

密码学介绍

An Introduction to Cryptography
中文第二版

[美] Jon Callas

杨新 编译

声明

发布信息

密码学介绍

版权信息

2009 年中文翻译版权归杨新所有。个人可以免费参阅，如果将本书用于任何商业用途，请为翻译版本的版权所有人支付版权费！本书中文参考内容的版权人也会从中获益！

许可证和专利信息

IDEA 算法是 Ascom Tech AG 专利（美国专利号 5,214,703）。CAST 算法是美国北方电信（Northern Telecom, Ltd）的专利。PGP 公司获得加利福尼亚大学专利许可（10/655,563）包括从通用区块加密算法（Conventional Block Cipher）由分组密码操作方法构造广义区块加密算法（Wide-blocksize block Cipher）的技术。PGP 公司拥有包含此内容的软件和包含专利的文档，或正在申请的专利。提供的软件和文档并没有给予任何使用专利的权力。

商标

PGP、PGP 标志、Pretty Good Privacy 和 Pretty Good 英文标志都是 PGP 公司注册商标。其它注册的和未注册的商标的所有权归原商标所有人。

许可

PGP 软件压缩代码拥有 Mark Adler 和 Jean-Loup Gailly 的免费 Zip 算法的使用许可

限制

本文如有任何主题内容的更改，恕不另行通知。

出口信息

PGP 出口可能是工业安全局时时刻刻所争论法律法规的主题，美国商贸部限制技术和商品出口

关于 PGP 公司

PGP 公司是一家世界级的安全软件公司。它是数据和电子邮件加密界的领军代表。产品基于统一密钥管理和策略结构。PGP 加密平台也提高了全套企业级加密方案。PGP 同时也为电子邮件、笔记本、台式电脑、实时加密、PDA、网络存储、FTP 服务、批量数据传送和备份的个性安全定制。PGP 方案已经被超过 3 万家企业、公司和世界范围内的很多政府采纳。其中包含 Fortune(R)（财富杂志）100 强公司中的 84% 和 Fortune(R) 全球 100 强企业中的 66%。同时 PGP 公司的创新、标准、解决方案在全球屡获殊荣。PGP 的解决方案可以帮助客户保护机密信息、客户数据、协助管理、数据验证和保护公司商标和声誉。联系 PGP 公司登录 <http://www.pgp.com>，电话：+1 650 319 9000

授权

本书的翻译作者授权读者可以免费下载、阅读、打印本书。同时禁止任何网站以字符页面形式转载本书。对于任何翻译的疏漏或错误，如果造成了你的损失，翻译者不承担任何责任。如果使用本文为任何商业目的（包含网站转载字符的行为在内的），您必须为翻译作者支付书籍的版权费。

联系翻译作者 电子邮件：loveship2002@gmail.com，MSN：loveship2002@hotmail.com。

序

本书翻译原文来自美国 PGP 安全软件公司的产品 PGP Desktop 9.9.0 软件包中的 Jon Callas 在 2006 年所著《An Introduction to Cryptography (Intro To Crypto.pdf(CRC-32:79EE7FEF))》，原文的意旨在于使所有普通人都可以看懂这本关于密码学的书籍，经过翻译和改编，所以本书的文字简单，适合初学者阅读。内容从密码学的历史到密码学在现实生活中的应用，而且书中没有深入讨论任何专业性的问题，也不会讨论算法的细节，否则不少人会看不懂，最多的就是一个名字、一个概念，甚至是一个比喻，也可能是一个简单的数学知识。有兴趣的人可以从你感兴趣的方面深入的了解密码学。原作者是处在美国的法律环境，美国的法律更新和加密技术发展一样快，文章内众多法律名称也为我们了解美国关于密码技术的法律做了介绍。

Cryptography 在英语中是密码术、密码学的意思，外国人认为的“密码”一词是个动词，意味加密，也就是加密数据，他们强调的是这个过程，中国人称的“密码”是开启秘密的那个口令码，它是使用密钥文件的钥匙。所以，外国人用词更加精确。其实“密码学”原义是包含数据加密和数据解密 2 个过程，而从中文字面上看就误解成了研究破解密码的一种学问，认为研究密码学的都是在破译密码，不完全对。破解密码的学问叫做：密码分析学！

注释中有大量本人的“非官方”说明，遇到不懂的请参阅注释！注释中还针对中国读者添加了一些中国的实例，推荐遇到有注释的环节都仔细看看，表达的意思可能就是另外一种。在前文出现的一些词汇你可能不是很明白，而在后面会有具体的介绍。原版中的尾注很不方便，所以我把尾注全部改为当页下方的脚注，读者可以很快找到当页里面自己想要的东西！注释的链接有不少英文内容的，所以我建议大家努力学习英语！推荐阅读本书的人具有简单的计算机知识和高等数学的底子，其实用到的地方也不是很多，你不会这些也不必担心，下方的脚注仔细一看就完全可以明白。没有翻译外国人的名字，名字只是字符代号而已，这样可以使你更容易的在互联网上搜索关于他们的故事。

本书的内容除了来自 Jon Callas 的《An Introduction to Cryptography》，有一小部分是来自 Bruce Schneier 的《应用密码学》，这是因为 Jon Callas 写的过于简陋，入门都谈不上，高度刚刚碰到门槛，不小心可能会绊着，所以我稍微提升了一点高度。

附录部分并不是原书的内容，本书的英文书出自 PGP 官方人员的手，在一些细节就会有特别的广告嫌疑。为了给读者一个更加宽阔的软件选择视野，我挑选了一些我认为比较可信的软件。有些文字几乎是复制粘贴上去的。因为实在找不出更好语句来阐述它们了。我想应该不会有人在免费打广告的前提下问我要稿费。

如果有人想尝试重新翻译，我很赞成，这里给想尝试的人的一些建议：在翻译这类书前，并不是只要有英文的功底。中国话——隔行如隔山，语言类的知识是无法解决这个问题的，推荐去图书馆参阅大量数学部分的知识，且包括密码学书籍的部分，并且要了解基本的计算机知识。我并非 PGP 官方技术人员，也不是英文翻译专业出生。翻译这本书纯粹是对英文和密码学的兴趣。尽管我做了很细心的校对，书中的错误是难免的，也可能出现很多翻译的错误。在这本“非官方”译本中有许多不足的地方希望 PGP 用户、广大热心网友、数学高人、密码学研究者指出。同时感谢你们对这本书的支持。

杨新
2010.1

翻译者致谢

感谢在书籍翻译过程中为我提供帮助和支持的一大群网友朋友！其中包括 PGP 的 QQ 交流群的“砖家”们。感谢老鼠妹妹找出了很多细致的错误。感谢小许子指出的很多语句不通顺的地方。感谢黎慈军收集的邮件加密软件的名单。感谢肥肥 208 的非官方技术支持。感谢心铃茹雪、老猫在 PDF 文档的加密和书签方面的指导。感谢本书的所有阅读者。

发布版本:

中文第一版正式版	2009.4.9 (精简版 CRC-32:4A1CA0AC, PDF/A-1a CRC-32:2F53EF59)
中文第二版正式版	2010.1.28

中文版第二版相对于中文版第一版的变化:

1. 更改排版使用的结构。
2. 添加大量的密码学历史的细节以及相关知识。
3. 为了简化发布工作, 只发布 PDF/A-1a 标准^a的 PDF, 可以在计算机和智能手机等^b其他嵌入设备上打开阅读。

^a ISO 19005-1 等级 A 规范 (PDF/A-1a) 基于等级 B, 但添加了“标签 PDF”中的一些关键属性: 它要求结构信息和可靠的文本语义, 以便保存文档的逻辑结构和自然阅读顺序。简而言之, PDF/A-1a 不仅确保文档在未来使用时保持外观的一致, 还确保其内容(语义)能够得到可靠的解释, 并可以被有功能障碍的人士读取。

^b从 Adobe Reader 7 开始支持正式发布的 PDF/A 标准, Foxit Reader 也已经完全支持这个标准。另外在智能手机设备中, 经过测试 Adobe Reader LE 2.5 和 mBrain PDF+ 1.75 (05) 可以完整显示所有字符。Adobe Reader LE 1.5 is not support this document!

目录

1	关于本书.....	- 1 -
1.1	什么样的人适合阅读这本书.....	- 1 -
1.2	万丈高楼平地起	- 1 -
1.3	密码学很难——但是它使梦想变的简单.....	- 1 -
1.4	说难不难, 说易不易	- 2 -
1.5	究竟什么是密码学?	- 3 -
1.5.1	隐写术	- 3 -
1.6	这本书的历史.....	- 4 -
1.7	原书作者特别致谢.....	- 5 -
2	为什么密码学那么重要?	- 6 -
2.1	走进缺口: 骇人的事件.....	- 6 -
2.1.1	笔记本丢失事件	- 6 -
2.1.2	无安全保护的网路资源.....	- 8 -
2.1.3	个人身份信息丢失.....	- 9 -
2.2	法律法规.....	- 9 -
2.2.1	含保密项的法律法规	- 9 -
2.2.2	复合法律法规.....	- 10 -
2.2.3	违规警告的法律法规	- 11 -
2.3	限制加密技术的法律法规	- 11 -
3	密码技术的不完全历史.....	- 13 -
3.1	人工密码技术.....	- 13 -
3.1.1	代替密码和换位密码	- 14 -
3.1.1.1	代替密码.....	- 14 -
3.1.1.2	换位密码.....	- 15 -
3.2	机械密码技术.....	- 19 -
3.2.1	转轮机	- 19 -
3.3	计算机密码技术	- 21 -
3.3.1	公钥密码技术.....	- 21 -
3.3.2	加密技术标准的提升	- 22 -
3.3.3	AES 标准	- 24 -
3.3.4	密码界的战争.....	- 25 -
4	加密技术的基础.....	- 28 -
4.1	基本部件.....	- 28 -
4.1.1	参量和变量	- 28 -
4.1.2	随机数字.....	- 29 -
4.1.2.1	伪随机序列	- 29 -
4.1.2.2	真正的随机序列	- 30 -
4.1.2.3	真随机数与伪随机数更加直观的区别.....	- 31 -
4.1.3	密钥.....	- 33 -
4.1.4	算法.....	- 33 -
4.1.4.1	区块大小.....	- 34 -
4.1.4.2	公钥加密算法的根本	- 35 -

4.1.4.3	因子分解法则.....	- 35 -
4.1.4.4	对数法则.....	- 36 -
4.1.4.5	密钥的长度	- 37 -
4.1.4.6	密钥需要多长?	- 38 -
4.1.4.7	不可破译的算法强度有多大?	- 39 -
4.1.4.8	真正不可破解的加密: One-Time Pads	- 42 -
4.1.4.9	One-Time Pads 的诱惑	- 44 -
4.1.5	Hash 算法.....	- 46 -
4.1.5.1	通用 Hash 算法.....	- 47 -
4.1.5.2	Hash 算法难度.....	- 47 -
4.1.6	数据完整性算法: 信息鉴别码和数字签名.....	- 49 -
4.1.7	证书机制.....	- 50 -
4.1.7.1	为什么使用证书机制?	- 50 -
4.1.8	信任和权限	- 51 -
4.1.8.1	直接信任.....	- 51 -
4.1.8.2	分级信任.....	- 51 -
4.1.8.3	积累信任.....	- 52 -
4.1.8.4	混合信任模式.....	- 52 -
4.1.9	证书区别.....	- 53 -
4.1.9.1	认证和证书	- 53 -
4.1.10	融合——从明文生成密文	- 53 -
4.1.11	分离——从密文还原明文	- 54 -
4.1.12	芝麻开门的技术含量	- 55 -
4.1.13	生物识别.....	- 59 -
4.1.14	从这里继续	- 63 -
5	未来的密码技术	- 64 -
5.1	名词到形容词, 语法到语义学	- 64 -
5.1.1	社会期望.....	- 64 -
5.2	数字签名与语义学.....	- 65 -
5.2.1	数字签名并非签名.....	- 65 -
5.2.2	认可的神话	- 67 -
5.2.2.1	高强度算法的反驳.....	- 68 -
5.2.3	签名与责任	- 68 -
5.2.4	现实的语义改变	- 69 -
5.3	加密技术可靠性	- 70 -
5.3.1	硬件加密与软件加密	- 70 -
5.3.1.1	硬件加密.....	- 70 -
5.3.1.2	软件加密.....	- 71 -
5.4	硬件升级.....	- 71 -
5.5	权限管理.....	- 72 -
5.6	数据销毁办法.....	- 73 -
5.7	保密增强技术.....	- 75 -
5.8	细微的改变	- 76 -
5.8.1	新型 Hash 算法.....	- 76 -

5.8.2	新型算法.....	- 76 -
5.8.2.1	结合加密和验证的算法.....	- 76 -
5.8.2.2	新算法和重新设计的算法.....	- 77 -
5.8.2.3	椭圆曲线算法.....	- 77 -
5.8.2.4	双线性映射算法.....	- 77 -
5.8.3	量子密码学.....	- 78 -
5.9	什么能够改变路线?	- 79 -
5.9.1	专利影响.....	- 79 -
5.9.2	科学虚构的技术.....	- 79 -
5.9.3	法律改变.....	- 80 -
附录	- 81 -
6	技术软件的介绍.....	- 81 -
6.1	加密压缩软件.....	- 81 -
6.1.1	WinZip.....	- 82 -
6.1.2	WinRAR	- 83 -
6.1.3	7Zip.....	- 84 -
6.1.4	UHARC	- 85 -
6.2	邮件加密软件.....	- 85 -
6.2.1	The Bat.....	- 86 -
6.2.2	Foxmail.....	- 86 -
6.3	数据安全删除软件.....	- 87 -
6.3.1	O&O Soft SafeErase.....	- 88 -
6.3.2	East-Tec DisposeSecure.....	- 88 -
6.3.3	Linux 类系统下的工具.....	- 89 -
6.4	系统级别加密软件.....	- 95 -
6.4.1	MicroSoft EFS.....	- 95 -
6.4.2	MicroSoft BitLocker	- 96 -
6.4.3	PGP.....	- 97 -
6.4.4	TrueCrypt.....	- 97 -
6.4.5	Utimaco SafeGuard	- 98 -
6.4.6	The GNU Privacy Guard	- 98 -
7	关于 PGP 的开创者.....	- 100 -
7.1	背景.....	- 100 -
7.2	PGP 的起源	- 100 -
后记	- 102 -

1 关于本书

真正的秘密是被保护在多变的结构之下的

——美国首席法官 John Roberts

1.1 什么样的人适合阅读这本书

一些喜欢密码学并对密码学感兴趣的初学者，他们想搞懂密码学到底是什么。本书有对密码学的内容最基本的解释、技术和专业术语，这些都基于基础密码学和 PGP 软件的一些特殊地方，如果你想了解密码学，这是一个好的开始，并且这本书中有大量链接到有用互联网站的超链接，那里有许多令你感兴趣的资源，如果你阅读的是电子 PDF 格式的文件，你可以直接点击这些链接跳转到参考资料。

1.2 万丈高楼平地起

密码学是信息技术的一个重要的组成部分。我们工作使用的系统都是信息化的时，它就可以作为我们要改造的物理实物。假设我寄给你一封信，我会把信纸放入一个信封内而避免被别人看到，我甚至可以在底部开口处签名，并且你可以从邮戳上得知这封信是从什么地方寄来的。在商业活动中我们彼此交换名片，或者我会给你出示客户卡片来证明我们的关系，其实就是这个客户卡片的意义。我们现在在互联网的所作的一切，诸如我们单独聊天时，或者其它我们之间可以说明的关系的活动^a，都使用了加密技术，加密技术给我们带来的不仅仅是将钢筋混凝土的世界虚拟化，加密技术还可以保证远距离通信中信息的安全性、秘密性、可靠性。这和我们用 0 和 1 所构造的计算机世界是一样的。

1.3 密码学很难——但是它使梦想变的简单

密码学难是有很多原因的。因为你要去了解很多基础知识。你小时候可能偷偷写过“小纸条”，你可能看到过报纸上的猜字游戏^b，但有时你自己可能看不出来，那么最简单的密码术你已经知道了。然而，密码学在近 30 年来的变化要比 3000 年来的变化大的多的多。这是因为现代密码学和计算机产生了巨大的联系。它最大的意义就是让你无法一眼识破秘密，它就像爱丽丝在仙境^c中一样。公钥加密技术——以 PGP 为代表的现代密码学技术——是与常识相反的^d。密码学不需要你学习你已经知道的一些事情，或者换句简单的话说，你要学会如何去解释你已经知道的一些新颖、巧妙想法的过程。

^a 活动：指计算机通讯诸如下载、上传、远程连接、浏览等点对点的计算机连接活动。

^b 密文：由明文经过加密算法加密后生成的东西，可以是文字、图案等，按照算法还原回去的就是明文。

^c 爱丽丝漫游仙境：一篇童话故事，主人公和他的朋友从一个兔子洞进到仙境。仙境里面有很多不可思议的东西。

^d 与常识相反：一般的常识是算出来的密文的算法都是按照自己设定的规则来实现的，这样别人不容易猜测破解，甚至还原为明文。这个算法为私有算法。公钥密码学的算法是公开的算法，不一样人计算有不一样的结果，不容易猜测。它的过程经得起数学的理论攻击，就像是一个多项式的分式的极限，我们永远无法一个一个带进去算出来这个结果，因为无穷无尽，但是数学中的 \lim （极限）算法就可以算出来极限。一般的常识是一个钥匙开一把锁，现在这把锁上锁需要的钥匙和开锁的钥匙不是相同的一把，只有拿到开锁的钥匙才可以打开锁。

密码学的困难还在与我们要解决的问题确实很难。我们要达到的目的是我们构建密码学系统时如何去叙述它和理解它。我们无论如何都很难去建造、很难解释技术构造，同样也很难理解。好消息是尽管是这样我们并不是孤独无助的，最好的加密技术通常是那些经常出错误的办法，因为经常出错，别人无法琢磨，不变的是我们必须回到原来的画板前去仔细审视我们自己的作品。这也就意味着在拥有大量加密技术经验的人前，你是微不足道的。听起来很夸张，不过它确实如此！

当我在写这本书的时候，我也在阅读使用了一些用了新加密技术^a的电子邮件，很容易解释清楚它的原理：我想给你发一个邮件的同时不希望在传输过程中被改变，这个电子邮件的发送过程是机器完成的，这个过程变成过程调用就是：

“你还没有验证”

“是的，你已经完成了”

“不，你没有完成”

“我发送一个验证码到邮件发送清单”

“不，它不是有效的验证码”

“是的，它是有效的验证码”

“不，它不是，你被欺骗了”

“它也是，我将不会再被调用了”

“是的，你可以继续坚持，你给我展示的是你没有证据证明你有权限，你是个傻子”

密码术很难，以至于它减少了人们对数学教授像兔八哥和古怪鸭一样的搞笑想象^b！

那就是为什么加密技术同样也是简单的，它太难了是因为别人有比你多的多的经验，但是他们也不敢确信他们的秘密不会有人破解。加密技术很简单是因为在你拿到之前不需要自己再次解释。世界上最好的密码是犯了很多错误的算法。它难就难在这个空间是在非常规的想法下构造的，因为这些事情是在变化之中的，而且是在争论之下的。这个正在进行的争论不仅仅是一个令人兴奋的技术学科，而且是一个令人愉悦的体育项目（因为争论也是需要力气的！）。

1.4 说难不难，说易不易

在了解加密技术的过程中，我们了解一些经常使用的而且非常重要的技术词汇。然而，不幸的是这些词汇没有严格的定义和形式。举一个例子：我们要讨论是一个困难问题或一个完美系统算法。尽管没有严格的数学定义和形式。一个难问题就是没有比猜很好的办法来解决的问题。一个完美系统就是在一个难问题的基础之上的系统算法^c！这可以让别人只能用猜的办法。

通俗点就是坚固的加密算法它的实用价值高，这也就是我们要用它来作为基础算法的原因。我认为分开讨论算法的实用性和强度才能体现出它的价值！

一个例子：假设我们要讨论一个完美加密的系统算法，它是在真实的理论数学的基础上建立的，它只有 3 个可以猜测的数据。你完全可以猜测所有数据并且验证它，可以很快完成。这是个保护这个秘密的简单而愚昧、并且很不实际的方法。但是，如果上面这个问题中的可以猜测的数据量的范围从 3 个一下子变成了 340,282,366,920,938,463,374,607,431,768,211,456 个（几乎是 2 的 128 次方），这时候这个算法就变得很实

^a 系统：文章中的系统一词是指密码算法的一系列过程，比如理论基础，实现过程等，就是关于这个密码算法的一切信息。

^b 解释：加密学的算法有些想出来的是很巧妙的，很难想象那么多古怪老头坐在房子里天天挖空心思想象出什么奇怪算法来刁难大众。作者使用了美国式的幽默。换句中国话说就是：你无法想象数学教授像刘老根一样会忽悠人，所有人被忽悠了，他的加密技术就高深了！当然数学界也有范伟一样的人。总是被愚弄，然而有一天，他开窍了.....

^c 解释：这个对于难问题的所有理论基础和探索过程。别人无法有其他的办法来解决这个困难的问题，除了猜。

用了,大家花费在验证上的时间会变的很长,甚至超过人的生命周期!^a

1.5 究竟什么是密码学?

我们已经懂了一些密码学的基础知识,我们来看一些新定义,密码技术(Cryptography)是一个密写的科学和技术,这个词汇首先来自希腊的“隐型书写(hidden writing)”,在那时候我们就已经在使用密码学了。原始的数据称之为明文(plaintext),使他变的不可读的过程称之为加密,我们把明文变成密文(ciphertext),并且又变回来的东西叫做算法(cipher),算法通常使用有一些秘密的东西作为算法的重要部分,这个秘密就是密钥(key),明文变成密文的过程叫加密(encrypting),反之是解密(decrypting)。我们通常习惯这样表示^b:

加密: 加密算法(明文, 密钥) = 密文
解密: 解密算法(密文, 密钥) = 明文

有的算法在加密技术里面叫编码(codes),编码仅仅是一个在符号(数字或字母)和信、文字等之间有关系的表,比如在计算机内用编码产生字符,比如英文字母A的就是阿拉伯数字49来代表的,这个编码称之为Unicode^c,这个只是个名字。在电报机的时代,把报文变成文字要依靠电报密码本(Codebooks)^d。密码和算法通常是在一起使用,在计算机普及化的今天也是这样的,因为所有加密信息都要重新编码。算法不同于编码的原因是算法有一个关键叫密码的东西用来加密和解密文件,而编码表是现成的,一切按照密码表的安排。如果一个编码表是秘密的,即不公开的,那么它就成了加密技术的另外一种形式,一种密写的形式。

加密技术学有一个姊妹学科密码分析学(cryptanalysis),它是用来破解密码的科学和技术。加密技术学(cryptography)和密码分析学一起才是真正的密码学(cryptology),他们都是我们通俗的说密码学,也就是加密学,我也常常对这个词感到疑惑。

1.5.1 隐写术

隐写术(steganography)^e来自希腊的“隐蔽书写(covered writing)”。加密学的字面意思并不是隐藏的书写,它并没有隐藏,只是读不懂,你可以看到加密后的文字,隐写术确实也是加密学。

隐写术其实是将秘密的信息隐藏在其它信息中,这样,真正存在的秘密就被隐藏了,而且不容易看出来。通常发送者写一篇普通的信息,然后在同一张纸上加上隐藏的秘密信息。历史上的隐写方式有隐形墨水(Invisible inks)^f。印记法(hollow heels in shoes)用小针在选择的字符上刺小的针眼,在手写的字符之间留下细微差别。在打印字符上用铅笔作记号、除了几个字符外,大部分的字符用格子盖起来等等。甚至还可能隐蔽在一幅画或一首乐曲中,而一般看不到。

^a 暴力破解法: 通过猜测密码来试破解文件, 如果密码很长、很复杂, 将花费很长时间! 密码的强度越高, 暴力破解的时间越长。

^b 这是数学的写法类似 $F(x,y)=G$, F 和 G 就是一个名字而已, 括号里面有 2 个字母表示这个 F 函数需要 2 个输入参数, x 和 y 是需要输入的 2 个数据的代号。

^c Unicode: 是一个电脑中的编码, 把人类所有的语言文字用数字进行了编码, 目前的计算机类设备已经都支持。访问: <http://www.unicode.org>

^d 电报密码本: 发送的电报码转换为数字, 再从电报密码本查到数字对应的文字。访问: <http://www.dtc.umn.edu/~reedsj/codebooks.html>

^e 隐写术: Neil F. Johnson 和 Sushil Jajodia, 《Steganography: Seeing the Unseen (隐写术: 看不见的技术)》, IEEE Computer, 1998 年 2 月, 26-34 页, 访问 <http://www.jitc.com/pub/r2026a.htm>。Neil Johnson 的站点 <http://www.jitc.com/stegdoc/>。

^f 用到的都是一些化学的技术, 一些液体为墨水涂到纸上后看不出字, 涂上另外一种液体就可以完全解密。现代刑侦取证手段中可以不涂上第二种药水就看到字体, 在红外线、或者紫外线等光照射下, 特殊的仪器可以分析出纸上的不同化学成分的划痕。

最近,人们开始在图象中隐藏秘密信息,用图象中的每个字节的最不重要的信息用密文信息代替。图象并没有怎么改变,大多数图象标准规定的颜色等级比人类眼睛能够觉察得到的要多得多,也就是人眼能够分辨的颜色是有限的,比如两种非常接近的红色,在视觉上我们已经无法分辨出来了,只是在颜色信息上可以分出来。这样加入了秘密信息后,秘密消息却能够在接收端用专门算法的软件剥离出来。用这种方法可在 1024×1024 像素的灰色刻度图片中存储 64K 字节的信息。能做这样技术的公开程序已有很多种。

Peter Wayner 的模拟函数也能使信息隐匿,这类函数能修改信息,使它的统计外形与一些其它东西相似:如纽约时报的题录部分、莎士比亚的戏剧、Internet 网上的新闻组。这类隐写术愚弄不了普通人,但却可以愚弄那些为特定的信息而有目的地扫描 Internet 的大型计算机^a。

而破解隐写术的叫隐写术分析学 (steganalysis)。

在编码、算法、隐写术之间的一个最大的区别是编码和隐写术是事先配置好的,就像是英文单词 transparent 用 Slater^b 编码表来表示就是 22611。一些特殊需要的编码书的内容也可能不一样,比如在 overhaul 编码表中是 15740。同样的,隐写术很难投入实际应用,因为它没有一个固定的标准来告诉人们在什么地方书写,什么地方去看,有的书写完干了就看不到了,或者是写了后被人当废纸扔了。隐写术必须是定制的,否则就有局限性。

相反的,现代密码学就凸显了它的优势,并且整个系统可以实现公共化,这些建立在适当密码学算法上的关键就是密钥,因此,这个加密技术系统可以被大家攻击、验证,和提高而不会威胁到使用它的人们的安全。

在 PGP 公司,我们实现程序的标准化和依据这个原则验证组件各个方面,我们提供给您我们软件的源代码^c来让你验证软件的算法安全性。公开的是 PGP 的算法的计算机程序代码,而非 PGP 公司商业软件的程序源代码,任何人都可以获得,并独自研究它的安全性。并且 PGP 公司广泛纳谏。2005 年中,平均每月有 2000 人次为 PGP 公司提出意见。

1.6 这本书的历史

Phil Zimmermann 设计第一个 PGP 软件时,软件包含描述 PGP 程序和基本操作很详细的文本文件,其中包含 PGP 消息实时加密数据格式,并且包含所有东西太多以至于它逐渐像的一本书^d。PGP 程序的资源在另外一本书^e中发布,并打开了 PGP 走向软件的道路。O'Reilly 在 1995 年发行了一本包含所有基础和历史内容的书^f。

最原始的 An Introduction to Cryptography (即本文的英文原版)发布于 1998 年的 PGP 6.0 中。这个 PGP 产品手册介绍了 PGP 加密系统的基本操作。我们发布程序的源代码在我们产品的手册中。目前,PGP 作为一

^a 搜索引擎可以搜索大量的字符数据,然而对于图像的内容搜索只能依靠图片的注释等信息,甚至是文件名。计算机识别图形内容的信息的技术也在发展,目前可以识别文字 (ORC, 光学字符识别),最直接的就是图形验证码,目前已经有出现可以识别简单的图形验证码。验证码的发明就是要区分人和计算机,随着这个技术的进步,Google 又研发了新的图灵算法来区分互联网上的人和计算机。

搜索引擎技术推荐查看《深入搜索引擎:海量信息的压缩、索引和查询 (Managing Gigabytes:Compressing and Indexing Documents and images Second Edition)》作者: [新]Ian H.Witten [澳]Alistair Moffat [新]Timothy C.Bell, 梁斌 译,电子工业出版社,540 页,书号: ISBN-13:978-1-55860-570-1

^b Slater: 由 Slater 发明一个文字编码,在 1880 加拿大的西北抵抗军中使用,访问: <http://homepage.usask.ca/~rhf330/tele.html>.

^c 源代码: 访问官方网站,免费获得源代码: <http://www.pgp.com/downloads/sourcecode/>。包含有 PGP Desktop 的源代码,还有 PGP 命令程序的源代码和 PGP 通用 GPL-修改源代码,你也可以找到 PGP 担保细则和特别要求在 <http://www.pgp.com/company/pgpassurance.html>

^d P. R. Zimmermann 著《The Official PGP User's Guide (官方 PGP 使用手册)》,麻省理工学院 (MIT) 1995 年出版,216 页,书号: ISBN 0-262-74017-6.,前不久刚脱销,应该可以在二手书店里找到。

^e P. R. Zimmermann 著《PGP: Source Code and Internals (PGP: 源代码和内部结构)》,麻省理工学院 (MIT) 1997 年出版,933 页,书号: ISBN 0-262-24039-4.包含 PGP2.6 源代码,脱销!

^f S. Garfinkel 《PGP: Pretty Good Privacy》, O'Reilly & Associates, 1995 年出版,393 页,书号: ISBN 1-56592-098-8.

个安全规则成为了 OpenPGP、OPGPMIME^a的技术标准。同样,大量的其它书籍出现也为程序员、技术工程师、科学家、数学家介绍核心加密程序的原理。这个技术并没有什么好描述的主题。有好奇心的读者可以随心所欲的了解,这就是这本书的意义!

自从那时候起许多事情发生了改变。美国出口条例只允许我们把源代码放在互联网上,而不是纸质图书中^b。PGP 软件现在已经发展成为被 100 多个国家作为紧密控制的技术。就在 10 年前,PGP 技术还很不现实,简直就是老祖宗的一个运动!而今天它变得很平常,商业和法律领域都要求使用数据加密技术。

尽管还有很多其它的资源可以参阅。但是本书作为一个好书去学习的地位是不容置疑的,这本书很有用、很热手!我们在 1991 年和 1998 年计算机世界改变时已经完成本书的修订工作!2009 年本书出现简体中文翻译版本!

1.7 原书作者特别致谢

Paulina Borsook 编辑文章、给予帮助支持、研究,并担当写作助理。Olivia Dillan 和 Will Price 坚持此书必须重写。Barbara Jurin 提供了她的最杰出的编辑技术。Phil Zimmermann 也参与编辑,并坚持本书必须以对话方式作为写作风格。Tom Stoppard 告诉我如果不用大量插叙的话就无法说清楚一些事情。

^a M. Elkins, D. Del Torto, R. Levien, T. Roessler, 著《MIME Security with OpenPGP (OpenPGP 的 MIME 安全)》, 访问:
<http://www.ietf.org/rfc/rfc3156.txt>, 这是 OpenPGP/MIME 的技术标准, 也就是 OpenPGP 成为复杂邮件信息的格式, 它依赖于 OpenPGP 格式。

^b 美国出口条例解释包含源代码的纸质图书是免税的, 这也就是说明我们可以合法的为读者发行纸质书籍, 那真是一件麻烦而繁琐沉重的过程, 很感激的是我们现在不再需要了, 请大家通过互联网来阅读!

2 为什么密码学那么重要?

如果你要把你的秘密展露给风, 你不应该责备风把你的秘密展露给树看。

—— Kalil Gibran 《Sand and Foam》

人是交流的动物。我们通常也是见什么人说什么话。在早一些时候, 我们不仅交谈, 而且低语交流。我们不仅写作, 而且传小纸条。在我们学会说话后不久, 我们也学会了见人说人话见鬼说鬼话!

密码学如此重要是因为它可以让一些事在表面上变成秘密, 它同样可以控制信息的通路, 指定什么样的人可以获得信息。你学习的加密已经超出了你的好奇心, 同样也是你的实际需要。在全球经济化下, 对信息的需求比货物的需求更大, 加密技术的本质就是悄悄话和大喊之间的区别。而且各种法律法规使它变的很重要, 丢失的数据时获得风险的同时也意味着法律的风险随之而来!

这里有许多重要的事。我们所做的并没有达到我们预期的目的: 刷牙、每 3000 英里加一次油, 确信我们的顾客信息表在笔记本内是加密的。如果你对自己和别人有其它疑虑, 为什么不使用数据加密技术呢? 下面的几个实例对你而言可能是重磅炸弹!

2.1 走进缺口: 骇人的事件

让我们来看看几个原本可以阻止的安全事件, 但最好不要发生在我们自己身上。看看最近的几个数据和备份丢失的实例, 下面几个实例你不愿意听到是因为在那些数据声明作废之前已经丢失了!

2.1.1 笔记本丢失事件

笔记本丢失的事件已经成为了危害社会安全的最大源头, 笔记本盗窃占了电脑盗窃案的 48%。台式电脑占了 26.7% 手持电脑设备^a占了 13.3%。

2004 年笔记本丢失至少造成了 670 万美元的损失, 那还只是设备费用的损失, 没有包含里面数据的价值^b。其实, PGP 软件的创始人 Phil Zimmermann 在火车站就丢过 2 个笔记本, 高兴的是 Phil Zimmermann 做了他所宣扬的事, 对于被偷的笔记本他没有任何顾虑^c。Phil Zimmermann 的笔记本经过加密, 更不用担心电脑信息会泄露! 潜在市场仍会继续增长: 2005 年国际数据公司 (International Data Corporation, IDC) 公布的统计数据显示, 在美国 50% 的电脑销量是笔记本, 这个数据是从 2004 年的 29% 增长上来的^d! 这确实是一个可观的数字, 也就是说超过一半的电脑是可以被轻易带走的。

没有人想丢笔记本, 越来越多的人认为笔记本存在的意义已经成为我们商业活动和日常生活的一个重要部分! 如果笔记本加密了我们就没有后顾之忧了。看看下面没有保护数据的笔记本被盗事件:

- 2004 年 6 月, 洛杉矶加利福尼亚大学官方代表发出警告, 一个被偷的笔记本中的 14500 名血液捐助者的个人信息存在风险。起因是 2003 年 11 月一个小偷撬开一个锁住的货车, 拿走了包含献血人的名字、

^a 手持电脑设备: 指智能手机 (SmartPhone), 个人数字助理 (PDA), 口袋电脑 (PocketPC), 还有其它嵌入式的设备。此类设备易用、易丢!

^b 根据计算机安全协会 2004 年 CSI/FBI 电脑犯罪安全统计数据。

^c 解释: 在美国买电脑很便宜, 美国一个上班族的一个月的工资可以买 4 台电脑。

^d 2004 年 11.8 日, Fitzgerald, Michael 在 CIO 杂志上发文 "How to Stop a Laptop Thief (如何阻止笔记本被盗)", 访问: <http://cio.idg.com.au/index.php/id:1973406143;fp:4;fpid:18>

生日、社保号码数据库的笔记本，数据库里面并没有包含血型等医药信息，官方人员并没有认为这是个人信息安全的损失直到 2004 年 5 月一次安全检查^a中发现。

- 2004 年 5 月，一个包含 100 多项正在进行的麻醉品管理局 (Drug Enforcement Administration, DEA) 调查信息的美国司法部检察长的笔记本从审计员汽车内被偷^b，他随即改口称这台电脑是偶然损坏后没留意就扔到垃圾堆，这台电脑内包含 400 页的案件数据，甚至可以推测出线人身份，值得关注的是 2 年后司法部检察长发布一个报告来抨击 DEA 和 FBI 没有保障的数据安全！
- 2004 年 3 月，一个硬盘驱动器（不是笔记本，可能是以前用过的）在运往银行去保管的过程中从一个调查公司雇员的汽车中丢失^c，调查信息包含美国保险公司的蓝盾计划 (Blue Shield) 和 Cigna 计划中的美国部分人的健康保险合格的检验报告，也就是这个原因 100000 个林荫健康联盟 (The Alameda Alliance for Health) 的成员的个人和健康信息连同那个硬盘一起消失的无影无踪。
- 2004 年 2 月，一台笔记本包含 Fargo^d 包含抵押人的姓名、地址、安全号码信息，从一个雇员在中西部停车加油时的汽车中丢失。^e
- 2004 年 1 月，包含通用汽车 (GMAC) 财政服务社 200000 顾客的姓名、地址、生日、安全号码、信用卡、婚姻状况、性别信息的 2 台笔记本，在亚特兰大^f 一个雇员的汽车中丢失。^g
- 2003 年 12 月，包含 43000 顾客的姓名、地址、安全号码信息的一台笔记本，在 Rhode 岛的主数据处理中心出来后丢失，银行的 CEO 表示 IT 部门有计划安装加密系统和丢失探测软件在他们所有的电脑上。^h
- 2003 年 11 月，包含 Fargo 的威尔斯 (Wells) 公司住房贷款顾客的姓名、地址、安全号码信息的一台笔记本，被小偷从银行顾问的办公室内偷走，但是小偷被抓获。据悉，该嫌疑人有伪造身份证的犯罪历史。ⁱ
- 2000 年 9 月，无线巨人高通 (Qualcomm) 公司的 CEO, Irwin Jacobs 在参加美国商业编辑作者界的会议时笔记本丢失，他称丢失的这台笔记本中包含组织所有有价值的电子邮件的信息，他只离开了 20 英尺远笔记本就不见了！^j

^a 2004 年 10 月，Becker, David 在 CNet 新闻网中发文“UCLA laptop theft exposes ID info (丢失的 UCLA 笔记本暴露了用户名信息)”详情：http://news.com.com/UCLA+laptop+theft+exposes+ID+info/2100-1029_3-5230662.html

^b 2004 年 6 月 7 日，新闻周刊题目：Missing: A Laptop of DEA Informants (DEA 笔记本信息的丢失) 访问：<http://www.msnbc.msn.com/id/5092991/site/newsweek/>

^c 2004 年 5 月，Lazarus, David 在《旧金山大记事》发文“Window smashed, data lost (车窗喝醉了，数据丢了)”，访问：<http://www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2004/05/12/BUG806JPV71.DTL&type=business>

^d 法戈，美国北达科他州东南部城市

^e 2004 年 4 月 16 日，Lazarus, David, 在《旧金山大记事》发文“Car thief whisks off Wells data (汽车小偷摸走了 Wells 公司的数据)”。访问：<http://sfgate.com/cgi-bin/article.cgi?f=/c/a/2004/04/16/BUGH865O141.DTL>

^f 美国佐治亚州首府

^g 2004 年 3 月 25 日，McDougall, Paul 在《信息周刊》发文“Laptop Theft Puts GMAC Customers' Data At Risk (笔记本贼把 GMAC 的顾客数据置于风险之地)”。访问：

<http://informationweek.securitypipeline.com/news/18402599;jsessionid=YWJ4ORVP2WZQJQSNDBGCKHY>

^h 2003 年 11 月 19 日，Mearian, Lucas 在《计算机世界》发文“Bank RI customer information stolen along with laptop (银行连同笔记本丢失顾客信息)”。访问：<http://www.computerworld.com/securitytopics/security/story/0,10801,88443,00.html>

ⁱ 2003 年 12 月 21 日，Lazarus, David, 在《旧金山大记事》发文“What's Next for Wells (Well 接下来会做什么呢?)”访问：<http://sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2003/12/21/BUGE73RAKL1.DTL>

^j 2000 年 9 月 18 日，《无线新闻》上文章“Qualcomm CEO Loses Laptop (高通的 CEO 丢失笔记本)”访问：

2.1.2 无安全保护的网路资源

这些世界上顶级的计算机科学学校, 甚至世界上最大的软件开发商应该知道, 或者宣布过客户的数据很安全的^a大企业, 他们的数据也经常不小心被修改。数据安全被破坏的事件是时有发生:

- 2004 年 8 月, 一名黑客非法进入位于美国加利福尼亚州西部城市 Berkeley 的加利福尼亚大学的计算机, 其中包含从 2001 年起参加了家园支持计划 (In-Home Supportive Services, IHSS) 的 140 万人的资助人和受助人的姓名、地址、安全号码、生日的数据库。^b
- 2004 年 2 月, 微软 Windows 2000 和 Windows NT 系统的源代码被发布在互联网上, 那是一个近 600MB 未经微软授权的代码包——对那些喜欢研究系统保密的相关漏洞的人来说这真是个好日子!^c
- 2003 年 8 月, 一个黑客进入位于 Berkeley 的加利福尼亚大学 Bancroft 图书馆一台包含姓名、地址和全世界 17000 名图书馆使用者的设备识别号码信息的服务器, 图书馆内有很多珍贵史料和史前古器物。没有人认为可以验证出贼的身份, 数据返回了 12 年前图书馆服务器的状态^d。
- 2002 年 2 月, 一个有意思的玩笑: 一个叫 Jeremiah Jacks 的年轻计算机程序员可以随意浏览 Guess.com 网站^e内的 20000 个顾客的名字、信用卡号、服务终止时间。为此, Guess.com 已经要求要把所有的顾客信息实时安全加密, 这个网站已经被联邦商务委员会 (Federal Trade Commission, FTC) 检查过, 需要创建一个可以使用 20 年的独立安全检查程序, 禁止随意索引所有的安全保密的数据。尽管这个网站没有被 FTC 通过。在 2003 年 6 月 Jacks 发现 PetCo.com 用了和 Guess.com^f一样不安全的数据库系统

Javelin Strategy、Research 和 the Better Business Bureau 共同发布了他们的 2006 年的身份盗用欺诈事件^g的总结报告, 该报告更新了 Javelin 和 BBB 2005 年的报告和 FTC 2003 年的总结报告^h。

这个报告体现出来的信息有:

- 美国受害人的数量从 2003 年的 1010 万人减少到了 2006 年报告的 930 万人。
- 合计损失从 2003 年到 2006 年从 532 亿美元增长到 566 亿美元。
- 受害人均损失从 5249 美元涨到 6383 美元。

<http://www.wired.com/news/business/0,1367,38855,00.html>

^a 例如一些网络在线存储服务。

^b 2004 年 10 月 20 日, Claburn, Thomas 在《信息周刊》上发文 “Break-In At Berkeley May Have Compromised Data Of 1.4 Million Californians (伯克利的中断可能导致有关 140 万人加利福尼亚人的数据被破坏)” 访问:

<http://informationweek.securitypipeline.com/news/50900323>

^c 2004 年 2 月 12 日, Lemos, Robert 在 CNet 新闻网发文 “Microsoft Probes Windows Code Leak (微软试探 windows 代码漏洞)”, 访问: http://news.com.com/2100-7349_3-5158496.html

^d 2003 年 11 月 23 日, Lazarus, David 在《旧金山大记事》发文 “Online breach at Bancroft (Bancroft 图书馆被互联网入侵)”。访问: <http://www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2003/11/23/BUG5D37C7T1.DTL>

^e 2002 年 3 月 2 日。Poulsen, Kevin 在安全焦点上发文 “Guesswork Plagues Web Hole Reporting (网络鼠疫扩散的猜测报告)”。访问: <http://www.securityfocus.com/news/346>

^f 2003 年 6 月 30 日 Poulsen, Kevin 在安全焦点上发文 “PetCo Plugs Credit Card Leak (PetCo 信用卡插件的漏洞)”。访问: <http://www.securityfocus.com/news/6194>

^g 身份欺诈事件: 泄露出的有关个人信息的数据被其他人拿去非法使用。比如伪造信用卡, 伪造身份证。或者使用他人名义消费。

^h 隐私权信息交流中心, 有多少身份盗用的受害者? 什么影响着受害者? 访问: <http://www.privacyrights.org/ar/idtheftsveys.htm>

- 每个受害者解决这些问题平均花费的时间从 2003 年报告中的 33 小时到 2006 年报告的 40 小时。

2.1.3 个人身份信息丢失

在近几年盗窃个人身份信息的定义已经改变。特别的是美国政府已经将打击犯罪上升到法律的高度。你可能验证身份到了泛滥的程度。如果某人用你的 iTunes^a账户够买了很多音乐而使用你的信用卡付钱, 这就是法律上的身份信息盗用(账号盗用), 还有很多身份信息盗用的其它形式。有这种犯罪形式, 罪犯不可能只使用你的信用卡号, 甚至包含你整个信息和记录的信用卡。然后把这些卡寄到其它地方。我们先称他们是小 i 和大 i。大 i 就是已经使用你的身份几个月的人, 当把恼火的欠债记录都给你看的时候, 你就发现了他。

大 i 的犯罪手段仍然是低技术含量的, 犯罪者至少见过你几次, 并且从你的付费、账单、预付卡等这样的信息上获得了你的个人信息。加密不能停止垃圾回收处理器的工作(因为他可以重新获得你的信息), 但是粉碎机可以。你还需要一个粉碎机, 他不仅仅是保护你, 粉碎掉没用的垃圾邮件确实是人生一大乐事!

