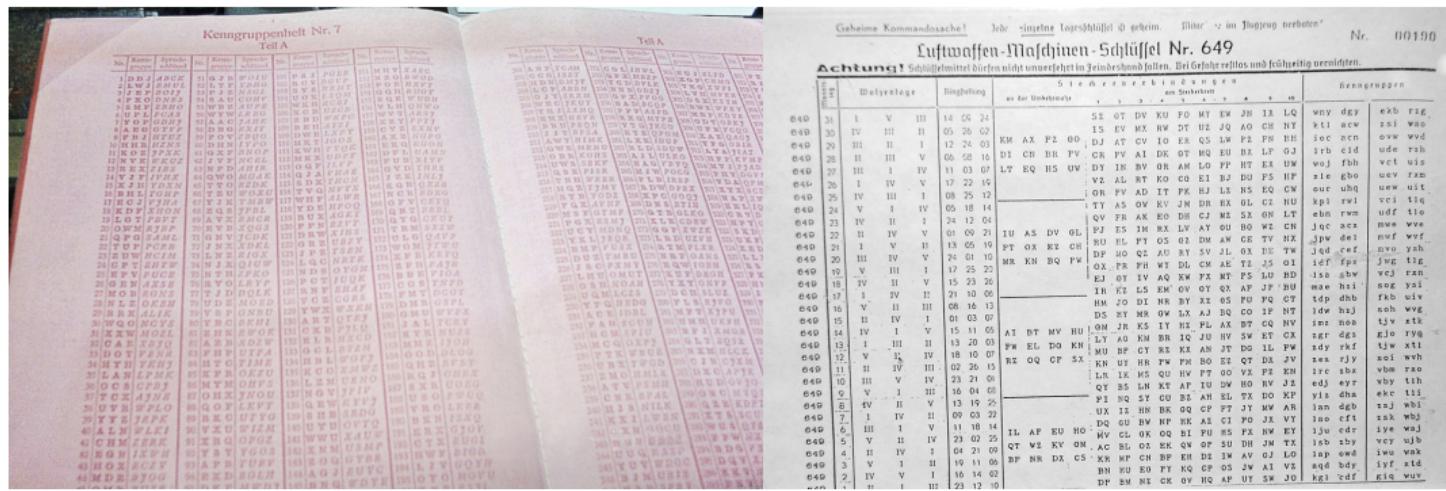


Enigma 密码机的使用

- 使用时，发信人首先调节三个转子的初始位置，转子的初始位置就是密钥，是收发双方预先约定好的。
- 然后键入明文，并把灯泡闪亮的字母依次记下来，最后把记录下的字母按顺序用电报发送出去。
- 收信方收到电文后，也使用一台 Enigma，按照原来的约定，把转子的位置调整到和发信方相同的初始位置上，然后依次键入收到的密文，灯泡闪亮的字母就是明文。
- 使用 Enigma 密码机解密和加密的过程完全一样，这就是反射板的作用，同时反射板的一个副作用就是一个字母永远也不会被加密成它自己，因为反射板中一个字母总是被连接到另一个不同的字母。

Enigma 密码机的设置

- Enigma 密码机每天都需要根据一份收发方共享的设置单进行设置，例如转轮选择与排列、转轮位置、连接板字母对等。



Enigma 密码机设置表

Enigma 密码机的设置

Luftwaffen-Maschinen-Schlüssel Nr. 649

Achtung! Schlüsselmitteil dürfen nicht unversehrt in Feindeshand fallen. Bei Gefahr restlos und frühzeitig vernichten.

Wortstellung	Wellenlage		Ringstellung	Steckerverbindungen an der Umkehrrolle										Kenngruppen					
	1	2		3	4	5	6	7	8	9	10	wny	dgy	ekb	rzg				
31	I	V	III	14	09	24		SZ	GT	DV	KU	FO	MY	EW	JN	IX	LQ		
30	IV	III	II	05	26	02		IS	EV	MX	RW	DT	UZ	JQ	AO	CH	NY		
29	III	II	I	12	24	03	KM	AX	PZ	GO	DJ	AT	CV	IO	ER	QS	LW		
28	II	III	V	06	58	16	DI	CN	BR	PV	CR	FV	AI	DK	OT	MQ	EU		
27	III	I	IV	11	03	07	LT	EQ	HS	UW	DY	IN	BV	OR	AM	LO	PP		
26	I	IV	V	17	22	19					VZ	AL	RT	KO	CG	EI	BJ	DU	
25	IV	III	I	08	25	12					OR	PV	AD	IT	PK	HJ	LZ	NS	

- 三个转子的选择与排列组合: $5 \times 4 \times 3 = 60$
- 三个转子的初始位置: $26 \times 26 \times 26 = 17576$
- 连接板字母对设置 (假如有 6 对):

$$C_{26}^2 \cdot C_{24}^2 \cdot C_{22}^2 \cdot C_{20}^2 \cdot C_{18}^2 \cdot C_{16}^2 \cdot \frac{1}{6!} = 100391791500$$

- 一共约为 10^{16} 种可能配置, 即一亿亿种可能!

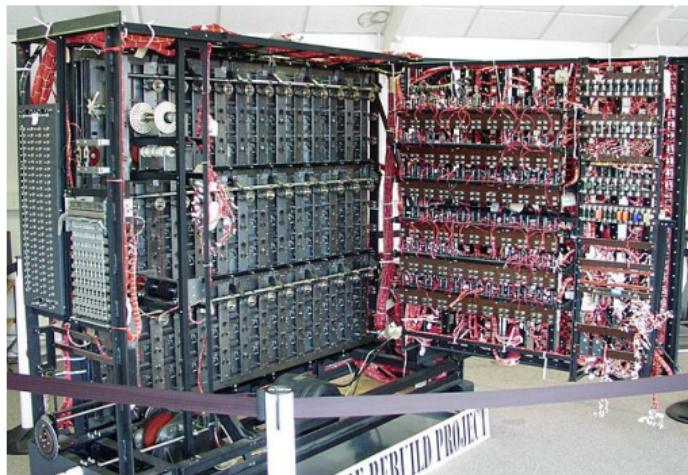
Enigma 密码机的破译



艾伦·图灵



布莱切利
公园



“炸弹”破
译机

小结

- 1 课程简介
- 2 应用举例
- 3 发展历史
- 4 基本概念
- 5 古典密码