据中国 CCTV 报道, 大 i 的这种犯罪手段在中国东南沿海发生过: 当顾客在刷卡机构消费刷卡时, 在刷卡的间隙用信用卡复制器备份数据到 IC 卡上, 同时记住密码, 然后用 IC 卡伪造信用卡, 卖到其它地方去提供非法消费, 一般案发率较高的是那些可以国际透支的信用卡。通常在邻国间非法使用!

2.2 法律法规

几乎讨论任何加密问题的时候都会包含涉及加密的各项法律法规。不管这些讨论究竟怎么变。近几年的讨论都是在法律法规中限制应用加密的方面的, 尽管还没有完成, 但是不阻碍我们使用加密技术。近几年的法律法规中一项最有意义的改变在加密技术应用方面都有很大改观, 主要体现在建议各个组织使用加密技术的方面, 甚至有时候也要求使用加密技术。

下面就是大量法律法规的概括, 请注意他的事件顺序, 看上去好像是越来越多了:

2.2.1 含保密项的法律法规

- 欧盟安全指导组织(European Union Privacy Directive, EUPD^b)和数据保护指导机构(Data Protection Directive, DPD)命令所有的欧盟成员国设立法律保护个人数据安全, 同时也规定非欧盟成员国商务活动时保护个人数据的最低安全标准。DPD 提出 8 个关键点中的第 7 个条款要求个人数据必须坚固安全。这也给其它国家保密项的法律法规的指定起了示范作用。
- 加拿大也有一些保密项的法律法规, 在保密法、个人信息保护、电子文档保密法(Personal Information Protection and Electronic Documents Act, PIPEDA)上有 2 点。个人权力也是被这些法律所保护的^c。
- 澳大利亚出台国家保密法规, 国家保护法规保护政府代理机构和使用税务文件的机构的个人信息^d。

^a iTunes: 苹果公司的网上音乐商店, 提供音乐 Mp3 文件等付费下载。

^b 欧盟安全指导组织的全文如下: <http://www2.echo.lu/legal/en/dataprot/directiv/directiv.html>, 欧盟中央的信息: http://europa.eu.int/comm/justice_home/fsi/privacy/, 美国人对于欧盟 DPD 措施的观点: <http://www.dss.state.ct.us/digital/eupriv.html>。

^c 加拿大保密项法律法规的情况说明: http://www.privcom.gc.ca/fs-fi/02_05_d_15_e.asp, 加拿大安全资源中心中的文档: http://www.privcom.gc.ca/information/02_06_01_e.asp, 加拿大保密法规内容: <http://laws.justice.gc.ca/en/P-21/index.html>。

^d 保密法和警方发言人网站发布法律的内容: http://austlii.edu.au/~graham/PLPR_australian_guide.html。

- 美国健康保险移植和责任法规（The US Health Insurance Portability and Accountability Act, HIPAA）要求美国健康公共事业部（Department of Health and Human Services, HHS）建立保险信息电子交换和个人身份验证信息的安全标准和技术指导。

值得注意的是 HIPAA 并不是严格意义上的保密法规, HIPAA 中的 P 代表的是英文 Portability(可移植), 而不是 Privacy (隐私, 安全, 保护), 它的目的是提高在保险信息记录和传输过程中的标准, 保证向前传输, 甚至是向后传输的时候都是被安全保护的, 它的过程已经很明确的指出使用加密技术。“是针对个人健康信息在传输过程中, 特别是在互联网^a传输时的安全” HHS 建议使用加密技术作为 HIPAA 技术守则的一部分。

- 日本个人信息保护法规（Personal Information Privacy Act, PIPA）将应用日本的任何一个公司或者至少 5 千人包括姓名、地址、性别、生日、电话号码、电子邮件地址的个人数据信息上。PIPA 要求每一个公司要有首席安全官（Chief Privacy Officer, CPO），否则处罚 30 万日元或者监禁最多 6 个月^b。
- 中华人民共和国于 2005 年 4 月 1 日《电子签名法》正式实施, 首个电子印章中心挂牌。法案充分借鉴了国外经验, 首次赋予可靠电子签名与手写签名或盖章具有同等的法律效力, 并明确了电子认证服务的市场准入制度。电子签名法规定, 民事活动中的合同或者其他文件、单证等文书, 当事人可以约定使用或者不使用电子签名、数据电文。当事人约定使用电子签名、数据电文的文书, 不得仅因为采用电子签名、数据电文的形式而否定其法律效力。根据我国电子商务发展的实际需要和实践中存在的问题, 借鉴联合国及有关国家和地区电子签名进行立法的做法, 我国电子签名立法的重点为: 确立电子签名的法律效力; 规范电子签名的行为; 明确认证机构的法律地位及认证程序; 规定电子签名的安全保障措施。法律规定, 电子签名必须同时符合“电子签名制作数据用于电子签名时, 属于电子签名人专有”、“签署时电子签名制作数据仅由电子签名人控制”、“签署后对电子签名的任何改动能够被发现”、“签署后对数据电文内容和形式的任何改动能够被发现”等几种条件, 才能被视为可靠的电子签名。法律还规定, 当事人也可以选择使用符合共同约定的可靠条件的电子签名。为保护电子签名人的合法权益, 法律规定, 伪造、冒用、盗用他人的电子签名, 构成犯罪的, 依法追究刑事责任; 给他人造成损失的, 依法承担相应的民事责任。

2.2.2 复合法律法规

- 瑞士银行建立了巴塞尔 2（Basel II）安全制度, 此举是减少在欧洲、美洲、亚洲和其它国际金融服务（Financial Services Providers, FSPs）中的风险, Basel II 设置了实施标准, 提升了 FSP 的信息技术的安全等级。
- SOX（Sarbanes-Oxley Act）是美国的一家会计公司, 公司出台新规要求操作透明化和他们的审计员和公司要承担责任。SOX 的 404 部分内部控制的税收管理办法中规定内部系统的测试（或升级为需要的标准）必须安全和坚固。SOX 的 404 部分挑战了 ISO17799 标准下的商业安全联盟（the Business Security Alliance）制定特别信息安全管理办法（the Information Security Governance Task Force），后者符合联邦信息安全管理办法（the Federal Information Security Management Act, FISMA）的要求。

^a 互联网的英文单词 internet, 当开头的字母 i 是小写时表示公司的网络, 也可以理解为内部网络, 当 i 变成大写字母 I 时, 表示的是万维网, 也就是我们说的互联网。

^b PIPA 官方和非官方的翻译: <http://www.privacyexchange.org/japan/japanmain.html>。

Sarbanes-Oxley 并没有明确提出保护财政报告详细信息, 也没有关于加密解决方案的具体命令。无论如何, 同时有 ISO17799 标准和 FISMA 的安全控制下的加密规则和电子签名可以用来避免数据被修改、数据的使用不当和造成损失。

- GLB (Gramm-Leach-Bliley) 是另外一个美国相同的法律, 它要求美国财政协会维护顾客机密安全信息, 特别要强调避免黑客的入侵。

尽管 GLB 的指导并没有要求加密顾客信息, 那么美国财政协会考虑数据加密也是理所当然的。联邦财政协会检查委员会 (The Federal Financial Institutions Examination Council, FFIEC) 推荐数据加密作为一个降低风险的技术。FFIEC 也已经考查过为什么团体组织不会采纳数据加密达到 FFIEC 所要求标准级别的原因

- 美国药监局 (US Federal Drug Administration, FDA) 标题: 21 号联邦电子记录规章 (21 Code of Federal Regulations Electronic Records, 21 CFR); 电子签名 (CFR 第 11 部分) 是政府文书工作销毁办法 (Government Paperwork Elimination Act) 的一部分, 21 CFR 第 11 部分提出在 FDA 所规定的工业中使用电子签名, 尤其是如何在医学设备、药品、生物工厂中实现电子记录。第 11 部分集中阐述了数据的可靠性和机密性、所需的安全、管理验证数据。

2.2.3 违规警告的法律法规

示范法的违规警告开始于加利福尼亚参议院议案 (California Senate Bill 1386, CA SB), 数据库安全警告办法 (the Database Security Breach Notification Act) 又叫 SB1386, 一些加利福尼亚组织的商业活动破坏了加利福尼亚民法, 无论是国有的还是私有的, 无论是不是本地, 加利福尼亚的这些组织的有任何有关影响到安全的违规都会被警告, CA SB1386 用于未批准的分散驾照、社会安全号码、信用卡、银行账户和图书馆卡号码等一些相似的数据。

尽管 SB1386 没有明确提出组织加密个人信息要求, 他表示法律将不会在加密的数据上应用, 换句话说, 如果个人信息加密了, 这些组织将会避免收到这个顾客安全问题的警告。

SB1386 就是 2005 年报道中很多笔记本丢失数据的原因, 他们也没有什么明显的动作。有些时候偷笔记本只是为了钱^a。但是 SB1386 成为了一个非常有意义的法律, 它只要求提醒人们数据已经丢失 (仍然可以提出诉讼), 没有什么处罚能够比得上公共督促更好。另外, 如果数据加密了, 它就提供了“出狱免费通行证”, 这个也促使了商业活动更加严谨。

当我写这篇文章的时候, 美国 23 个州都已经出台相关法律, 而且有在美国国会上的讨论。在澳大利亚、日本和其它国家也有类似法律出台, SB1386 的影响如此之大, 我不能准确的描绘出世界情形的变化, 只能说加密数据是一个好主意并开始在世界范围内变的引人注目了。

2.3 限制加密技术的法律法规

在 90 年代后期, 加密学技术被认为是一项军事技术, 这作为一个和武器的有一样的规章之下的东西, 在几年之后这逐渐被改变, 尤其是在美国和发过这些技术有紧密限制的国家。

^a 我自己的健康也与电脑盗窃事件中的加利福尼亚 75 万人的记录一起“丢”了。在一个内部工作的调查后我生病了。电脑作为一个东西丢了——他们并不是为了数据, 而只是为了卖“二手设备”换钱。当他们受到国家的关注时, 事情变了, 他们最后把自己也送了进去。

加密技术被认为是双用途的技术。一点是他可以同时提供民用目的和军用目的。它不再是一个军需品，不过是天使和魔鬼的化身，就像是游戏机和核技术^a！无论如何，在 1990 年到 2000 年政府对一些应用加密技术已经有了很大的改观，法国是限制使用加密技术最大的国家也变成了自由化的国家。美国没有任何在加密技术在使用限制，但是有对加密技术出口限制。美国出口条例在 2000 年已经被放宽。

国际范围内，出口加密技术仍然被控制在《瓦森纳协定（Wassenaar Agreement）》下，《瓦森纳协定》是一个控制双用技术的条约，条约现在仍在不断变化。总的来说，不管是不是有国际恐怖主义，法律法规都将会越来越健全！

写这篇文章的同时（2006 年），像 PGP 这样的加密软件可以随意销售和在互联网上下载，但必须在美国出口限制黑名单上 7 个以外的国家：古巴、伊朗、伊拉克、利比亚^b、北韩（朝鲜）、苏丹、叙利亚共和国，自由使用加密技术的障碍是美国限制出口黑名单上那些越来越少的名字。一些进口限制的国家有大规模的潜在市场，但是他们害怕窥探他们的公民（比如中国^c）而拒绝进口。

限制加密技术的法律法规仍然存在，但是出于实际应用的目的这些都已经不算什么了！

^a 技术的两面性，游戏机可以放松自己，也可以玩物丧志；核技术通常用来制造使用目的不同的 2 个东西：核电站和核武器！高速电脑被认为是双用途的，东西也必须物有所值，高性能计算机以作为游戏电脑而以结束告终。对于高的性能游戏电脑用途的众人讽刺情形在一些很奇怪的出口争议后消失了。

^b 利比亚作为规范化外交关系的典范已经从限制国家中去除，你看到的利比亚可能已经从限制出口的黑名单中删除了。

^c 尽管无法在中国的正规销售渠道获得这些技术，很多的公司和个人在国外购买，主要的用途也是和国外的一些商业机构交流。目前中国增加了对安全产品的研发力度，相对 WiFi 技术的 Wapi 技术就是一个体现。

3 密码技术的不完全历史

历史学家所记载的就是历史

——Sir Leigh Teabing

能够推测的密码技术历可能是 David Kahn 的《The Code breakers: The Story of Secret Writing^a》(电码译员: 加密书写的故事)。电码译员受鼓励而成为加密安全专家。书涵盖了编码和算法的历史和通过第二次世界大战从埃及人那设计和破解。Kahn 的其它书籍作为了解密码学历史的来源也值得去阅读。

然而 Kahn 缩小了《The Codebreakers》中的精华部分的, 这是今天加密技术中我们所感兴趣的, 也是最精密的部分。甚至我们所看重的其它算法的历史关于任何使用价值方面的信息 Singh 的《The Code Book^b》(密码书)》书中也没有提及。那也就是为什么我要写这部分的原因。我将揭示在其它地方被掩盖的东西。我并不是概括 Kahn 的书, 我将谈到一些他没有提及的、有侧重的、油腔滑调的绕过的一些内容。

3.1 人工密码技术

加密技术的历史几乎和书写的历史一样老。没有人可以说清楚他们开始的具体时间。我认为这时只有 3 个人知道如何去读和写, 他们中的 2 个人想写点什么东西来互相交流, 而不想让第 3 个人阅读。我只能假设。我还可以打赌如果第 2 个或第 3 个抄写员看的多了变聪明了, 他也会明白内容, 他们可能会把一些有惯用手法的地方做标记^c。很多这样的事情过去了后, 国王和富商发现了小纸条^d中的巨大力量。如果你把一些信息写下来, 然后交给你的大使、将军、购买者和其它可以信任的人。信息(能看懂的没有, 在世界上懂的密写的人没有几个)并不能阅读, 如果信息半路被截取, 内容仍然不回泄露, 因为看信息的人不知道这个信息到底是什么。

当然这个变成了桌面上一个只有专家才可以动手的鸡肉。技术专家也在不断钻研设计新的加密手段让别人无法阅读。这样加密学才成为一个学科, 而不是抄写员的小聪明。

就像是生物学的个体的摘要^e, 语系在加密学的范围内发生变化。当我是小的时候, 我就知道传小纸条的时候使用密写, 在传送的时候不必担心被里面和自己作对的人阅读出其中的信息。当我学会希腊文字来写纸条的时候, 发现它更像一个编码而不是加密, 但是它是学习密写的一个好的开始。当你开始你学会典型编码问题的时候你会觉得它很简单。打断一下, 我要说明一下, 希腊字符里面没有英文中的“J”, two 代表“O”, 2 种方法表示“C”。其它人可能注意到了其它的编码问题, 比如: 可以用一个符号代表“TH”“CH”, 但是没有好办法来表示“SH”。^f

^a D. Kahn, 著《The Codebreakers: The Story of Secret Writing》(代码破解: 密写的技术) Simon & Schuster 公司 1996 发布, 书号: ISBN 0-684-83130-9 (1967 版本的更新)。他把密码学(加密与密码分析的组)归功于阿拉伯人。事实上, 确实是他们创造了“加密法(cipher)”一词。关于密码学的早期主要著作是 15 世纪早期由阿拉伯科学家 al-Qalqashandi 完成的百科全书的第 14 卷, 这个也是最早的著作之一。

^b S. Singh 著《The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography》(密码书: Mary 的秘密革新, 苏格兰女王, 量子密码), Doubleday 公司 1999 发行, 书号: ISBN 0-385-49531-5。

^c 速记员通常使用自己的一些符号来记录一些有规律的文字, 来缩短记下文字所花的时间, 在书写时间上占大优势。

^d 政府和商人总是给技术发展提供资金。他们彼此交换小纸条, 从仅仅成为通用技术到远程通信技术中使用。远程交流的价值对政府、军事、商业和其它也有作用。

^e 一般生物学中指门、纲、目、科、属、种。

^f 中国的人工密码技术可以追溯到古代, 主要是诗词流行的时候, 一些特殊的诗词表达了一些特殊的意思, 其实这些诗词表达的意思已经是读者抽象出来的意思, 仁者见仁智者见智, 什么样的解释都会出现, 这时只有真正了解作者本人的才可以完全解密这些信息, 其它人不过是望文生义罢了。那时候一些特殊的填字游戏、灯谜游戏都是这个类型。近代中国引入的大量外来的

3.1.1 代替密码和换位密码

在计算机出现前, 密码学由基于字符的特殊密码算法构成。不同的密码算法是字符之间关系, 比如互相代换或者是互相之间换位, 好的密码算法是结合这两种方法, 每次都进行多次运算, 增加复杂程度。

现在事情变得复杂多了, 但原理还是没变。重要的变化是算法针对比特信息了, 而不是对字母进行变换, 实际上这只是字母表长度上的改变, 从 26 个元素 (26 个字母) 变为 2 个元素 (0、1 代码)。大多数好的密码算法仍然是代替和换位的元素组合。

3.1.1.1 代替密码

代替密码就是把明文中每一个字符替换成密文中的另外一个字符。接收者对密文进行逆替换就恢复出明文来。

在经典密码学中, 有四种基本类型的代替密码:

(1) 简单代替密码, 或单字母密码.

就是明文的一个字符用相应的一个密文字符代替。报纸中的密报就是简单的代替密码。

(2) 多名码代替密码.

它与简单代替密码系统相似, 唯一的不同是单个字符明文可以映射成密文的几个字符之一, 例如 A 可能对应于 5、13、25 或 56, “B” 可能对应于 7、19、31 或 42, 等等。这样增加了破解时被猜测的复杂程度

(3) 字母代替密码.

字符块被成组加密, 例如 “ABA” 可能对应于 “UIY”, ABB 可能对应于 “YGG” 等。

(4) 多表代替密码.

由多个简单的代替密码构成, 例如, 可能有 13 个被使用的不同的简单代替密码, 单独的一个字符用来改变明文的每个字符的位置。

著名的凯撒密码就是一种简单的代替密码, 将字母按顺序连成一圈, 它的每一个明文字符都由其右边第 3 个字符代替 (A 由 D 代替, B 由 E 代替.....W 由 Z 代替, X 由 A 代替, Y 由 B 代替, Z 由 C 代替)。它实际上更简单, 因为密文字符是明文字符的环移, 并且不是任意的置换。

ROT13 是建在 UNIX 系统上的简单的加密程序, 它也是简单的代替密码。在这种密码中, A 被 N 代替, B 被 O 代替等等, 每一个字母是环移 13 所对应的字母。ROT13 并非为保密而设计的, 它经常用在互联网 Vsenet 电子邮件中隐藏特定的内容, 以避免泄露一些难题的解答等等。

简单代替密码是很容易破译的, 因为它没有把明文的不同字母的出现频率掩盖起来。在好的密码分析者重构明文之前, 所有的密文都由 26 个英文字母组成。

多名码代替密码早在 1401 年最早由 Duchy Mantua 公司使用, 这些密码比简单代替密码更难破译, 但仍不能掩盖明文语言的所有统计特性, 用已知明文攻击, 破译这种密码非常容易, 只是密文攻击要难一些, 但

技术, 在近代中国历史的战争中, 大量的情报人员使用人工加密的一些技术, 诸如使用一些书籍、小说作为编码本, 每一个信息交换点最上下级使用的编码都不同。在这些军方密码人员中, 有不少能力非同一般的人, 现代的不少影片都披露了当时不少的内幕。

在计算机上也只需几秒钟。

多字母代替密码是字母成组加密，1854 年普莱费尔发明了这种密码。在第一次世界大战中英国人就采用这种密码。字母成对加密。希尔密码是多字母代替密码的另一个例子。有时你会把 Huffman 编码用作密码，这是一种很不安全的多字母代替密码。

多表代替密码由 Leon Battista 在 1568 年发明，在美国南北战争期间由联军使用。在计算机的帮助下他们容易破译，许多商用计算机保密产品都使用这种密码形式。第一次在 1586 年发表的维吉尼亚密码和博福特密码均是多表代替密码的例子。

多表代替密码有多个单字母密钥，每一个密钥被用来加密一个明文字母。第一个密钥加密明文的第一个字母，第二个密钥加密明文的第二个字母等等。在所有的密钥用完后，密钥又再循环使用，若有 20 个单个字母密钥，那么每隔 20 个字母的明文都被同一密钥加密，这叫做密码的周期。在经典密码学中，密码周期越长越难破译，使用计算机就能够轻易破译具有很长周期的代替密码。

滚动密钥密码（有时叫书本密码）是多表代替密码的另一个例子，就是用一個文本去加密另一个文本，即使这种密码的周期与文本一样长，它也是很容易被破译的。

3.1.1.2 换位密码

在换位密码中，明文的字母保持相同，但顺序被打乱了。在简单的纵行换位密码中，将原文的空格去除，这样是为了增加原文单词的分辨难度，因为有的英文是 2 个单词合成的。下面把字符转换为了大写，你可以每个字符大小写任意，明文以固定的宽度水平地写在一张图表纸上，下面是每行 10 个字符，密文按垂直方向读出，写成一行就是密文。

解密就是将密文按相同的宽度垂直地写在图表纸上，然后水平地读出明文。

原文: Cryptography has a long and fascinating history
明文: CRYPTOGRAPHYHASALONGANDFASCINATINGHISTORY
排列: CRYPTOGRAP
HYHASALONG
ANDFASCINA
TINGHISTOR
Y
提取: CHATY RYNI YHDN PAFG TSAH OASI GLCS ROIT PGAR
密文: CHATYRYNIYHDNPAFGTSAHOASIGLCSROITPGAR

这就是过去的太平盛世中人们一直在把编码转变的算法。在历史上著名的 Julius Caesar 被选为这个算法，用起来很简单，将字母按顺序连成一圈，就是把字母按照字母表中那个字母后面的第 3 个字母来替换（A 由 D 代替，B 由 E 代替.....W 由 Z 代替，X 由 A 代替，Y 由 B 代替，Z 由 C 代替）。它实际上更简单，因为密文字符是明文字符的环移，并且不是任意的置换。这个密码 3 在这个成为了一个不同凡响的数字。这并不是一个好的算法因为密码^a最高只有 26，因为 27 和 1 是一样的，很容易让密码破译人员写出所有可能的可能情况。

在第一次世界大战中，德国人所用的 ADFGVX 密码就是一种换位密码与简单的代替密码的组合。在那个时代它是一个非常复杂的算法，但还是被法国密码分析家 George Painvin 所破。

虽然许多现代密码也使用换位，但由于它对存储要求很大，有时还要求信息为某个特定的长度，因而比较麻烦。代替密码要常用得多。

^a 相似的 Caesar 作为罗马帝王，有一些信明显丢失了字母“J”和“U”。

Bruce Schneier^a过一句名言: 有 2 种加密方法: 一种是防止你妹妹偷看你的文件, 还有一种是防止政府看你的文件。Robert Morris Sr 给出建议: 在信息安全方面有 2 个基本事实: 你的敌手想看你的文件的期望度和你关心这个文件的程度。如果你的敌手不关心你的文件, 你也没必要花这么大力气去保护。同样的如果你的对手非常急于得到你的文件, 你就要花费大力气来保护。综上, 观察者说这里还有第 3 种情况, 也是 Bruce 没有提及加密方法: 不让你妹妹看你的文档(她很失望没看到你的文件), 但是又可以让你的老师知道内容(他想阻止你在课堂上传纸条)^b。

在另一个没什么好感的老师和我的好妈妈的帮助下, 我开始另外一种形式的编码——Gregg 速记法。Gregg 速记法在编码结构上有很多的优势。首先, 符号不是英文字母, 而是使用了花体^c, 其次, 他不是任何语言的字母, 他是语音代码加密法。对手可能教会我们书写笔记, 但是他没有注意我们每个人写的看上去都不一样。他只注意我们在听写的时候的内容的完整性和精确性。速记法可以使我自己完全阅读, 而使他人一点都看不懂^d。

加密成为一个通用技术被我首先应用在了我的妹妹的身上。它确实也是一个在千百年来合理的对付国家政府的高级工具。破解密码的攻击者可能只有猜测或者获得一封加密算法。不幸的是他们一直在这么做。在时间上, 妹妹还是比国家政府更好对付。这个技术的领先在使用你自己的符号系统来进行普通应用。

现在描述下古代加密技术进步——也经常没有进步。加密技术是一个个人使用的技术。一个加密学家将会设计出几套算法和密码。密码破译学家也会设计出相应的编码和解密算法。一些密码学家会写关于密码算法原理的书籍。一些解密学家也会写一些破解密码原理的书籍。这些书通常有限的提及关键内容, 或是故意将一些东西隐藏。有些天资高的密码破译学家也许有神助一般有魔法力量, 他们找出了密码中的小秘密。人工加密技术通常被复杂的算法所约束的。在密码编码时出现的错误和计算出错都很危险。第一次世界大战^e的出现使这个到达了顶峰。这时期有许多书介绍算法的工作原理和如何被破解。顶级加密机构已经将他们的技术编成书来作为机构培训人员的标准教材。这个学问的本身被认为是绝密和专利, 而其它国家并没有同等级别的专家。一般来说, 第一次世界大战后加密技术成为了一种黑手技术, 并成为了一门科学。第二次世界大战末期的加密技术在实际应用中也有很大的进展。人工加密技术有时使用计算机程序, 那也就是为什么有些算法是繁琐、价值不高的破解。密码破解专家特别依赖一些比其它出现频率更高的字母。密码破译学家从出现频率较高的词汇来猜测密文中的分布。英文中用最普遍的是 ETAOIN SHRDLU。事实上, 统计加密数据中提及的词汇正是密码分析学家的工作。

加密学家可以通过下面的几点来对抗密码分析学家的统计:

1. 使信息更短。

如果你的信息很短, 密码分析学家将只能得到很少的数据。说比做当然要容易。毕竟所有的加密点都达到了才能够安全通信, 否则不要进行通讯。使信息更小的逻辑结论是根本不要发送他们。

2. 减少统计结果。

加密学家可以加入一些没有任何意义的数字来改变从密文中统计数目。举个例子为什么不用 1 到 100 来代替 1 到 26 的数字编码过的字母, 或有多 个“E” 而只有 1 个“Q”。这是唯一一个最有用的来对付密码分析学家获得更多文本的办法。有人更狠的是使用复式字符组, 他们漫不经心的使用这个技术。

^a Bruce Schneier, 《Applied Cryptography: Protocols, Algorithms, and Source Code in C, second edition (应用加密学: 协议, 算法, 和 C 语言源代码, 第二版)》, John Wiley & Sons, 1996; ISBN 0471117099.

^b 这也建议 Bruce 不要有妹妹, 至少不是像我妹妹这么聪明的。

^c 花体: 这里指一种看上去是乱画的字体, 类似中国的草体, 可能很草。换句话说中国字写的不好的人写的字有加密技术的“科技含量”。

^d 注意对我妹妹来说还是很容易失效。我的敌人(妹妹)也会接近妈妈和速记法书籍。

^e the Aegean Park Press 有很多好书源: <http://www.aegeanparkpress.com/desc.html>, 这些书包含来自美国、英国、法国和意大利的文本数据, 它们中的一些书只是在最近的几年才撤出货架, 它们包含历史和密码分析, 涵盖第 1 次世界大战和其它时期的, 例如一个密码系统的书被 1876 年美国总统竞选候选人竞选时使用。如果你有足够的兴趣阅读这些历史而不厌烦, 在它们的书目中将有一些东西可以使你着迷。

3. 使用代表符号不仅仅是为了密文,更是为了全部语句。

这个技术就是为什么编码书在人工高级编码学中广泛使用的原因。这个途径使密码分析学专家更加步履艰难,因为他们只有更少的数据去尝试。甚至只有最好的办法:合并这个技术和上面的复式字符组。

4. 加密只是密文信息的一部分。

它可以减少分析学家得到的密文的体积,换句话说如果得到尽可能多的密文,就可以得出上下文的关系。密码破译者经常得出上下文的关系而提出加密的论点。一个信息像:来自 9001 的外交头面人物会见来自 9049 的贸易部长,讨论正在进行的 9964 的紧张关系。接合了报纸可以提供的 9001 代表美国、9049 是德国、9964 是伊拉克,特别是大于 9000 的号码这些片段将会给密码分析学家很大帮助

5. 使用多密码。

如果不同的部分使用不同的加密手段这会使密码分析工作难上加难。这也使得通讯伙伴的生活更加困难。

密码分析学是在不知道密钥的情况下,恢复出明文的科学。成功的密码分析能恢复出信息的明文或密钥。密码分析也可以发现密码体制的弱点,最终得到结果。密钥通过非密码分析方式的丢失叫做泄露。

对密码进行分析的尝试称为攻击。荷兰人 A.Kerckhoffs 最早在 19 世纪阐明密码分析的一个基本假设,这个假设就是秘密必须全受控于密钥。Kerckhoffs 假设密码分析者已有密码算法及其实现的全部详细资料(当然,可以假设 CIA(中央情报局)不会把密码算法告诉 Mossad(摩萨德^a),但 Mossad 也许会通过什么其他方法推出来)。在实际的密码分析中并不总是有这些详细信息的。

应该如此假设:如果其它人不能破译算法,即便了解算法如何工作也是徒劳的,如果连算法的知识都没有,那就肯定不可能破译它了。所以我们就要从密码分析攻击开始。

常用的密码分析攻击有四类,当然,每一类都假设密码分析者知道所用的加密算法的全部知识:

(1) 唯密文攻击。

密码分析者有一些信息的密文,这些信息都用同一加密算法加密。密码分析者的任务是恢复尽可能多的明文,或者最好是能推算出加密信息的密钥来,以便可采用相同的密钥解出其他被加密的信息。

(2) 已知明文攻击。

密码分析者不仅可得到一些信息的密文,而且也知道这些信息的明文。分析者的任务就是用加密信息推出用来加密的密钥或导出一个算法,此算法可以对用同一密钥加密的任何新的信息进行解密。

(3) 选择明文攻击。

分析者不仅可得到一些信息的密文和相应的明文,而且他们也可选择被加密的明文。这比已知明文攻击更有效。因为密码分析者能选择特定的明文块去加密,那些块可能产生更多关于密钥的信息,分析者的任务是推出用来加密信息的密钥,或者是导出一个算法,此算法可以对用同一密钥加密的任何新的信息进行解密。

(4) 自适应选择明文攻击。

这是选择明文攻击的特殊情况。密码分析者不仅能选择被加密的明文,而且也能基于以前加密的结果修正这个选择。在选择明文攻击中,密码分析者还可以选择一大块被加了密的明文。而在自适应选择

^a注:以色列的情报组织。

密文攻击中, 他可选取较小的明文块, 然后再基于第一块的结果选择另一明文块, 以此类推。

另外, 还有至少三类其它的密码分析攻击。

(5) 选择密文攻击。

密码分析者能选择不同的被加密的密文, 并可得到对应的解密的明文。例如: 密码分析者存取一个防篡改的自动解密盒, 密码分析者的任务是推出密钥。这种攻击主要用于公开密钥体制。选择密文攻击有时也可有效地用于对称算法。有时选择明文攻击和选择密文攻击一起称作选择文本攻击。

(6) 选择密钥攻击。

这种攻击并不表示密码分析者能够选择密钥, 它只表示密码分析者具有不同密钥之间的关系的有关知识。这种方法有点奇特和晦涩, 不是很实际。

(7) 软磨硬泡(Rubber-hose)攻击。

密码分析者威胁、勒索, 或者折磨某人, 直到他给出密钥为止。有时称行贿为购买密钥攻击。这些是非常有效的攻击, 并且经常是破译算法的最好途径。

已知明文攻击和选择明文攻击比你想象的更常见。密码分析者得到加了密的明文信息, 或者贿赂某人去加密所选择的信息, 这种事情时有发生。如果你给某大使一则信息, 也可能发现该信息已加密了, 并会被送回他的国家去研究。此时你会去贿赂某人; 密码分析者也许知道, 许多信息有标准的开头和结尾。

加密的源码特别脆弱, 这是因为源代码中会有规律地出现一些关键字^a, 比如: `#define`, `struct`, `else`, `return` 等。加了密的可执行代码也有同样问题^b, 如: 调用函数、循环结构等等。已知明文攻击(甚至选择明文攻击)在二战中已被成功地用来破译德国和日本的密码。David Kahn 的书中有此类攻击的历史例子。

不要忘记 Kerckhoffs 的假设: 如果你的新的密码系统的强度依赖于攻击者不知道算法的内部机理, 你注定会失败。如果你相信保持算法的内部秘密比让研究团体公开分析它更能改进你的密码系统的安全性, 那你就错了。如果你认为别人不能反汇编你的代码和逆向设计你的算法, 那你就太天真了^c。

最好的算法是那些已经公开的, 并经过世界上最好的密码分析家们多年的攻击, 但还是不能破译的算法。美国国家安全局对外保持他们的算法的秘密, 但他们有世界上最好的密码分析家在内部工作, 你却没有。另外, 他们互相讨论他们的算法, 通过严格的审查发现他们工作中的任何弱点。

密码分析者并不是总去深入领悟算法的。例如: 在二战中美国人破译日本人的外交密码——PURPLE (紫密) 就是例子, 而且美国人一直在做这种事。如果算法用于商业安全程序中, 那么拆开这个程序, 把算法恢复出来只是时间和金钱问题; 如果算法用于军队的通讯系统中, 购买(或窃取)这种设备, 进行逆向工程恢复算法也只是简单的时间和金钱的问题。

那些因为自己不能破译某个算法就草率地声称有一个不可破译的密码的人要么是天才, 要么是笨蛋, 不幸的是后者居多。千万要提一味吹嘘算法的优点, 但又拒绝公开的人, 相信他们的算法就像相信骗人的包医百病的灵丹妙药一样。好的密码分析家总会坚持严格审查已知算法, 以图把不好的算法从好的算法中剔除出去。

^a 计算机编程中对一些特殊的英文设置为保留字符, 比如 C 语言中的 `IF`、`Then`、`For` 等都是程序结构设计的一些特征字符, 计算机编程的编译器依靠这些字符来编译程序, 所以这些字符其他的变量, 或者是什么其他的名字是不允许使用的, 否则编译器就不能准确的识别和生成新的程序。

^b 计算机上所有的执行文件就可以被反编译, 虽然不能逆向还原为原来的代码文件, 表示成汇编语言代码是完全可以看懂程序思想的。一些特殊的结构都可以看到很明显的一些标志, 比如调用函数、循环结构。了解和学习汇编语言推荐《汇编语言》, 作者: 王爽, 清华大学出版社 2003 年, ISBN 7-302-07195-0。

^c 1994 年 RC4 算法就发生了这种情况。

一个人工高级加密的有趣事是 **Jefferson Wheel** (**JWheel**^a) 算法的名字是他的发明人美国哲学家、总统 **Thomas Jefferson** 的。他组成了一个由混杂字母表组成的环。密码是这个构成这个环的轮环。在环上拨号来加密信息，然后发送给对方一行密文。它是完全使用环上的密码来提高安全性的技术。所有技术使得人工加密技术包含了解密学家的技巧和密码分析学家的技巧。当然，变的太聪明也给密码分析学家提供了不少的帮助^b。

3.2 机械密码技术

欧洲的“黑暗时代”中，科学艺术，包括密码学，进展非常缓慢，当欧洲走出这个阴影的时候，编码和加密进入了一个快速发展的阶段。政府成立了大型的间谍网络，并且开发不少加密法便于他们之间的通信。发现并阅读敌方的密码信息也十分重要。因此出现了一个新的政府组织（在今天已经是一个重要的部门），称为“**black chamber**（保密局）”。保密局的工作时截获并且解密重要信息。一个例子就是英国的解密部门，在 18 世纪，这个部门主要是阅读美国和欧洲的加密邮件。

直到 20 世纪，美国才开始筹备并且设置官方的保密局。这不代表美国不重视加密技术。

第一次世界大战中，美国设置了自己的保密局，称为 **MI-18**，由 **Herbert O. Yardley** 领导。战争结束后仍然在纽约领导这个部门。并且发挥了很大的作用。1929 年美国保密局被解散，任务由陆军和海军负责。

第一次世界大战之后，人们开始考虑一种新的方法来实现加密。加密技术变成了一个科学，技术专家改进了最复杂的人工加密算法^c，目标当然是设计一种坚不可摧的算法。那就意味着这个目标设计的加密机器遵循一些人工加密中使用的原则，他的复杂性和可靠性足以破坏掉任何人工密码分析的算法。

3.2.1 转轮机

在 20 年代中，人们发明各种机械加密设备用来自动处理加密。大多数是基于转轮的概念，机械转轮用线连起来完成通常的密码代替。

转轮机有一个键盘和一系列转轮，它是 **Vigenere** 密码的一种实现。每个转轮是字母的任意组合，有 26 个位置，并且完成一种简单代替。例如：一个转轮可能被用线连起来以完成用“F”代替“A”，用“U”代替“B”，用“L”代替“C”等等，而且转轮的输出栓连接到相邻的输入栓。

例如，在 4 个转轮的密码机中，第一个转轮可能用“F”代替“A”，第二个转轮可能用“Y”代替“F”，第三个转轮可能用“E”代替“Y”，第四个转轮可能用“C”代替“E”，“C”应该是输出密文。那么当转轮移动后，下一次代替将不同了。

为使机器更安全，可把几种转轮和移动的齿轮结合起来。因为所有转轮以不同的速度移动， n 个转轮的机器的周期是 $26n$ 。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位置号。

最著名的转轮装置是 **Enigma**。由 **Arthur Scherbius** 和 **Richard Ritter** 在 1918 年设计^d。**Scherbius** 的商业计划是卖给需要加密信息的银行、律师和一些喜欢的人。直到几年后德国政府看到 **Enigma** 正是它们加密信息所想要的。**Enigma** 在第二次世界大战期间被德国人使用。它由 **Arthur Scherbius** 在美国申请了专利，德国人为了战

^a Jefferson Wheel 算法有很详细的描述在: http://www.monticello.org/reports/interests/wheel_cipher.html，你可以阅读怎样设计你自己从书中学来的东西。纸杯（paper cups）信息: <http://www3.brinkster.com/Redline/crypt/jefferson.asp>

^b HarperCollins 著《The Man Who Broke Napoleon's Codes (破解了 Napoleon 算法的人)》，Mark Urban 出版, 368 页, ISBN: 006018891X。这是关于 **George Scovell** 的故事，一个在惠灵顿陆军官员破解了被法国在伊比利亚使用的加密技术的密码和算法。这是一个非常有趣的故事，因为被描述为提高人类加密技术的科技被法国人用于一个或其它的用途，**Scovell** 逐步击败了它们。你可以看到当一些不注意的工具被实际用来使密码破译学家的工作更简单时，他就变的很有意思了。

^c J. R. Childs 著《General Solution of the ADFGVX Cipher System (ADFGVX 加密系统的通解)》，Aegean Park 出版, Book C-88, 245 页, ISBN: 0-89412-284-3。ADFGVX 算法是德国在第一次世界大战中的一个加密算法，6 个开头的字母组成了它的名字。ADFGVX 也不同于我以前任何描述过的任何一个。他并不是替换字符，而且重新整理字母并排序。用 2 个密文生成一个明文。我这么叫它是因为 ADFGVX 6 个字母成对出现

^d 有书籍称是 **Enigma** 是 **Arthur Scherbius** 和 **Arvid Gerhard Damm** 发明。

时使用,大大地加强了基本设计。

Enigma 有三个转轮,从五个转轮中选择。转轮机中有一块稍微改变明文序列的插板,有一个反射轮导致每个转轮对每一个明文字母操作两次。像 Enigma 那样复杂的密码,在第二次世界大战期间都被破译了。波兰密码小组最早破译了德国的 Enigma,并告诉了英国人。德国人在战争进行过程中修改了他们的密码。英国人继续对新的方案进行分析,他们是如何破译的。尽管如此,在 20 世纪 90 年代初 Enigma 也被广泛的用于大多数政府^a。

机械加密系统通常是由齿轮、转子组成,要加密的原数据作为档位。Enigma 使用 3 个或 4 个转子,而在 NEMA^b 密码机中增加到了 11 个转轮。

这个转轮的提升在加密技术中是很成功的,你可以很惊奇的在其它书中读到这些, Enigma^c 最后还是被日本的 PURPLE 密码机^d破解了。美国的 Hagelin M-209 机器也被破解^e了。因为,加密机把人工解密技术从商业中带出来,这是机械加密技术的成功标志。现在仍然有很多专业密码机在使用。当然,人类有进步,也早已破解了机器加密。机器加密的结果导致了密码分析技术的改变。在机器出来之前,密码分析专家都是从大学语言学院出身的,而并不是数学学院。她们在语言填字游戏等中有不错技术。机器加密技术的提高意味着密码分析学家开始向数学统计学家和专业密码工程师进化。

同样,机器加密还做着一个小事:与新发明的电子计算机抗衡。

像许多其它东西一样,计算机开始也并没有立刻用来应用加密技术。很多机械的或电子的计算机都是用电话线和中继器连接起来。密码分析学家必须使用这些设备来对付机械加密机。机器加密的问题是用手来解决实在太大了。机电系统没有一个可行的计算机办法破解 Enigma,或是日本的 PURPLE 和 Hagelin。德国的用来加密高层通话的 Lorenz 需要比机械系统更强能力的设备。因此,第一个可以设计编程的计算机是用来破解 Lorenz 的。那台计算机叫 Colossus^f。

Colossus 并不是一个完全的程序化的计算机,但是 Zuse^g 的密码机 Konrad 就是。你的程序被 Colossus 反向切换到程序自身。总之,所有的程序的加密都开始被破解,而 Lorenz 更为突出。但他的速度制约了它在多方面的效用。Colossus 是快速的。一个 Intel Pentium 处理器以主频大约 2Ghz 的速度运行 Colossus 模拟器才可以赶上一个真正的 Colossus。与它相反,Zuse 的机器编程后可以下棋^h,而且能够达到每小时 2000 次计算速度,而不是 20 亿次每秒。

目前有 10 台 Colossus 出产。战后在 60 年代还在 Bletchley Park 使用的 2 台电脑被送到了英国 Cheltenhamⁱ。剩下的都被邱吉尔将军下令摧毁,文件也全部焚毁。Colossus 在计算机科学界和密码学界是一个 50 年的神话,因为它是由 Alan Turing 设计的。这是由 Tony Sale 领导的 Bletchley Park 团队用 Tony Sale 的 the Bletchley Park

^a BletchleyPark.net 的 Enigma, 访问: <http://www.bletchleypark.net/stationx/enigma.html>。Hartmut Petzold 的《The Enigma rotor-type ciphering machine of the German Armed Forces (德军的转子加密机 Enigma)》德国博物馆, 访问:

http://www.deutsches-museum.de/ausstell/meister/e_enigma.htm

^b NEMA: 是 Neue Machine (Neue 机器) 或 New Machine (新机械) 的缩写。

^c 一个不错的、值得去看的电影叫《Enigma》。导演: Michael Apted, 剧本由 Tom Stoppard 从 Robert Harris 的小说改编, 访问: <http://imdb.com/title/tt0157583/>。那就是英国密码分析学家的神话诞生的地方。情节在历史上已经无根可查,但仍是一个可以鼓励人的神话。无论如何加密学不像那些电影基于一些历史或者在历史中 Enigma 的细节一样。他是由 Bletchley Park Trust 直接帮助。其中一个生产者 Mick Jagger 是加密学的怪才,他有收集密码机的爱好。如果你想看看那个时代加密学家、密码分析学家和其它天才的兴趣,好好看看吧。中国也有很多加密学的高手,甚至破解高手,推荐看一个叫《暗算》的电视连续剧。这部影片是中国密码界的一个缩影。

^d Frank B. Rowlett 著《The Story Of Magic, Memoirs of an American Cryptologic Pioneer (魔法般的故事: 美国加密先锋的回忆录)》Aegean Park 出版, Book C-81, 266 页, ISBN: 0-89412-273-8, 这本书是讲述 Frank B. Rowlett 的故事, 他的团队破解了日本的 PURPLE 加密机。这是一本特别好的一本书, Rowlett 也描述如何使用这个工作的细节。不像很出名的 Enigma 密码分析。从来哪个没有可以达到 PURPLE 的水平,在被破解之前日本人的技术确实很高。只有 PURPLE 是 Rowlett 津津乐道的加密算法。

^e Wayne G. Barker 著《Cryptanalysis Of The Hagelin Cryptograph (Hagelin 算法密码分析)》Aegean Park 发行, Book C-17, 223 页, ISBN: 0-89412-022-0

^f 英文意思: 巨人。访问这里了解这个 Colossus: <http://www.bletchleyparkheritage.org.uk/ColRbd.htm>

^g Prof. Horst Zuse, 《The Life and Work of Konrad Zuse (Konrad Zuse 的工作和生活)》访问: <http://www.epe-mag.com/zuse/>

^h Zuse 大吹说在当时他设计了这个机器,而 Colossus 设计不出来。他写的下棋程序就长达 60 页。

ⁱ Cheltenham: 切尔滕纳姆, 英国英格兰西南部城市。

Trust^a 和 Heritage 密码算法 (Codes and Ciphers Heritage Trust^b) 重新建造一个计划使用的 10 年的产物。

它在世界上机器加密和计算机加密的影响上很突出: 加密被认为是过去几个年头中政府法律对于公民涉及加密不满范畴的一个反击形式。

机器加密在一定程度上为第二次世界大战做出了贡献。事实上, 机器加密在战后更加成功。以瑞士人为例: 11 转轮的叫 NEMA 的 Enigma 在 90 年代早期仍在使用。基于转轮技术的流式密码在今日仍具影响力。

我们所关注的计算机的广泛传播带来了另外一个加密技术的革新, 这个作为他们感兴趣的東西, 人工加密和机械加密是一段重要的历史, 眼下将密码学实际应用才是最要的。

3.3 计算机密码技术

我认为第一次世界大战末就是机器加密历史开端标志。我认为 1975 年是近代计算机密码技术历史的开端。我选择 1975 年的原因是: 公钥加密算法和 DES (the Data Encryption Standard, 数据加密标准) 算法的发展。他们每一个都代表加密学本身最本质的改变。包括实现方法, 甚至是我们描述它的语言都发生了革新。现在应用的加密技术也正是将加密算法计算机化后的产物。

3.3.1 公钥密码技术

我们通过了解加密学的历史, 使用加密技术的难度和密钥分布不广泛的问题是一大难题。最好的加密算法是加密的强度达到的。如果对方可以轻易的获得或推出密钥, 那么加密算法^c毫无强度可言。这也是我们不希望发生的, 所以一个好的算法至关重要。

密钥分布的困难度限制了密码技术的使用精度。如果你看到一些老间谍电影中一个人手腕拿着一个公文包的样子, 你就遇到了在公共环境下的密钥分布的问题。

在 20 世纪 70 年代早期, 一大批年轻的数学家和密码学家开始考虑密钥分布的问题, 和讨论如何解决这些问题。Ralph Merkle、Whitfield diffie 和 Martin Hellman 每个人都在单独研究这一问题, 有时他们也会在一起讨论一些多样的设计方法。有趣的是, Ralph Merkle 开始试着去证明分布密钥是不可能实现的。他们中的 3 个人继续原来的研究。diffie-Hellman 在今天仍在使用。几年过去后。另外 3 个科学家 Ron Rivest、Adi Shamir 和 Len Adelman 继续名为 RSA 计划的工作^d。

公钥加密改变了加密技术原先所做的, 并用试验用 2 个密钥的方法解决了密钥分布问题。而在这之前加密或者是解密都是只用一个密钥 (即中国人最开始说的密码)。公钥加密使用了 1 对密钥: 一个用来加密数据和一个用来解密数据。另外, 你不可能从加密密钥推测出解密密钥, 因为加密密钥的任何信息都无法让你知道解密密钥, 它毫无理由的成为了一个秘密。你可以把加密密钥发布在报纸上, 或直接告诉别人, 或者划在浴室的墙上。“这个是用来加密一个秘密的密钥! 而不是解密的密钥”。

这就是为什么这种新式的密码加密叫做公钥加密。一个密钥可以完全公开而不用担心威胁到算法的安全。使用公钥, 等于我们把公文包的把手换成了软背带。我们不需要他交出我们中的每一个的原密码。我使用你的公钥, 你使用我的公钥。

就像我前面说的: 很多加密是靠直觉的, 因为人类已经做了一种形式的加密, 或者另外一种就是我们的书写, 这个公钥加密的完美原理在有些人认为是不可思议的。但是, 他确实是真的!

^a The Bletchley Park Trust 的信息可以在这里找到: <http://www.bletchleypark.org.uk/>

^b The Codes and Ciphers Heritage Trust 的信息可以在这里找到: <http://www.bletchleyparkheritage.org.uk/>

^c 事实上, 密码分析学家所作的就是在算法本身的逆向找出加密的密钥。

^d 据称早一些时候英国 GCHQ 的加密专家想出了与 RSA 很像的公钥加密算法。但是他们不发布和应用这个, 所以他们仍然不入流。CESG 拥有他们称为“non-secret encryption (非秘密加密算法)”的一些技术, 但是我写这个的时候并没有找到这些信息, 这不能证明他的真实性。最简洁的描述是 Bruce Schneier 的《Non-Secret Encryption, Crypto-Gram of May 1998 (非秘密加密算法, 1998 年 5 月的密码)》访问: <http://www.schneier.com/crypto-gram-9805.html> .

首先, 公钥加密引起了一大堆语言问题, 加密的密码直接就是可以公开的密码。这个加密的密码有时候我们称公共密码 (public-key) 和另外一个秘密密钥 (secret-key), 我认为秘密密钥叫私钥 (private-key) 比较好。也就意味着我们可以用这个词的缩写来代替我们前面提到的词汇, 我们下面开始讨论 2 个以 P 开头的东西。原来的这些词在一些简短的纸条上书写是很不方便, 因此, 我们叫他们 Pu (public, 公共) 和 Pr (private, 私有)。作为一个好的加密算法, 我们使用不同的字母称他们为 Public -key (P, 公有密钥) 和 Secret-key (S, 私有密钥)。

这里有一个最大的语言问题。我们如何称最开始需要加密的文件? 密码源? 我们已经使用了单词 public (公共的)、private (私有的)、secret (秘密的)。另外 2 种密码叫原始对称密钥加密算法 (original cryptography symmetric-key cryptography) (因为这个密钥同时可以用来加密和解密) 和非对称公钥加密算法 (asymmetric cryptography for public-key cryptography) (因为他不是对称的, 也就是不同的, 加密和解密的密码都不同)。

这可能不是最好的论据。但是如果你读了更多信息, 你就会看到他们都在被使用。现在, 我用公钥加密算法来区分公用密钥和对称加密算法。我先看公钥和私钥。我发现这些项目是很饶舌的, 这是个暗喻。

公钥加密算法如何加密? 用违背直觉的和垂直螺旋的思维方式考虑后, 认为这 2 个密钥是没有联系的, 那就是他为什么近年才发展。如果没有计算机技术的发展, 公钥加密算法也无法投入实际应用。公钥加密算法是基于一些有相关性的数学算法上的, 但是很难上手。几乎没有人想去直观了解这些数学推导。乘法很简单, 但是长除法很难。数字因式分解也许更难。一个数的幂计算很简单。但是开根和对数就难了。公钥加密算法的不同简而言之就是使用的是非对称正运算和逆运算。如果用手, 你是把 391 进行因式分解要比乘以 17 或 23 难的多。而计算机算这个不是很困难的事, 所以我们都使用很大的数字, 现在 PGP 的最小的密码长度都是 1024-bit (比特^a) 的, 他有近 300 个数字的长度。而我们推荐的是使用 2048-bit 到 4096-bit, 那就意味着大约 600 到 1500 个数字的长度! 要和这么多的数字计算就不能没有计算机。公钥加密技术也是计算机加密技术。

3.3.2 加密技术标准的提升

就像我先前提到的, 1975 年就已经有了数据加密技术标准 (the Data Encryption Standard, DES) 的更新。就像在商业交流中机械加密的安全一样。计算机加密标准也变的很完善。1973 年, 美国政府需要设计一个算法用于联邦政府加密无分类数据, 他们在 1974 年第二次制定。IBM 称这个基于老算法的数据新算法叫 Lucifer^b。在创建 DES 的过程中的一部分。美国国家安全局 (the National Security Agency, NSA) 修改了 DES, 他的密钥从 8 字节 (64-bit^c) 的数字缩短到 7 字节 (56-bit) 的数据, 同样也改变了一些算法的内部结构 (特别是数据结构 (data structures, S-boxes))。此外 NSA 也没有任何解释为什么这么改, 只是说为了提高 DES 标准。美国政府在 1977 年联邦政府信息处理标准中 (Government Federal Information Processing Standard, FIPS), FIPS-46 文件中批准 DES。

NSA 对 DES 的改变引起争论。为什么说密码长度从 64-bit 缩小到 56-bit 是“提高”? 难道缩小后的密码长度是一种提高? 人们对于这个说法的安全性感到质疑, 特别是安全程度不能用直觉来判断。这样许多年后, 密码学家对 DES 有了许多不好的观点。无论参议院的特别调查组怎么调查, 他们结束了论断。民间密码破译人员推测 NSA 不仅削弱了 DES 算法的强度, 而且他们似乎确信我们想不到他们用 DES 内部结构技术的加密信息算法强度会优于我们可以达到的技术^d。值得一提的是密码内部系统中的秘密入口叫做后门 (backdoor)。

^a 存储设备中的最小信息容量单位。

^b 这个英文单词是魔鬼、恶魔、撒旦的意思。

^c 简单说明一下这个字符和比特位如何转换, 一个英文字符占 1 字节数据, 在计算机内存中使用 8 位的 2 进制代码来表示, 样子一般是这样的: 00101010, 你会发现这个数最大是 11111111, 也就是 10 进制的 255, 加上 00000000 是 256 种数字, 所以 Ascll 有 256 个符号 (包含控制符), 但是中国汉字太多了, 完全超过 256 个字符了, 我们就使用 16 位的二进制来表示, 也就是 2 字节。

^d 这里有大量的讨论关于 NSA 在发展中怎样直接改变了 DES。DES 团队的一个成员 Walter Tuchman, 他说 NSA 什么都没有改变。

这有必要担心一下, 人们设计的算法中有后门的这个措施是有意行为, 使得人们更容易根据自己的入口破解加密信息。

尽管如此, 可见 DES 的重要性。

你要知道, 在人类的历史中, 加密技术都是聪明人做的事。国家的密码权威需要控制你和别国其它加密学家和密码分析学家的水平。在机械加密时代, 这个例子就是改变密码位数: 密码学家用加密技术制造加密机, 如果你有的正确的机器, 你就有了相对的安全。一个算法标准想法的关键是如何让所有人接受。因为如果我们隐藏东西用了相同的办法, 你不能够决定使用标准还是隐写术。我们知道去哪里可以找到, 并看到一个设计拥有公共组件的密码技术(甚至我们一点不懂如何设计加密), 这是违背了加密技术的原则的。这就是 19 世纪密码学家 Auguste Kerckhoffs 的 Kerckhoffs 原则中的——一个好的加密算法的由来。这个加密算法结构就不是一个秘密, 只有密码是秘密^a。

DES 并不符合 Kerckhoffs 原则, 但是他向那里发展了。美国政府厌恶放弃他们的加密标准的人。1991 年时, 在政治环境中出现公开藐视 DES 技术的声音。个人加密专家也在破解 DES 中取得进展, 结果也不是那么坏。在那几年密码技术专家 Eli Biham 和 Adi Shamir 发布了一篇论文关于破解 DES 的方法^b。他们开发出一种叫微分密码分析学(differential cryptanalysis)的方法。他们使用新技术展示了 DES 标准下的 S-boxes 如果使用随机的方法会比想象中的有多么安全。显然的一些涉及设计 DES 的人知道微分密码分析学是什么。他们同时也展示了 64-bit 下 DES 并运用随机数字化的 S-boxes 比 56-bit 的 DES 算法强度要弱。显然这个(56-bit)已经比更长的密钥(64-bit)和更强的结构的算法更好。但是所做的已经做过了, 他们认为 DES 中没有什么特别用意, 尽管还是有些人这么认为。

1994 年, 一个 DES 的设计者 Don Coppersmith 应大众要求公布了 DES 的 S-boxes 的设计标准, IBM 早就知道了在设计 DES 的时候的微分密码分析。Coppersmith 说 NSA 说服 IBM 不公开这个技术秘密。因为它在很多双用途的破解密码中很有用。所以在 Biham 和 Shamir 发现微分密码分析技术之前, NSA 和 IBM 就早都知道了。但这是他们独立发现的。

DES 是在普遍存在的技术中被讨论最多的。因为这个过程使大家都注意到他了, 我们有比其它的知道更多关于它的东西。对于所有的缺陷, 它确实是一个好加密算法, 特别是在他的那个时代。今天我们所知道的很多都是从 DES 的研究中来的。DES 最大的问题就是他的密钥长度, 56-bit 太小了。当这个问题开始凸显的时候, DES 成为了一个新的算法 Triple-DES^c的组成。它把 3 个 DES 操作合成了, 支持 2 个或 3 个 56-bit 密钥^d。当然 Triple-DES 比 DES 慢 3 倍。用起来让人相当恼火。比如, 当 DES 密钥变成 2 倍还是 3 倍的时候, Triple-DES 才和单个 DES 相同。

这个情况导致了另一个加密标准, 由 DES 用 5 年时间制定了 FIPS。它在 1983 年又继续了 5 年。同样在 1988 年和 1998 年再次得到延续。这时候 Biham 的 Shamir 发布了破解 DES 的时候, 5 年的时间已经过了 2 次。尽管在 DES 设计上有担心。他短小的密钥仍然是被嘲笑和轻视的对象, 那么就找一个替换它的东西吧!

另一个成员 Alan Konheim, 要求 S-boxes 完全改变。参议院在 IBM 关于小的密码长度一样有很好的强度的报告下结束了争论。它直接影响了 S-box 的设计。我相信 NSA 有一些间接的改变, 可能和你的看法不同。

^a Auguste Kerckhoffs, La 军事加密专家、军方 Journal Des sciences (Journal 数据加密) 专家。1883 年军情数据:

<http://www.petitcolas.net/fabien/kerckhoffs/>, Alors 这篇文章在法国, 包含了 Kerckhoffs 原则的内容, 在接下来的几十年一个加密算法也该达到的密码安全意旨是这个安全算法系统应该是 white boxes (白盒), 而不是 (white boxes) 黑盒。能看到内部结构是白盒, 黑盒由输入输出判断, 是看不到内部的。

^b Eli Biham 和 Adi Shamir, 《Differential Cryptanalysis of DES-like Cryptosystems (DES 类密码学系统的微分密码分析)》, 在《Journal of Cryptology》上, 卷 4, 3-72, IACR, 1991。还有他们的《Differential Cryptanalysis of the Data Encryption Standard (DES 的微分密码分析)》, Springer, 188 页, ISBN 0-38797-930-1。

^c Triple 是三重的意思。

^d 这个三重的方法可以被其它任何算法使用, 而且可以使算法更强, 然而你不可否认两重加密就可以变的更强。这有一系列称为 “meet-in-the-middle” 的攻击来验证双重加密不比本身强度更高。

3.3.3 AES 标准

在 90 年代中期, NIST 开始考虑 DES 的替代。他们告诉密码专家和工程师他们需要一个新的标准算法。在建议中, 由竞选的方式来选择和委命下一代加密算法。在 1997 年 1 月 NIST 他们开始转变 DES 算法到新的 Advanced Encryption Standard (AES, 高级加密标准^a) 算法。有很多标准竞争 AES 算法标准^b, 在那年的 9 月他们公布了候选算法的结果, 公布了 AES 的实际需求。它包括:

- 候选算法是区块加密, 而不是传统的流式加密。
- 候选算法使用 128bit 的区块。
- 候选算法必须使用最小 128bit 长度的密钥, 并且可以向上扩大, 且可以使用 128、192、256bit 密钥。
- 候选算法至少和 Triple DES 算法一样快, 最好和 DES 算法差不多。
- 候选算法必须免费, 没有知识产权约束。可以提名为专利算法, 但是如果被选, 必须成为免费使用许可。
- DES 被设计为使用至少 5 年, 最后超过了。在这个竞争中, 起码最早要使用到 2003 年。这个替代品必须考虑使用 25 年到 50 年之间。

2 个最主要的要求并不意外。如果 DES 的替代算法比 Triple DES 算法慢, 人们将不会使用。同样的如果有使用的法律限制, 将不会被使用。但是除了这 2 点的也突出。当时算法使用都使用 128-bit 长度的密码和 64-bit 区块大小^c。加密学家并没有简单的使用 AES 的要求。

共计 15 种算法被提交作为 AES 的候选算法。NIST 举行了 3 次会议, 第一次是 1998 年夏天。

回头看看 AES 的过程, 直接指出了 DES 的问题是很惊奇的事情。DES 是一个公共事业计划, 同样也是一个安全公共事业计划, 围绕设计与结构所展开的需求、过程、决定和讨论是个秘密(直到现在也是)。甚至今天, 不小心也可以引起 DES 的争论。一个选择新标准算法公开的、可共享的、竞争的过程, 是最好的选择的方式。

最大的问题是怎么管理这个竞争。NIST 说最初 5 个一组而且会减小入口。然后再从中选择 AES 标准。他们说社团介入将会称为重要的一部分。他们将做出一个选择, 而不是光投票。在我和 NIST 人员说话的时候, 他们称所标注过的算法在处理上并没有和其它的不一样。

在这个复杂原因下, 也有 NSA 包含的问题。NIST 说 NSA 将会做出决定。他们说 NSA 会提供注释了技术帮助。大多是通过密码分析技术。换句话说, 如果 NSA 知道破解了一个算法, 那这个肯定没戏了, 详情肯定不会被公布(甚至是 NSA 自己破解的)。

那么多选者一个算法的建议中, 有一个建议是很简单的, 那就是速度。用密码分析破解他们的速度, 最快的算法的通过密码分析破解可能已经用在 AES 上了。另一些人隐约注意到破解速度远比安全重要。

人们也认为有政治因素在里面, 我想起了在政府中有地位、有 DES 的开发历史的 IBM, 他们确信他们提交的方案 MARS 能够赢得胜利。我认为这个优势使 IBM 处于不利地位, 这个最重要的原因是无论是胜者都必须彼此建立信任。因为 IBM 的特殊关系, 他们很显然毫无疑问成为了胜者。剩下的人实际是在障碍物下,

^a Daemen, Rijmen, 《The Design of Rijndael: AES - The Advanced Encryption Standard (Information Security and Cryptography)》(Rijndael 的设计: AES-高级加密标准(信息安全和密码学)), Springer-Verlag, 2002.

^b NIST 启动 AES 的计划可以在这找到: <http://csrc.nist.gov/CryptoToolkit/aes/>, AES 的要求可以在这里找到: http://csrc.nist.gov/CryptoToolkit/aes/pre-round1/aes_9709.htm

^c 这有大量的例子, 方形算法使用 128-bit 的区块。也有很多算法使用多种长度的密钥。但是密钥长度超过 128-bit 将不会增加安全。例如: Bruce Schneier 的 Blowfish 算法, 使用的密钥是 448-bit 的, 但是 Schneier 自己也不推荐使用超过 128 的。用来最为 SSL 连接的 RC4 密码算法, 可以使用到 2048-bit 的密钥, 还有约无论你哪怕是使用 600-bit 的密钥, 一些诡异的密码学家质疑你最多使用了 128-bit 的密码。尽管如此, 一个很重要要知道的常识: 密钥大小超过 128bit 时, 与其说是设计需要, 不如说是智力炫耀(就像密码战争中向政府的藐视)。

当然这里也有国家因素在里面。“美国政府不会选择外国人设计的算法^a”。每个人共谋提出相同和相反的意见。一个非美国人设计的不可靠的算法成为了人们谈论的最耀眼的话题，换句话说 NSA 已经破解了它。这也不是他们的错。

1999 年 NIST 测试的算法减少到 5 个。他们其中 3 个不仅比 Triple-DES 快，而且比 DES 还快。这 3 个算法也是没有知识产权的。这也就是这最后的 5 个之中的 3 个现在很出名的原因了。这 3 个是 Rijndael^b、Twofish^c 和 Serpent^d。剩下的 2 个是 MARS 和 RC6。

Rijndael 的速度是最快的。Serpent 慢一些，但是更安全。Rijndael 有很革新的设计。但是在密码学研究里面，革新并不意味着获得赞扬。Serpent 的设计者很快指出：难道速度是决定标准吗？并且他们可以除去一些额外的安全系数后提升到比 Rijndael 还快的速度。Twofish 的设计者指出他们可以调节 Twofish 算法细节，使它和 Rijndael 一样快，或者和 Serpent 一样安全。每个人都写出过程，并且作报告来展示无论有什么要求他们自己都是最好的。

我觉得三个算法之中任何一个都是不错的。我对 Rijndael 有些接触。Rijndael 是一个非常完美的算法。它的内部使用了有趣的几何学算法。它的速度也非常快。我不同意他们所谓最快和最安全的一些言论，但是我们也要考虑这个，当然，快固然好。Twofish 深深的吸引了我。如果是我做决定，我肯定会选 Twofish，因为它拥有更传统的设计。无论我们在 15 年后发现了 Rijndael 它有多么大的缺陷，我也想选 Rijndael。在 OpenPGP 标准^e中，我们把 Twofish 写入标准。现在你也可以在 PGP 软件能中选择使用这个算法。

NIST 在 1999 年第二次 AES 会议中举行了民意测试投票。他们在会上告诉人们给这 5 个项目进行投票。如果你读了 AES 竞争中的其它条目，你会称这个投票叫竞选。但是注意它不是！这时 NIST 很清楚他们会考虑密码专家的意见，获胜者的投票不是很重要，但是仍然考虑一大堆聪明人观点所体现的价值。在 2000 年的第 3 次 AES 会议中又有一次投票。

最后，NIST 选择 Rijndael 作为 AES 标准算法，它嘲笑了投票的方式。无论最后什么样的决定，它成为了一个有趣的决定。在技术上 Rijndael 确实是大胆的设计，也没有特别强调他的加密速度，这是一个对很多人有利的因素。在政治上，2 个比利时密码学家设计了美国标准。它使 AES 变得国际化，因为它的参选者很难为民族主义来辩护。当进行民意投票选择时，NSA 都限制公开密码破译的过程。

最重要的是，AES 的竞选称的上是密码学上的技术。他开始于技术，提供了通用加密设计参数的要求。5 个参赛的算法都是好算法，也不用担心害怕使用他们其中的任何一个。它也说明了密码技术安全设计的公开化，并与全世界分享。它同样有 Kerchoffs 原则的部分——你无需对安全性有任何设计上的秘密。

AES 中的密码设计者告诉 NIST 他们测试其它算法的竞选工作很顺利，这次有了新的散列算法标准（hash algorithm standard）。NIST 已经举行了 2 次领先的高级哈希标准（Advanced Hash Standard，AHS^f）会议，一次在 2005 年 10 月，一次在 2006 年 8 月。接下来的几年将会很有意思。

3.3.4 密码界的战争

没有人在密码界战争中死亡。并没有火力射击，墨水倒是用了不少。这些人在 19 世纪 80 年代和 90 年代

^a 我不记得谁说的这个，但是我记得有人在喝咖啡的时候曾经说过。

^b 如果你不是佛兰德（Flemish）人和荷兰人就不能正确念出 Rijndael 的发言；你不可能把 R 和 ij 在一起正确的念出。你可以用很接近的原来发音的英文 rain-doll 或 rhine-doll 代替。我觉得 rain-doll 更像佛兰德语的发音，而 rhine-doll 更像荷兰语的发音。这是根据我的比利时和荷兰的密码专家朋友的发音念的。我选择 rain-doll 是因为我好像听到他的发明人这么说过。

^c Twofish 英文意思：双鱼。

^d Serpent 英文意思：大蛇

^e Peter Gutmann 的《The Crypto Gardening Guide and Planting Tips（密码学构造指导和注意事项）》，2003 年 2 月出版，访问：http://www.cs.auckland.ac.nz/~pgut001/pubs/crypto_guide.txt，Peter Gutmann 是一个出名的研究者，他做的不仅仅是加密。他是 PGP 2 的开发者之一，而且拥有很多其它成就。他同时还有敏锐的智慧，不惧怕使用它。他所写的一切就值得去读读。如果你在关注密码学就应该看看这个，看过他的一些提示后你也会变的明智。

^f 阅读高级 Hash 标准最好的地方是：<http://www.nist.gov/hash-function>，也期待它对社会的改变。

在政府和每个人之间讨论、争论加密技术。你可以在 Steven Levy 的书《Crypto》^a中找到关于密码界战争的详情。有很多因素直接导致密码界战争不可避免的爆发:

- 政府认为密码技术是他们的领域。无线电技术使得密码技术对他们来说很需要。加密专家和密码破译专家说的所有方面都被称为战略科学。况且, 加密技术如何影响世界的声音只是在民用公开化后开始出现的。
- 计算机已经成为鲜明特色的加密技术, 使得密码技术成为公民生活中加密技术的实际应用。公钥加密技术解决了传统密钥分布的问题, 才使得在公民使用加密技术成为可能。其次, 像 DES 这样的密码标准有潜在的互用性。
- 机械加密仍然很有用。政府在到处使用的密码机是第二次世界大战的密码机的延续。更好的是使用了更多的转子和其它更新的技术, 但是还是相同的东西。政府认为加密技术成为一种战争的技术, 并且用法律法规^b控制像战争中的其它工具一样。
- 像这种发生在第二次世界大战中的, 有关阴谋或间谍的技术仍在继续。密码机被盗、编码书和密码术被盗。密码分析学中一个称为“后门”的简单描述漏洞的词汇也危及到了密码机。

密码机的一个最大丑事是一个密码机供应商瑞士加密公司 (the Swiss company Crypto AG), 伊朗政府指出他的密码机很有风险。显然, 德国和美国已经把后门设计在了密码机^c上。

- 最具讽刺性的是, 接合美国设计 DES 的过程, 发布了关于加密技术设计开发的图书, 甚至包含密码学趣事的书。从沉迷于密码技术 Martin^d Hellman 到 Kahn 的《the Codebreakers》。
- 政府在世界活动中所做的是高压、威逼、强制。这在公民世界中没有赢得什么掌声。

我发现了密码界战争中计算机的发明很有意思, 它是在密码机的基础上设计的。现在回头看看, 它似乎又长、又慢。

在密码战争的中期, PGP 软件的设计、分类加密在一片喧哗声中赢得了喝彩。PGP 软件的原始设计人 Phil Zimmermann, 在向美国政府的一个关于 RSA 数据安全的申诉后接受破坏出口限制条例的调查。在调查完毕后, PGP 公司成立。当遇到任何接触出口限制的部分时, 都使用条约中的这句: 印刷材料, 包含原始数据, 在这些法规中豁免。那就意味着如果你要在加密软件中印刷源代码, 它会被法律豁免。如果印刷的数据被扫描进计算机, 并且集成在软件里面, 那么这个软件在法律上就可以出口。PGP 公司和互联网的联合正好就是 1998 到 2000 年美国批准的获许销售 PGP 软件不违反出口法规的根据。

这并不令人吃惊, 这也就是为什么密码界的战争最后以失败告终。密码学就是数学, 是在长远看来这是和危险技术一样会带来社会问题。1999 年放宽加密技术控制。在这之前法国对该技术有超过其它国家最严格的限制。从那以后法国立刻变成了最自由的国家。2000 年美国放宽了出口限制条约。当然, 直到今天自由化都

^a Steven Levy 著《Crypto (秘密人员)》: 在数字时代加密技术人员如何应对政府保密条规, Diane Pub Co 发行, 356 页, ISBN: 0-75675-774-6.

^b 密码技术政策的简明历史, 访问: <http://www.nap.edu/readingroom/books/crisis/E.txt>

^c 关于 Crypto AG 的丑事这里有很多信息. 最好从这篇文章开始: <http://www.aq.net/Kalliste/speccoll.htm>, Der Spiegel 的评论可以在这里找德国文版: <http://jya.com/cryptoag.htm>, 英文版: <http://jya.com/cryptoa2.htm> Baltimore Sun 的文章: <http://jya.com/nsa-sun.htm>

《The Covert Action Quarterly (季刊: 隐蔽活动)》: <http://mediafilter.org/caq/cryptogate/>。

^d Martin Hellman 表述他如何涉及加密技术: <http://www-ee.stanford.edu/~hellman/crypto.html>, 其它也在该网站。一个很重要的了解密码界战争的重要性: <http://www-ee.stanford.edu/hellman/breakthrough.html>。

有很大改观。加密技术仍然是做为双用技术来分类的, 包含军事化和民用化的意思, 密码界战争就这样结束了。

4 加密技术的基础

没有人可以在它人之上建立安全。

——Willa Cather

为了这本书的目的。我们将会只看计算机加密技术的部分。因为这才是最实际的介绍，人工加密技术的方法对我们并不是没有价值。同样的机械加密不再有价值。我们直接跳跃到什么是计算机加密技术，密码规则和系统才是我们要创造和使用的。

我们将自底向上的接近问题。首先我先描述一下加密系统和组件然后讨论怎样把他们放在一起。我们同样也看看现实世界 PGP 软件使用的规则和其它大型系统使用的规则。

4.1 基本部件

我们使用设计密码系统（就是加密系统）的里面有很多组件。

4.1.1 参量和变量

不像其它专业技术，加密技术最终是要人们参与在一起的过程，一些人们之间讨论的一直是我最感兴趣的。当我拟定概要和问题的時候，我们就开始设变量，这些变量本身就是一种语言。特别在三维几何数学中。我们使用变量字母 x 、 y 和 z 来表示； x 是横坐标线， y 是竖坐标线，和 z 是高。如果我们添加角度时，我们使用 θ 、 ϕ 、 ψ 来表示。数学的变量分之中，不仅使用罗马字母，还使用希腊字母。

在密码学中，我们设主变量是参数。我们不使用 x 、 y 或者是 α 和 β ，我们使用 Alice^a 和 Bob。这种表示在很多方面都更新了。抓住这个事物的要点是实际运用。Alice 和 Bob 比 A 和 B 表述更为精准，这就是为什么是这 2 个，只是传统语言的标准变量。第一次在 Ron Rivest 关于 RSA 加密算法系统的文章中提到，但是，我确信 Alice 就是 Lewis Carroll 写的那个 Alice，长大后 Carroll 喜欢这个，她也加入了密码学的研究，我们很少使用第三个参数，所以，这里没有什么其它标准，有时候就叫 Carol，有时候是 Charlie。我们偶尔还认为：“恩～Dave？Delia？Doris？”还有其它的参数的角色。

一般认为 Eve 是一个偷听者。就是这个学科的幽默之处。Mallory 总是被夹在中间，他和 Alice 说话时说自己是 Bob，和 Bob 说话时又称自己是 Alice。如果你在网上找找，你会发现一个下面的表。像变化多端的、详细的、易忘记的、不一致的 Victorian 算法也有这样含义：Bob 给了 Alice 一朵颜色特别的玫瑰。我只记起了 Alice、Bob 和 Eve。Mallory 也是很好记的。如果你想看看其它的，找一些双关用语的名字或首字母，就会找出在这个规则中每个人的角色。

表格 1 密码学中的人物角色

名字	属性
Alice	第一个参加者
Bob	第二个参加者
Carol	第三个参加者

^a John Gordon 著《The Alice and Bob After Dinner Speech（宴会演说后的 Alice 和 Bob）》，Zurich Seminar 提供，1984 年 4 月，<http://download.org/etext/alicebob.html>。

Dave	第四个参加者
Eve	窃听者
Mallory	恶意的主动攻击者
Trent	值得信赖的仲裁者
Walter	监察人，在某些协议中保护 Alice 和 Bob
Peggy	证明人
Victor	验证者

4.1.2 随机数字

我们使用随机数字来创建密钥的一部分，它还是我们的加密系统其它的部分。随机数字通常用在统计系统，但是密码学的随机数字和统计系统里面的随机数字有很大的不同。密码学中的随机数字必须是统计系统中的一种，最重要的是它也必须是无法猜测的。我们通常称这种性质叫熵^a (entropy)，计算机化的人们已经把它固定在了 bit (比特) 里面，它也是 2 的幂^b。一个简单计算 bit 的方法：10-bit 就是 1000 的一个因子。如果你听到 30-bit 的熵，猜一下大概是 $1000 \times 1000 \times 1000 (=1,000,000,000)$ 中的那个 1 的概念。

为什么在很多关于密码学的书中还不厌其烦地谈论随机数产生呢？随机数产生器已嵌入在大多数编译器中了，产生随机数仅仅是函数调用而已。为什么不用编译器的那种呢？不幸的是，那些随机数产生器对密码来说肯定是不安全的，甚至可能不是很随机的。它们中的大多数都是非常差的随机数。

随机序列产生器并不是随机的，因为它们不必要是完全随机的。像计算机游戏，大多数简单应用中只需几个随机数，几乎无人注意到它们，然而密码学对随机数产生器的性质是极其敏感的。用粗劣的随机数产生器，你会得到十分奇妙的相关和奇怪的结果。如果安全性依赖于你的随机数发生器，那么你得到的最后的东西就是这种奇妙的相关和结果。

随机数产生器不能产生随机序列，它甚至可能产生不了乍看起来像随机序列的数。当然，在计算机上不可能产生真正的随机数。Donald Knuth 引用 John Von Neumann 的原话：“任何人考虑用数学的方法产生随机数肯定是不合情理的”。

计算机确是怪兽：数据从一端进入，在内部经过完全可预测的操作（固定的算法），从另一端出来的却是不同的数据；把同一数据在不相干情况下输入进去，两次出来的数据是相同的；把同样的数据送入相同的两个计算机，它们的运算结果是相同的。计算机只能处理一个有限的状态数（无论如何是不是一个大数，都是有限的），并且输出状态总是用过去的输入（因为已经设计好定义域）和计算机当前状态的确定的函数。这就是说计算机中的随机序列产生器是周期性的，周期性的任何东西都是可预测的，就像是地球的周期、轨道可以预测一样。如果是可预测的，那么它就不可能是随机的。真正的随机序列产生器需要随机输入数据，计算机不可能提供这种随机输入。

4.1.2.1 伪随机序列

最好的计算机能产生的是伪随机序列产生器，什么意思呢？许多人试图形式化的方式定义它，但我不赞成，随机数序列是看起来是随机的序列，序列的周期应足够长，使得实际应用中相当长的有限序列都不是周期性的。就是说如果你需要十亿个随机数据，就不要选择仅在一万六千比特后就重复的序列产生器。这些相对短的非周期性的子序列应该尽可能和随机序列没有多少区别。

如果一序列产生器是伪随机的，它应该有下列的性质：

^a 中文读音 shang，读音为一声。科学技术上泛指某些物质系统状态的一种量度，某些物质系统状态可能出现的程度。

^b 通俗点例如：幂就是 a 的 b 次方，也是 b 个 a 相乘。a 和 b 都是数字。a^b

(1) 看起来是随机的

这表明它通过了我们所能找到的所有随机性统计检验。

人们在计算机上已经做了许多努力来产生好的伪随机序列, 学术文献中有很多讨论伪随机序列产生器和各种随机性检验的, 但所有这些产生器是周期的, 都不可能例外。然而周期大于 2^{256} 比特的随机数序列, 能够大量得到应用。

这里的关键问题还是那些奇妙的相关性和奇怪的结果。如果你以某种方式使用它们, 则每个随机数序列产生器都将产生这些相同的结果。这正是为什么密码分析者用它来对系统进行攻击的原因。

密码学意义上安全的伪随机序列:

密码的应用比其他大多数应用对伪随机序列的要求更严格。密码学的随机性并不仅仅意味着统计的随机性, 虽然它也是其中的一部分。密码学意义上安全的伪随机数, 还必须具有下面的性质:

(2) 它是不可预测的。

即使给出产生序列的算法或硬件和所有以前产生的比特流的全部知识, 也不可能通过计算来预测下一个随机比特应是什么。

密码学意义上安全的伪随机序列应该是不可压缩的……除非你知道密钥。密钥通常是用来设置产生器的初始状态的种子。

像任何密码算法一样, 密码学意义上安全的伪随机序列产生器也会受到攻击, 就好像加密算法有可能被破译一样, 破译密码学意义上安全的伪随机序列产生器也是可能的。密码学讲的都是关于如何使产生器抵抗攻击。

4.1.2.2 真正的随机序列

现在我们走进哲学家的领域, 真有随机数这样的东西吗? 随机序列是什么? 你怎么知道序列是随机的? “101110100”比“101010101”更随机吗? 量子力学告诉我们, 在现实世界中有真正的随机性。但是在计算机芯片和有限理想的确定世界中, 这种随机性还能保持吗?

暂且不说哲学。从我们的观点来说, 如果一个随机序列产生器具有下面的第三条性质, 它就是真正随机的:

(3) 它不能可靠地重复产生。

如果你用完全同样的输入对序列产生器操作两次(至少与人所能做到的最精确的一样), 你将得到两个不相关的随机序列。

满足这三条性质的产生器的输出对于 One-Time Pads、密钥的产生和任何其它需要真正随机数序列产生器的密码应用来说都是足够好的。难点在于确定真正的随机数。如果我用一个给定的密钥, 用 DES 算法重复地对一个字符串加密, 我将得到一个不错的、看起来是随机的输出。但你仍不可能知道它是否真的随机数, 除非你租用美国国家安全局的 DES 破译专家。

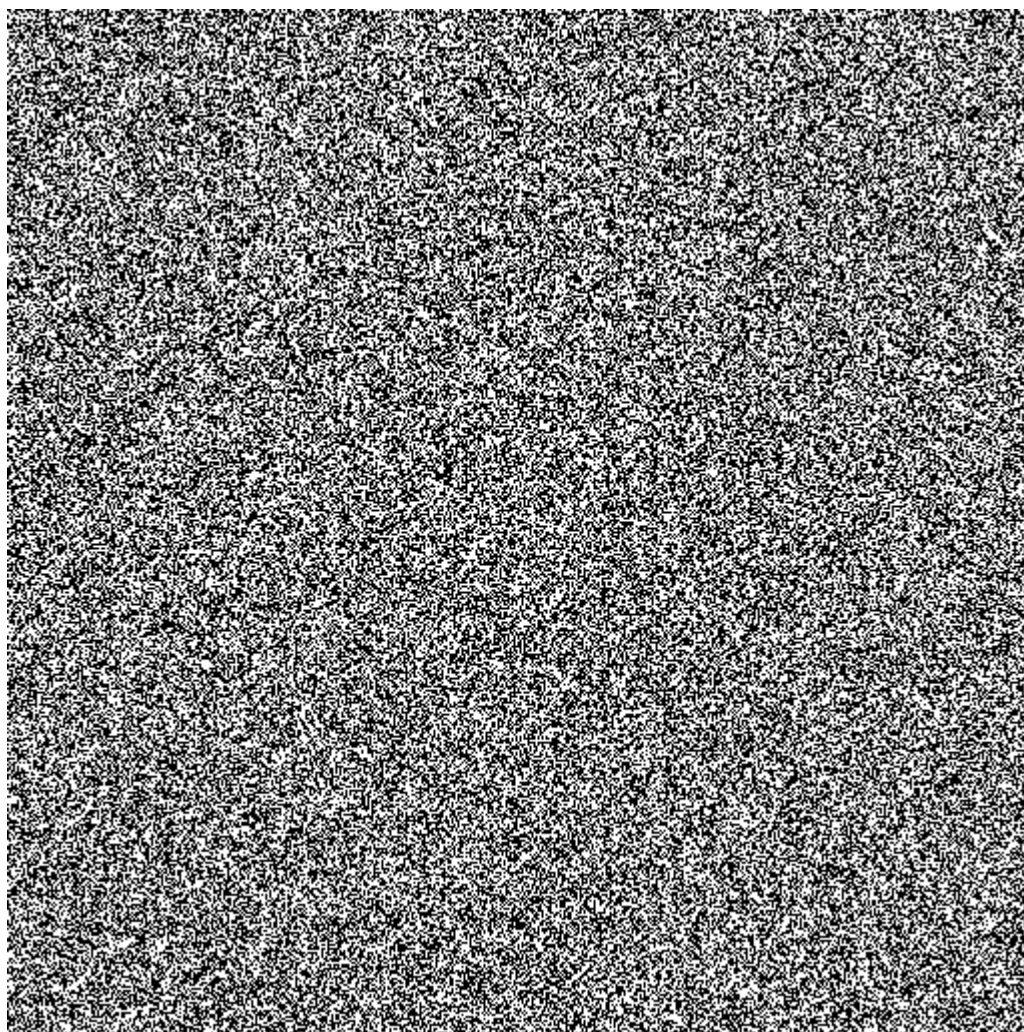
我们花费了大量的努力来为加密技术创造出随机数字。当我们使用 PGP 软件的时候, 这个随机数字会由 PGP 后台程序生成。也根据你使用计算机的方式。他们随机数生成程序会根据你输入时的按键的频率、速度和鼠标的移动来产生随机数字, 并确保没有其它任何办法猜出这个从那里来的。这个不是你实际按下的键盘

的那个按键的字母，也不是你移动的鼠标到什么地方而决定的，而是你的键盘如何按键、鼠标如何移动。如果你按住一个按键并重复，每次的结果也会同步改变。这个过程将会定位作用在底层，不管你是好好的输入的，还是乱七八糟的打的。它的目的就是根据你和别人意识不同，从而运动不同，进而随意产生一些数字也是不同的。

4.1.2.3 真随机数与伪随机数更加直观的区别^a

解释了很多，有些人还是不是很明白。下面有一些更加直观的例子来说明这个问题。我使用图片直观显示，可以让您对平时常用的伪随机数函数有更深入的理解。真随机数发生器（true random number generators, TRNGs），是利用不可预知的物理方式来产生的随机数，例如硬币：翻转，Random.org 就是利用大气噪音（atmospheric noise）来生成随机数的，而大气噪音是空气中的雷暴所产生的。伪随机数发生器（pseudo-random number generators, PRNGs），是计算机利用一定的算法来产生的。

下面把 Random.org 的真随机数发生器产生的数据绘制成位图图片给大家展示：（手机等移动设备请放大观看）

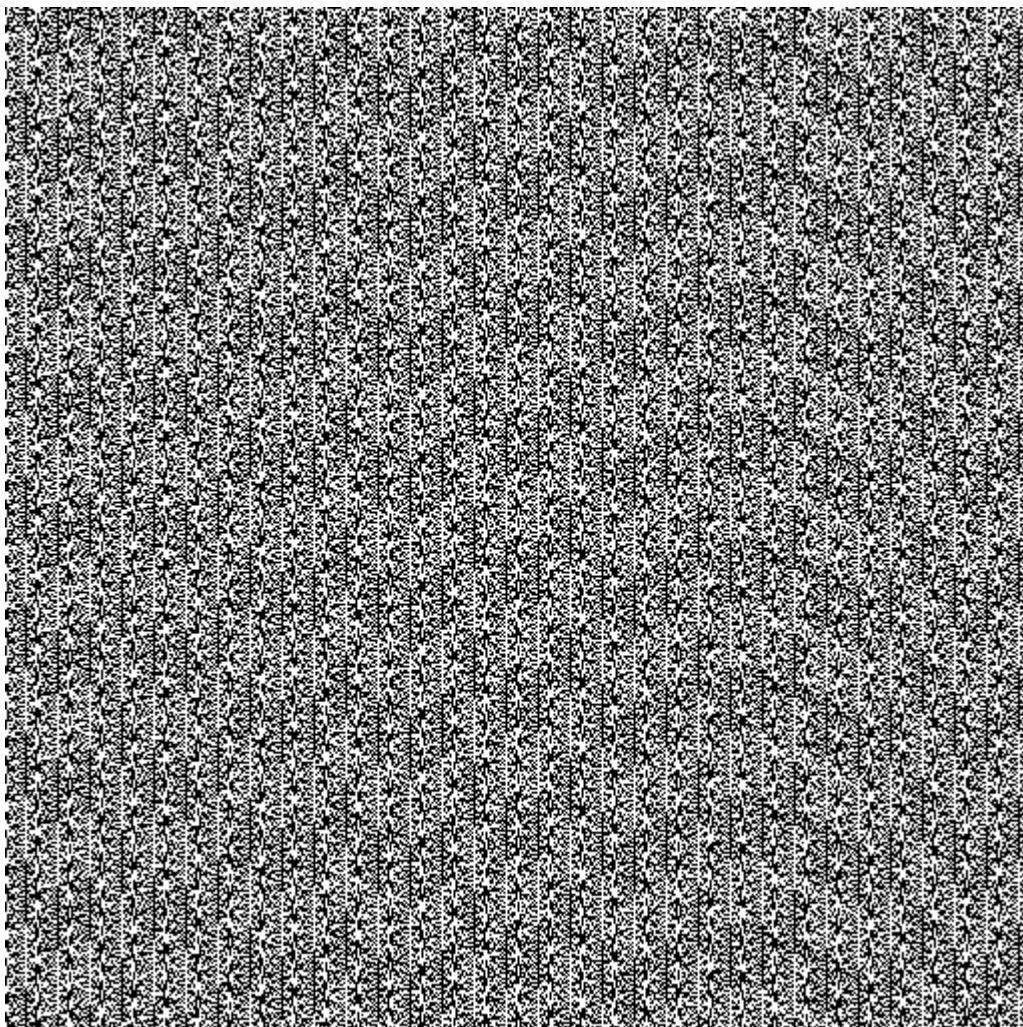


图表 1: Random.org 真随机数发生器

^a 真随机数与伪随机数 Pseudo-Random vs. True Random. 原文来自: <http://www.boallen.com/random-numbers.html>

看上去有点像电视机没有频道的时候都是雪花的感觉^a，好像看不出什么特别的规律。我们看下一张不一样的。

这张是 Windows 下 PHP 的 `rand()` 函数产生的数据绘制的随机图片。你似乎发现了什么东西！



图表 2: Windows 下 PHP 的 `rand()` 函数产生的图片

太明显了，上面的图片可以看到有竖线，而且宽度都差不多，很有规律，这是怎么回事呢？

实际上也并不是所有的伪随机数发生器效果都这么差的，只是恰好在 Windows 下的 PHP 的 `rand()` 函数是这样。如果是在 linux 下测试相同的代码的话，所产生的图片会看不出明显的条纹。在 Windows 下如果用 `mt_rand()` 函数替代 `rand()` 函数的话效果也会好很多。这是由于 `mt_rand()` 用了 Mersenne Twister (马其塞旋转^b) 的算法来产生随机数。PHP 的文档^c还声称：`mt_rand()` 产生随机数值的平均速度比 `libc` 提供的 `rand()` 快四倍。

Linux 操作系统提供本质上随机（或者至少具有强烈随机性的部件）的库数据。这些数据通常来自于设备驱动程序。例如，键盘驱动程序收集两个按键之间时间的信息，然后将这个环境噪声填入随机数发生器库。

随机数据存储在熵池中，它在每次有新数据进入时进行“搅拌”。这种搅拌实际上是一种数学转换，帮助提高随机性。当数据添加到熵池中后，系统估计获得了多少真正随机位。

测定随机性的总量是很重要的。问题是某些量往往比起先考虑时看上去的随机性小。例如，添加表示自

^a 你也可以通过访问这里来自己产生图片：<http://www.random.org/bitmaps/>

^b 马其塞旋转请访问：http://en.wikipedia.org/wiki/Mersenne_twister

^c PHP 文档细节：<http://cod.ifies.com/2008/05/php-rand01-on-windows-openssl-rand-on.html>

从上次按键盘以来秒数的 32 位数实际上并没有提供新的 32 位随机信息, 因为大多数按键都是很接近的。

从 `/dev/random` 中读取字节后, 熵池就使用 MD5 算法进行密码散列, 该散列中的各个字节被转换成数字, 然后返回。

如果在熵池中沒有可用的随机性位, `/dev/random` 在池中有足够的随机性之前等待, 不返回结果。这意味着如果使用 `/dev/random` 来产生许多随机数, 就会发现它太慢了, 不够实用。我们经常看到 `/dev/random` 生成几十字节的数据, 然后在许多秒内都不产生结果。

幸运的是有熵池的另一个接口可以绕过这个限制: `/dev/urandom`。即使熵池中沒有随机性可用, 这个替代设备也总是返回随机数。如果您取出许多数而不给熵池足够的时间重新充满, 就再也不能获得各种来源的合用熵的好处了; 但您仍可以从熵池的 MD5 散列中获得非常好的随机数! 这种方式的问题是, 如果有任何人破解了 MD5 算法, 并通过查看输出了解到有关散列输入的信息, 那么您的数就会立刻变得完全可预料。大多数专家都认为这种分析从计算角度来讲是不可行的。然而, 仍然认为 `/dev/urandom` 比 `/dev/random` 要“不安全一些”(并通常值得怀疑)。

Windows 下沒有 `/dev/random` 可用, 但可以使用微软的“`capicom.dll`”所提供的 `CAPICOM.Utilities` 对象。

4.1.3 密钥

密钥是密码学中的秘密。密码学的安全是依据用来创建、使用、保护、删除时的密钥决定的。在 PGP 软件和其它计算机加密系统中, 这个密钥就是最高级的数字, 当然你也可以选择字母什么的。有三个主要的方式产生一个密钥:

1. 原始密钥 (raw keys) 是从随机数生成器中生成的比特字符串, 有很多我们使用的就是, 事实上, 它就是原始密钥。
2. 推导关键词 (Derived keys) 是从其它的一些东西里面产生的, 比如, 当我们用字符密钥加密的时候, 我们并不是直接使用字符密钥, 而是从你的字符密码中导出一些实际关键字 (Actual key)。
3. 构造关键字 (Structured keys) 是从一些随机数字中产生出推导关键词的一种形式。比如 RSA 公钥需要一个数学构造, 我们用原始随机数字比特流找出最符合数学构造的数字。

是不是有点晕了? 无论它们怎么生成, 密钥就是加密技术安全的关键。只要他们都被保密了, 你就拥有了系统所提供的安全。如果你看不住你的密钥, 你也就没有任何安全性可言了。密钥的大小和下面的算法, 以及你所依赖安全性的要求有很大的关系。

4.1.4 算法

编码就是算法, 那个我们用来加密和解密的处方——公式。它也是密码系统的一部分, 就像在 PGP 软件中使用的标准制式, 已经被所有人知道了。我们很想设计一个安全算法, 但是你会不信任一个安全算法。它让你非常容易的设计一个算法, 而不用担心会被破解。要做到一个算法别人不能破解这是很难的。那也很难保证算法不泄露。特别是你在计算机上运行时。很多人喜欢逆向工程系统 (reverse-engineer systems), 你把算法隐藏在里面, 他们用这个就可以找出来。许多广泛用途的算法都是保密的, 但是可以说现在它们之中没有一个是秘密的。最好还是把精力用在一些其它的事情上。

我们前面讨论时, 有 2 个算法: 公钥密钥算法 (public-key ciphers) 和对称密钥算法 (symmetric ciphers)。为什么我们要把这 2 个类型实际应用算法分开说? 如果公钥加密算法有超过对称密钥算法的很多优势, 那我

们为什么还为对称密钥算法浪费口舌？

原因纯粹是为了实际用途。公钥加密使用的密钥通常比对称密钥大好几倍，但是也慢了几倍。公钥加密算法比对称密钥算法至少慢了 1 万倍，那就是我们使用的 2 种不只是速度操作的算法：公钥加密发送对称密钥，对称密钥有更好的灵活性和速度。

所有算法有 2 个大小，一个是密钥的大小，一个是算法一次加密数据的大小，我们称为数据的区块大小（block size）。你下面将会同时介绍它们。

4.1.4.1 区块大小

了解公钥算法中的区块大小和密钥大小是很容易的。用这个密钥长度的大小加密数据的一个同样大小的数据区块。所以如果你有 2048-bit 的公钥，它每次就会加密 2048-bit（256 字节^a）的数据。

而对称密钥算法有许多选项。算法的区块大小与密钥的大小毫无关系。今天通常使用算法是 64-bit 或 128-bit（8 或 16 字节）的数据。AES 算法使用的区块大小是 128-bit，这也作为 AES 竞争标准的一部分，比如说：Twofish。老一代的包括 Triple-DES、CAST 和 IDEA 的算法使用 64-bit 的区块。这也有系列我们称作流密码（stream ciphers）的算法，操作单个字符，甚至 1-Bit。在 SSL^b系统中使用的 RC4 算法就是一种流密码算法，事实上这个很少使用。因为它单次加密单组数据产生的一些有趣的安全特征，流密码可以认为是一个和区块算法不同的单独算法。

直到最近几年，所有的密码算法使用流密码或者其它的密码。Caesar 密码是流密码，Enigma 也是流密码，那个时期的密码都是这样的。同样公钥加密让我们想出老方法的一种新形式，区块算法也让我们设计了老加密方法的新形式。有趣的是，同样的事情发生在了计算机密码学中。计算机使区块加密变的简单。所以我们要区分出区块算法和流密码算法。

你可能会想为什么你想一次加密一个区块而不是一个字符？在流密码里面区块密码有一个寻址问题。流密码出现的问题是密码破译学家可以用这个猜出明文的一部分信息。它帮助了解密，也帮助了解密。Napoleon 的密码为什么被破解就是这个原因，Enigma 也是这样破解的。同样这也就是几年前 WiFi 网络中的“WEP (Wired Equivalent Privacy)” 密码被破解的原因^c。改变的越多，淘汰的也就越多。

明文的数据你可以猜一下，或者称这种行为叫 Crib，这就返回了人工加密的时代。Crib 这个单词有多个英文意思，在这里的是在尝试的时候进行欺骗，并不是放小孩的婴儿床的意思。一个基本的常识：今天的已经破解的明文，并不都是用欺骗法来破解的，这个词很直接。已知明文的攻击中，攻击者认为它是创建明文和密文的过程中有很有用的攻击方法。

区块算法是一个可以找到明文中危险地址的方法，因为它们只操作明文一个区块。所有混合明文中 16 字节的比特数据是很难被猜测欺骗^d的。

但是，如果我们只加密区块这样是还不够的。我们要把密码破译人员的工作变的更难！破译人员仍然可以知道相同的密文来自相同的明文。所以就要做 2 件事来阻挠他们分析数据得过程。首先，我们把一个区块的输出和下一个区块的输入混合。这步称为链接（Chaining），有很多方式进行链接。链接使的区块连接变的不同，或者更加精密。如果密文变成一样得了，那很幸运，能碰到一样的并不多见，那并不是原文的一个明显的结构。其次，我们使用随机数据来加密一个区块的数据。这个在计算机中叫一个初始化向量（initialization

^a 关于 Bit（比特位）到 byte（字节）的转换：比特位除以 8 就是字节。 $2048 \div 8 = 256$

^b SSL (Secure Socket Layer)，SSL 协议位于 TCP/IP 协议与各种应用层协议之间，为数据通讯提供安全支持。SSL 协议可分为两层：SSL 记录协议（SSL Record Protocol）：它建立在可靠的传输协议（如 TCP）之上，为高层协议提供数据封装、压缩、加密等基本功能的支持。SSL 握手协议（SSL Handshake Protocol）：它建立在 SSL 记录协议之上，用于在实际的数据传输开始前，通讯双方进行身份认证、协商加密算法、交换加密密钥等。访问：<http://baike.baidu.com/view/16147.htm>

^c 这并不是唯一一个破解 WEP 的方法，WEP 是一个研究的好材料，因为它就有很多破解的办法，和它如何简单的避免攻击。所有的问题来自应用中使用了流密码，应该使用区块加密。WEP 是一个很好的例子来解释一个很强的算法怎么会使系统变弱的。

^d 同样有其它的办法。一个最大的问题是不再重新使用密钥。计算机和公钥加密学的发展使它变的简单。

vector, IV)^a。IV 并不是秘密。它负责转换明文, 并不需要像一个随机密钥那么严格。另一方面, 它的合理的任意性与计数器的一些东西是相反的。用 IV 链接区块并加密输出数据来避免被猜测知道原文(前提是这个算法设计的很好; 现在有很多算法都是被明文数据攻击破解)。这 2 个添加进加密数据就可以使明文没有什么可以模仿的了。甚至可以多重加密(假设你改变了至少一层的密码和 IV)。

即使这样, 很多密码相同的算法也没有使用链接的技术。比如: 加密最基本的应用形式之一: 硬盘加密。这种算法比使用链接的算法强度要弱。甚至是使用了强度很高的加密算法, 它和每个数据块的联系也是不多的, 它是一个小问题, 仅仅是一个小问题。所有的 PGP 软件都使用了很多种链接的设计方法。在 PGP 软件的网络共享软件中, 我们使用了一种新形式的链接方法叫 EME 模式^b。EME 是 AES 下的“可变”模式。无论里面有什么数据, 它都可以使我们的磁盘区块可以正确的读取。

在 AES 竞争前, 几乎所有的算法都使用 64-bit 的区块大小。当 NIST 开始进行 AES 竞争, 它要求新算法的区块大小是 128-bit。使用 128-bit 是包括了链接的, 任何 2 个密文的区块都一样大。甚至密码破译时每个区块有一样的机率来拖延密码攻击的时间。2 个区块有相同密文的概率问题叫做 The Birthday Problem (生日问题)。

如果一个房间里 2 个人有相同的生日的机率有多大? 如果你仔细看人的生日数字, 有 50% 的机率相同。可能是生日数目的平方根。结果是 $\sqrt{365}$ 也就是约等于 23。在 64-bit 区块加密算法中, 你加密 2 个区块使用相同的机率的值和你加密 $\sqrt{2^{64}}$ 大小数据的相等, 算出来是 2^{32} (约 40 亿) 的区块, 也就是 320 亿字节的数据^c。对于用一个密钥和 IV 加密来说确实是很大的数据, 但是也不是你想象的那么大。2006 年 DVD 有约 47 亿字节的数据。在超高速的网络可以在相对很短的时间内传输大量数据。这也就意味着在高级规则中, 有必要关心阻止一个密钥变成另外一个密钥。

在另一方面, 如果你把数据区块大小变成 128-bit, 当你加密 $16 \times \sqrt{2^{128}}$ 数据的时候会遇到一次生日问题攻击^d的碰撞。大约是 295,147,905,179,352,825,856 字节 (256Eb)。这个大的文件我们现在几乎遇不到, 概率已经被暂时移除了, 我们也不必太担心。

也就是说, 在加密到一定大小的数据后, 一定会出现一个漏洞问题, 我们把区块大小改变后, 把明文数据增加到足够大, 或者说是我们现在不可能用到的大小, 这个问题也就在以后有必要使用这个大小时候会出现。那时候我们还可以继续扩大。比如再从 128-bit 变到 256-bit, 那是以后的事情。

4.1.4.2 公钥加密算法的根本

就像我以前说的, 公钥加密依据的是数学理论, 而很容易来回运作。今天有两个我们使用的实用法则: 因子分解法则 (The factoring family) 和对数法则 (the logarithm family)。

4.1.4.3 因子分解法则

因子分解法则^e中使用了两个质数 p 和 q , 把它们乘起来是 $n = p \times q$, 使用 n 来做一些明文的一些数学计算, 在我谈论 hash (哈希) 算法的时候会拿密文来详细介绍生日问题攻击。如果有人看到了密文, 而且如果你知道 p 和 q , 你可以很容易的算回去而获得原文。难就难在你只知道 n 。当然只有密钥的所有者知道 p 和

^a 这个小知识在《解析几何》中有数学介绍, 这里的就不要那么深入了, 意思就是所有区块的数据都要经过一个转换, 或者叫经过一个变换, 这个变换的依据是根据一个依据——初始化向量来做的。

^b Shai Halevi 和 Phillip Rogaway 论文《A Parallelizable Enciphering Mode (一种可平行的加密模式)》, 访问: <http://eprint.iacr.org/2003/147>

^c $2^{32} \times 8 = 34359738368 \text{ byte} = 32 \text{ Gb}$ 的数据, $k/M/G/T/P/E/Z/Y$ 相邻之间转换进率是 1000 或 1024。B 是 byte, 字节的意思。一般生活中是 1000, 计算机科学里面是 1024, 存储设备制造商用 1000 而不是 1024。所以会比看到这么大的数字小一点!

^d 在讨论 hash 算法的时候会介绍生日攻击 (birthday attacks) 的详情。

^e 在密码学中, RSA 系统的安全性取决于分解一个位数很大的数字的计算难度, 比如一个 200 位的数, 由两个各为 100 位左右的质数构成, 除非该质数是已知的, 否则, 即使最快的计算机, 利用目前的技术, 要分解这样一个数也需几十亿年的时间。

q 是什么, 这样系统就很安全。当然, 2 个基本质数, 最好大小相似一些。如果你要一个 1000-bit 的 n , 你可以让 p 和 q 每个是 500-bit 的大小。

现在有 2 个因子分解法则: RSA (名字是它的设计者 Ron Rivest、Adi Shamir 和 Len Adleman 的名字命名的:), 和 Rabin (作者 Michael Rabin)。RSA 是今天广泛使用的公钥加密系统。Rabin 仅次于 RSA, 同样使用了平方和平方根。

Rabin 比 RSA 有更多的数学理论为基础。我们认为破解 RSA 和因子分解一样难。因为缺少条件也没有理由去想象。另一方面, 已经证明出了破解 RSA 的计算量和因子分解是一样难的。可是当攻击者说服密钥的拥有者加密一些不是密文的数据时, 但是如果选的只是字符串, 这里仍然可以选择密文攻击。这个垃圾信息的解密过程可以泄露出质数 p 和 q 。很多年过去了, 这个被认为 Rabin 的一个致命缺陷。最近, 我们又对这个攻击进行了大量的试验。Rabin 是一个相对 RSA 算法外可选的, 仍然是网络资源中值得选择的算法^a。Rabin 作为一个好的密码算法, 仍然被大量使用, 因为它在技术和密码学资源上没有 RSA^b引人注目, 也没有超过 RSA 的实际优势。

例如, 在 PGP 软件算法中可以选择使用 Rabin 加密算法, 但是只有很少的原因时提供给用户使用 Rabin 算法。只有在其它人的软件支持 Rabin 算法时才会考虑加密它们使用这个算法。密钥和 RSA 使用的并没有什么不一样的或比它快的地方。因此, 我们认为, 简单的期望比任何都重要。

4.1.4.4 对数法则

基于对数法则算法的成员有很多, 最基本的算法是 Diffie-Hellman, 这个算法的名字是它的设计者 Whitfield Diffie 和 Martin Hellman 的名字组合而成的。Diffie-Hellman 的安全性和公式 $m = g^x$ 难度相当。你很难找出 x 是多少。Diffie-Hellman 算法称: 交换算法密钥比公钥加密算法更高明。Diffie-Hellman 让我们想象这个密钥, 它是个数字, 当然, 当我们使用密钥的时候注意到我们使用的是数学方法计算其他人的密钥的。这个过程叫做 ephemeral^c, 它的参数中没有一个是常量。

Diffie-Hellman 在 Elgamal 的变化后成为一个公开密钥算法, 名字是它的开发者 Taher Elgamal^d的名字命名的。Elgamal 加密算法基于 Diffie-Hellman 算法, 以 RSA 密钥的产生方式来产生它的静态密码, 你可以用这个来签名并加密。Diffie-Hellman 是 PGP 软件中叫 Elgamal 的密钥 (DSS 签名就是 Elgamal 签名的一种, 我们一会要详细介绍)。

所有的加密都使用了模数算术来计算。模数算术听上去很难, 其实比字面上看起来的要简单的多。如果你使用过计数器或汽车的里程表, 你就用过了模数算术。它不过是数学的一点点知识, 比如: 12 小时制计时法中现在 11 点, 你有一个 3 小时的会议, 你会在 2 点时候出来, 注意不是 14 点^e。相似的, 你不能从里程表中读出这个破机器是开了 200 万英里还是 100 万英里^f。在表的例子里面, 我们计算数字用 12 (或 24) 的最大量程的数字, 而对于里程表他只有 100 万的最大量程。超过了就从 0 开始。

在这个公钥加密系统中, 我们用质数^g作为量程基础, 我们重复循环操作的时候使用质数。不会再有循环。

^a 网络影响 (The network effect) 传真影响 (fax effect) 是电话、传真机、和其它加密技术使用的设备的总称。

^b Neal R. Wagner 著《The Laws of Cryptography: Rabin's Version of RSA (加密规则: Rabin 的 RSA 标准)》, <http://www.cs.utsa.edu/~wagner/laws/Rabin.html>, Wagner 称这个网站“过时了”但是仍然很有用。他写的书的草案可以在这里找到: <http://www.cs.utsa.edu/~wagner/lawsbookcolor/laws.pdf>

^c 英文含义: adj. 生命短促的, 短暂的, 瞬息的; n. 短命的东西。在这里应该是使...结束的方法。

^d 如果你研究过公开密钥算法, 你会看到 Taher 的姓是“El Gamal” (2 个单词) 或“ElGamal”。他使用这个在其它上面拼写, 但是不久后使用“Elgamal”是因为他更容易记忆, 这也影响他给别人发邮件用“Taher L. Gamal.”。

^e 或者 24 小时制计时法中 3 小时的会议开始于 23: 00, 02: 00 结束, 而不是 26: 00。

^f 如果里程表一圈最大是 100 万, 跑完 100 万会归零; 跑 200 万, 也就是 2 圈, 还是 0。这样从数字上无法判断到底跑了多少。

^g 只能被本身和 1 整除的数是质数。质数又叫素数, 也叫梅森素数。例如, 2, 3, 5, 7 等是质数, 1, 4 ($=2 \times 2=1 \times 4$), 6 ($=2 \times 3=1 \times 6$), 8 ($=2 \times 4=1 \times 8$), 9 ($=3 \times 3=1 \times 9$), 10 ($=1 \times 10=2 \times 5$) 24 ($=1 \times 24=2 \times 12=3 \times 8=4 \times 6$) 等不是质数。质数的数目是无穷的。目前已经找到的最大质数是: $2^{43112609} - 1$ 。

这个最简单的例子: 12 小时制的表, 如果你重复加 4, 你就得到了一个循环。其中 12 小时制有 4 个循环: [12, 4, 8]、[1, 5, 9]、[2, 6, 10]和 [3, 7, 11]。用质数作为基础数字, 将不会发生循环, 不信你可以试试。为什么我们这么关心循环呢? 因为它可以让密码破解者更容易破解。

在数学中就必须看的更远, 你会发现(例子^a) $g^x \bmod p$ 就是 g^x , 而如果它不是质数, 那么就可以用一个小的数字来替代计算。全都用质数 p 进行 \bmod 运算, 这样可以让我们摆脱看起来的混乱^b。

模数算术, 特别是使用质数为底, 还有另外一些有趣的数学性质: 你可以用整数进行特殊的运算。 $\sqrt[5]{5}$ 不是整数, 但是 $\sqrt[5]{5} \bmod 11 = 1$ 。另外模数算术总是保持固定大小数字不变。如果 p 是 1000-bit 的数字, 任何数学计算都要使用 1000-bit 的数字。这个性质说明在数学中整系数的计算和浮点型中的一样。除此之外, 加法和减法也一样。我们可以幂运算和根运算, 或者其它一切的算法。我们称这种形式为有限域^c, 一个有限数的数字允许我们进行一些无限的数字运算。

二次方曲线、椭圆曲线等的一些特殊性质, 有很多公钥算法使用他们其中的曲线来替换一些直线型的数。通常情况下, 如果没有特殊要求就不使用。它们有一些特殊性质, 但是实际上看上去不是这样的, 也可能是工程学的领域里面的一些特殊性质。比如: 你可以使用二次方曲线来使用 RSA 算法, 密钥也不小, 计算机也没有偷懒, 数学计算上也没有用很多内存, 但是还是出问题, 为什么呢?

有一个例外, Diffie-Hellman 是基于椭圆曲线算法^d。首先的, 密钥比较小。其次, 计算机计算也很快。这更实际, 也更吸引我们。美国政府表示下一个 15 年到 20 年他们需要一个椭圆曲线算法的移位算法, 因为椭圆曲线算法下的密钥更小, 计算更快。

4.1.4.5 密钥的长度

算法也有很多大小不同的密钥。公钥算法和对称算法中不同大小的密钥之间有很多不一样的地方。你注意到公钥加密的密钥来自数千 bit 的大小, 对称算法也是这样的, 有什么不同呢? 它们之间有关系吗?

对于对称算法来说 128-bit 足够了。事实上, 越大越好。这里有一个破解单独的 128-bit 密钥的解释。

想象一下地球被绿色的草地覆盖。每一颗草代表一台可以每秒计算十亿密码的计算机, 而很多的草构成了一个群, 每个计算机集群破解一个 128-bit 密钥需要 1 千年。

这个描述是足以让人晕倒的指数函数的幂运算的结果。假设一个破解者照上面所说的, 需要用计算机装满一个空地, 不考虑有没有足够的电力供给计算机群和冷却计算机群^e。我不认为这个密钥可以在 1 千年内破解。可能是我们担心的过度了, 但是计算机邮件的更新, 处理速度大幅提高后, 这也不一定。^f

为什么我们为 256-bit 密钥的 AES 和其它一些算法而费心? 最好的回答: 为一些可能被发明的科学虚构^g技术设置一些障碍。当 DES 开发的时候, 打算只使用 5 年, 最后用了 20 年。AES 设计的强度要求至少可以持续使用 50 年。

更大的密钥需要公钥加密技术来保持与对称算法平衡。就像木桶原理: 算法强度和它最弱的那项的强度相同。一个密码系统的强度和它的算法的强度相同。所有你使用公钥密码的长度来匹配对称加密的密钥, 你会获得更大的密钥的强度, 表格 2 给出了 NIST 推荐的多种加密系统的密钥大小平衡。你可以看到推荐的

^a Mod 运算: 是两个整数相除取余数, 例如: $7 \bmod 3 = 1$, 即 7 除以 3 等于 2 且余 1, 你会发现这个余数一直比除数小。

^b 很有趣, 老 Caesar 算法, 移位 (the shift-by-N) 算法 也是模数算术的一种, 每个字符都使用 $c = p + k(\bmod 26)$ 替换, 密文的字符 c 是来自明文 p 。

^c 域的概念请参阅《近世代数》中的域的章节。

^d Certicom 在商业化椭圆曲线加密技术中有大量的经验, 这有一个网站说明:

http://www.certicom.com/index.php?action=ecc_tutorial_home, 包含他们在椭圆曲线算法的数学指导。

^e 计算机集群需要的电力很多, 而且运算时电子线路发热量很大, 如果不散热, 可能减少电子元件的寿命, 或者直接烧毁。这也就是为什么普通的计算机机箱内部都有散热器的缘故。

^f 目前运算破解的工具软件主要是利用 CPU (包括一些多核处理器), 有些算法也支持利用显卡 GPU 运算来破解密钥。

^g 我不认为说它们是科学虚构的技术是轻蔑语。我还可以想起, 登月也是科学虚构的, 但没有向往过去。有些技术虽然没有出现在人们眼中, 但是不能说没有人拥有这个技术。我们只能虚构一些“高科技”, 并且做好抵御他们的手段。

3,000-bit RSA 或 DSA 密钥和 128-bit 对称加密的强度相同，你也会发现椭圆形算法，我们只要 256-bit 公钥。

表格 2: NIST 公布的密码安全平衡

对比加密强度						
算法名称	比特位大小					
对称算法	56	80	112	128	192	256
Hash 算法	160		256		384	512
MAC	64	60	256		384	512
RSA/DSA	512	1024	2048	3072	7680	15360
椭圆曲线算法	160		224	256	384	512

很值得注意的是 256-bit 对称密码的强度一样的，有 15,000-bit RSA 或 DSA 密钥强度相当，但是曲线密钥只要 512-bit。这就是为什么美国政府开始推崇椭圆曲线算法的原因。如果你想使用采用大密钥的 AES 算法加密安全平衡，你需要大的公钥，使用整数的生成算法，这些密钥完美的变大了。

4.1.4.6 密钥需要多长？

答案并不是一个固定和准确的，它要视情况而定。为了断定你需要多高的安全性，你应该问自己一些问题：

1. 你的数据价值有多少？
2. 你的数据要多长的安全期？
3. 攻击者的计算资源情况怎样？

一个顾客清单也许只值 600RMB；一起令人痛苦的离婚案件的财政数据也许 30 万 RMB；一个大公司的广告和市场数据应该值 3 百万 RMB；而一个数据取款系统的主密钥价值可能会超过亿元。

在市场经济贸易的世界里，保密只需要几分钟而已。在报纸行业，没准今天你的秘密将是明天的头条标题。产品研发信息或许需要保密一到两年。根据美国法律，美国人口普查数据要保密 100 年。

公司的贸易秘密是那些竞争公司最感兴趣的；对敌军来说军事秘密是值得感兴趣的。你可以这样说明你的安全需求。例如，

密钥长度必须足够长，以使破译者花费一亿美元在一年中破译系统的可能性也不超过 $\frac{1}{2^{32}}$ ，甚至假设破解技术在此期间每年有 30% 的增长速度。

下面给出了对各种信息的安全需要的估计。

信息类型	时间	最小的密钥长度要求
军事下达信息	几分钟 到 几小时	56-64-Bit
产品发布、合并、利率	几天 到 几周	64-Bit
贸易秘密	几十年	112-Bit
氢弹秘密	>40 年	128-Bit
间谍的身份	>50 年	128-Bit
个人隐私	>50 年	128-Bit
外交秘密	>65 年	>128-Bit

美国普查信息	100 年	>128-Bit
--------	-------	----------

但是未来的计算机处理能力是难以估计的^a, 但这里有一个比较保守的经验方法: 计算机设备的性价比每 18 个月翻一番或以每 5 年十倍的速度增长^b。这样, 在 50 年内最快的计算机将比今天快 10^{10} 倍^c, 且这些数字仅对于普通用途的计算机而言; 谁能知道某种特制的密码破译机在下一个 50 年内如何发展呢?

假定一种加密算法能用 30 年, 你就能对它是多么安全有一个概念。现在设计的一种算法也许直到 2000 年才会普遍使用, 也许在 2025 年仍然将会运用它来为那些需保密至 2075 年或更晚的信息加密。

换句话说, 如果你现在使用的密钥的长度比你查到的密钥长度大或者是大得多的话, 你不用担心会有什么令人吃惊的技术能够危害到你的信息安全。

4.1.4.7 不可破译的算法强度有多大?

PGP 软件目前还没有集成椭圆曲线算法, 也没有提供 15,000-bit 的密钥; 它只有 4096-bit 的密钥。你很在意密钥的大小吗?

首先, 了解到 128-bit 对称密钥已经足够对应未来的短时间。最新统计^d数据展示究竟多长的密钥可以使用到 2050 年, 你其实应该抛弃 109-bit 对称密钥、4047-bit RSA/DSA 公钥和 206-bit 椭圆曲线公钥, 因为只要花费 44.4 亿美元就可以制造一个一天可以破解一个密钥的机器。这个统计也推测出 DES 的使用期限是 1982 年, 这还是一个保守的估算。你害怕吗? 在理论上, 所有的密码都可以被破解, 只是速度问题而已!

隐含的提出了一个新问题: 多少比特尺度的密钥才够? “你想要多长的密钥来保证数据的安全?” 如果你打算发一个信息给你的股票经理人, 你不需要 128-bit 那么长的密钥。怎么说它都适合你。你可以直接给他说就可以了。这么麻烦的做法谁都不愿意。

AES 拥有 256-bit 的密钥而且速度不是很慢, 这是令人非常高兴的模式, AES-256 比 AES-128 慢了约 20%, 而且在替换 AES 一些东西后就比它慢了。因此, 那就是人们使用 AES-256 的原因是完全出于市场推广的考虑, 并非出于安全的考虑^e。

我们一直要权衡这些因素。所以没有必要使用 AES-256。它确实是个好算法, 所有人认为应该使用 256-bit 密钥的安全性。只不过比 AES-128 慢一点。因此, 推荐使用这个。

现在一些年轻人使用 256-bit 密钥。这也就意味着你需要多花费 20% 的计算机性能损耗, 如果你想使用 AES-128 而别人要求你使用 AES-256, 不是很令人郁闷的吗?

也许也没有滥用别人的算法那么坏。AES 之前, 也有很多设计的算法使用可变长度的密钥, 很多已经确定使用 128-bit 密钥, 而且支持更大的。直到 AES 的要求是设计大密钥为主要目的时, 这也不是为了以后而做的, AES 使你觉的很不爽。AES 参选中的其它算法也是很不错的选择。PGP 软件支持的其中之一: Twofish^f, 使用 256-bit 的密钥。它真是个不错的算法。

但我们考虑使用一种公钥的大小时, 其它的一些影响我们也会考虑。最主要的是速度。今天, 几乎都认

^a 在我翻译书的时候 (2009.8) 我已经接触到了拥有 4 核计算核心的桌面处理器。

^b 摩尔定律: 计算机性能 18 个月性能翻一倍。或者, 计算机价格 18 个月贬值一半。

^c 计算过程: $2^{\frac{50 \times 12}{18}} = 2^{\frac{100}{3}} = 108226394096809 \approx 10^{10}$

^d Arjen K. Lenstra 和 Eric R. Verheul 著《Selecting Cryptographic Key Sizes (如何选择密钥的大小)》,《Journal of Cryptology Volume 14, Number 4, (加密期刊: 第 4 卷第 4 本)》 255 到 293 页, 2001 年。也可以访问: <http://www.keylength.com/>, 为参数的设置和最符合你的密钥大小的假设做了相互说明。

^e 举个例子: 中国产的部分抽屉锁, 锁的用料越来越薄, 质量越来越差, 在外表上面却做了很多文章, 厂家都在价格竞争。

^f Bruce Schneier、John Kelsey、Doug Whiting、David Wagner、Chris Hall 和 Niels Ferguson 著《Twofish: A New Block Cipher (一个新算法)》, <http://www.schneier.com/twofish.html>, 注意他们描述的是 128-bit 的算法, 想了解 192 和 256-bit 的模式。看 128-bit 的就够了。

为 1024-bit 的密钥对公钥加密算法来说太小了。(报告提示可以参阅 2001 中期的技术) 如果你使用更长的密钥, 在你使用一些小设备的时候就要咬牙切齿了。

我有一个运行 PGP 软件的黑莓(BlackBerry)手机, 如果使用一个 3,000-bit 的密钥就需要一段时间去运行。这个破玩意使用的是 40MHz 的 80386 处理器。返回到 1991 年, 当 PGP 开始设计的时候, 我们就是针对台式电脑设计的, 没有去考虑一些移动设备的性能。如果你的手机使用 4096-bit 这么大的密钥, 你最好还是改变到 2048-bit 的大小, 否则你会对等待加密的时间感到厌倦。你不会注意到 5 年前的笔记本电脑加密速度和现在的电脑的加密的性能上有什么不一样的地方, 哪怕是很旧的电脑, 加密的时候速度都不会很慢。下面有一个加密算法的加密速度的一个测试, 也许你看看这个就明白了。

硬件测试环境:

中央处理器: Intel Pentium4 2G Hz

内存: 512MB DDR400

硬盘: 西部数据 IDE 接口 7200 转硬盘, 分区格式为: FAT32

测试软件: Benchmark

表格 3: 加密基准速度测试 (按平均速度递减排序)

算法	加密速度 (MB/S, 兆每秒)	解密速度 (MB/S)	平均速度 (MB/S)
AES	33.9	28.6	31.2
Twofish	31.0	29.6	30.3
Serpent	19.3	19.5	19.4
AES-Twofish	17.0	14.1	15.6
Serpent-AES	12.8	11.6	12.2
Twofish-Serpent	12.1	11.5	11.8
Serpent-Twofish-AES	9.1	8.4	8.8
AES-Twofish-Serpent	9.1	8.4	8.8

这是一台 2003 年的主流水平的计算机, 距离今天 (2009) 已经有 6 年了, 这样的测试平台已经可以说明一切, 这样的设备的加密速度并不低。我们也可以购买速度更快的计算机设备。毕竟用来加密的计算机不是去为了进行天文计算。我们看一下主流的笔记本便携式计算机移动设备的速度。

便携式计算机硬件测试环境:

处理器名称: Mobile DualCore Intel Core 2 Duo T7100, 2000 MHz (10 x 200) 原始频率 1800 MHz

主板: Intel GM965 (Centrino (Santa Rosa) 兼容平台)

内存: 2GB Kingston DDR2 677 (2X1GB)

硬盘: Hitachi Travelstar 5K500.B 320 GB 5400 RPM SATA2

测试软件: TrueCrypt 6.1a 便携版本

缓冲大小: 10MB

表格 4: 便携式计算机加密基准速度测试 (按平均速度递减排序)

算法	加密速度 (MB/S, 兆每秒)	解密速度 (MB/S)	平均速度 (MB/S)
AES	131	131	131
Twofish	114	119	116
Serpent	58.9	59.7	59.3

AES-Twofish	60.2	56.8	58.5
Twofish-Serpent	38.2	40.2	39.2
Serpent-AES	39.8	40.5	40.1
AES-Twofish-Serpent	29.9	30.6	30
Serpent-Twofish-AES	30.1	22	24.6

无疑一些小的移动设备以后的处理速度会变的更快。前 10 年的台式电脑和这十年的手机会有一些新用途, 就像下个 10 年的门锁和电灯泡一样, 会变的又快又多。你如果也想这些东西确保安全, 但是不会愿意花费超过 10 秒的时间来加密它。

目前 (2009), 很多的手机厂商提升了手机硬件的 CPU 性能, 不仅仅是 CPU 频率的提升^a, 一些厂商已经注意到手机在生活中通讯的地位是一些笔记本无法做到的。如果我们查找一个联系人并且发送文件, 需要打开笔记本去做的话, 在一些紧急关头就显得很薄弱。另外, 影响个人身份信息安全的设备, 除了笔记本之外, 还有手机。他们开发了相应的保密程序直接嵌入手机, 除了加密联系人数据库之外, 还有短信数据库, 有的厂商的产品需要安装第三方软件^b, 有的厂商甚至在手机系统中直接嵌入数据加密的软件^c。现代无线电技术使用新的 3G 技术, 以及速度更快的 2.4G 的无线网络 (WiFi, Wapi^d) 技术。这些技术使得手机的安全性大幅度增强。

^a 目前的手机除了打电话、发短信等一些简单的事物外, 增加了游戏、音乐等多媒体处理能力, 那就需要一个强劲的中央处理器来处理这些数据。截止到编写书为止, 已经出现手机处理器达到 1G Hz 以上的产品, 达到奔腾 III 处理器的频率。

^b 这些软件需要根据手机系统来选择, 系统一般分为 Windows Mobile、Symbian、Linux 类的系统。由于运行环境不同, 软件就不能通用了。根据我个人经验, 使用 Windows Mobile 系统的手机的硬件能力要相对强一些, 当然更为费电, Symbian 的性能差一点, 电池使用的时间会长一些。

^c 例如 Nokia 在 E71、E66、E63 等商务手机中加密系统集成成了 XTS 模式的 AES 算法和 128-bit 密钥。透明加密的使用过程不会对手机的运行速度有太大的影响。具体访问:

http://nfb.online.nokia.com/Page%20Content/Mobilize%20your%20business/Knowledge%20center/Datasheets/DeviceMemoryCardEncryption_Datasheet_Global_Default.pdf

^d WiFi (Wireless Authentication Privacy Infrastructure), Wapi 使用的都是 802.11b 技术, 硬件是相同的, 不同的是用户访问的验证方式不同, 中国制定的 Wapi 标准的安全性更高。一般升级设备的操作系统系统就可以支持。无需更换硬件设备。WAPI 安全系统采用公钥密码技术, 鉴别服务器 AS 负责证书的颁发、验证与吊销等, 无线客户端即移动终端与无线接入点 AP 上都安装有 AS 颁发的公钥证书, 作为自己的数字身份凭证。当移动终端 MT 登录至无线接入点 AP 时, 在使用或访问网络之前必须通过鉴别服务器 AS 对双方进行身份验证。根据验证的结果, 持有合法证书的移动终端 MT 才能接入持有合法证书的无线接入点 AP, 也就是说才能通过 AP 访问网络。这样不仅可以防止非法移动终端 MT 接入 AP 而访问网络并占用网络资源, 而且还可以防止移动终端 MT 登录至非法 AP 而造成信息泄漏, 实现“合法终端通过合法接入点访问网络”。无线局域网鉴别与保密基础结构(WAPI)系统中包含 WAI 鉴别及密钥管理和 WPI 数据传输保护。WAPI 采用国家密码管理局批准的公开密钥体制的椭圆曲线密码算法和秘密密钥体制的分组密码算法。

项目		WEP	WAPI	IEEE 802.11i
鉴别	鉴别机制	单向鉴别 (AP 鉴别 MT)	双向鉴别 (AP 和 MT 通过 AS 实现相互的身份鉴别)	现相互的身份鉴别) 单向和双向鉴别 (MT 和 Radius 之间), MT 不能够鉴别 AP 的合法性
	鉴别方法	开放式系统鉴别 (或共享密钥鉴别)	身份凭证为公钥数字证书; 无线用户与无线接入点地位对等, 实现无线接入点的接入控制; 客户端支持多证书, 方便用户多处使用	用户身份通常为用户名和口令; AP 后端的 Radius 服务器对用户进行认证
	鉴别对象	客户机	用户	用户
	密钥管理	无	全集中 (局域网内统一由 AS 管理)	AP 和 Radius 服务器之间需手工设置共享密钥; AP 和 MT 之间只定义了认证体系结构, 不同厂商的具体设计可能不兼容
	算法	64 bit RC4	192 位椭圆曲线算法 (ECC192)	与具体的协议有关
	安全漏洞	鉴别易于伪造	未查明	用户身份凭证简单, 被盗取后可任意使用
加密	密钥	静态	动态	动态
	算法	64-bit RC4	128-bit SMS4	128-bit AES 和 128-bit RC4

安全是一个广泛的技术，不仅仅是长的密钥更好；你要考虑我们是保护什么，因为目前只有一种最完美安全的加密系统。我们下面就要讨论。

4.1.4.8 真正不可破解的加密：One-Time Pads

One-Time Pads（一次性密码本^a，OTP）是目前唯一安全的加密系统。注意，对于这个我用了“完美”一词来形容。一次性密码本“真的很完美”。或许我应该说“那是相当的完美！”。它是由 Major Joseph Mauborgne 和 AT&T 公司的 Gilbert Vernam 在 1917 年发明的。One-Time Pads 是第二次世界大战^b英国使用后出名的，也被俄国（包括前苏联和俄国）使用。

比较容易理解 One-Time Pads 的算法。是不是乏味的，用了就知道了。表格 5 和表格 6 中，我已经设计了一个 One-Time Pads。这个表里面使用了 26 个英文字母，加一个空格符号组成了 27 个字符。我们把这个字符用 0 到 26 的 27 个数字标号，当然你也可以自己随便排，甚至可以把中文 1 万多个汉字一个一个排，也可以把全世界的文字都排一下。我这样排列只是为了容易看懂，里面的顺序是可以打乱的，但是不能重复。密码本有 100 个空格，你也可以制作 1000 的格子，里面的整数可以随便输，没有什么限制。你想象一下在纸上的密码本是什么样子的。当然，你编的这些都要保密。

表格 5：字母编号样表

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	2	3	4	5	6	7	8	9	10	11	12	13	14
O	P	Q	R	S	T	U	V	W	X	Y	Z	空格	
15	16	17	18	19	20	21	22	23	24	25	26	0	

表格 6：One-Time Pads 样表

1 到 200，用一次就删除一个									
17	23	14	0	7	10	22	9	6	18
3	16	11	15	17	16	21	5	8	11
6	12	23	1	18	6	19	14	16	11
22	6	13	25	1	0	4	11	24	5
17	10	23	23	11	15	10	12	2	6
24	19	21	26	2	5	15	10	7	9
26	2	22	17	10	0	11	8	3	6

^a One-Time Pads 的翻译很多，还有串流加密法本、一次一密乱码本，不管怎么翻译，它就是一个名字，我就不翻译名字了。

^b Leo Marks, *Between Silk and Cyanide: A Codebreaker's War* (密码制造者的战争), 1941-1945, 免费发布, 624 页, ISBN: 0-684-86422-3 (硬皮), 0-684-86780-X (纸质), 很难说这书是一本好书。Leo Marks 是一个引人注目的人，他的父母经营着伦敦最出名的书店，书店位于 Charing 十字路 84 号。他是个电影剧本作者，偶尔去做演员（包括在《The Last Temptation of Christ》（耶稣最后的诱惑）中的撒旦的声音），他在第二次世界大战时期，也就是在 22 的边缘为 SOE 进行代码安全检查。

当这本书在美国发行的时候我扫了一眼，当时 the dot-com 倒闭了。我几乎没有读，是因为我气恼的是我们在军事和银行中使用了太多的安全技术，而在现实中却没有多少。从它开始的段落中，确实是一些冷笑话和歇斯底里、无缘无故的生气。故事很不错。Marks 文章的一些地方让我重新开始考虑思考实现安全性的方式。

Marks 教会我勇敢，信息安全火线后面有大批的难民，然而他们都被遗弃了，那里曾经是他们赖以生存的，但是他们不听。他发现了在每一个错误的过程上都有愚蠢的官僚主义存在。这个题目来自 Marks 使用过的“丝织布上的 one-time pads 表（易藏，易燃）安全算法。他决定是使用丝绸还是氰化物。如果你在商业安全领域，你可能不会给人们发送不确定是否可以完全被毁灭的信息，他们可能也没有为突袭来的商业风险做好准备。我知道了人们无法让他们自己做一个安全的决定，甚至特别是如果他们赖以生时。

这是个很愚蠢和微不足道的说法，但是在 PGP 系统中的许多都吸取了来自 Leo Marks 的教训。它不仅仅是个好书，而且使我在安全领域受 Marks 的内在化影响而变得更加专业。

18	19	19	26	1	11	14	24	12	19
18	20	7	7	4	8	1	11	25	0
9	3	14	16	4	17	6	1	16	1

按照下面的规则加密: (为了方便你们看懂, 我举一个例子: N)

1. 把明文的一个字母放入密文中。

例子: 现在就是 N

2. 转换这个字母用下一个单元格的数字, 比如 A 用 2 转换就是 C (其实就是原来的字母变成加一个数字后的字母, 这个在英语中是移动的意思)。注意空格是第 27 个字母。用数学表达就是: 第 i 个密文字符 = 第 i 个明文字符 + 表中的第 i 个数字。 i 代表 1, 2, 如果结果超过 27 了就除以 27 取余数, 总之这个结果要小于 27。

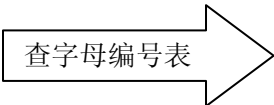
例子: 我看到 One-Time Pads 样表第一个格子 (第一行第一列) 是 17, N 在字母表中的号码是 14。17 + 14 = 31, 31 超过了 27, 于是 $31 - 27 = 4$, 在字母表中找到 4 对应的字母 D, 第一个密文就是 D! 其实如果相加超过 27 都要减去, 这个 27 的数字是根据第一个字母表有多少个元素规定的, 上面的字母表有 27 个元素。

3. 对明文的字符按照次序重复上面的过程。如果加密的文字超过了表的体积, 就使用下一页表, 如果不够自己还可以再造表。而且可以不断的造表。
4. 删除使用过的 One-Time Pads 表。为了让他保密, 最好是吃了或烧了。(如果你 2 个都做, 顺序很重要, 这样使它不容易引起别人对这个密码本的兴趣。), 但是在这之前, 你需要确保解密者已经拿到了 One-Time Pads 表, 否则, 他无法解密。

现在可以尝试还原密文。给你出一个小练习, 密文是:

DKJ PBVBNCITAVP TZKGX HAYUGRP NMDBRPGV LMMVAOWDCJOQSB SNN

对练习提示: DKJ 在上面的字母表中的数字是 4, 11, 10, 减掉表里面第 1, 2, 3 个数字, 不够就加 27, 意思就是不够就加 27。总之不能得到负数的结果

4+27-17=14		14=N
11+27-23=15		15=O
10+27-14=23		23=W

结果就是 NOW, 你可以根据这个把剩下的都算出来。

你知道整个信息有 55 个字符, 但是这些还不能继续。我会注意到在这个信息结尾的一些空间。我举个例子, 我考虑到了添加一个终止符号“STOP (停止)”在这个字母表里面, 就像以前电报一样。我用一个终止符“STOP”结束信息, 或者全部都是“STOP”“STOP”。我可以找出 100 个密码表的字符来加密 NYAH NYAH NYAH, 然后我什么也得不到。

为什么是完美的加密呢？注意加密用的密钥是表里面提供的。每一个单元格的数字是随机的，我们用一个密钥来和密文唯一的一个字母来运算。假设随机数字真的是随机的，整个密码表是无法猜测的，等于密文是无法猜测的。整个密文中，密文的大小和密钥的大小是一样长的。因此密码分析学家就无法使用图形、统计、和其它方法从密文中获得信息。甚至即使我告诉你编译这个密码的方法。有很多经验的密码学家会分析出信息的第一个是包含 3 个字母的单词。但没有泄露任何关于整个词汇的特别信息。

特别说明，One-Time Pads 不外乎是一个大的不重复的真随机密钥字母集，这个密钥字母集被写在几张纸上，并一起粘成一个乱码本。它最初的形式是将 One-Time Pads 用于电传打字机。发方用乱码本中的每一密钥字母准确地加密一个明文字符。加密是明文字符和 One-Time Pads 密钥字符的模 26 加法。

每个密钥仅对一个信息使用一次。发方对所发的信息加密，然后销毁乱码本中用过的一页或用过的磁带部分。收方有一个同样的乱码本，并依次使用乱码本上的每个密钥去解密密文的每个字符。收方在解密信息后销毁乱码本中用过的一页或用过的磁带部分。新的信息则用乱码本的新的密钥加密。

有很多 one-time pads 的表。它们保密过程都几乎完美，它们令人眼花缭乱。同样也是很乏味^a的，但是它们也很有意思，也很吸引人。

4.1.4.9 One-Time Pads 的诱惑

设计出这么完美的算法并不容易，当你使用 one-time pad 时，还要考虑很多的操作问题。

- 表必须是随机的。真正的随机，不是伪随机的。如果不是真正的随机。密码分析学家可以使用非随机的过程进行猜测。研究这部分的时候，我发现一个网站进行一些像我这样例子的 one-time pad 加密，但是它使用的是伪随机信号发生器。当 Alice 做的时候，她就损失了她的绝对安全。这个系统的安全性和那个发生器的安全性一样，发生器就是密钥，如果 Eve 猜测到这个发生器，Eve 就可以翻译所有电文。
- 这个表必须保密。Alice 和 Bob 必须保证没人可以得到它们。如果 Eve 贿赂 Bob 的看守人 Hattie 或 Alice 的门卫 Justin，Eve 就可以很容易的破译电文。那个表需要绝对保密。我觉得吃了或烧了并不是很滑稽的。有时候必须这么做。在第二次世界大战 Leo Marks 使用 one-time pads 在丝绸布上画表，这就是为了烧毁的时候更方便，而且燃烧后只有很少的灰残留。但是也没有听说谁会把这个敏感的表印在容易销毁的闪光纸(Flashpaper^b)上的。
- Alice 和 Bob 必须有和发送信息时的一样多填充表上的数据。他们必须有很多页的表来加密数据。这也就意味着，他们必须要新的表寄通过信任的信仆送过去。大概推测一下，Alice 的门卫 Justin^c 可能会泄露编码表，一般只有他会做，毕竟密码表比信息要大的多。

在互联网上，保存足够大的表的信息更困难，因为如果以前没有发过，这是很难安心的发送 one-time pad 表的，压缩也没有办法，因为你不可能压缩随机数字。

^a 上面的密文的明文是：NOW IS THE TIME FOR ALL GOOD CRYPTOGRAPHERS TO COME TO 。（英文意思：优秀密码破译人员们来的时候到了。）

^b 又叫火纸，如用火点燃它，火光强烈，一瞬即逝，无影无踪，燃烧时无烟，燃烧后不留任何灰烬，是增强魔术效果的一种辅助品。将一般的纸张变成 flashpaper，主要原理是将纸的纤维素变成硝化纤维素，纤维素进行硝化反应。硝化过后的纸，燃点较低，比较容易燃烧。原本的纸需要比较高的温度才能燃烧，硝化过后，燃烧时需要的温度较低，因此只要有一点的热，就可以让整张纸烧起来。

^c 我一直认为 Alice 的门卫应该叫 Justin Case。

- 他们应该用完密码表。Alice 和 Bob 必须停止发信息，直到他们拿到新的密码本。一个著名的故事^a，俄国人扔了一本后又重新制作了一本。NSA 利用他们的这个失误，解密了信息。这并不容易，他们为此花了 10 年的时间，当密码破译者花了他们毕生而破译成功的时候，破译者几乎快疯了。尽管如此，这也能够做到，如果你不想达到这么高的安全性，为什么要不辞辛苦的达到这个安全标准？

另外的问题是为什么我用这么多诱惑的词汇来描述这些，从而引起你注意 one-time pads 呢？它确实提供了强大的加密安全。问题是这个完美的成果诱使人们去想工作安全性是不是也是完美的？一个链条的结实程度只和它最弱的那一个链节的一样^b。一个系统的安全是和它的组成部分中最弱的一样，而不是最大的那个。

如果不是 one-time pad 的出现，Alice 和 Bob 使用像 PGP 软件一样的加密系统，他们权衡简单操作的密码算法的完美性。区别就在于如果密码表到了 Eve 的手里和公钥到了 Eve 手里那就不一样了，那就是公钥加密比 one-time pads 安全的原因。

One-Time Pads 的想法很容易推广到二进制数据的加密，只需由二进制数字组成的 One-Time Pads 代替由字母组成的一次一密乱码，用异或代替 One-Time Pads 的明文字符加法就成。为了解密，用同样的 One-Time Pads 对密文异或，其他保持不变，保密性也很完善。

这听起来很好，但有几个问题。因为密钥比特必须是随机的，并且绝不能重复使用，密钥序列的长度要等于信息的长度。One-Time Pads 可能对短信息是可行的，但它决不可能在 56Kbps 的通信宽带工作上工作。你能在一张 DVD-ROM 中存储 4.8GB 的随机二进制数。但有一些问题：首先，你需要准确地复制两份随机数比特，但 DVD-ROM 只是对大量的数据来说是经济的；其次，你需要能够销毁已经使用过的比特，而 DVD-ROM 没有抹除设备，除非物理毁坏整张盘。数字磁带对这种东西来说是更好的媒体。

即使解决了密钥的分配和存储问题，还需确信发方和收方是完全同步的。如果收方有一比特的偏移（或者一些比特在传送过程中丢失了），信息就变成乱七八糟的东西了。另一方面，如果某些比特在传送中被改变了（没有增减任何比特，更像由于随机噪声引起的），那些改变了的比特就不能正确地解密。再者，One-Time Pads 不提供鉴别。

One-Time Pads 在今天仍有应用场合，主要用于高度机密的低带宽信道。美国和前苏联之间的热线电话据传就是用 One-Time Pads 加密的。许多苏联间谍传递的信息也是用 One-Time Pads 加密的。到今天这些信息仍是保密的，并将一直保密下去。不管超级计算机工作多久，也不管半个世纪中有多少人，用什么样的方法和技术，具有多大的计算能力，他们都不可能阅读苏联间谍用 One-Time Pads 加密的信息，除非他们恰好回到那个年代，并得到加密信息的 One-Time Pads。

PGP 世界是基于 Phil Zimmermann 的算法，PGP 团队创造和提高了自动工作时安全性。对于我看来，在软件各方面的安全性都有很大提升，因此我不再考虑如何加密的问题。

对于 one-time pad 的诱惑我没有任何防备。如果把它投入实用那将会很酷。我想起一些小说里都有的情节：主人公炫耀一些美丽的东西，某人辛勤的劳动忽略了抱怨，但是会适当对工作投入更多的热爱。通常这类故事是喜剧，最后主角和她的心爱终成眷属。这个故事我认为是 Shaw 写的 Pygmalion，书中 Eliza 向 Higgins 示爱。结尾很不和谐，但是比《My Fair Lady》好，他们快乐的生活直到最后。One-time pads 不仅完美，而且是高成本维护的，我们要不断的更换密码本。这个技术在以前和现在都可以很好的被使用。但是同对于公钥加密算法和高级对称算法的 128-bit 足够的密钥传输来说，它真的太大了。除非你的对手有非常多的计算机，那才可能会使你担心。256-bit 很好了——这是真正的科学虚构推测的技术。

^a 1943 年美军开始一个叫 VENONA 破解苏联外交信息的计划。1980 年完成，1995 年就丧失了使用价值。NSA 站点有这个计划的内容：<http://www.nsa.gov/venona/index.cfm>

^b 也就是木桶原理：一个木桶能够装多少水取决于最短的那个木片的长度。另外延伸一下，密码学这个水桶，不仅和木条的长度有关，而且和木桶的底有关。装再多水的木桶，底子漏水，这个木桶在使用过程中的效率会是很低的。

4.1.5 Hash 算法

哈希算法（Hash functions^a）是密码学主要的一部分。这是我们加密人员与泛滥的破解技术抗争的主力，我们知道他们最不喜欢就是密码图形。

一个 hash 算法提供了可变长度的输入字符串^b和固定长度的结果。输入的很简便就是“hash”的意思，这个词不是人名的缩写。你可以用 hash 来输入数据，固定长度的字符串允许我们使用 hash 值来引用实际字符串本身。

因为 hash 算法使用长的字符串，再变成一个短的。不可避免有 2 个字符串通过 hash 算法会得出一样的结果，这个在密码学中叫“碰撞”。举个你可以明白 hash 值的例子，假如 Jon Callas 和 Jane Cannoy 他们名字的 hash 值都是 JC。碰撞是了解 hash 算法很重要的部分，我们将会比特（bit）的单位上有更多的介绍。

尽管缩写是一个很简单描述原文的方式，缩写造成了密码学目的的 hash 算法的错误。密码学的 hash 算法有很多用在加密技术中的属性。

- 很难逆向运算 hash 算法。据 hash 知识，没有一个好的办法找到 hash 值对应的那个字符串。我们已经知道了 hash 算法会丢失数据，创造了一个简单的相对性。这个相同的性质也是名字缩写的：除了 JC 没其它的信息，不能找出我的名字，是 Jon Callas？是 Jane Cannoy？还是？
- 一个 hash 值，它应该很难确定一个本来的字符串。这个性质是缩写遗漏（initials lack）。看缩写的时候如果知道名字的匹配是很简单的。在密码学中，我们想找出源信息和这个结果之间的联系，他们之间的关系是尽可能不透明的。
- 确定一个源字符串，我们根据这个字符串的 hash 值很难找出第二个字符串。很难有效的改变字符串获得一个碰撞。也很难改变“我同意支付 100 美元”到“我同意支付 500 美元”而获得碰撞。注意这 2 个字符串之间只有 1 位不同。
- 也很难找出碰撞的 2 个字符串的 hash 值。

这个算法在很多不同的事情上给了我们灵活的想法，这里有一些例子：

- 当你在 PGP 软件中输入密码的时候，我们使用 hash 算法来生成一个密钥。中间的过程就是 hash 算法，通常一遍遍的使用来降低破解者的暴力破解的风险。
- PGP 软件的随机数生成器在传入数据后，会根据你键盘和鼠标的移动时时更新。这样使得观察者不确定这个值，也没有不变的随机数字。我们使用 hash 算法消除观察者的数据中的不均匀性。
- 随机数生成器使用 hash 算法产生输出值。这个过程 PGP 软件也做了。
- 文件完整性算法，使用 hash 算法可以很快的检查文件。比如：你可以保留文件的 hash 列表在你的电脑上。hash 数据库中的值也变了，你就看到计算机内的文件变化了。软件分布系统站点通常有分布的

^a Hash 算法有时也叫信息摘要算法。并不是每个数据都参与这个运算，只是选取了一些特殊位置的数据进行运算。可以说是一个简易的快照。

^b 我们在计算机科学中把计算机里的一个字称作字符，比如‘x’。连起来的字符就是字符串，比如：“你能看懂 x=123 的意思。”。

文件 hash 值列表, 所以人们拿到文件的时候可以和列表比对, 审核文件的完整性^a。

- 复杂密码系统使用 hash 算法创建数据完整性作为它的一个系统组件, 我们稍后会了解这个。

注意几乎所有算法现在都在被广泛使用, 这有一个假设它们不会发生碰撞。如果 2 个密钥发生了 hash 碰撞, 任何一个密钥都可以解密文件。如果 2 个软件包有相同的 hash 值时, 一个肯定被误认为是另外一个。

4.1.5.1 通用 Hash 算法

表格 4 列出了一些 hash 算法的共同点, 特别是 PGP 使用的。

表格 7: 通用 Hash 算法

名称	大小 (Bits)	描述
MD5	128	MD5 是 hash 系列算法中的最低标准, PGP 软件在 PGP5.0 版本以前使用。MD5 的脆弱性在 1996 年第一次出现。MD5 是 MD4 的改进, PGP 软件不再使用它的原因是它是第一个被破解的通用 hash 算法
SHA-1	160	SHA-1 是 MD5 的改进, 由 NIST 设计, 解决 MD5 的问题后被广泛使用。
RIPE-MD/160	160	RIPE-MD/160 是一个和 SHA-1 差不多的 Hash 算法。设计 RIPE-MD/160 为了改善超过 MD5。它被 Reseaux IP Européens(RIPE)组织设计, 而不是美国 NIST 我们认为它的安全性和 SHA-1 差不多。
SHA-256	256	SHA-256 是美国 NIST 最新设计的新 Hash 算法。也属于“SHA-2”的类型, 它有和其它不同的内部结构, 但和其它 hash 算法的基本结构都是一样的。
SHA-512	512	这是“SHA-2”算法的一种, 和 SHA-256 差不多。
SHA-384	384	SHA-384 有比 SHA-512 更小的输出。一般不常用 SHA-384 是因为除了大小以外没有任何优势, 如果我们需要比 SHA-256 强度高的算法, 我们会直接选 SHA-512。同样 SHA-224 是 SHA-256 缩小版。

4.1.5.2 Hash 算法难度

目前(2006 年中期), 我们知道了 hash 算法系列在使用上并不是很完美, 他们中的一些确实不完善。这个问题到 2004 年的夏天变的明朗了, 中国山东大学王小云教授宣布她和她的团队在一些 hash 算法^b中发现碰撞。这时 RSA 名字中的“S”的这个人 Adi Shamir 说, “上星期, 我还认为 hash 算法是我们认为最好的部件。现在则认为它是我们的部件中是最差的。”在 2005 年初, 王小云的攻击延伸到了第一次幸免的 SHA-1^c。

我们仍然在应对这个问题。他们中的所有都绕着 hash 碰撞, 2 个字符串生成了一样的 hash 值。一个数学的分支: 组合数学 (combinatorics) 中的一个叫归档原理 (Pigeonhole Principle^d) 的公理。最简单的归档原理的解释是: 如果你有 13 个鸽子而只有 12 个笼子, 至少有一个里面装有 2 个鸽子。很显然的, 不是吗? 那就是为什么这是公理的原因!

^a 比如本书的原版的 PDF 文件的 CRC-32 校验码为 79EE7FEF, 如果新版本中的这个数值改变了, 我就可以确定原作者或者 PGP 公司对文件进行了改动。可能是发布了新版本的书, 也可能是换了一本。

^b 王小云、冯登国、来学嘉和于红波的关于 hash、MD4、MD5、HAVAL-128 和 RIPEMD 的碰撞理论: <http://eprint.iacr.org/2004/199>

^c 王小云等文章《Finding Collisions in the Full SHA-1 (找到全部 SHA-1 的碰撞)》在《Advances in Cryptology — CRYPTO 2005》, LNCS 3621, Springer, 2005, ISBN 3-540-28114-2, 17-36 页。更多关于王小云教授的信息请上 google 搜索。

^d 或称作分类原理, 信箱原理, 格孔原理。

如果你应用这个公理到 hash 算法, 考虑 16-bit 的 hash 计算。再考虑整个 16-bit 的字符。依照归档原理, 至少有 2 个字符串会有一样的 hash。事实上, 还有一大堆一样的例子。这个碰撞和鸽子汇集问题是一样的。如果这个碰撞是均匀分布的(这对 hash 算法来说也正确), 一个 hash 值那会有 256 个碰撞, 然后根据归档原理, 至少 1 个 hash 有至少有 256 个碰撞。

找出一个碰撞应该和猜一样容易, 但是有多难呢? 回答这个问题又引发另外一个有趣的数学问题叫生日问题。在谈论区块大小的时候我们谈到过的。和 Alice 有相同生日的人的概率是 $1/365^a$ 。但是如果你有一房子满满的人, 和另外一个人生日发生碰撞的概率有多大? 特别的, 有多少人机率均等也就是房间中有 2 个人的生日是一样的呢?

这个问题的一般回答和找出 hash 碰撞的是一样的。我们认为生日是比另外一个名字缩写问题更好的 hash 问题, 但远远不完美。尽管如此, 生日是一个公平的任意分布的^b。对于生日来说, 原来生日^c碰撞的机率是大约 23 个人中的偶数。通常, 机率是偶数的大约是选项数字的平方根。

我确定你注意到我使用了一个很含糊的词“大约”。这是因为答案不是准确, 只是接近平方根。大概的说, 碰撞的机率是:

$$\text{prob}(\text{pigeons}, \text{holes}) = 1 - \frac{\text{holes}!}{(\text{holes} - \text{pigeons})! * \text{holes}^{\text{pigeons}}}$$

其中, Prob 是机率的意思, 只表示函数名, pigeons 是鸽子, holes 是笼子洞。Pigeons 和 holes 都是输入变量的名字。

省下你的数学运算。如果你解决了鸽子数目的问题, 结果的机率是一个洞的 2 个鸽子里面每一个都有 $\frac{1}{2}$ 的概率。可以算出约为 $1.2 \times \sqrt{\text{holes}}$, 对于我们使用的那个问题来说, 我们也可以认为等于 $\sqrt{\text{holes}}$ 。特别是当我们去处理一个非常大的数字的时候, 这样去推测很方便, 这个方法也是理论数学中被惯用的手法。

所以, 如果我们有一个 n-bit 的 hash 算法, 如果我们有 $2^{\frac{n}{2}}$ 个字符串的碰撞的机率相等。也就是说, 160-bit 的 hash 运算只有 80-bit 的安全性。 2^{80} 是很大的一个数字。大约 2 倍的阿伏伽德罗常数。阿伏伽德罗常数^d是摩尔体积的分子数, 或者用一个方便的东西表示, 就是一汤勺水中水分子的数目。那是个很大的数。王小云带着报告参加了 2004 年密码界峰会, 她震撼了密码界。她没有用一张纸来展示如何碰撞, 她仅仅只用了他们中的一部分。就像你看到的, 因为碰撞很难发现。仅仅有 128-bit 的 hash 算法中的一部分中有碰撞, 也就意味着碰撞已经出现了。对于密码分析家的主要问题是 “她知道我们不能做什么?” 6 和月以后, 她的技术扩展到攻击质数的 160-bit 的 hash 算法。

这就是我们在最后 2 年所总结的:

- 王小云是最优秀的密码分析专家。她有着其它数学家没有的基础数学洞察力; 她非常迅速的成为世界上为数不多的、最优秀的 hash 算法密码分析专家。
- 一些其它的理论工作不是去进行应用实际, 而是更多的思考。例如: 在王小云报告前几个月, John Kelsey

^a 我们忽略闰年的 2 月 29 日。

^b 原来人在出生的月份中 8 月比其它月多, 这甚至有一些据说关于 11 月的感恩节信息的信号在 8 月的第 3 个星期, 他也说明了更多的人喜欢在星期二出身, 这就导致了不同的选择。

^c 一个讨论生日问题很好的文章在: <http://mathforum.org/dr.math/faq/faq.birthdayprob.html>, 一些你可能用到的数学解释在这里统计好了: <http://mathworld.wolfram.com/BirthdayProblem.html>, 或直接点 <http://mathworld.wolfram.com/BirthdayAttack.html>。

^d 该常数及其知识属于化学部分, 摩尔任何物质所含基本单元(分子, 原子, 离子等)数。它是物理学和化学中的基本常量之一, 其值由实验测定, 为 $N_A = 6.0221367 \times 10^{23} \text{mol}^{-1}$ 。

和 Bruce Schneier 展示^a了寻找一个给定的字符串的 SHA-1 碰撞, 你可以 2^{106} 次工作来代替 2^{160} 次, 但是你需要一个长达 2^{60} 字节的信息。在王小云展示我们的工作中的漏洞前, 这是个有趣的但不实用的故事。现在, 我们中的一些人怀疑是结构问题上的一些漏洞。我们目前还不知道。

- 有很多议案关于如何修改剩下的算法来抵抗王小云的攻击。他们都非常棒, 但是一个明显的问题是, “明年有什么攻击, 这个修正可以解决吗?” 当然, 这个问题是不能回答的。我们不可能反对未知的攻击来保护我们的算法。无论如何, 其中的很多议案确实是解决的好办法。一个简单的技术诸如当进行 hash 运算时使用每双字节 (用 AABBC 来代替 ABC), 或者插入 0 比特在每 4 个字节后面^b, 或者添加随机数据在准备 hash 运算的数据之前, 用这些办法解决了已知的问题。
- 我们开始考虑一个如何设计一个好的 hash 算法的想法。在 2005 年 10 月, NIST 主持了一个关于 hash 算法的工作组。密码专家开始考虑想出一个如何设计一个好的 hash 算法的想法。第 2 个工作组在 2006 年 8 月开始计划。同样也有像 AES 相似的竞争方式来产生一个新的 hash 算法。
- 工程师的观点中也有一些好的想法。在 PGP 团队中, 我们已经发扬了首创精神。

在 PGP 团队, 我们开始转移 MD5 到 1997 年的水平。PGP 5.0 开始从 MD5 向 SHA-1 发展, 保持 MD5 的唯一目的是为了向后兼容性。PGP 8.0.3 介绍了这个技术支持, 也可以在阅读中找到, 但是没有 SHA-256、SHA-384 和 SHA-512 的算法。PGP 9.0 开始从 SHA-1 向 SHA-256 发展。

4.1.6 数据完整性算法: 信息鉴别码和数字签名

我们下一个题目是讨论验证数据是否完整性的, 也就是确定这个数据就是你想要的那个数据, 所有数据只不过是一些数据。就像加密, 有 2 种基本的方法检查数据完整性 (data integrity), 使用对称密钥和非对称密钥, 他们叫信息鉴别码 (Message Authentication Codes, MACs) 和数字签名 (Digital Signatures)。

你可以用一个算法或 hash 算法完成信息鉴别码; 后者叫 HMACs。数字签名比信息鉴别码的功能更为强大, 但是信息鉴别码更快。当一个信息鉴别码的使用者用对称密钥的时候, 任何人知道都这个密钥可以重写数据, 信息鉴别码则需要那个才可以进行鉴别数据的完整。这 2 个部分合作的非常好, 当他们在堆组件中时就不那么好了。与此相反, 数字签名需要一对密钥私钥的那一半, 任何人有公钥都可以验证他。这个性质允许数字签名可以用一个签名人来发布文件和多个验证机。这个技术被用来检查软件中的信息是没有被更改的, 是来自制定者的源。这个在通信和数据存储方面很有用。MACs 可以像 SSL 协议一样阻止攻击者偷偷的修改数据比特流, 或添加什么东西删除什么东西。

我们用来数字签名的算法是 RSA 和 DSA。RSA 可以签名也可以加密。DSA 有趣在它是一个只能签名的算法。同样有我们前面提到过的 DSA 的椭圆曲线算法的 ECDSA。它可以做到创建用 Elgamal 算法的数字签名, 但是这还不是典型。在此同时, OpenPGP 标准^c允许 Elgamal 数字签名, 但是自那以后就删除不用了。

如果你在使用了数字签名后准备一些数学的实际研究, 你就会看到一些关于签名的也是加密的描述, 但是加密用私钥加密比公钥好。这对 RSA 来说是对的, 但是不适合 DSA 和 Elgamal。原因是 Elgamal 签名使用中

^a John Kelsey 和 Bruce Schneier, 《Second Preimages on n-bit Hash Functions for Much Less than $2n$ Work (在 n 比特 hash 算法中 2 次逆向少于 $2n$ 次计算)》<http://eprint.iacr.org/2004/304>。

^b Michael Szydlo 和 Yiqun Lisa Yin, 《Collision-Resistant usage of MD5 and SHA-1 via Message Preprocessing (使用 MD5 和 SHA-1 通过信息处理来抵抗碰撞)》<http://eprint.iacr.org/2005/248>。

^c 不幸的是, 没有任何描述 tar 文件格式的描述, 尽管它被普遍使用, 事实上它是 POSIX 1003.1-1990 标准的一部分。PGP Zip 的 tar 工具 使用和 gnutar 相反的算法, 访问: <http://www.gnu.org/software/tar/>, 它不完全是 POSIX 的 tar 标准。使用了 gnutar 打包, 在目录 dist/src/下的 tar.h 文件中描述了 tar 的结构, 但是不幸的是, 这个代码就是最好的文档。

有一些很奇怪的问题。设计 DSA 签名就是为了减少 Elgamal 签名问题，他有比 Elgamal 签名和 RSA 签名更短的其他优势。

无论你使用什么算法来创建签名，数据本身没有被像写字一样的签名，可以说是在数据在 hash 算法上面的签名。我们做这个原因有 2 点。第一，当你回忆加密的时候，公钥加密算法在数据的区块上操作，你必须操作数据的大小比密钥的大地方(这是最多的问题)，多次重试以后。你可以用数字签名来签名无论有多大的数据，还可以把这些分卷组合起来。第二，这将会非常慢。所以体积比较小的目标文件会非常快。只有 hash 算法表现良好，这也是合理之中的事情

那就是为什么密码员关心 hash 的长度和安全。数字签名混合了原始签名和 hash 值。如果这里出现的问题，数字签名中也会出现问题。

数字签名很重要是因为他提供了数字完整性和在验证的数据上加密。Hash 算法的一项功能是可以告诉你数据是否被改变，但是你必须分开去验证的所有数据，得到 hash 表。如果你是个验证者，使用数字签名你需要数据和他的签名。你同样有签名人的公钥，你需要签名者中任意一个人的签名。作为一个签名者，你需要把私钥保存好，你签名的时候不需要对签名的文件特别担心。有趣的是，签名人也无需保存私钥。也可以签名后删除签名密钥中的私有部分。甚至签名部分的数字签名因为错误而丢失都可以完成验证。

4.1.7 证书机制

公钥就像电话号码；你需要找到你准备找的那人的。他们的数据就像我们所了解的一大堆元数据（metadata）中的电话号码。如果你不熟悉元数据，它就是资料的数据。拿书的例子来说：不仅仅有书的目录，也有标题、作者、版本印刷信息、修正和装订、ISBN 号、主题、类型等等。这些就是书的元数据^a。现在拿电话号码做例子：元数据包含响铃时显示的姓名、家庭住址、办公室、手机号和传真号等等。如果这个例子变成密钥的时候，它也有元数据，就像电话号码是和一个人的名字有关联的一样。也有公钥的另外一种元数据，就像说明它怎么被使用的。加密？签名？还是 2 者都进行？何时创建？何时失效？

我们把密钥信息放入数据的一小点中，也就是证书元数据的，我们也称为密钥，特别是在打算使用 OpenPGP 的时候。按照 Whitfield diffie 的说法称为 PGP 实际密钥，而不是证书。这个“密钥”是一个比“证书”更好的词。听起来很好，说起来也更方便。比行话“证书”很通俗，即使它不是很准确（我一会会讨论）。OpenPGP 证书叫做密钥。我更倾向于使用证书一词，因为我想使人们明白 OpenPGP 的密钥，它其实就是一个证书。

证书不过是合并密钥和密钥的元数据的数据。它可以是一种格式或其它的，我们只有密钥就没什么用。我们想用密钥和元数据工作，特别因为元数据包含了怎么使用密钥的重要信息。这几年中有很多人尝试摆脱证书，有些成功了，有些没有。证书仍然是主流的密钥，因为你想要的是密钥，而不是元数据。

4.1.7.1 为什么使用证书机制？

我们认为密钥（只是一个什么都没有的密钥）是 Alice 的。密码破译人员是一种警惕性的生物，我们担心我们最明显的事情会暴露给他们。Bob 如何知道这是 Alice 的密钥？如果 Alice 给的 Bob，那就明显了，就像 Alice 给了 Bob 她的电话号码。使用电话时我们可以查号码簿。我们在号码簿上找别人的号码。这个号码簿汇编了名字和他们的号码。我们也有和号码簿一样的相似物，那就是密钥，我们必须确信密钥是对的。

数字签名允许我们以非常明智的手段做一些工作，为什么不找到 Alice 和她的密钥的元数据，然后签名？你可以验证这个签名，完成数据完整性检查而确定数据是否正确的。我们使用了这个词“正确”，就像是逻辑

^a 我们可以考虑几点。书的标题是元数据？或者是书本身？我看到了很多其它方面的谈论。其它人认为是索引和个元数据的目录，但是我不认为。对我来说它们似乎是书本身的一部分。你应该想到是书本身和书上的一切东西，你就明白元数据了。

推理出正确(意思是过程是正确的)和真(意思是事实是正确的)^a。你有一个正确的方式确定他不错误的。你也有一个正确的证书,但是不是“准确”来形容。例如 Alice 改名了,她的证书也会不准确,但是仍然有效。

我们创建证书主要有 2 个语法。他们只不过是数据的一种格式,和图片有很多格式的道理一样。我们以后会讨论的更多,现在 2 个格式是 OpenPGP 和 X.509。今天 PGP 软件可以用任何一种证书工作,只要能用,格式都不重要。

一个证书可以用任何密钥签名。证书可以被它自己持有的密钥签名。我们称这样的为自签名证书 (self-signed certificates)。例如: Alice 给 Bob 她自签名证书的密钥,这就是他要的。他随时都可以用这个证书要验证信息是否被改变。证书也可以被除了 Alice 以外的人的密钥签名。也许 Bob 自己签名,但是一般是第 3 方 Charlie 去签名这个证书。在 OpenPGP 的世界,我们称他为第三方委托介绍人 (third-parties Trusted Introducers),但是在 X.509 世界,Charlie 就像是认证授权 (Certificate Authority, CA)。

4.1.8 信任和权限

信任是一个有趣的词汇。它涉及很多内容的很多事情。既然我们使用这样一个小的、严格的定义。信任就是委托机构确定证书是否是准确的形式。信任模式是我们信任的一种范围比较宽的模式。我提到了一些令人混乱的因素,但是它使你想得更有意义。我们看看信任模式的基础。

4.1.8.1 直接信任

直接信任是最直截了当的信任。Bob 信任 Alice 的证书是因为这是 Alice 直接给她的。它是现在最好的信任模式,简单在我已经描述了。名字还没有给出。这并不是最简单的信任模式,但是目前他是最好的唯一值得信任的模式!

人们使用像 OpenPGP 这样的密钥数字指纹(它是 OpenPGP 密钥的 hash 值)在电子邮件和商业中的直接信任。我就这么做;如果你拿到了我的商业名片,那就是我的数字指纹。你可以认为这个密钥就是真正的密钥——你可以确定证书是对的。

就像我前面说的,直接信任的问题是它没有明确互联网中的一些范围。那不是意味着它没什么用,只是说它不包含在支持的以内。

4.1.8.2 分级信任

分级信任也是直接的。在分级信任中,你可以确定被严格信任的人签名过的是正确的。就像是 Bob 认为 Alice 的证书是正的,因为它被 Charlie 签名的。Bob 信任 Charlie 就像是权威部门的签名。

有很多公司机构把 Charlie 的事作为自己的任务,比如 GeoTrust、VeriSign,使用广泛信任的是 CA 证书,他们是值得信任的,因为他们在创建证书的过程中有很多努力。他们中的一部分有相对多数量的信任密钥。这些密钥本身都是根证书 (Root Certificates, 是自签名证书)。根证书的密钥是为另外一个证书签名。在我们信任证书以后,这个可以扩展到其它领域。

为了验证证书,我们可以关注一系列的证书。Alice 的证书是被一个 Jack 创建的证书再签名的证书签名的。许多大型公司、政府等有他们专门创建和维持的证书分级制度。

X.509 是一种用在 SSL 链接上的分级信任。如果你访问 Amazon.com,服务器有一个我们可以信任的根证

^a 这里的“真”说的是一个命题在客观上是正确的,比如: $1+1=2$ 是真命题,月亮是圆的也是真命题,而 $1+1=3$ 就是假命题,月亮是弯的是假命题,但是月亮看上去是弯的就是真命题。而前文的“正确”说的是过程,比如你计算一道题花了很久,但是答案错了,检查后会说:过程是正确的,或者方法是正确的。

书签过名的证书。这些商业认证授权都是 2 到 3 层的深度。

你可能问“我们如何信任根证书？”回答是：直接信任。在你的浏览器中已经有了大量的根证书。你认为它有效的依据是你安装的软件中已经包含了这些根证书^a。

分级信任是直接的，也有风险。如果管理机构出现错误，而错误又很大。一个臭名昭著的事件，微软在代码签名系统中使用分级信任系统，可信的代码。这是由 VeriSign 提供分级的服务，几年前，一些卑鄙的拥有 VeriSign 证书自称是微软雇员的人开发的软件并伪造了微软的证书签名，幸运的是，问题很快被控制。他们放弃了出现问题的微软给程序签名密钥的证书。

4.1.8.3 积累信任

分级信任的缺点：脆弱的级别。导致产生我们最后的信任模式：积累信任 (cumulative trust)。积累信任考虑了很多因素，并且决定证书是否在这些因素上面有效。

最广泛的使用积累信任的是 PGP 网络信任。在网络信任中，Bob 可以分配多变的信任等级给别人(他可以接受认证授权，甚至是 Charlie 和 Dale)。如果他们有足够的点数，Bob 认为证书是准确的^b。在 OpenPGP 中，我们有 3 个等级：访问者、非委托人(0 点)、部分信任(1 点)、完全信任(2 点)。如果他们有足够的证书到 2 点^c，这时认为证书有效。

由一个人接受给另外一个人也是值得信任的。例如，Bob 可能认为 Zelda 的证书是正确的，因为 Alice 签名了，他信任 Alice 认为她的证书是对的，因为 Charlie 和 Dale 都被 Alice 鉴定了。

密码学家 Ueli Maurer^d开发了一个允许大量信任分配的 PGP 网络信任扩展。在这个模式下 PGP 积累的信任有效的许可范围是 0、0.5、1。然而，他允许把任何部分的许可给访问者。你可以给一个访问者 0.9 的权限，而另一个 0.15。

累计信任是围绕直接信任和分级信任的，这个也很容易实现。

4.1.8.4 混合信任模式

没有任何的信任模式是以单独的形式被广泛使用的。每个都有很多优势。甚至直接分级，这个管理方式是 Bridge CAs (间接认证)。这个认证授权的方式像一个访问者对另外一个访问者的方式。

在 PGP 世界我们也使用混合模式。PGP 全球目录 (Global Directory) 是以一个大的网站信任为基础的迷你层次。如果 Alice 提交她的密钥到 PGP 全球目录，全球目录发送电子邮件到她的密钥捆绑的电子邮件上，如果她回复这个邮件，这个地址会出现在 PGP 全球目录中。PGP 全球目录会每 6 个月发送一次跟踪邮件。每 2 个星期验证一次她的密钥。

另外 PGP 提供域名等级信任目录。如果域名设立 OpenPGP 密钥服务器在 keys.域名上，比如：keys.pgp.com，很多 OpenPGP 系统(包括 PGP Desktop、PGP Universal 和 Hushmail) 会自动在这个站点寻找密钥。这个域名提供给信任的访问者域名等级密钥。比如：在 keys.pgp.com 上我们有目录，这里有这个域名 pgp.com 密钥。密钥已经被我签名了，这样你可以自动认为我的密钥是正确的。注意，这个接合了网站信任的其它部分和域名的访问者的直接信任，这样就有了自己的迷你分层机制。

^a 例如在微软的 Windows 系统中，IE 浏览器的 Internet 选项—内容—证书里面就是所有的证书，其中包含根证书。

^b 一个例子，如果一个人要加入一群人的计划之中，如果其中有部分人信任这个人，可能人数很少，如果超过了一个标准，比如 2/3 的人，那么这个人被认为是可信的。

^c PGP 软件中，也设置有 1 点的界限。

^d Ueli Maurer, 《Modelling a Public-Key Infrastructure (模型化公钥基础结构)》，1996 年欧洲研究讨论会在计算机安全(ESORICS'96)的会议录，Springer-Verlag 在《Computer Science》上的讲稿，1146 卷，325-350 页，1996 年 9 月

<http://citeseer.ist.psu.edu/maurer96modelling.html>

4.1.9 证书区别

前面说过,有 2 种证书的数据格式: X.509 和 OpenPGP。说它们相同也不相同。许多产品可以和 2 种一起使用,如 PGP。

他们之间的不同会有越来越少的联系。类似的就像有多样的图片格式,他们有理由存在。但是当你打开网页和照片册的时候,这些图片格式的区别就消失了。另外,一种格式总是和另外一种有一点点相似。如果人们使用比一般证书还多信息的证书,其它人会接受这个。我们来看看 X.509 和 OpenPGP 到底有什么不同。

4.1.9.1 认证和证书

我前面描述 X.509 是一个简单的基本结构,他们包含密钥和密钥的信息,他们在一起才可以签名。OpenPGP 证书更复杂: 他们比一个密钥有更多的信息,比一个数据块有更多信息,比一个签名更多。不管这个,形式是不对称的。你可以说明 OpenPGP 证书包含一组 X.509 证书。

例如 www.pgp.com 的 SSL 证书包含公钥、网站的主机名(www.pgp.com)、一些其它信息(比如签名人没有认出签名中证书组成的密钥),签名是 GeoTrust 认证授权的。

用来为电子邮件加密的有基于公钥算法的 OpenPGP 证书。它包含信息(邮件地址如 jon@pgp.com),一个签名把它们全都结合了。这个签名是自签名,由公钥自己生成。还有另外一个签名,是 PGP 全球目录(<http://keyserver.pgp.com/>)。还有一个 Phil Zimmermann 制作的,一个 Will Price 的,一个 Jeff Moss 的,还有很多其它人的。注意每个人的都是他的 X.509 证书签名。这个包含大量证书,也代表了我有的其它人的电子邮件地址。比如 jcallas@pgp.com 和 jon.callas@pgp.com。

这些自签名证书允许我们做一些有趣的声明,比如什么地方存储密钥的参数,在这个参数里我的密钥告诉我什么时候加密会用到它,我希望你选 AES-128,如果你不选 AES-128 就去试试 AES-256。如果还不是,那 Twofish 算法怎么样呢? 这些参数允许密钥的主人告诉人们谁用这个,他是什么。在写 X.509 证书的时候这之中没有等量,但是人们仍然使用这个。

还有更多,一个 OpenPGP 证书超过一个的公钥的容量。证书中的主密钥必须是几种签名密钥(DSA 或 RSA)之一。也有其它的密钥来做加密或签名。你只能使用 DSA 密钥来签名,所以你的主密钥是 DSA,你还必须有另外的密钥来加密,加入一个 Elgamal 密钥或者 RSA 密钥。或者可以使用 RSA 密钥来加密和签名,但是密码学家认为这个不是个好的形式。所以我们使用独立的密钥来加密和签名。也可以附带添加其它的密钥来签名。一般识别出主签名密钥来签名,这表明已经识别出了附属密钥。

这种措施的好处是软件可以提供统一的数据采集标准来支持多个目的、多个名字和多个密钥。同时无需用户知道这么多的信息。

4.1.10 融合——从明文生成密文

使用证书有 2 种基本结构来加密信息, OpenPGP 格式^a和 S/MIME^b的核心 CMS 格式^c。不管在格式上有多少实际的不同,都是加密和签名的典范。事实上,我描述的最基本的方法是所有的加密和签名是如何工作的。

^a Jon Callas, Lutz Donnerhacke, Hal Finney 和 Rodney Thayer 著《OpenPGP Message Format (Open 消息格式)》,RFC2440, 这本书是 OpenPGP 数据加密规则的核心,介绍了如何实时加密数据。访问 <http://www.ietf.org/rfc/rfc2440.txt>。

^b 还有一个格式是 TLS。T. Dierks and C. Allen, 《The TLS Protocol Version 1.0 (TLS 协议标准 1.0)》,在 RFC2246 中 访问: <http://www.ietf.org/rfc/rfc2246.txt>

^c R. Housley 《Cryptographic Message Syntax (消息加密方法, CMS)》,RFC3852, 访问: <http://www.ietf.org/rfc/rfc3852.txt>。

像 PGP NetShare (网络共享) 这样的新系统使用了相同的基本方法。甚至低等级的系统^a权限, 诸如 PGP Virtual Disk (虚拟磁盘) 和 PGP Whole Disk Encryption (全盘加密) 使用相同的基本方法。甚至一些网络系统诸如 SSL 和 VPN (虚拟专用网络^b) 中也使用。当然, 一些操作步骤有些遗漏。它可以去加密数据、签名数据或者 2 者都进行。很多电子邮件是被签名了, 而不是加密了, 但是很多文件被加密了, 而不是被签名。无论如何, 你与遇到的加密系统都会在最基本的加密数据方法中使用一些特殊手段。

1. 以明文开始。
2. 我们需要一些数据的格式处理。比如电子邮件中, 短语会被适当包装, 行尾被转换为互联网格式的标准等等。在源数据^c上, 我们不对二进制数据做任何改变。
3. 为信息数据创建数据完整性对象。这意味着计算一个 MAC、一个数字签名或者一些等价的。总之。没必要都进行像 MAC 一样的对称完整性对象和数据签名, 但是也可能都做。
4. 如果需要则压缩数据。默认下 OpenPGP 压缩数据。CMS 则不。
5. 用像 AES 的对称算法加密以前的数据。我们可能用一个签名添加到加密的压缩数据的上。
6. 找出我们要解密数据的公钥。
7. 把对称密钥加密成每一个都有公钥的形式。
8. 格式处理源数据。也就是使 CMS 对象变成 S/MIME 邮件信息, OpenPGP 信息从电子邮件格式中能够获得, 等等。

方法的结果是一些像俄国套娃^d, 或者像洋葱一样一层一层。下面的步骤就是封装前面的步骤。

4.1.11 分离——从密文还原明文

当我们收到这些信息的其中之一, 我们必须分离密文和产生明文, 下面就是我们要做的。

1. 从密文的外层开始。

^a CPU 指令系统(用于控制 CPU 完成各种功能的命令)的特权级。在 CPU 的所有指令中, 有一些指令是非常危险的, 如果错用, 将导致整个系统崩溃。Intel 的 CPU 将特权级别分为 4 个级别: RING0, RING1, RING2, RING3。关系是一环包一环的结构。Windows 只使用其中的两个级别 RING0 和 RING3, RING0 只给操作系统用, RING3 谁都能用。如果普通应用程序企图执行 RING0 指令, 则 Windows 会显示“非法指令”错误信息。尽管有 CPU 的特权级别为保护。

^b VPN 的英文全称是 Virtual Private Network, 虚拟专用网络。我们可以把它理解成是虚拟出来的企业内部专线。它可以通过特殊的加密的通讯协议在连接在 Internet 上的位于不同地方的两个或多个企业内部网之间建立一条专有的通讯线路, 就好比是架设了一条专线一样, 但是它并不需要真正的去铺设光缆之类的物理线路。这就好比去电信局申请专线, 但是不用给铺设线路的费用, 也不用购买路由器等硬件设备。VPN 技术原是路由器具有的重要技术之一, 目前在交换机, 防火墙设备或主流操作系统等软件里也都支持 VPN 功能, 一句话, VPN 的核心就是在利用公共网络建立虚拟私有网。访问:

<http://baike.baidu.com/view/19735.htm>

^c 即数据的机器代码, 我们用 WinHex 或 UltraEdit 可以查看, 一般显示为 16 进制的代码。

^d 外形像不倒翁, 中空, 一般是从大到小有很多个, 因为可以一个在一个里面装起来, 最后变成一个而得名: 俄国套娃。访问:

<http://baike.baidu.com/view/279751.html>

2. 移除源数据上面的外壳。二进制密文是放入电子邮件、网络服务和其它的传输方式之中的。
3. 在信息中扫描公钥列表, 寻找我们的私钥。
4. 使用适当的公钥算法解密关于密钥的信息。
5. 使用密钥和适当的对称密钥算法解密加密的信息。
6. 如果信息是压缩的, 就进行解压缩。
7. 如果有数字签名或 MACs, 立刻验证它们。
8. 满腹成就感地看着明文。

依靠我们处理的这些密文详细过程, 在这些数据流中有许多变化的成分。在加密磁盘的过程中, 当磁盘挂载上后, 我们打开对称密钥进行读写。在电子邮件加密中, 我们在加密前后对数据的进行全部的运算, 在我们解密的时候没有做。甚至更坏的, 一旦我们完成, 我们把更复杂的像电子邮件的字体、图片、超链接来这样的东西也进行加密。这就是我们使用的密码学的所有部件, 并且把它们都集成窄了密码系统中。

4.1.12 芝麻开门的技术含量

我们谈论了很多的加密的算法都使用了密钥来打开密文, 我们不仅仅要考虑算法的安全性, 我们还要考虑密钥的安全性, 很多软件中密钥的安全性由口令来决定, 比如在 PGP 中, 你都要输入口令才可以继续。

我们有时候绞尽脑汁、挖空心思的去考虑密码, 尽量设计的复杂、困难, 有人的推荐是把数字、字母、标点符号混合起来, 一些乱七八糟的东西全部混合起来。你晕了吧, 我也晕了, 但是我们仔细一想, 为什么要考虑那么复杂的密码? 这是有原因的。下面我们就过来讨论口令的安全性。

我们在前面说到过破解密文的办法有一种“万能”的办法, 为什么说是万能的办法呢? 因为除了这个办法以外没有更好的办法了, 这种办法就是“暴力破解法”, 我们使用全部的密码组合来重试。大家应该见过那种数字密码的密码箱, 我们现在做一个小实验, 如果这个密码是 3 位的数字, 会出现多少种情况? 我们现来做一个简单数学概率论的题, 看第一位数字, 它可以由 0 到 9, 一共 10 个数字, 第二位、第三位也是, 这里的计算方法是把每一位可能的情况相乘, 这个密码的情况有 1000 种, 完全可以花费一个下午的时间来慢慢打开这个箱子, 心急的人可能说“整那么麻烦干什么, 有榔头不就完了!^a”, 我们的目标是完美的打开这个秘密, 也有专门的办法来对付这么性急的人^b。

当然, 我们的密码不只 3 位, 有的网站规定上不少于 6 位。我们也不仅仅是使用数字, 也使用英文字母, 数字加上英文后, 密码的取值范围变的很大, 如果加上符号, 这个密码的可能情况就更多了。起码我们要知道: 10 数字, 26 字母, 52 字母 (大小写), 62 字符 (数字+大小写), 96 字符 (包括标点符号和数字符号), 这样就加固了密码的安全性, 设计一个复杂的密码确实让我们绞尽脑汁, 如果完成了, 我们这时候不是就大功告成了。我们担心我们的密码会泄露, 我们担心我们会说漏嘴, 我们担心我们说梦话, 我们担心精神类的药物^c。破解人员只能靠猜来解决问题了。

^a 该句出自电影《疯狂的石头》。

^b 小说《达·芬奇密码》中有介绍: 他将信息写在一张纸草制成的纸上, 然后将它卷成小卷和一小玻璃瓶醋同时放进“密码箱”。可想而知, 如果“密码箱”被砸, 里面的醋瓶子也会破裂, 流出来的醋将使纸草立即溶解。

^c 硫喷妥钠, 又名戊硫代巴比妥, 是一种对大脑和脊髓里的受体产生作用的巴比妥酸盐。英国的精神病学家把它作为治疗恐怖症的处方药。这种药物具有麻醉作用。硫喷妥钠可削弱一部分大脑的活性, 消除它的抑制作用, 使人不由自主地开口说话。澳

- 英国媒体近日针对用户密码进行了一次匿名性大规模统计，并公布了最常用密码 Top10。统计结果显示，有 1.8% 受调查者采用了 Top10 中的密码，也就是说，只要按照这份 Top10 排行，黑客们就有 1.8% 的几率猜破谜底，触及你的隐私数据。

第十名.thomas，使用率 0.99%，英国最常用的名字。

第九名.arsenal，使用率 1.11%，英国最著名的球队之一。

第八名.monkey，使用率 1.33%，六位，简单容易拼写。

第七位.charlie，使用率 1.39%，同样是英国非常常用的名字。

第六位.qwerty，使用率 1.41%，这个，还用多说么？

第五位.123456，使用率 1.63%，最简单的六位密码。

第四位.letmein，使用率 1.76%，让我进来吧。

第三位.liverpool，使用率 1.82%，英国最著名的球队。

第二位.password，使用率 3.780%，我本人就认识不止一个采用“password”作为 password 的专业人士。

第一位.123，使用率 4%，如果允许使用三位密码，123 是绝大多数人的首选，hao123 的火爆也是一个例证

- LockDown.com 近日公布了一份采用“暴力字母破解”方式获取密码的“时间列表”。列表中按照密码长度、密码组合数量以及破解攻击模式来进行破解。

如果你用一台双核心 PC 破解密码，最简单的数字密码，六位（例如银行密码）瞬间搞定，八位 348 分钟，十位 163 天。普通大小写字母，六位 33 分钟，八位 62 天。数字+大小写字母，六位一个半小时，八位 253 天。数字+大小写字母和标点，六位 22 小时，八位 23 年。由此可知，对于我们普通人来说，考虑到容易记忆以及长度，八位数字+大小写字母密码是最安全方便的选择。另外，千万别用 6 位密码。

攻击模式解释：

A 级：10,000 Passwords/sec，在 Pentium 100 上破解 Office 密码的速度

B 级：100,000 Passwords/sec，在 Pentium 100 上破解 Windows Password Cache (.PWL Files)的速度

C 级：1,000,000 Passwords/sec，在 Pentium 100 上破解 ZIP 或 ARJ 压缩文档的速度

D 级：10,000,000 Passwords/sec，双核心处理器 PC

E 级：100,000,000 Passwords/sec，工作站级破解，或多个 PC 协同破解

F 级：1,000,000,000 Passwords/sec，大型计算机网络/超级计算机

- 一位安全专家在研究了 34000 名 MySpace.com 成员的注册信息后发现，互联网用户使用的密码比以往更加安全了，最长的竟然有 32 位之多。

在 Wired News 发布的一篇文章中，Counterpane Internet Security 首席技术官 Bruce Schneier 称，调查用户的平均密码长度是 8 位，其中 81% 同时包含数字和字母，而那个 32 位的超长密码是“1ancheste23nite41ancheste23nite4”。

Schneier 透露说，最常用的五个密码是：password1、abc123、myspace1、password 和 blink182(一个乐

大利亚新南威尔士的刑事犯罪学家和审问专家迈克尔·恩德斯说：“说谎非常困难，相当费神。任何东西只要可以削弱一个人执行这种脑力劳动的能力，就有可能被认为是一种‘吐真药’。”关于一个人在硫喷妥钠作用下提供的信息的可靠性问题是，一个人在药物麻醉的状况下，比他在正常情况下更容易受提问者的暗示影响。因此，一些被用药麻醉的人可能会做出错误答复，因为他们是根据一个有预谋或无意识的暗示做出的回答。药物信息访问：<http://baike.baidu.com/view/218531.htm>

队), 只有 3.8% 的密码是词典中的某个单词, 另外 12% 是单词加一位数字, 而这个数字有三分之二的几率是 1, 就像上边排名第一的 password1。

Schneier 调侃说: “以前我们认为最常用的密码是 ‘password’, 现在变成了 ‘password1’。谁说用户们的安全意识没有提高?” 随后他又认真地指出: “严肃地说, 密码的确越来越好了。只有不到 4% 是某个单词、绝大多数都是字母和数字的混合, 这让我很惊讶。”

Schneier 还表示, 密码的有效期(寿命)超过了其有效性。黑客的密码破解能力越来越高, 人们却始终不愿意记住一大堆密码, 而是更喜欢在不同的地方使用一个通用密码, 就连比尔·盖茨也不例外。不过还有一个问题, Schneier 是如何得到这些密码的呢? “嗨, 能告诉我你的密码么?” 显然不是这样。事实上, 他和一位安全界同事合作, 创建了一个虚假的 MySpace 登陆页面, 迷惑这 34000 人“乖乖交出”了自己的密码——典型的钓鱼欺诈。很显然, 密码的安全性不仅在于长度和复杂程度, 更在于主人的自我保护意识。

- 微软安全政策部门经理 Jesper Johansson 在澳大利亚黄金海岸 AusCERT 大会上的演讲中提出一项反传统的安全理念: 要让员工把密码写出来, 而不是仅仅保存在大脑。很多安全信息部门的重要守则之一就是“永远不许将你的密码写出来”, 事实证明, 这一观点是弊大于利, 有可能导致更严重的安全隐患。很多程序的漏洞会导致密码的泄露, “有多少安全规定中提到‘严禁写出密码, 违者必惩’?” Johansson 说道, “我认为, 这一规定大错特错。密码管理规则应该告诉你, 将你的密码写下来, 记录在案。我个人需要使用 68 个不同的密码帐户, 依靠记忆是不可能全部记住的。如果不让我将密码写下来, 我会如何做呢? 很显然, 我会为所有帐户设置同样的密码。只要有一个密码被破解, 整个安全体系都会瓦解。”

看来大家都害怕密码泄露, 如果你有很强的记忆力, 我有一个好方法推荐给你, 看看下面 2 个密码:

EG05p8DVkoS29CI54Bz83PnyD65KxQLjyTh6FjkIX0pM1n624Y58n9bPS7IUO5G0

ERA - GUN - zero - five - pen - eight - DOG - VOW - key - orb - SEA - two - nine - CUP - INK - five - four - BIT - zen - eight - three - PEN - net - yap - DOG - six - five - KEY - xmas - QUIZ - LEG - jug - yap - TIA - hat - six - FLY - jug - key - leg - XMAS - zero - pen - MAN - one - net - six - two - four - YAP - five - eight - net - nine - bit - PEN - SEA - seven - leg - UNIT - ORB - five - GUN - zero

l1hQ3TXE69JsDO63ZT059MF2I8e0m41j8ptA3R58FP7Nxm654R8KB39XryOE1NVd

leg - one - hat - QUIZ - three - TIA - XMAS - ERA - six - nine - JUG - sea - DOG - ORB - six - three - ZEN - TIA - zero - five - nine - MAN - FLY - two - INK - eight - era - zero - man - four - one - jug - eight - pen - tia - AIR - three - RAT - five - eight - FLY - PEN - seven - NET - xmas - man - six - five - four - RAT - eight - KEY - BIT - three - nine - XMAS - rat - yap - ORB - ERA - one - NET - VOW - dog

你看明白了吗? 可以记住吗? 每一个英文字母都由一个特殊的单词来决定, 有的是单词的首字母, 有的是单词的意思, 有的还包含大小写, 这个方法的好处是除了可以记住密码, 还可以背英文单词, 甚至如果有人威逼你说出口令, 一大堆的英文之后, 是无法确定口令的具体字符的。当然你也可能加入一些符号, 比如 ^0^^a这样的代码。

这时候, 很多人开始抱怨他的记忆力差, 根本记不住密码, 或者是写密码的纸条到哪去了。大家有没有

^a 是不是很像一个开怀大笑的人脸?

想过, 如果一个人搞到了你的密文和算法, 你想的是“一切都完蛋了”, 甚至是骗到了密码, 我们是否可以摆脱文字密码的束缚。

这里要稍微介绍一下暴力破解法的原理。破解者收集被破解着的信息, 再由专门的软件根据收集到的信息生成密码的一个数据库, 有点像字典, 将所有产生的密码都一个一个输入进去解密, 这就是所谓的“字典攻击”。

Daniel V.Klein 用这个系统能够破译一般计算机上 40% 的口令。试图登录时, 他并不是一个口令接一个口令的试验, 他把加密的口令文件抄下来然后进行离线攻击。下面是他所试验的:

- 1) 用户的姓名、简写字母、帐户姓名和其他有关的个人信息都是可能的口令, 基于所有这些信息可以尝试到 130 个口令。对于一个名叫“Daniel V.Klein”, 帐户名为“Klone”的用户, 用来尝试口令的一些词是 klone, klone0, klonel, klone123, dvk, dvkdvk, dklein, Dklein, leinad, nielk, dvklein, danielk, DvkkvD, DANIEL-KLEIN, (klone), KleinD 等等。
- 2) 使用从各种数据库中得到的单词。这些单词是男人和女人的姓名名单(总共约达 16,000); 地点(包括像“spain”、“spanish”和“spaniard”这样的排列也全被考虑在内); 名人的姓名; 卡通漫画和卡通人物; 电影和科幻小说故事的标题、有关人物和地点; 神话中的生物名字(从《Bulfinch 神话故事》和神话动物字典中产生出的); 体育活动(包括球队名、一些浑名, 和职业队名称); 数字(比如“2001”和写出的“twelve”); 一串字母和数字(“a”“aa”“aaa”“aaaa”, 等等); 中文音节(选自汉语拼音字母或在英文键盘上输入中文的国际标准系统); 《圣经》的权威英译本; 生物术语; 公用的粗话(如“fuckyou”、“ibmsux”和“deadhead”); 键盘模式(如“qwerty”、“asdf”和“zxcvbn”); 缩写(如“roygbiv”——彩虹的七种颜色和“ooottafagvah”——帮助记忆头部十二条神经的东西); 机器名称(可从 letc/hosts 中获得); 莎士比亚作品中的人物、戏剧和地点; 常用的犹太语; 小行星名称和 Klein 以前出版的技术论文中搜集到的单词。综上, 每个使用者可以考虑超过 66,000 个独立的单词(舍弃字典内外复制的那些)。
- 3) 第(2)步得到的单词的不同置换形式。这包括使第一个字母大写或作为控制符, 使整个单词大写, 颠倒单词的顺序(不管前面有无大写), 将字母“O”换成数字“0”(使得单词“scholar”变作“sch0lar”), 将字母“I”换成数字“1”(使单词“scholar”变成“scho1ar”), 以及进行同样操作将字母“Z”换成数字“2”, “S”换成“5”。另一种测试是将单词变为复数形式(不管它是否为名词), 非常聪明的将“dress”变为“dresses”、“house”变为“houses”, 并且“daisy”变为“daisies”, Klein 并不考虑复数规则, “datum”可以变为“datums”(不是“data”)“sphynx”变为“sphynxs”(而不是“sphynges”), 同样地, 将后缀“-ed”, “-er”和“-ing”加到单词上, 如“phase”变为“phased”, “phaser”和“phasing”。这些附加的测试使得每一位使用者可能的口令清单增加了的 1,000,000 个单词。
- 4) 从第(2)步得到的单词的不同的大写置换形式, 不考虑第(3)步。这包括所有单字母的单个大写置换(如“michael”可换为“mIChael”, “miChael”, “michAel”等等)双字母大写置换(“MIChael”, “MiChael”, “MicHael”……“mIChael”, “mIcHael”等等)。三字母置换, 等等。对于每一个使用者, 单字母置换增加了大约 400,000 个单词, 双字母置换增加 1,500,000 个单词, 三字母置换增加至少 3,000,000 个单词。必须要有足够的时间来完成测试, 测试完成 4, 5, 6 个字母的置换没有充足的计算机“马力”是不可能的。
- 5) 对外国用户要尝试外语单词, 对有中文名称的用户要使用中文口令来进行特别的测试。汉语拼音字母组成单音节、双音节或三音节的单词, 但由于不能测试确定它们是否是实际存在的, 所

以要启动穷举搜索。(在汉语拼音中共有 298 个音节, 158,404 个双音节词, 稍多于 16,000,000 个三音节词。) 一种类似的攻击方式, 就是穷举构造出来的可以发音但并不存在的单词, 可以很容易的被用于英语中。

- 6) 尝试词组。自然测试所耗费的数量是令人惊愕的。为了简化测试, 只有在用户口令生成字典中存在, 且仅有 3, 4 个字母长的才被测试。即使这样, 词组数目也有千万。

当字典攻击被用作破译密钥文件而不是单个密钥时就显得更加有力。单个用户可以很机灵地选择到好密钥, 如果一千个人各自选择自己的密钥作为计算机系统的口令, 那么至少有一个人将选择攻击者字典中的词作为密钥。

运气好的话, 在 n 次之后你就可以看到打开了。运气不好的话, n 就趋近正无穷了。一般这个过程由专门的软件去进行, 所以速度有了很大的提高。有方法提高这个“运气值”, 如果这个人设置的口令码是由生日、电话号码、身份证号码、车牌号、幸运数字等等组成的话, 一个很熟悉他的人就可以获得一切信息^a, 经过软件排列组合后可以大大提高口令码的命中率。

所以我在这里的建议是这个密码要和自己没有任何关系, 然后使用数字、符号、字母(可以同时使用大小写)自由的搭配, 长度尽可能的长, 当然要在你能记住的前提下, 然后可以在相隔一段时间后更换口令码的字符。确保这个秘密永远是属于你的。如果真的记住这个很困难的话, 我们还有下面的技术。通常是一种和几种组合起来使用的。

4.1.13 生物识别

生物识别技术指通过人类生物特征进行身份认证的一种技术, 这里的生物特征通常具有唯一的(与其他人不同)、可以测量、或可自动识别和验证、遗传性或终身不变等特点。生物识别的核心在于如何获取这些生物特征, 并将之转换为数字信息, 存储于计算机中, 利用可靠的匹配算法来完成验证与识别个人身份的过程。你可以在很多的电视电影中看到这些技术。

生物识别的涵义很广, 大致上可分为身体特征和行为特征两类。

身体特征包括: 指纹、掌型、视网膜、虹膜、人体气味、脸型、甚至血管、DNA、骨骼等; 行为特征则包括: 签名、语音、行走步态等。生物识别系统则对生物特征进行取样, 提取其唯一的特征转化成数字代码, 并进一步将这些代码组成特征模板, 当人们同识别系统交互进行身份认证时, 识别系统通过获取其特征与数据库中的特征模板进行比对, 以确定二者是否匹配, 从而决定接受或拒绝该人。

下表对四类主要的人体生物特征的自然属性进行了比较

表格 8 人体生物特征的自然属性比较

自然属性	虹膜	指纹	面部	DNA
唯一性	因人而异	因人而异	因人而异	亲子相近同卵双胞胎相同
稳定性	终身不变	终身不变	随年龄段改变	终身不变
抗磨损性	不易磨损	易磨损	较易磨损	不受影响
痕迹残留	不留痕迹	接触时留有痕迹	不留痕迹	体液、细胞中含有
遮蔽情况	可戴手套面罩	不能戴手套	不能戴手套	无

从上表列出的特性可以看出, 某一应用领域可能特别需要某种生物特征, 如刑侦应用与指纹识别、亲子

^a 不排除你的身边的人是间谍。

鉴定与 DNA 等。与其他生物特征相比,虹膜组织更适合于信息安全和通道控制领域。例如,虽然多种特征都具有因人而异的自然属性,但虹膜的重复率极低,远远低于其它特征。又如,容易留痕迹可以给刑侦带来很大方便,但痕迹易被他人利用来造假,则不利于信息安全。再则,虹膜相对不易因伤受损,更加大大减少了因外伤而导致无法进行识别的可能性。下面有一些生物特征识别方式:

1. 指纹识别^a

指纹是指人的手指末端正面皮肤上凸凹不平产生的纹线。纹线有规律的排列形成不同的纹型。纹线的起点、终点、结合点和分叉点,称为指纹的细节特征点。指纹识别即指通过比较不同指纹的细节特征点来进行鉴别。由于每个人的指纹不同,就是同一人的十指之间,指纹也有明显区别,因此指纹可用于身份鉴定。

指纹识别技术是目前最成熟且价格便宜的生物特征识别技术。目前来说指纹识别的技术应用最为广泛,我们不仅在门禁、考勤系统中可以看到指纹识别技术的身影,市场上有了更多指纹识别的应用:如笔记本电脑、手机、汽车、银行支付都可应用指纹识别的技术。

2. 静脉识别^b

静脉识别系统就是首先通过静脉识别仪取得个人静脉分布图,从静脉分布图依据专用比对算法提取特征值,通过红外线 CCD 摄像头获取手背静脉的图像,将静脉的数字图像存贮在计算机系统中,将特征值存储。静脉比对时,实时采取静脉图,提取特征值,运用先进的滤波、图像二值化、细化手段对数字图像提取特征,同存储在主机中静脉特征值比对,采用复杂的匹配算法对静脉特征进行匹配,从而对个人进行身份鉴定,确认身份。全过程采用非接触式。

3. 虹膜识别^c

虹膜是位于人眼表面黑色瞳孔和白色巩膜之间的圆环状区域,在红外光下呈现出丰富的纹理信息,如斑点、条纹、细丝、冠状、隐窝等细节特征。虹膜从婴儿胚胎期的第 3 个月起开始发育,到第 8 个月虹膜的主要纹理结构已经成形。除非经历危及眼睛的外科手术,此后几乎终生不变。

虹膜识别通过对比虹膜图像特征之间的相似性来确定人们的身份,其核心是使用模式识别、图像处理等方法对人眼睛的虹膜特征进行描述和匹配,从而实现自动的个人身份认证。英国国家物理实验室的测试结果表明:虹膜识别是各种生物特征识别方法中错误率最低的。

从普通家庭门禁、单位考勤到银行保险柜、金融交易确认,应用后都可有效简化通行验证手续、确保安全。如果手机加载“虹膜识别”,即使丢失也不用担心信息泄露。机场通关安检中采用虹膜识别技术,将缩短通关时间,提高安全等级。

4. 视网膜识别

视网膜是眼睛底部的血液细胞层。视网膜扫描是采用低密度的红外线去捕捉视网膜的独特特征,血液细胞的唯一模式就因此被捕捉下来。

视网膜识别的优点就在于它是一种极其固定的生物特征,因为它是“隐藏”的,故而不可能受到磨损,老化等影响;使用者也无需和设备进行直接的接触;同时它是一个最难欺骗的系统,因为视网膜是不可见的,故而不会被伪造。另一方面,视网膜识别也有一些不完善的,如:视网膜技术可能会给使用者带来健康的损坏,这需要进一步的研究;设备投入较为昂贵,识别过程的要求也高,因此角膜扫描识别在普遍推广应用上具有一定的难度。

5. 面部识别^a

^a 指纹识别(fingerprinting) 访问: <http://baike.baidu.com/view/7245.html?wtp=tt>

^b 静脉识别的原理访问: <http://baike.baidu.com/view/1315106.htm>

^c 虹膜识别的原理访问: <http://baike.baidu.com/view/831985.htm?func=retitle>

面部识别是根据人的面部特征来进行身份识别的技术, 包括标准视频识别和热成像技术两种。

标准视频识别是透过普通摄像头记录下被拍摄者眼睛、鼻子、嘴的形状及相对位置等面部特征, 然后将其转换成数字信号, 再利用计算机进行身份识别。视频面部识别是一种常见的身份识别方式, 现已被广泛用于公共安全领域。

热成像技术主要透过分析面部血液产生的热辐射来产生面部图像。与视频识别不同的是, 热成像技术不需要良好的光源, 即使在黑暗情况下也能正常使用。

6. 手掌几何学识别^a

手掌几何学识别就是通过测量使用者的手掌和手指的物理特征来进行识别, 高级的产品还可以识别三维图象。作为一种已经确立的方法, 手掌几何学识别不仅性能好, 而且使用比较方便。它适用的场合是用户人数比较多, 或者用户虽然不经常使用, 但使用时很容易接受。如果需要, 这种技术的准确性可以非常高, 同时可以灵活地调整性能以适应相当广泛的使用要求。手形读取器使用的范围很广, 且很容易集成到其他系统中, 因此成为许多生物特征识别项目中的首选技术。

7. DNA 识别

人体内的 DNA 在整个人类范围内具有唯一性 (除了同卵双胞胎可能具有同样结构的 DNA 外) 和永久性。因此, 除了对同卵双胞胎个体的鉴别可能失去它应有的功能外, 这种方法具有绝对的权威性和准确性。DNA 鉴别方法主要根据人体细胞中 DNA 分子的结构因人而异的特点进行身份鉴别。这种方法的准确性优于其它任何身份鉴别方法, 同时有较好的防伪性。然而, DNA 的获取和鉴别方法 (DNA 鉴别必须在一定的化学环境下进行) 限制了 DNA 鉴别技术的实时性; 另外, 某些特殊疾病可能改变人体 DNA 的结构组成, 系统无法正确的对这类人群进行鉴别。

8. 声音识别

声音识别属于行为识别的范畴。声音识别主要是利用人的声音特点进行身份识别。声音识别的优点在于它是一种非接触识别技术, 容易为公众所接受。但声音会随音量、音速和音质的变化而影响。比如, 一个人感冒时说话和平时说话就会有明显差异。再者, 一个人也可有意识地对自己的声音进行伪装和控制, 从而给鉴别带来一定困难。

9. 步态识别

步态识别技术现还处在初期阶段, 其发展还面临许多艰难的挑战。这项技术的最新进展在由美国国防先进研究项目代表设立基金研究通过人体语言确认人的身份的美国科研机构中。其理论是每个人以相同的方式生活, 都有自己专一的信号或指纹, 每个人也有自己专一的走路步伐。其技巧是收集人体语言并把它转化为计算机能识别的数字。

一种方法每个人建立“运动信号”来识别。他们从拍摄人走路或跑步的方法开始研究每个人的运动信号, 再利用计算机上的模拟照相机捕捉和储存这一运动行为 (用软件工具除去冗余最终只以数字形象储存物体的一系列轮廓)。之后只要一个人把他的整个走路过程拍摄下来, 指令计算机就能根据储存的形象确定这个人的身份。通过系统很好地归纳所有不同的步伐后, 据称现已获得 90%~95% 的正确匹配。”

另一种方法则是使用结构分析方法去测定一个人的跨步和腿伸展特性。

这两种技术迄今所有的数据库形象是两维的, 并很大程度上取决于照相机的角度。当一个系统企图采用不同的角度去比较同一个人两个镜头时, 就会出现困难。很大程度上直接限制了它的发展!

^a 面部识别的原理: <http://baike.baidu.com/view/246859.html>, 其它信息:

<http://www.hudong.com/wiki/%E9%9D%A2%E9%83%A8%E8%AF%86%E5%88%AB%E7%B3%BB%E7%BB%9F>。

^b 掌纹识别的信息访问: <http://baike.baidu.com/view/1888819.html>

10. 皮肤芯片

这种方法通过把红外光照进一小块皮肤并通过测定的反射光波长来确认人的身份。其理论基础是每个具有不同皮肤厚度和皮下层的人类皮肤,都有其特有的标记。由于皮肤、皮层和不同结构具有个性和专一特性,这些都会影响光的不同波长,目前 Lumidigm 公司开发了一种包含银币大小的两种电子芯片的系统。第一个芯片用光反射二极管照明皮肤的一片斑块,然后收集反射回来的射线,第二个芯片处理由照射产生的“光印”(light print)标识信号。相对于指纹(Fingerprinting)和面认(Face recognition)所采用的采集原始形象并仔细处理大量数据来从中抽提出需要特征的生物统计学方法^a,光印不依赖于形象处理,使得设备只需较少的计算能力。

生物识别技术是目前最为方便与安全的识别技术,它不需要记住复杂的密码,也不需随身携带钥匙、智能卡之类的东西。生物识别技术认定的是人本身,这就直接决定了这种认证方式更安全、更方便了。由于每个人的生物特征具有与其他人不同的唯一性和在一定时期内不变的稳定性,不易伪造和假冒,所以利用生物识别技术进行身份认定,安全、可靠、准确。此外,生物识别技术产品均借助于现代计算机技术实现,很容易配合电脑和安全、监控、管理系统整合,实现自动化管理。

生物特征识别系统在利用个人特征来鉴别或验证用户身份时,如果有生物特征被察觉或检测到是“有噪音的”(比如指纹中带有疤痕或者因感冒而改变声音时),这个生物特征识别系统的性能可能会受到损害,此时的匹配评分计算是不可靠的。这个问题可以通过安装多种传感器捕捉不同的生物特征来解决,这也被称为生物特征融合或多模态生物特征识别系统。

基于多模态或多生物特征融合的解决方案代表了一个新兴趋势,某些应用会比单一方法的识别系统具有更好的技术性能。国际标准化组织(ISO)和国际电工委员会(IEC)已经联合公布了《信息技术—生物特征—多模态和其他多生物特征融合》(ISO/IECTR24722:2007),该方案能融合多种生物指令,以保证在一种生物特征失真的情况下,仍能顺利识别。新的 ISO/IECTR24722:2007 不但包含了目前就多模态和多生物特征融合做法的描述和分析,它还研讨了需求、可能的路径和标准化来支持多生物特征识别系统,以提高其通用性和实用性。这项最新的 ISO/IEC 技术报告提供了多模态和其他多生物识别系统的总的概述,并给出了关于多生物特征融合的一个参考——生物特征识别系统需要一种以上的生物模态。

多生物特征解决方案的潜在好处是能延伸到人体进入控制区域从而获取敏感数据。这样,就能使生物特征识别系统更安全,入侵者用人造物或模仿品来同时骗过多生物特征基本上是不可能的,而个人在某项特征不便时亦可灵活调换。

在国内,由清华大学丁晓青教授组研制的 TH-ID 系统多模式生物特征(人脸、笔迹、签字、虹膜)身份认证识别系统已通过教育部组织的专家鉴定。能够实现在复杂背景下的图像和视频人脸自动检测、识别和认证,在人脸、笔迹、签字、虹膜的识别认证技术上取得了重要进展,在整体上达到了国际领先水平。

该系统包括两大部分内容:人脸、笔迹、签字和虹膜四种生物特征的身份认证(识别和验证)的四个子系统 and 利用多种生物特征的多模生物特征融合的身份认证系统。他们构建了基于统一数据库的人脸、笔迹、签字、虹膜四种生物特征的多模生物特征身份识别认证系统,能够进行融合模式的选择,进行各种可能的模式融合。可以有效克服单一生物特征常有的缺陷,极大地提高了身份认证的准确度,从而也为生物特征身份认证的实际应用打下了坚实的基础。

目前,生物识别技术在生活方面主要有三大应用方向:

1. 作为刑侦鉴定的重要手段;
2. 满足企业安全、管理上的需求(例如物理门禁、逻辑门禁、考勤、巡更等系统,已经全面引入生物识别技术)

^a 参阅“Face Recognition”/TR Nov 2001

3. 自助式政府服务、出入境管理, 金融服务、电子商务, 信息安全(个人隐私保护) 方面。

生物识别应用之发展潜力和背景, 在现阶段的中国, 主要体现在以下几个方面: 首先, 巨大的人口基数, 以及越来越频繁的流动性。这其中不论静态管理还是动态控制, 身份识别当然是首要因素。其次, 经济全球化背景下, 中国产生的数量庞大、规模超凡的世界工厂的安全和管理, 亦是生物识别的用武之地。另外, 经济全球化带来更直接的影响, 是频繁的个人身份认证的需求。再次, 电子商务和电子政务的演变和普及中生物识别, 是现阶段及可预见的将来最佳的解决方案。

在所有的技术中, 现阶段更受瞩目的并迅速发展是人脸识别。它目前主要有三种应用模式:

1. 人脸识别监控, 即将需要重点关注的人员照片存放在系统中, 当此类人员出现在监控设备覆盖的范围中时系统将报警提示。此种模式主要应用在奥运通道安检、地铁等需要实时预警的地点。
2. 人脸识别比对检索, 即利用特定对象的照片与已知人员照片库进行比对, 进而确定其身份信息。能够解决传统人工方式工作量巨大、速度慢、效率低等问题, 可应用在网络照片检索、身份识别等环境。适合于机场等人员流动大的公众场所, 但需要大型数据库的支持。
3. 身份确认, 即确认监控设备和照片中的人是否是同一人。可广泛应用于需要身份认证的场所, 如自助通关、银行金库、门禁以及需要实行实名制管理的业务, 如银行业务等。

据悉 2008 年北京奥运会全面运用人脸识别系统人群, 快速辨认和甄别恐怖分子和其他可能引发犯罪的人员, 以防止其进入相关的敏感区域。

在生物特征识别领域近些年的发展中, 国际上虽然制定了一些标准, 但远不够完善。鉴于我国目前的技术在国际上也属于领先水平, 因此在现阶段我们有必要抓紧制定国内标准, 掌握主动权。在可预见的将来, 我们有理由对自己有信心, 对我们所在的生物特征识别领域有信心! 中国在这一方面必将取得喜人的成就!

4.1.14 从这里继续

我们已经看了如何把这些部件组合成复杂的密码系统, 也看到了原文如何变成不可阅读的密文, 还看到了转化保护口令的众多手段。当然, 还有很多细节我们没有去看, 有一些其它的过程已经被我们忽略了。它们中的许多也仅仅是一些我们见过的系统框架的变化而已。

例如, 我们没有关注你的私钥是如何加密你的密码的。这是高级课题。无论如何, 我们已经看了所有要做的组件, 可以去考虑它们了。获取密码时使用 `hash` 算法获得对称密钥, 然后加密私钥。很简单! 当然实际要比这个详细的多, 如果这种东西使你着迷了, 你就回去找这些信息。我已经用了很多链接(举例)告诉你 OpenPGP 如何工作, 你只要上网在浏览器中打开这些链接, 查找"string to key"在 RFC 2440 中。你马上就可以获得信息。

5 未来的密码技术

预测是很困难的，特别是关于未来的预测。

——Niels Bohr

我们已经看到密码学的由来，但是如何发展？当然，这很难说，但是我们可以讨论它的趋势。我们也谈论什么能够影响当前潮流的趋势。这给出了我们发展的方向、什么使我们发生语言错误，有多大的可能。

本章节我们要谈论一些未解决的问题：如何出了面对的挑战。在它们之中，我们忽略了一些干扰因素，如果不忽略，它们也就什么都不是了，留下一大片空间，使它表面的尘土被拂去，但当我们返回前面的主题时。真诚的提出这些问题，但还没有好的方法去解决。所作的仍旧是权衡。看看我们在章节中有趣的和令人惊奇的问题。

5.1 名词到形容词，语法到语义学

我们持续的动力使得产生更多的密码技术，会有越来越多的密码技术被植入更多的地方。我们使用密码技术将会改变我们的生活，这是一个大的趋势。它在名词的含义上越来越少，形容词上的含义越来越多；新东西会越来越少，技术的修改和更新变的越来越多了。这就是所有技术的自然发展。伴随着伟大技术的复杂化，很多互相交错的学科也出现了。

100 年前一个广播是新出现的，那是个极具吸引人的东西。它是无线电，一个专有名词用来做定冠词^a。现在，她渗透了，它的英文是小写的形容词，不是大写的名词。我口袋有一个无线电话可以通过无线连接和电脑通话^b，比替代短距离线的技术——蓝牙（Bluetooth）要好的多。我的电脑是通过无线网络连接到互联网的。无线电技术是一个重要性质，而不是它外表和内部一样有趣。使用无线电技术的设备已经开始蔓延，使用无线电来解决一些琐碎的东西，比如替换键盘信号线、鼠标信号线。这在以前是很复杂的。现在只有一个还使用的有线，那就是电源。我们的笔记本可以使用无线电力^d，这个发展就会成功。燃料电池（fuel cells）替代传统电池也许是除去线的方式之一。

在密码技术里面，相似的改变是我们不用关心语法，而更多关系语义^e。密码技术是一个关于语言 and 如何使用这个语言的技术，这个结构变的更微妙。密码技术的内部和本身是无趣的；而我们用密码技术做出的事情是很有趣的。

5.1.1 社会期望

未来初始动力是我们期待信息能像文明一样的被对待，和更多的依靠它。我们期待人们买东西更小心。我们期待健康会更好，还有更多复杂的任务，时刻都被保护，因为健康信息必须是有求必应。各类组织都必

^a 广播开始是很多个扬声器中间有线连接的，像现在家庭影院音响线路一样。现在的广播是无线电技术的，也就是无线电广播、收音机。

^b VoIP, Voice over Internet Protocol。一种由 IP 网络传送话音的技术服务。电话和信号站用 2.4Ghz 的无线频率传输。范围比较小（<100m）

^c 在我们这个时代的互联网（Internet）是小写的名词也是这样的发展。仅仅引起争论的是讨论互联网的信息而不是互联网。

^d 无线电源出现在 2008 年秋季 IDF（Intel Developer Forum，英特尔信息技术峰会）上。Intel 展示了一套由无线发射器和无线接收器的系统。靠近发射器一定距离，接收器就有电力。与 WiFi 技术不同，它能感知距离，超出范围将不再尝试发送电能。2-3 英尺内传输效率约 75%

^e 语义学：研究字符或字符组同其含义之间的关系(而与它们的解释方法和用法无关)的一门科学。

须保护好自己的数据信息。

欧洲是法律和社保最好的区域。目前, 到来的美国违规警告法律条文在保护信息方面显得有深远意义。他们没有规定操作程序, 在操作错误时只有透明性。这些所有的需求都基于一个主观点: 加密的信息就是受保护的信息。加密技术变成了一个保护包含信息本身等所有的运载工具。

未来密码技术仍然以人为本, 我们也依靠法律、法规和社会期望。现在还没有明确的一些东西。形式上, 法律法规通过细化和讨论将相对比较完善。最好我们现在就已经有初步草案。会有很多毛边让我们去修剪。那些毛边就是未解决的问题, 现在我们来看看它们到底是什么。

5.2 数字签名与语义学

我们使用密码技术去数字签名的过程中有很多问题。我们在未来将会更多的使用数字签名, 不仅仅是代替手写签名, 还有很多数据保护方面的。

5.2.1 数字签名并非签名

数字签名是我们保护数据时的灵活的工具。名字上有一个问题, 它不是签名。签名是个密码技术上的动作, 不是你签信用卡时的笔迹。你签了以后就意味着产生了法律效力。你签什么是次要的: 就像传说中的一样 X, 佐罗的 Z, 或其它的。我对在一些现金出纳机的签名扫描仪觉的很麻烦。我在里面尽可能快的写了一个“签名”, 如果签名是一系列线段时, 它被识别的就不是很好。在我看来其它的方式和签名是一样的。噢~是的, 我感觉我侥幸逃脱了什么, 因为没有人用那个叫我。没有人注意那个是不是和我卡后面的签名相同。有趣的是这个签名很少用来比对。我知道的很多人的签名在安全方面没有拿去检查, 尽管卡后写的“必须检查签名”。你可以看到一个令人高兴的话题讨论在这^a, John Hargrave 找到令人不可容忍的挑战来推测签名。

他不是一个胡闹, 这个词和我们观察是相反的。签名是一个技术, 不是一个事物。比较签名不仅仅是必要的安全检查, 对比也是很重要的^b。

Alice 想对数字信息签名, 并送给 Bob。在 Trent 和对称密码系统的帮助下, 她能做到。

Trent 是一个有权的、值得依赖的仲裁者。他能同时与 Alice 和 Bob(也可以是其他想对数据文件签名的任何人)通信。他和 Alice 共享秘密密钥 KA, 和 Bob 共享另一个不同的秘密密钥 KB。这些密钥在协议开始前就早已建好, 并且为了多次签名可多次重复使用。

(1) Alice 用 KA 加密她准备发送给 Bob 的信息, 并把它传送给 Trent。

(2) Trent 用 KA 解密信息。

(3) Trent 把这个解密信息和他收到 Alice 信息的声明, 一起用 KB 加密。

(4) Trent 把加密的信息包传给 Bob。

(5) Bob 用 KB 解密信息包, 他就能读 Alice 所发的信息和 Trent 的证书, 证明信息来自 Alice。

^a John Hargrave 的《The Credit Card Prank (信用卡的闹剧)》<http://www.zug.com/pranks/credit/>。

^b 我认为为什么说签名是一个很简单的风险回报事情的原因是: 职员欺诈的可能有多大。一个大客户的误报可能导致他们以后在其它地方购物, 误报——一直被认为是错误收费。这不是职员承担赔偿责任的费用, 顾客比职员要多的多。没有其它的证据, 这个职员最好不承担这个责任。与你想的相反, 如果有人签名“Mickey Mouse (米老鼠)”更像是有人想让职员掉受处分, 而不是欺诈。

Trent 怎么知道信息是从 Alice 而不是从其他人冒名顶替者那里来的呢? 从信息的加密推断出来。由于只有他和 Alice 共享他们两人的秘密密钥, 所以只有 Alice 能用这个密钥加密信息。

这和文件签名一样好吗? 让我们看看我们需要的特点:

- (1) 这个**签名是可信的**, Trent 是可信的仲裁者, 并且知道信息是从 Alice 那里来的, Trent 的证书对 Bob 起着证明的作用。
- (2) 这个**签名是不可伪造的**。只有 Alice(和 Trent, 但每个人都相信他)知道 KA, 因此只有 Alice 才能把用 KA 加密的信息传给 Trent。如果有人冒充 Alice, Trent 在第(2)步马上就会察觉, 并且不会去证明它的可靠性。
- (3) 这个**签名是不能重新使用的**。如果 Bob 想把 Trent 的证书附到另一个信息上, Alice 可能就会大叫受骗了。仲裁者(可能是 Trent 或者可存取同一信息的完全不同的仲裁者)就会要求 Bob 同时提供信息和 Alice 加密后的信息, 然后仲裁者就用 KA 加密信息, 他 马上就会发现它与 Bob 提供的加密信息不相同。很显然, Bob 由于不知道 KA, 他不可能提供加密信息使它与用 KA 加密的信息相符。
- (4) **签名文件是不能改变的**。Bob 想在接收后改变文件, Trent 就可用刚才描述的同样办法证明 Bob 的愚蠢行为。
- (5) **签名是不能抵赖的**, 即使 Alice 以后声称她没有发信息给 Bob, Trent 的证书会说明不是这样。记住: Trent 是每个人都信任的, 他说的都是正确的。

如果 Bob 想把 Alice 签名的文件给 Carol 阅读, 他不能把自己的秘密密钥交给她, 他还得通过 Trent:

- (1) Bob 把信息和 Trent 关于信息是来自 Alice 的声明用 KB 加密, 然后送回给 Trent。
- (2) Trent 用 KB 解密信息包。
- (3) Trent 检查他的数据库, 并确认原始信息是从 Alice 那里来的。
- (4) Trent 用他和 Carol 共享的密钥 KC 重新加密信息包, 把它送给 Carol。
- (5) Carol 用 KC 解密信息包, 她就能阅读信息和 Trent 证实信息来自 Alice 的证书。

这些协议是可行的, 但对 Trent 来说是非常耗时的。他不得不整天加密、解密信息, 在彼此想发送签名文件的每一对人之间充当中间人。他必须备有数据库信息(虽然可以通过把发送者加密的信息的拷贝发送给接收者来避免)。在任何通信系统中, 即使他是毫无思想的软件程序, 他都是通信瓶颈。

更困难的是产生和保持像 Trent 那样的网络用户都信任的人。Trent 必须是完善无缺的, 即使他在 100 万次签名中只犯了一个错误, 也将不会有人再信任他。Trent 必须是完全安全的, 如果他的秘密密钥数据库泄露了, 或有人能修改他的程序代码, 所有人的签名可能是完全无用的。一些声称是数年前签名的假文件便可能出现, 这将引起混乱, 政府可能倒台, 混乱状态可能盛行。理论上这种协议或许是可行的, 但实际上不能很好运转。

一个数字签名则不是技术, 而是一个物。它是我们期望只能个人产生的东西, 它也是真正的数学化的、

密码计算的。我们假定这个私钥的“主人”是唯一“知道”密钥的人，我们来看看：人类不可能知道密钥。那个密钥的拷贝是采取了一些技术的，技术也是不知道密钥的^a。而且人类也无法做这个创造密钥的数学题。现在我们有机器可以替我们完成，也就意味着我们需要对这个软件做的正确的事情有绝对的信任。在我们使用墨水前，我们可以很容易的通过检查纸张来发现；当我们的文档变成数字文档时，这个检查就不现实了。

我们中的很多人认为数字签名不是技术，我将会证明给你看。在 PGP 公司，我们给图片签名。实际上是往图片中嵌入墨水产生的图像。当我旅行时一些商函需要寄出去，PGP 公司可以写下来用我的签名图像在上面打印，这样表示是这个从我手中发出的。这个过程也没什么错误，它没有给你授权？确实，当我在飞机上时，我用黑莓手机（装有 PGP 软件）给议会发送文本信，他们怎样把信头和信尾接合？它的定义延伸了，因为数字签名是一个技术，不是一个物。

一个数字签名更像一个蜡封，而不是文字签名。当他们担心数字签名的安全时，我们涉及的一个封闭的事物。可以伪造一个相似签名吗？封印本身可以提取到其它文档中吗？如果我们回来的及时，最好还是找一个数字签名的签名封印，但是我们时找不到数字签名的相似物的。因此我们必须注意它们两者是不一样^b的。

表格 9：各种算法区别

算法	机密性	验证	完整性	密钥管理
对称加密算法	YES	NO	NO	YES
公钥加密算法	YES	NO	NO	YES
数字签名算法	NO	YES	YES	NO
密钥共识算法	YES	可选	NO	YES
单向 Hash 函数	NO	NO	YES	NO
信息验证密码	NO	YES	YES	NO

5.2.2 认可的神话

无论你什么时候读关于数字签名的东西，总是在说数字签名提供认可的特性，签名人不可能说他们没有在上面签名。它是个神话，至少是个数学家的幻想。如果作为一个幻想，不是一个坏的幻想。Leibniz 乐观的支持我们决定讨论开始的和我们所说的并形成这些逻辑观点，“让我们一起计划”每人都处理一小部分争论就可以转移，就像是 Godel 毁灭 Leibniz 的学说^c，看看这个严正主张，什么是认可？

举个例子，Bob 有一个 Alice 密钥数字签名的文档。Alice 闯入 Bob 办公室说“看，我知道那个是我的签名，我不否认，但是我没有签过那个信，我没有做过，我也不知道是怎么回事！”。

Alices 说的是真的吗？她真的没有给文档签名？和这个认可的意思和信任的意思很接近，如果她使用 1024-bit 的 RSA 密钥和 160-bit 的 hash，她就没有说实话。她说实话的机率^d是 2^{-80} 。这里有更多的安全方法，作为认可依据的一个设定值。我们假设只有 Alice 有她的私钥。我们假设 Alice 的软件没有错误、系统没有间

^a George A. Miller, 《The Magical Number Seven, Plus or Minus Two (魔法数字七, 加或减 2)》: 一些我们在一些信息处理能力方面的限制, 最初发表在《The Psychological Review (心理回顾)》1956 年, 卷 63, 81-97 页, Stephen Malinowski 的文章, 参阅 <http://www.well.com/~s malin/miller.html>。

^b 这个混淆不是新的。你使用图章盖一个章形成一个蜡印章和签名的道理一样，我不说那也是很明显的。Jon Callas 和 Bruce Schneier 的《Why Digital Signatures Are Not Signatures (为什么数字签名不是签名)》的工业标准, 2000 年发布。Bruce 的站点有文章 <http://www.schneier.com/crypto-gram-0011.html#1>。

^c 关于数学系 Godel 的信息, 访问: <http://baike.baidu.com/view/551541.htm>。

^d 注意, 数学中的机率就是一个客观的数字, 只表明宏观现象的预测。哪怕是 0.000001% 的机率, 这个事件仍然可能发生。没准就是下一个! 另外还有一个词“几率”, “几率”和“机率”有区别: 正统来说应为“机率”, 而“几率”则是“机率”之误。不过语言本身就是创造出来的, 所以用的多了之后, “几率”也就转正了。“机率”是数学概率论的基本概念, 是对随机事件发生的可能性的度量。

谍软件什么的。我们假设她的密钥上的密码有至少 80-bit 的强度。这也只是开始。我们也假定当 Alice 签名的时候她知道自己签的是什么文件。

实际上, 她被黑客攻击的机会, 或者是点错了按钮, 或是她女儿乱点的, 或者是秘书做的, 或者她认为给什么东西签名很完美。它的机率就比 2^{-80} 高了。无疑是软件使用 Alice 的密钥做的签名。数学上这是个合理的解释。也不容易判断软件是不是在 Alice 的控制之下, 或者它是和 Alice 的知识、了解、认可后完成的。它取得了决定权。我们必须从大处入手。如果签名信息是订购一本《the Brooklyn Bridge》, 我们可以以 10 美元一顿午饭的代价否认订单。

认可仅仅是一个数学想法。如果某人遗弃了这个签名, 我们要求委托检查总目录, 把无用的移除。数学中这个提供人类授权的行为是个神话。

5.2.2.1 高强度算法的反驳

我考虑认可的时候还有一个有趣的事。如果 Alice 的签名有 80 bits 的强度, 我们怀疑人机对话装置可能有间谍软件。或者是小孩、也可能是家里其它的问题。我们经常问“Alice 说谎或她被黑客攻击了可能吗?”, 但是他什么时候用 256 bits 强度的密钥签的名? 逻辑上我们不会, 是否考虑了不到 2^{-80} 而高于 2^{-256} 的概率? 人们对安眠药、做饭、购物有很坏的反应^a, 其它国家的人可能开了一个玩笑吗? 如果机率有 2^{-240} , 我们不考虑备用解释?

是的, 我被外星人恶作剧了, 我认为这是要点。如果我们担心外在, 强度更高的密钥可以使系统更安全。因为 Alice 愿意请我吃饭, 我进入了密码的阴影区域。她坚持不使用签名的原因可能就是她待 Coyote^b如友的原因。我会帮她扫描系统间谍软件和惊讶她女儿的幽默感, 但是如果把那些假定移除, 未必会有一个在那, 2^{-256} 是一个很小的数字。复杂系统中的加密强度不必要使系统更混乱, 也可能使它更混乱。你不可能摆脱加密的外表, 无论你使用多强的加密手段。

5.2.3 签名与责任

数字签名最早的建议应用之一是用来对禁止核试验条约的验证。美国和前苏联互相允许把地震测试仪放入另一个国家中, 以便对核试验进行监控。问题是每个国家需要确信东道国没有篡改从监控国家的地震仪传来的数据。同时, 东道主国家需要确信监测器只发送规定的需要监测的信息。

传统的鉴别技术能解决第一个问题, 但只有数字签名能同时解决两个问题。东道国一方只能读, 但不能篡改从地震测试仪来的数据; 而监督国确信数据没有被篡改。

数字签名的使用把责任给了签名人。如果你相信认可这就是一个合理的事情。也可以理解——他不想推卸责任和让某人冒险了? 如果你是银行, 你的顾客使用信用卡, 你持有风险。如果你的顾客使用数字签名, 那么就持有风险。

当然, 在经济活动中顾客的方面是不重要的, 当银行和商人假设是我的责任。作为顾客, 我同样是密码员, 我发现了令人忧虑的低安全系统保证我的信用卡安全。但是高安全系统像数字签名就受我吹捧。签名的安全性有多大不重要, 风险都是我的, 特别是从间谍软件到对自己信任的人的有风险。Stewart Baker 叫这个“Grandma picks a bad password, Grandma loses her house (奶奶挑了一个坏密码, 于是把房子丢了)”问题。

在低风险和低责任的系统下, 只有疯子会选择一对公钥。这个情形是对我们很不利的。数字签名被植入金融系统后, 可以提升整个系统安全性, 和减少欺诈和操作失误中的风险。但是在我们的世界中操作系统有

^a 在这个特定情况下, Alice 的心睡了, 但身体还在动, 就像梦游。无论索赔缘由有多么充分, Alice 无疑要为签名负责。但是在合同法上带来了一个有趣傻瓜。

^b Christopher Moore, 《Coyote Blue, Perennial Books》, ISBN 0-06073-543-0. 没有任何密码知识在他书中, 你应该看看他的热喜剧小说。

漏洞, 导致产生了利用寻找信任机制的欺诈和偷窃犯罪。未来加密技术必须重新考虑法律层面, 配置数字签名。无论是 Alice, 还是平民老百姓都会想用。

5.2.4 现实的语义改变

我们可以找到现实中反垃圾邮件签名系统的数字签名的语义学的例子, 密钥域名识别邮件 (DomainKeys Identified Mail, DKIM^a)。Miles Libbey 说 Yahoo! 很好的描述 DKIM 的用途“我们 Yahoo! 想知道, 我们顾客的一个从 eBay^b 网来的信息, 它确实就是从 eBay 网来的, 甚至进行校友会讨论” (充分披露: 我是一个 DKIM 规范的作者)。

一个 DKIM 的签名者在邮件流中为未经许可的邮件负责声明。对于一个 DKIM 验证者无论这个意味着什么他可以使用这个签名。这是一个帮助处理信息的有价值的信息。为了发送者, 它会很快的把消息收入收件箱。另外, 也会很快的分拣信息到垃圾邮件文件夹中。

DKIM 签名是在发送管理员域名和接受管理员域名之间的首要会话。首先是服务器对服务器的会话。尽管, 客户机可以验证和签名数据, 这不是你首要的目的。签名者责任的范围被限制了; 签名者没有透露信息的内容。它也是一种邮戳, 服务器的状态是“我把信息放入邮件流中”。

注意那就是电子邮件前面意义的改变。DKIM 绝不第一个尝试用数字签名来签名邮件的。很多以前的讨论集中在。结束用户自己的签名信息。特别是用 OpenPGP 或 S/MIME。事实上, 在我涉及 DKIM 以前, 我写了一个《PGP CTO Corner (PGP 首席技术官的窘境)》的文章, 是关于使用数字签名阻止垃圾邮件^c的, 讨论这些不是个好主意, 因为管理操作很难。还有对只能模糊证明的语义危险的承诺。如果我为我所有的邮件签名, 这样来展示它们不是垃圾邮件。如何区分空闲的想法 (或者说是草案), 我准备把我的名字放在什么上面?

DKIM 在以下方面改变了电子邮件的自然语义:

- 一个 DKIM 签名分开了信息内容和传输层的关系, 它也分开了最终用户和接受域名。管理的域名是最终用户的经理人, 他有责任让这些先到域名服务器。域名服务器那时会拿起最终用户的争论。
- 这个声明是域名对域名的。Yahoo! 有延伸这个项目到最终用户的计划(这样你可以看到 eBay 的信息有 eBay 的签名), 但是这个没有什么必要。
- DKIM 签名在把信息放入邮件流中, 并指出了一个有限的责任。从那之后, 什么都可以发生。比如, 我拿着 Miles 的 eBay 信息的事到我的校友会。我大学的服务器信号一闪发给我 20 个 eBay 的信息, 这是依据 eBay 的委托义务。的确, eBay 也可能找不到这个一闪的信号。
- DKIM 签名不是档案签名。电子邮件客户端可以验证 DKIM 的签名, 看完下面这个后你就不会去做那个了。签名的密钥存在 DNS^d 中, 签名可能被签名域名服务器移动。我们的设计者认为这有一个密友 (privacy-friendly) 的功能, 而且并不限制。我们认为邮件验证可以选择性的促成邮件变成记录在案的通讯, 我们不希望那样。DKIM 结构是短暂的签名密钥, 而不是永久的。我们认为域名对域名是 DKIM 的一方面, 也就是称之为密友的功能。
- DKIM 基于密钥, 而不是证书。它是流线型的, 小型信任模式。这个密钥是否在这个域名的 DNS 中, 如果有, 一个验证者从 DNS 获得签名密钥。DKIM 密钥不会收回; 他们仅仅获得 DNS 的回应。

^a DKIM 的联系可以在这里找到: <http://www.dkim.org/>

^b 易趣网站, 一个和淘宝网差不多的网站, 主要提供网上购物。

^c Jon Gallas, 《Crypto and Spam (密码学和垃圾邮件)》, <http://www.pgp.com/library/ctocomer/cryptoandspam.html>。

^d DNS 是域名系统 (Domain Name System)。里面有一个连接主机名和域名的目录。比如转换 www.pgp.com 为数值网络地址 (IP)。

DKIM 的所有小方面有意图的特征不同于其它的签名系统。DKIM 的声明在时间上有限制,在范围上有限制,承担责任上仍有限制,仍然提供一个有用的目的: 允许邮件接收者验证邮件源。

5.3 加密技术可靠性

安全系统是一种提供安全的系统。地址威胁不同于其它的威胁,很多威胁是非智商威胁,闪电、刮风和地震是非智能的威胁。黑客、间谍和小偷都是(高)智能的威胁。有一个在中间的叫半智能威胁,范围从昆虫到害虫再到肉食动物都有。我们对应的威胁是按照智商的程度来说的。避雷针解决了闪电起火的威胁,闪电不会干扰到避雷针。黑客和间谍的发展入侵到我们开发的方案对策中,我们也同样开发阻挠黑客的技术。我们必须考虑他们不让我们安装避雷针,或者是阻止我们解决这个问题。

让我们来分辨 2 种广泛类别的安全系统: 抵抗智能威胁的安全系统和抵抗非智能威胁的安全系统。

我们的密码员必须掌握抵抗高智商攻击的技术,像 PGP 一样的安全软件可以很安全来抵抗智商攻击,用户必须对整个系统的可靠性小心: 如果你忘了你的密码或者丢失了私钥,你已经丢失了信息。密码系统面对的一个挑战是确保系统可以抵抗智能攻击,而不会降低整体系统的可靠性。

保护密码系统的安全性要确保没有密钥的用户产生错误。密码系统的可靠性要确保正确的用户有密钥。未来密码技术的要点是在临界状态下的积木式密钥管理系统,但是如何确保正确的人总是有密钥,不该拿的人永远没有密钥。

5.3.1 硬件加密与软件加密

5.3.1.1 硬件加密

直到最近,所有加密产品都是特定的硬件形式。这些加/解密盒子被嵌入到通信线路中,然后对所有通过的数据进行加密。虽然软件加密在今天正变得很流行,硬件仍是商业和军事应用的主要选择。例如,NSA 只对硬件加密授权使用。为什么这样是有原因的:

首先是速度。正如我们在第三部分看到的那样,加密算法含有很多对明文比特的复杂运算,没有哪类这样的操作能在一般的计算机上进行。两种最常见的加密算法,DES 和 RSA 在普通用途的微处理器上运行没有效率可言。尽管一些密码设计者不断尝试使他们的算法更适合软件实现,但特殊的硬件将一直获得速度之胜利。

另外,加密常常是高强度的计算任务。计算机微处理器对此效率不高,将加密移到芯片上,即使那个芯片仅是另一个处理器,也会使整个系统速度加快。

硬件流行的第二个原因是安全性。对运行在一般的、没有物理保护的计算机上的某个加密算法,Mallory 可以用各种跟踪工具秘密修改算法而使任何人都不知道。硬件加密设备可以安全地封装起来,以避免此类事情发生,防篡改盒能防止别人修改硬件加密设备。特殊目的的 VLSI 芯片可以覆盖一层化学物质,使得任何企图对它们内部进行访问都将导致芯片逻辑的破坏。美国政府的 Clipper 和 Capstone 芯片都被设计成防篡改,芯片设计成这样就使 Mallory 不可能读到未加密的密钥。

IBM 发明了一种用来加密主机数据和通信的加密系统。它包括用防篡改模块保存密钥。

电磁辐射有时会暴露电子设备内正在处理的东西。可以将加密盒子屏蔽起来,使得信息不致泄露。通用计算机也可以屏蔽,但却是个复杂得多的问题。美军称这类操作为 TEMPEST,这个课题远远超出本书范围。

硬件流行的最后一个原因是易于安装。大多数加密应用与普通计算机无关。多数人希望加密他们的电话会话、传真或数据链路。将专用加密硬件放在电话、传真机和调制解调器中比放在微处理器或软件中便宜得

多。

当加密数据来自计算机时, 安装一个专用加密设备也比修改计算机系统软件更容易。加密应该是不可见的, 它不应该妨碍用户。对于软件要做到这点的唯一办法是将加密程序写在操作系统软件的深处, 这很不容易。另一方面, 就是初学者也能将加密盒插在他们的计算机和外接调制解调器之间。

目前, 市场上有三类基本的加密硬件: 自带加密模块(可完成一些如银行口令确认和密钥管理等功能), 用于通信链路的专用加密盒以及可插入个人计算机的插卡。

一些加密盒是为一些具体的通信链路设计的, 如 T-1 加密盒设计成不加密同步比特。用于同步或异步通信链路的加密盒是不同的。较新的一些加密盒趋向于处理更高的比特率和高通用性。

即使如此, 许多加密设备也有一些不相容问题, 购买者应该小心注意这些差别, 并了解他们的特殊用处, 避免自己购买的加密设备不能满足要求。特别要注意, 硬件类型、操作系统、应用软件、网络等方面的限制。

PC-板加密器通常将所有写到硬盘上的东西进行加密, 并且可以配置以将写到软盘和串口的东西都加密。并不为这些板卡屏蔽电磁辐射或物理干扰, 因为如果计算机不受影响, 保护这些板卡是没有意义的。

越来越多的公司开始将加密硬件设备安装到他们的通信设备上。保密电话、传真机和调制解调器都可买到。

虽然有多少种设备、就有多少种不同的解决方案, 但这些设备的内部密钥管理通常是安全的。一些方案在一种场合比在另一场合更合适, 购买者应该懂得哪类密钥管理与加密盒相结合, 哪类是自己所期望的。

5.3.1.2 软件加密

任何加密算法都可以用软件实现。软件实现的不利之处是速度、开销、和易于改动(或操作)。有利之处是灵活性和可移植性, 易使用, 易升级。可以不花一分钱将他们容易地复制下来, 并安装在许多机器上。他们也能和大型应用如通信或字处理程序相结合。

软件加密程序很大众化, 并可以用于大多数操作系统。这些是用于保护个人文件; 用户通常必须手工加解密文件。密钥管理方案的安全性是重要的: 密钥不应当储存在磁盘的任何一处(甚至也不应该写在处理器与磁盘交换数据的内存中)。密钥和未加密文件在加密后应删除, 许多程序对这点都很草率, 但用户必须仔细选择。

当然, Mallory 可以一直用无用的东西替换软件加密算法, 但对大多数用户来说, 这不是什么问题。如果 Mallory 能够潜入办公室将加密程序修改掉, 也能将一个隐形摄像象机置于墙中, 搭线窃听电话线路, 或者将一台 TEMPEST 检测仪放于墙下。如果 Mallory 确实比一般用户更强有力的话, 那么用户早在游戏开始之前就输掉了。

5.4 硬件升级

硬件升级后的密码技术是为了应对很多原因: 安全性需要提高, 硬件设计者认为需要, 硬件现在更加的便宜和高性能。硬件的提高会有很多优势。使用硬件加密可以对可靠性产生不利影响, 特别是如果用硬件来存储密钥。

如果我们提供给你一个有一个硬盘的笔记本, 笔记本在读写的时候加密数据; 这个磁盘加密等于我们的全盘加密。我们同样提供使用安全芯片存储密钥的的硬盘, 在用户输入密码后硬盘会开启。这个过程增加了硬盘的安全性硬盘本身不保存密钥和解密数据。安全芯片^a拥有加密过的密钥, 如果没有用户的密钥这个是无

^a 指 TPM 安全芯片, TPM 芯片可以以模块的形式插在主板的接口上, 也可以直接焊接在主板或集成到网卡中实现加密功能。TPM 芯片之所以安全, 最大的优势是可以通过硬件算法对数据、密钥文件进行存储和加密。和单纯的软件加密有很大的不同, 软件加密生成的二进制密钥文件是存储在硬盘上的, 而且很容易被黑客获得并破解, 而 TPM 芯片不仅对软件生成的密钥二次加密并存储在自己的芯片寄存器中, 而且还限制程序的可访问性, 确保能提供高级别的安全保护。PC 有了这种功能, 加密过的硬盘

用的。

作为一个安全系统, 这个发展有个有趣的特点。如果你丢失了笔记本, 没有人可以使用你的硬盘(假如他们不知道你的密码)。如果你把硬盘从笔记本拿出来和扔掉, 也没有人可以获得上面的数据, 因为他们需要存储在芯片内的密钥。密码术也给了你粉碎你旧硬盘的等价方式。

然而, 这个新系统比你旧的那个可靠性差些。如果你忘记了密码就有一个很大的风险, 你已经丢失了你的数据, 这很可能, 也许不可能称为问题。在大型组织中, 小的影响可以变得很大。我会无中生有的推出一个数据: 某人可以完全记起他(她)的密码有 99.9% 的机率。我自己会接受这个机率, 如果我是一个大公司的 CIO^a, 我不得不期待一些令人头痛的问题。如果有 10 万人在我公司里, 其中有 100 人今年忘了密码, 也就等于 100 个硬盘失效, 否则我也不会有这么多烦恼, 那就是为什么我需要一个密钥管理系统来获得安全, 现在已经有可靠的办法去管理这个系统。

还有另外一种可靠性损失, 是在硬件本身里。安全硬件本身也可能会失效, 如果失效, 也意味着你丢失了硬盘。另外, 如果这个硬盘是你笔记本的一部分, 出问题了, 这部分需要更换, 你的秘密也就被替换了。如果是视频微型芯片、电源线、USB 端口, 或其它部件失效了, 你也可以拿出你的硬盘。

最近一些新的硬件系统已经重新设计, 变的更加可靠。这些硬件系统的早期版本中, 它们不能从芯片到芯片迁移。如果这个系统的目的是限制另外一个机器的数据, 它工作过程很好, 但是对通用加密系统不好, 最新版本的系统允许密钥被备份。

尽管这样, 这个选择权创建了操作的灵活性, 这在以前是不存在的。你, 最终用户, 必须不止是备份你的数据, 还要备份你的安全硬件。另外, 你做到了把你的钥匙锁到车里一样的事情, 但是我们聪明的密码员也告诉你无法撬开。未来这些系统可靠性会提高, 安全性也一样。软件系统像 PGP 一样的软件已经有了可靠部件, 未来将会考虑集成化。

5.5 权限管理

密码学好在是粗糙的控制。如果你不能解密数据你就不能使用数据, 如果你解密了, 它就是你的。很多人期待一个细致的控制, 这就是前面提到的项权限管理。要更注意 DRM (Digital Rights Management, 数字权限管理^b), 我可以流利的描述出确实你没有通过错误的扬声器播放音乐。许多权力管理都围绕 DRM 作为一个娱乐的应用而争论。因此, 相似的系统应用与文档的通常也叫 ERM (Enterprise Rights Management, 企业权限管理)。

权限管理系统只和礼貌的攻击方式抗争。如果被保护的目录要被看和被听, 一个专注的攻击者可以找到进去的办法, 攻击者可以对扬声器录音或屏幕截图。尽管如此, 权限管理系统被选择是因为大量可能的攻击者没有足够的耐心去破坏系统, 或者畏惧保护器的本事。

ERM 系统可以创建文档、邮件、还有其它可以被创造者使用的其它文档的计划。它极具吸引力的原因是我们希望我们不是去发送邮件, 或者我们不想阅读其它人阅读的文档。每个人都像保护管理的最终权力; 每个人愿意在接收端。

当我们使用权限管理系统, 这是多么有意义的情形。但是也有许多影响社会广泛的权利管理制度, 他们将直接影响他们使用多少, 这里有些例子:

- **权限管理鼓励和增强了政治行为。** 如果人们发送邮件后可以取消。这就是权限管理的一个卖点: 减少

或数据, 只要数据离开了现有 PC 的 TPM 芯片, 数据都将无法正常识别, 因为 TPM 芯片用于加密的保密码属于商业密码, 都是高度保密的, 没有这个保密码密钥无法将数据还原成明文。但是就 TPM 芯片本身而言, 要完全发挥它的作用, 还必须依靠软件的配合才能实现。如普通的加密软件配合 TPM 芯片可以实现高级别的安全保护功能。中国政府为了安全考虑, 这类芯片不允许使用国外产品, 目前国内的有兆日和联想自主开发的“恒智”。

^a Chief Information Officer: 信息总管, 掌管公司的计算机化业务。

^b DRM 的更多介绍访问: <http://baike.baidu.com/view/47310.htm>

阻碍。潜在的阻碍让一些人承担了责任。如果他们写写东西就会立刻看到，他们也会这么做。一些人受到诱惑后，在会议上的暗示了主持，但是没有去承诺，否则，这样的会议会变的很多。

- **人们将要找到权限管理的路。**记着：权限管理是抵御常规攻击者的唯一选择。如果你用一个挑衅的信息激怒某人，他（她）可能就不礼貌了，我们生存的在一个有电话机监听的世界。如果有人破坏了你和他的邮件和邮件回复的约定，你很可能会处理下一封邮件，也就是他发给你的电子邮件截一张图片。当客户服务代表告诉顾客（理所当然应该知道）：顾客会拿到一张邮件内容的图片，并且把它粘贴网上。权限管理中的将会出现一些很差劲的事情暴露在公众眼中。
- **暴脾气的人可以保护文档。**人们用邮件来遮盖他们的轨迹。他们可以创建一些讨厌的邮件的图片，来防止伪造没有的邮件的轨迹—— 权限管理系统会擦去所有轨迹。
- **很多商业中不允许像文档一样的东西泄露。**比如，财政服务机关必须为所有通讯存档。他们必须提供所有文件、邮件、甚至一些内部信息供调查者检查。这些商业公司将会阻止权限管理的下的文件到别人公司去。

很多安全的优势提高了可追踪性和责任性。这些社会内涵已经涉及到无论是不是我们需要那样跟踪的地方。但是，权限管理的一个趋势是低责任。有很多有用的和适合的区域。有些人也会拒绝权限管理，因为他们有自己的管理和法律责任。在有些地方，权限管理被一些滥用系统的弊端的人说污蔑。它是否能成为标准是由如何发展能平衡用户和弊端来决定的。

5.6 数据销毁办法

在大多数的计算机上删除一个文件时，该文件并不会真的被删除。删除掉的唯一东西就是磁盘索引文件中的入口，磁盘索引文件用来告诉机器磁盘上的数据在哪里。许多软件供应商不失时机的出售文件恢复软件，它们可以在文件被删除后将其恢复。

还有别的方面的担忧：虚拟内存意味着你的计算机可以随时将内存读写到你的磁盘。即使你没有保存它，你永远也不知道你正在运行的一个敏感的文件是什么时候写到磁盘上的。这就是说，即使你从来未保存过的明文，计算机也可以替你做了。并且如 **Stacker** 和 **DoubleSpace** 这样的驱动器级的压缩程序会使得预测数据是怎样存到磁盘上，且存到哪里更加困难。

为了删除某个文件，让文件恢复软件都不能读，必须对磁盘上文件的所有比特进行物理写覆盖。根据美国国家计算机安全中心：写覆盖就是将不涉及安全的数据写到以前曾存放敏感数据的储存位置，为了彻底清除储存介质，DoD^a要求先用一种格式进行写覆盖，然后用该格式的补码，最后用另一种格式。例如，先用 **0011 0101**，接着用 **1100 1010**，在接着用 **1001 0111**。写覆盖的次数根据储存介质而定，有时依赖信息的敏感程度，有时对不同的 DoD 部分要求。无论怎样，在最后没有用不涉及安全的数据写覆盖之前，彻底清除就没有完成。

你可能必须删除某个文件或清除整个驱动器，你也应当清除磁盘上所有没有用的空间。

大多数商用程序声称实现了 DoD 标准覆盖三次：首先用全 **1**；接着用全 **0**；最后用 **1-0** 格式重复进行。按照我的一般的偏执狂级别，我建议覆盖一个被删除的文件需要 **7** 次：首先全 **1**；其次全 **0**；其余 **5** 次用密码学安全的伪随机序列。最近美国国家标准和技术研究所对电子隧道显微镜的研究表明即使这样也是不够的。说实话，如果你的数据的确有足够大的价值，还是相信从磁性介质上完全清除数据是不可能的吧！将介质烧掉或切碎；买张新磁盘要比丢失你的秘密便宜得多。

^a DoD 美国国防部 5220.22-M C 清除和处理矩阵，DoD 5220.22-M/ NISPOM 8-306 中指定的标准。

表格 10: 美国国防部 5220.22-M C 清除和处理矩阵

媒体	清除	清洁	细则
磁带			a. I 型磁器消磁。
类型 I	a 或 b	a、b 或 m	b. II 型磁器消磁。
类型 II	a 或 b	b 或 m	c. 用单个字符覆盖可寻址的存储单元。
类型 III	a 或 b	m	d. 用字符、他的补码和随机字符覆盖可寻址的存储单元并进行校验。不推荐使用该方法处理包含绝对机密的媒介。
磁盘			e. 用字符、他的补码和随机字符覆盖可寻址的存储单元。
Bernoullis	a、b、或 c	m	f. 每个覆盖在内存的存储期要长于分类数据的存储期。
软盘	a、b、或 c	m	g. 移除所有电源包含电池。
非移动硬盘	c	a、b、d 或 m	h. 用随机图样覆盖所有位置，所有带有二进位 0 的位置，所有二进位 1 的位置。
移动硬盘	a、b、或 c	a、b、d 或 m	i. 执行擦除每一个的厂商数据表。
光盘			j. 执行 i 以上，然后 c 以上，总计 3 次。
多次读写	c	m	k. 按照厂商的推荐执行擦除紫外线辐射。
只读	m, n		l. 执行 K 以上，但通过三个要素增加。
只写一次，多次读（缓慢）	m, n		m. 毁坏—破坏、焚烧、研成粉、撕碎或融化。
存储器			n. 只要包含机密情报必须毁坏。
动态随机存储器 (DRAM)	c 或 g	c、g 或 m	o. 运行 6 页不保密的文本（测试可接受的字体）。
电可擦写的编程存储器 (EAPROM)	i	j 或 m	p. 带状物必须被毁坏。压盘必须被清除。
电可擦除的编程存储器 (EEPROM)	i	h 或 m	q. 检查和/或测试屏幕表面检查已被破坏资料的迹象。如果显示出来，阴极射线管必须被毁坏。
可擦除可编程只读存储器 (EPROM)	k	l 然后 c 或 m	
Flash EPROM (FEPRM)	i	c 然后 i 或 m	
可编程只读存储器 (PROM)	c	m	
磁泡存储器	c	a、b、c 或 m	
磁芯存储器	c	a、b、c 或 m	
镀磁线	c	c 和 f、或 m	
磁抵抗的存储器	c	m	
非易失随机存储器 (NOVRAM)	c 或 g	c、 g 或 m	
只读存储器 (ROM)		m	
静态随机存储器 (SRAM)	c 或 g	c 和 f、g 或 m	
设备			
阴极射线管 (CRT)	g	q	
打印机			
喷墨打印机	g	p 然后 g	
激光打印机器	g	o 然后 g	

关于清除和处理安全标准 DoD 5220.22-M 的更多资料请参见美国国防部秘密警察服务网站（第 8 章）：
http://www.dss.mil/isec/change_ch8.htm

在我们一般使用的存储设备中, 不管是保存时间最长的存储介质——光盘, 还是使用范围最广的——闪存, 和应用最多的——磁盘, 及一些特殊领域的智能 IC 卡等介质。对使用切割, 或者挤压、拉伸、燃烧的办法可以解决, 或者再通过二次碾压技术对介质进行彻底粉碎, 粉碎不彻底的话仍然有还原的希望。

对于闪存、智能 IC 卡这些基于电子线路的设备, 有人使用高压击穿的方式销毁, 但是击穿并不是破坏了所有的数据, 只是一部分而已, 芯片可以通过一些特殊的半导体“开芯”工艺重新接上线路, 也许仍然可以读取, 这么做是极其危险的。

对于磁介质来说, 高温下磁性会消失, 这个方法对其它的一些磁性设备来说也是有用的, 不过燃烧的一

些过程会产生污染环境的有害气体。我不推荐。一般不到燃烧点, 磁性物质就失去了磁性。当然也提醒千万不要把磁盘扔到水里面, 硬盘几乎是封闭结构, 想让硬盘生锈的主意是很愚蠢的。

5.7 保密增强技术

还有另外一个神话要被打破: 这里有方式能在安全和隐私上取得平衡。通常的争论的是社会有一个两头装有东西的天平, 一头是保密性, 一头是安全性^a——我们在讨论天平究竟可以转多远。这也可以在降低保密性的前提下提高安全性。甚至是再比较低的秘密性下, 只是看起来好像安全性提高了^b——任何人都可以进入这个像飞机场的地方看看里面到底有什么。

这个用保密性来换安全性的简单设想并不现实。这里有很多方式设计一个系统使秘密性和安全性都不会丢失。这个主题完全可以写一本书。但也有加密系统创造不一样的链接保密和安全的天平, 这是一些测试。

- 可能第一个秘密增强安全方面有意义的是 David Chaum 做的封闭签名 (blinded signatures)。数字签名的这种签名的方式, 不只是一个人, 而且可以不是签名者, 都知道签名的详细信息。例如, 它有签名时变的很有用, 也就是说“这个人超过 21 岁”, 我会把“超过 21 岁”最为一个密钥, 在检查后这个人确实是超过 21 岁后, 并且确定这个签名。这个签名可以被验证, 但是我无法说清楚签名本身的详细内容是什么。封闭签名可以信任已确认的人, 或者完全是匿名的人。

如何用封闭签名的规则要依据他们本身。如果我签名的是一个图片, 你可以对比这个人的图片和签名本身, 然后这个人超过 21 岁了, 判断它是真的。这又成为了另外一个安全信任, 像 PGP 密钥和证书, 次级安全保护是系统的一部分。

你毫无疑问已经找出了这个情况的问题。如果这个东西给了另外一个人会发生什么呢? 是不是也可以让别人自由的看? 如何确认我们签名的东西有意义? 如何让验证者接受我们做的这些?

尽管如此, 破坏基础的人在封闭签名中改革, 你可以用数字签名来声明能证明这点, 但是不像其它不切题的信息。保护秘密的目的是确保我们接受到的部分是我们想要的信息^c。

- 基于 Chaum 的工作, Stefan Brands 创建了数字信任 (Digital credentials^d), 这个被添加进了创建很多的保护部分的选项的概念里面。

一个反对者说: 保留有秘密技术的人会恐惧隐私引诱持有者误用信任。因此, 这有一些办法在系统中实施责任制。比如, 你给一个超过 21 岁的人信任, 但是如果这个信任是错误的, 你可以撤回它, 正好我不知道你的客户。这个可能加固了责任性。它也使得信任更具价值, 因为, 所有合作者知道这里的实施细则和责任。同样, 这个秘密使的更少的特殊的人被排除在例外。它许可你使用一贯政策和强

^a 前面的“保密性”是指这些消息的加密的安全性。后面的“安全性”是指保护、管理这些秘密的手段, 比如很多级的访问验证。这样设计的目的是如果房间的门被打开了, 但是我们的秘密还在房间内的保险箱内。如果房间的门很坚固, 我们的保险箱可以选择价格便宜的, 当然如果门和保险箱都很坚固, 那固然很好!

^b 任何人都没有任何秘密了, 大家彼此都安全了。秘密和安全是反比的关系。秘密越多你就越害怕泄露, 安全性就更差。

^c David Chaum 《Achieving Electronic Privacy (成就电子隐私)》, 《美国科学家》1992 年 8 月版, 96-101 页,

http://www.chaum.com/artides/Achieving_Electronic_Privacy.htm, David Chaum, 《Security Without Identification: Transaction Systems to Make Big Brother Obsolete (安全无需识别: 事务处理系统把老大哥给废了)》, 《计算机协会通信》28 卷第 10, 1030-1044 页; 1985 年 10 月, http://www.chaum.com/artides/Security_Without_Identification.htm

^d Stefan Brands 重新考虑了公钥的下层结构和数字证书; 《Building in Privacy (秘密中建立)》, 麻省理工学院出版, ISBN 0-262-02491-8. PDF: http://www.credentica.com/the_mit_pressbook.php

制公平。这个里面的秘密性保护过程没有减少丝毫的安全性，反而还提高了！Brands 的数字信任解决了很多问题。他们也最大程度的创建了秘密性和安全性的独立。

- Lea Kissner 和 Dawn Song 在秘密性保留过程中有新的改革^a。想法虽然简单，但极具力量。Alice 和 Bob 各有一个数据库。他们想知道他们数据库中彼此没有的项。这有很多可以做到这个目的的程序。美国政府有一个名单，这些人不想乘飞机^b。航线有乘客列表。谁在这 2 个列表上？医院有在中心治疗的病人的名单；州政府知道谁在享受福利。研究人员如何研究高收入人群、疾病人和法律保护下的每个组？或者两个公司如何知道彼此共享的顾客，和想帮助下相同利益的顾客，但是谁都不想放弃他们的顾客列表？

还有许多重要隐私保护技术的新形式，因为它解放了我们考虑安全的方式。它使我们不用担心在提高安全性后会有隐私损失后的道德问题^c。

5.8 细微的改变

在他们发展的一点中，所有的技术达到人们可以平稳使用的程度，甚至通过机械的前进和提高。例如，在近 50 年中空中客机，只改变了空中旅行的一点点而已。你可以花费 5 到 6 个小时坐在一个狭窄的位置上从美国西海岸到东海岸，而且吃很差的食物，试着忽略想要管住孩子的人们的声音。有时也会背着行李徒步旅行，不考虑低安全性、低价格和退步的服务，坐飞行感受在半个世纪的过程中都是差不多的。而在这之下，我们坐的飞机和早期的飞机确实很大不一样。

相似的，密码技术渗透的更深，外在没有多大变化，但是内部有了很大的变化。这些发展对密码专家和工程师来说是很有趣的，但是对用户就不是。他们到底开发什么了？

5.8.1 新型 Hash 算法

Hash 算法是密码学家可以达到很多目的的灵活算法。在 2004 年，我们知道我们做的还不够。2 年以后，我们知道了我们还不清楚要设计好的 hash 算法。我们喜欢描述我们怎样设计一个好的算法。在几年后，我们要开发新的算法，并且在现实中展开他们，但是你不会看到多少变化。

5.8.2 新型算法

密码学家经常会想出新的算法。同样也有变的比以前更好用的旧系统。当你读到这个新通过的加密方法变化的开发笔记，对于最终用户有一些选择，因为首要的改变是加密的语义学，而不是语法。

5.8.2.1 结合加密和验证的算法

当我们构成数据的时候，我们使用分开的算法来加密和显示数据没有被修改。新算法中已经接合这些步

^a Lea Kissner 和 Dawn Song 《Privacy-Preserving Set Operations》,CMU-CS-05-113,2005.6.

<http://www.cs.cmu.edu/~leak/papers/set-tech-full.pdf>, 我在 the PGP CTO Corner 上写了关于 Kissner 和 Song 工作，访问：

<http://www.pgp.com/library/ctocorner/sets.html>。

^b 名单并不只是一列人，有许多私人秘密问题在不想坐飞机的人的列表上，因为不仅仅是人的名单，还要把他们都分清楚。

^c 一个最显然的例子就是中国的“艳照门”事件。

骤,也就是用单一算法加密,让解密的人知道数据是没有被动过的。新算法的优势是即使数据是流动的也可以加密,在网络协议方面出现了一个更高效的提高,特别是对实时流式数据(streams of live data)。

一个和公钥算法相似的工作叫签名加密系统(signcryption systems),它可以同时完成签名和加密。

5.8.2.2 新算法和重新设计的算法

今天,我们有很多类算法,特别是 AES 标准竞争后的结果。其它的一些候选算法也是不错的。5 个中的 3 个: Rijndael (AES)、Twofish 和 Serpent。甚至如果你使用它们在知识产权是也没有争议的。

同样,在加密和密码分析方面都有所进展。最近密码分析的工作并没有集中在算法的输出分析上,而是在操作上。这个观察叫侧槽分析(side-channel analysis)。它与算法在时间上一起运行,或者讨论计算机有多少能力可以达到这个效果。全部加密算法的内存信息会被提供给侧槽分析作为数据。

在密码学建造和破坏过程中,密码专家要开发对侧槽分析有免疫的新算法。你可能会期待,密码分析学家也会开发出新的侧槽分析和新的方法去应用侧槽分析来破解新的情况。

5.8.2.3 椭圆曲线算法

俗话说:一条链子的强度和最弱的一个节的强度一样。这个对密码学来说是对的,我们使用相同安全强度的不同部件。今天,一般我们使用在相同强度上比对称密钥更长的公钥加密系统。我们也开始使用 256-bit 强度的对称密钥,例如 RSA 和 DSA,一个 3,000-bit 公钥与一个 128-bit 对称密钥平衡,但 15,000-bit 公钥和 256-bit 对称密钥平衡。15,000-bit 是很大的密钥了。但是在一些小型的设备如手机、安全门中也是出于这个考虑,美国政府鼓励使用椭圆形加密算法。在椭圆形加密算法中,512-bit 公钥和 256-bit 对称密钥相同。椭圆形加密算法在下一个 5 到 50 年里会出现,也会因为专利限制而出现问题。

5.8.2.4 双线性映射算法

除了对称算法的发展,还有公钥算法的发展。最有趣的是使用双线性映射(bi-linear maps)。同样也使用数学上还有许多不同的传统椭圆曲线算法。它允许从主密钥创建一系列的密钥。每一个系列的其中之一都是一对公钥;由一个私钥和一个公钥组成。例外,任何主私钥的主人不能向其它主私钥学习。这样,他们共享对称密钥的很多性质,同样也有公钥系统。目前有很多算法:

- 用户创建单一主公钥和许多废除的密钥,当需要去存储单一密钥时可以用与很多目的。尽管存储设备很便宜而且会更便宜,仍然有人想要,因为它可以简化分级系统中的数据管理。例外,主密钥的所有者可以把密钥对给别人,这样主人就可以和其它人交流了。
- 服务器创建单一主公钥,系统的用户使用机构的发行密钥。这个装置创建了从主密钥上来的第三方密钥系统。任何有主密钥的人和主系统都可以很好的协同工作,但是仍受主密钥所有者的监视和控制。

当次级密钥在别人手里的时候会出现一些隐私问题,因为这都是用一些明显的后门设计的加密系统。有很多案例可以说明这些出口不是问题。比如在可操作的安全下,主系统或虚拟机都可以有自己的密钥。当加密主系统的接触点最小的时候,计算机集群也有统一的密钥。

- 双线性映射可以让他们知道的主密钥的所有的任何东西。为什么他们还要用 hash 计算任意字符串呢?

这个方法让我们创建可制定密钥的身份基础系统 (identity-based encryption systems)。当然, 这意味着我们相信 hash 算法是好的, 因为一个名字的 hash 碰撞意味着会有很多密钥发生碰撞。目前, 我们还没有发现哪个 hash 算法的弱点有安全威胁。

双线性映射是到来的最有趣的发展, 因为是简化的和复杂的密钥管理。他们在最终用户复杂的密钥管理下向简化发展。发行只需要一个密钥, 但是最终用户可以每个发布至少 1 个密钥, 通常这个密钥是很多的。密码学最难的地方是密钥管理, 这不是个小问题。没有能力的用户可能在密钥管理方面觉得很有难度。双线性映射系统有许多复杂的撤销问题, 因为撤出或废除一个主密钥是没有办法知道这个密钥在哪里的。

在用双线性映射设计身份基础加密系统的时候, 问题甚至变的更坏, 因为名字是密钥部分最难的, 还有证书问题。讨论相对简单的“Jon Callas”的密钥, 但是谁拿的“John Doe”的密钥? 而且有其它的复杂问题。即使我们知道谁 2006 年为止拿到了“John Doe”的密钥, 谁在 2106 年为止拿“John Doe”密钥? 不幸的是, 密钥管理最重要的是密钥和源数据的接合。名字只是源数据的一部分而已, 还有很多其它部分, 也包括有问题的部分。我们与错综复杂的帐号名、入口名、用户名和标签名一起生活, 因为我们生活中不只和名字一起使用。

新的双线性映射不仅打开了提供前进的空间, 还有更多的讨论。

5.8.3 量子密码学

量子密码^a是一个新的令人兴奋的学科, 尽管它的名字不是密码学, 或许这就是令我气恼的事。但是密码学是用数学科学把信息变的不可读的一个技术。量子密码就应该叫做量子加密; 它是使用量子的选择来阻止信息被截取的方式。

量子算法的基本原理就是爱因斯坦的玻粒二象性: 光子可以同时存在于许多状态。一个典型的例子就是, 当光射到银白色的镜面时, 它会有波一样的特性, 既可以反射也可以传播, 就象波浪撞击一堵带有缺口的防波堤, 有的会翻回去, 有的却可以穿过去。然而对光子进行测量时它又表现出粒子的特性, 有一个唯一的被测量的状态。

这个技术使用量子缠上光子的选择性, 把它们接合在一起。如果访问者不是传输中两个主要的光量子其中之一, 他们在传输的时候就丢失了数据。这样任何一个光子在这两部分之间的轨道运行的状态, 观察者都无法获知。

这就是一个戏剧性的简化。尽管如此, 量子密码已经允许成为可选择的密码技术。如果我把一封信直接交到你手里, 而从我手里到你手里的过程也没必要再来多余的加密。

Peter Shor 阐述了一个基于量子力学的因子分解机器设计模型。不象一般的计算机在某一特定时刻可以认为有一个单一、固定的状态, 量子计算机有一个内部波动函数, 这个函数是所有可能基状态的联合重叠。计算机在单步运算中通过改变整套的状态值来改变波动函数。在这个意义上, 量子计算机是基于经典的有限状态机改进而成的: 它利用量子特性允许在多项式时间里进行因子分解。理论上可以用来破译基于大数分解或离散对数问题的密码体制。

舆论一致认为, 量子计算机与基本量子力学定理是可和谐共存的。然而, 在可预见的未来制造出一台量子因子分解机基本上是不可能的。其最大的障碍是非连贯性问题, 因为它容易导致叠加后的波形丢失某些特性, 从而使计算失败。不连贯性会使运行在 Kelvin 下的计算机仅 1 纳秒后就死机。另外, 制造一台量子因子分解设备需要超大量的逻辑门, 这使得制造不太可能。Peter Shor 的设计需要一部完整的模取幂计算机。由于没有内部时钟, 数以千万甚至上亿的独立门被用于分解密码上非常大的数, 如果 n 个量子门有很小的错误概

^a 量子密码学的更多信息, 访问: http://baike.baidu.com/view/192896.html?tp=0_11

率 p (显然 $p \leq 1$), 则每成功运行一次所需实验^a次数就是 $\frac{1}{(1-p)^n}$ 。量子门的数目可假定按被分解的数的长度 (比特) 呈多项式增加, 那么, 实验次数将随该长度呈超级指数增长——这比用试除法进行分解还要糟糕!

所以, 虽然量子因子分解法在学术上非常令人兴奋, 而且它有不错的市场前景^b, 但它在不远的将来被用于实践却是不可能的。别说我没有提醒你!

5.9 什么能够改变路线?

原因的一部分是 Niels Bohr 说的一句很对的话: 很难预测不可预见的未来。对于未来的预测结果会有轻微的不合理。不管那个, 百搭牌^c可以给你很好的原谅自己的机会。

5.9.1 专利影响

我签名描述的很多密码系统都是专利。事实上, 密码学中很多有趣的改革已经获得专利, 或者正在获得专利 (只要他们申请的快)。一方面, 专利是很不错的, 因为发明人自信且可以拥有他们的专利, 而且迅速的致富。另一方面, 所有权通常减缓了这个密码技术的实施。加密技术只能够很多前进的时候就已经衰弱, 这就是因为专利问题。专利可以创造财富, 也可以毁灭一个优秀的算法系统。

5.9.2 科学虚构的技术

“科学虚构”这个词不是轻蔑的意思。好像还是不久以前, 去月球是科学幻想小说中的情节。今天, 破解算法成也成了一个科学幻想小说的一个细节。

今天, 人们从一些现实中的一些东西来创作科学幻想小说。他可以戏剧性的选择密码学的道路。在这个里面是最明显的技术是量子计算机, 有对量子计算机积极的研究, 但是现在仍然不知道量子计算机能做的一些详细的事情。

我们都认为量子计算机^d可以计算一些令人兴奋的平行度总值。而且 Peter Shor 设计了可以在量子计算机上面运行的数学计算的算法, 但是不适合今天的计算机。在他们这些人的其中, 他已经设计了因数的方法和计算离散数学的算法^e。如果你可以设计这么一个机器, 它就完全可以破解我们今天用的公钥算法。

^a 简单解释这个《概率论》中的公式, 比如: 5 个黑盒子里每个都有 10 个乒乓球, 其中这 10 个里面有一个是黄色的, 9 个是白色的, 每次从 5 个盒子里摸出一个球。伸入一个盒子拿出一个黄色球的概率是 0.1, 摸出一个白色球的概率是 $1-0.1=0.9$, 那么我们从 5 个盒子里都拿出来的是白球的概率是 $(1-0.1)^5=0.59049$, 你会发现都拿出来黄色和斗拿出来是白色的概率 $0.1^5 + 0.9^5 = 0.5905 \neq 1$, 因为还有同时抓出 5 个球中同时有白球和黄球的事件, 那么如果要抓出 5 个都是白球至少需要进行的次数是 $\frac{1}{(1-0.1)^5} = \frac{100000}{59049} = 1.693508780843 \approx 2$, 这个结果只是估计值, 不是真实实验结果, 没准你永远都摸不到。那就只能相信运气一说。这个和买彩票中奖的概率小但是一样有人中的原因一样。《概率论》通常被用作估计事件发生的概率, 讨论可行性。概率小不意味不发生。和发生与否无关, 只是一种数量上的统计预测。

^b 国际首个量子密码通信网络在中国测试成功, 国际上首个量子密码通信网络日前由我国科学家在北京测试运行成功。这是迄今为止国际公开报道的唯一无中转、可同时、任意互通的量子密码通信网络, 标志着量子保密通信技术从点对点方式向网络化迈出了关键一步。访问: <http://news.mydrivers.com/1/80/80803.htm>

^c 我认为在这里应该是指塔罗牌, 这个塔罗牌有时候是用来占卜的, 访问: <http://baike.baidu.com/view/946.htm>

^d Jacob West, 《The Quantum Computer (量子计算机)》, 介绍: <http://www.cs.caltech.edu/~westside/quantum-intro.html>。A. Barenco, A. Ekert, A. Sanpera 和 C. Machiavello, 《A Short Introduction to Quantum Computation (量子计算介绍)》, 源自 La Recherche, 1996 年 11 月。A. Barenco 修改, 访问: <http://www.qubit.org/library/intros/comp/comp.html>。中文介绍: <http://baike.baidu.com/view/18645.htm>

^e Peter W. Shor, 《Algorithms for Quantum Computation: Discrete Logarithms and Factoring (量子算法: 离散算法和因数)》, 1994

然而, 还有人以一个叫做后量子计算加密 (post-quantum-computing cryptography) 作为工作。有公钥方案, 就像 hash 一条链一样, 他会量子计算机的攻击免疫。还有一些实时加密系统, 也无法让后量子计算。

相似的, 一些人用了 DNA 或其它生物学的指令集系统。有趣的是 Leonard Adleman, 他就是 RSA 中的 A, 是一个 DNA 计算的研究者。理论上, DNA 计算机可以解决一些普通计算机无法解决的难题, 像因数分解。还有远距离粒子, 它可以改变电子和光学计算机的运作。

任何这些发展都可以把我们的预测的结果扔进垃圾桶, 这些东西我们一点都没有想到, 我们只是不知道未来会给我们带来什么。

5.9.3 法律改变

在法律影响^a方面改变了一些东西, 就像量子的状态不可预见一样: 人们考虑这个的方式。10 年前, 期待密码技术能有一个行之有效的办法管理它。世界经济是依靠像货物的数量增长的数字来发展的。普遍存在的密码技术成为了发展中的一部分。2001 年恐怖分子的攻击也没有改变这个趋势, 目前也没有要改变的迹象。很多呈现在我们前面的都涉及确保信息被保护, 就意味着密码技术是主流。

当然, 这个社会趋势也被可以改变。这个变化也会改变自由贸易、全球化的态度和更多对经济的影响。我认为这个变化还不可能发生, 但是它可能发生, 如果发生了, 我只希望有人可以回来看看我写的东西, 添加一些不可预见的事情。

年计算机科学基金会第 35 界论文会, 124-134 页. <http://citeseerist.psu.edu/14533.html>。

^a 这里有全世界保密法相关政策访问: http://austlii.edu.au/~graham/PLPR_world_wide_guide.html

附录

6 技术软件的介绍

工欲善其事，必先利其器。居是邦也，事其大夫之贤者，友其士之仁者。^a

——《论语·魏灵公》

在介绍了一大堆之乎者也的东西后，我要介绍一些实际的东西了。其实，无论密码学有多么高深，只有用户们用了说好，这样这个技术才是成功的。本书的附录部分在原书中没有，只是翻译者针对中国用户来写的。

下面介绍的软件中，这些软件有的你已经经常在用，有的你可能听说过，有的似乎很陌生。有没有比这些软件好的技术呢？当然肯定有，但是为什么不使用呢？很显然，有些特殊的算法在设计出来后，如果是一个很好的技术，这个设计者就会去申请专利，然后大把大把的钞票进入自己的腰包，技术人员一下变成了商人，如果一个技术可以让一个一辈子打工的技术人员变成一个暴发户，这也是值得的。换句话说，真正的高手在民间，而你看到的正如冰山一角^b。目前的技术而言，安全性已经不错了。

信息技术是一直发展的，我认为最好的系统永远是下一个，这点我们毋庸置疑。技术当然是越新的越好。但是我们的硬件不可能天天去更换，硬件和操作系统软件是配合使用的，我们不能给 486 装 WindowsVista，我们也不能给有 4 核处理器、4Gb 内存、1Tb 硬盘的电脑安装 PC DOS 7.0。使用计算机要以“够用就好”的目的，不用去盲目的追求最好的硬件。换句话说，给你一台顶级配置的电脑，你难道用它进行天文计算？或是玩玩模拟器上的《超级玛丽》？使用新东西的代价是脱离大众，你无法找到最新的驱动，以及软件开发商也需要一个缓冲期来开发支持新的硬件技术的软件。

目前，全球使用的主流系统为微软的 Windows（视窗）系列和开源的系统 Linux^c。无论你使用的是 Windows，还是 Linux，密码学中的算法都一样，不过是运行的环境不同了。这个和我要表达中文“我爱你”和英文“I love you”一样。语言变了，环境变了，意思不变。

加密技术侧重的是算法，无论硬件平台如何，无论操作系统如何，算法的思路和过程不会变，就像是全世界的语言表达中哭和笑是一样的。下面介绍的一些软件可能只有 Windows 平台下的，也可能有 Linux 平台下的，也许还有 MacOS 平台^d下的软件。

每种我都介绍几个，当然还有很多，你也可以去找，都是可以找到的。这部分说明的目的在于让大家从软件的角度了解安全技术的原理。这些软件的排名没用先后，也没有什么特别推荐的，技术的水平永远是相对的，也许很多年后，还有更好的出来。这些仅仅做一个参考。

6.1 加密压缩软件

有一种保护秘密的办法是把秘密分成好几份，分别由几个人拿着。大家在一起的时候，数据就可以解读了。在密码学中，我们也使用一些专门的算法把一些无用的数据分插入明文中，这样可以达到混淆试听的目的。

^a 器：工具。要做好工作，先要使工具锋利。比喻要做好一件事，准备工作非常重要。

^b 由于冰和水的密度不一样，漂浮在水面的冰山总是只露出了一小部分的体积。水面上下的体积之比大约是 1:9。

^c 了解 Linux 系统访问：<http://baike.baidu.com/view/1634.html?wtp=tt>

^d 了解 MacOS 的原理可以访问：<http://baike.baidu.com/view/24778.html>

的。在计算机中虽然也使用这样的办法,但是密文会变的非常大,我们也害怕在 56kbps^a的调制解调器上这个数据传送过程很漫长。所以数据压缩技术解决了这个问题。我们可以压缩数据,甚至是乱序插入无用代码后再压缩数据。

我们生活中经常使用到压缩软件,除了可以节省硬盘空间,有的还可以附加密码。这个压缩技术又可以写一本厚厚的书了。可以肯定的是,对于一些不同的数据必须使用不同的算法,这也是最科学的一种方式。压缩的速度和效率与这个压缩算法有最直接的关系。压缩的时候也使用到一些像字典的数据,这点有点像密码学的意思了。

有时候我们可以认为压缩技术是密码学的一个特殊分支。不同就在于压缩技术是把字典和密文放在一起了,密码学就会分开它们。这和它们的用途有关,压缩技术的目的是使数据更小,而密码技术是让你看到密文就感到困惑。

压缩技术就像是充气皮艇,当我们走路的时候我们可以把气放出去,当我们过河的时候我们在把它吹大。或者你也可以把它比喻成其它东西。

这里简单的介绍一下压缩的分类,有损压缩^b和无损压缩^c。你可以参阅一些书籍^d来具体了解。一般来说,压缩的过程中输入输出的时候如果进行有损压缩,然后使用加密算法,解密还原回去的时候,数据当然发生了改变。这是有损压缩的定义决定的。而无损压缩后加密的数据经过还原可以变成一模一样的。在加密压缩数据的过程中,为了保持密文的完整性,我们一般使用的是无损压缩。你当然不希望解密后的明文中最关键的部分被省略了。下面从最开始流行的一个软件开始介绍。

6.1.1 WinZip

首先进入大众眼中的是 WinZip^e,你在到处都可以看到介绍:这个软件是一个强大并且易用的压缩实用程序,支持 ZIP、CAB、TAR、GZIP、MIME,以及更多格式的压缩文件。其特点是紧密地与 Windows 资源管理器拖放集成,不用离开资源管理器而进行压缩或解压缩。它被 PC Magazine 杂志授予最佳精品实用程序大奖。在微软的一些系统中已经完全支持不用其它软件就可以打开 Zip 文件。这个原因是软件厂商取得了 Zip 算法的使用权。

WinZip 为 Zip 文件提供了二种加密方法:

- AES 加密

有二种不同强度的加密方式: 128-bit AES 和 256-bit AES。数据安全依赖的不仅仅是加密方法的强度,也同样依赖于你的密码强度,包括因素有: 密度长度和复杂程序,以及你提供的密码保护方法,以保证你的密码不透露给未经授权的第三方。

- 标准 Zip 2.0 加密

这是一种旧的加密技术,提供了一定量的保护,用以防止文件被没有密码同时想确定文件内容的临时用户打开。可是, Zip 2.0 加密格式的强度相当弱,不要期望它来提供对单独使用特定密码恢复工具的

^a Kbps, 千波特率, 换算为 $\text{Kbps} = \frac{\text{A}}{8} \text{ kb/s}$, 56kbps 也就是 8kb 一秒的数据, 由于有网络线路的信号衰减问题和网络数据包的报头, 算出来的是理论值, 实际值会比理论值低一些。

^b 有损数据压缩, 方法是经过压缩、解压的数据与原始数据不同但是非常接近的压缩方法。有损数据压缩又称破坏型压缩, 即将次要的信息数据压缩掉, 牺牲一些质量来减少数据量, 使压缩比提高。这种方法经常用于因特网尤其是流媒体以及电话领域。访问: <http://baike.baidu.com/view/128147.htm>

^c 无损数据压缩 (Lossless Compression) 指数据经过压缩后信息不受损失, 还能完全恢复到压缩前的原样。访问: <http://baike.baidu.com/view/156047.htm>

^d Khalio Sayooo 的《Introduction to Data Compression (书籍压缩技术介绍)》, 这可就不是这么一本普通的“介绍”了。你可以访问这里阅读英文版: <http://www.verycd.com/topics/380781/>。

^e WinZip 官方网站: <http://www.winzip.com/>

保护。

你不应该依靠 Zip 2.0 加密来为你的数据提供强壮的安全保护。如果你的数据有重要安全需求,你应该考虑用 WinZip AES 加密来代替。Zip 2.0 加密仅有的超过 AES 加密的优势是它能被大部分 Zip 应用程序所支持,包括早先版本的 WinZip。使用这个技术加密的文件,可以被任何知道正确密码的人解压缩,而且几乎可以被任何 Zip 应用程序访问。加之,Zip 2.0 加密被 WinZip Self-Extractor 2.2 以上版本和 WinZip Self-Extractor 专业版(包含在 WinZip 中)所支持;上面描述的 AES 加密方法不被任何一个自解压程序所支持。

WinZip 的 AES 加密工具相对早先的 Zip 2.0 加密是一个重大进步,它可以帮助许多需要的 WinZip 用户保护自己的数据,防止他们的私密信息被未经授权的人查看。

WinZip 的 AES 密钥生成技术信息中,当你用 WinZip 进行 AES 加密时,你输入的密码将转化为适当长度的密钥(128-bit 或者 256-bit, AES 密钥的长度依赖于你的指定)。这是使用一个定义在 RFC 2898 (作为公用密钥密码系统标准#5 也可用)的 1000 的循环数,通过 PBKDF2 运算来完成的。WinZip 使用 8-byte Salt 值来用于 128-bit AES 加密, 16-byte Salt 值用于 256-bit 加密。

“Salt”值用于 WinZip AES 加密的目的是用来为每个文件生成不同的加密密钥,即使多个文件使用相同的密码进行加密。使用 8-byte salt 值用于 WinZip 的 128-bit 加密的结果是,如果有大约 40 亿个文件用相同的密码加密,才会有二个文件加密所用的密钥相同。只有获得二个使用相同密钥的文件副本才有可能解密它们的内容,因此它可以有效的延缓文件被解密的可能性。这也就是为什么推荐你加密数量非常大的文件用 WinZip AES 加密方法(也就是说,文件总数在百万,例如: 2000 个 Zip 文件(每个文件包含 1000 个加密文件)使用相同的密码进行加密的原因),你使用 256-bit AES 密钥,使用 16-byte 的 Salt 值,加密效果要比 128-bit AES 密钥,使用 8-bit Salt 值的效果更多。

作为处理的要点的部分,在 RFC2898 中必须调用一个随机函数;WinZip 使用 HMAC-SHA-1 函数用于这个目的,因为它是一个很好的运算法则,已经被广泛地使用了好多年。PBKDF2 函数重复调用 HMAC-SHA-1,生成一个 160-bit 散列值作为结果,用一个相当复杂的方法混合输出,最后产生一个 128-bit 或 256-bit 的加密密钥作为结果。

如果你使用 256-bit AES 加密,实际 HMAC-SHA-1 生成的是一个 160-bit 的结果,意思是说不管你指定的密码是什么,用于加密密钥的搜索空间不可能达到理论的 256-bit 最大值,并且不能保证超过 160 位。具体论述在 RFC2898 文件的 B.1.1 段落中有信息介绍。

6.1.2 WinRAR

随着技术的发展,出现了 WinRAR^a,除了支持 WinZip 支持的所有文件以外,支持 7Z、ACE、ARJ、BZ2、CAB、GZ、ISO、JAR、LZH、TAR、UUE、Z 等多种类型的压缩文件,可以估计压缩,还支持 WindowsNT 类系统在信息安全和数据流方面的功能。存在一系列的 RAR 版本,应用于数个操作系统环境: Windows、Linux、FreeBSD、DOS、OS/2、MacOS X。

WinRAR 支持 ZIP2.0 格式使用私有加密算法。RAR 压缩文件使用更强大的 AES-128 标准加密。如果你需要加密重要的信息,选择 RAR 压缩文件格式会比较好一些。为了确实的安全性,密码长度请最少要 8 个字符。不要使用任何语言的单词作为密码,最好是任意的随机组合字符和数字,并且要注意密码的大小写。RAR 压缩文件密码的最大长度是 127 个字符。较长的密码被裁切为此长度。

RAR 文件中能够添加到压缩文件中的文件数量,取决于可用内存和文件名的长度。每对 RAR 压缩文件添加一个文件,占用 128 个字节的内存。所以,如果你要压缩百万个文件以上的的话,推荐你要有 128MB 的内存。WinRAR 已经测试过可以管理超过一百万个以上的文件。

一个 RAR 压缩文件,或是在 RAR 压缩文件中的任何一个文件大小,它的大小限制为 8,589,934,591GB,如

^a WinRAR 的官方网站: <http://www.rarlab.com/>

果创建的压缩文件大于 4GB，你必须使用 NTFS 磁盘格式，旧式的文件系统不支持如此巨大的文件。ZIP 压缩文件中的任何一个文件大小限制为 2GB。常规状况下，RAR 压缩文件格式比较适用于在经常访问数 GB 磁盘空间的繁重任务。

我发现了一个很有趣的现象，很多软件下载网站，在提供 WinZip 下载的地址后面，提供了一个 RAR 的压缩包，也就是网站的工作人员把 WinZip 的安装包使用 WinRAR 压缩，这说明什么了？看来市场已经到处挤兑 WinZip，甚至 WinRAR 已经可以完全取代 WinZip，廉颇老矣，尚能饭否^a？

6.1.3 7Zip

7-Zip^b和上面 2 个软件最大的不通是这个软件免费。它是一款号称有着现今最高压缩比的压缩软件，它不仅支持独有的 7z 文件格式，而且还支持各种其它压缩文件格式。此软件压缩的压缩比要比普通 ZIP 文件高 30-50%，因此，它可以把 Zip 格式的文件再压缩 2-10%。7-Zip 是一款开源软件。大多数源代码都基于 GNU LGPL 许可协议^c下发布。AES 代码基于 BSD 许可下发布。unRAR 代码基于两种许可：GNU LGPL 和 unRAR 限制许可。

7z 式支持强大的 AES-256 加密，最高支持 16000000000GB 的文件压缩。

7-Zip 软件的帮助中对 AES 加密攻击时间的举例，设想一个用户每秒钟重试密码 10 次，一个组织拥有 10 亿美元的资金，可以达到每秒重试 100 亿次密码：

表格 11：破译密码需要的时间

密码长度	单用户攻击破译时间	集群式计算机攻击破译时间
1	2 秒	1 秒
2	1 分钟	1 秒
3	30 分钟	1 秒
4	12 小时	1 秒
5	14 天	1 秒
6	1 年	1 秒
7	10 年	1 秒
8	19 年	20 秒
9	26 年	9 分钟
10	37 年	4 小时
11	46 年	4 天
12	55 年	4 个月
13	64 年	4 年
14	73 年	13 年
15	82 年	22 年
16	91 年	31 年
17	100 年	40 年

^a 出自《史记·廉颇蔺相如列传》，意思是廉颇老了，还能吃下饭吗？表示到了晚年还要为国出力。这里表示对 WinZip 的发展产生疑问。

^b 7Zip 的官方网站：<http://www.7-zip.org/>

^c GNU LGPL 信息：这一函数库是免费软件，您可以遵照免费软件基金会出版的 GNU，次要通用公共许可协议条款来修改和重新发布这一程序，或者用许可协议的第二版，或者（根据您的选择）用任何更新的版本。发布这一函数库的目的是希望它有用，但没有任何担保。甚至没有适合特定目的而隐含的担保。更详细的情况请参阅 GNU 次要通用公共许可协议。您应该已经和函数库一起收到一份 GNU 次要通用公共许可协议的副本。如果还没有，写信给 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

没准 4 核 CPU 或者更多核的处理器普及后, 只要把破解算法的源代码在多核处理器上优化一下, 性能就可以翻倍。所以选择一个好密码(或者我应该叫口令)对于数据安全十分主要。

需要提醒的是, 在极限压缩的情况下, 7-Zip 软件的内存占用会达 600 多 MB, 如果你没有 1GB 或 2GB 的内存那就很小心了, 如果你那样做了, 会得到一个压缩失败的提示。解压过程也需要压缩时那么大的内存空间, 这个是很夸张的。不过对于一般的用途完全可以满足。

6.1.4 UHARC

UHARC 的超强压缩甚至可以让 RAR、7z 汗颜, 除此之外它还支持创建自解压缩档案。受到一些特殊算法的支持。

由于受到 WinRAR 本身的限制, 很多文件还是无法压缩得更小, 这时我们可以试试 UHarc 这个号称世界上压缩比最高的软件。这是一款极品压缩软件, 作者 Uwe Herklotz, 其超强压缩比可以让 ZIP、RAR、ACE、7-Zip 汗颜, 越大的文件, 越能体现出 UHarc 压缩的优势, 不过 UHarc 提高了压缩率, 却是以花费更多的时间为代价的。我第一次接触 UHARC 是从某某盗版软件的安装界面中看到的。黑黑的命令提示符窗口让我以为这个软件是从 DOS 时代过来的。目前互联网上的很多盗版游戏的安装包内都有这个解包过程。

UHARC 的技术没有公开化, 我也没有联系到作者, 究竟它的加密是什么原理实在说不清楚, 这个不作为推荐的软件, 不是很可靠, 但是肯定的是使用了私有算法。胆大的可以试一下。

确实, 文件压缩的越小, 在解密的时候需要的更多的技术来支持, 如果有人想看看专业的压缩软件的效率究竟会有多高的话, 可以参考这个网站^a。上面列举出了网站测试人员测试后的结果, 你可以找到世界上最高压缩率的软件。需要注意的是, 你可以找到很多所谓的破解 RAR 文件密码和破解 ZIP 文件密码的软件, 其实使用的都是暴力破解法, 不断的测试新生成的密码。你如果仔细阅读芝麻开门的技术含量那节的话, 选择一个好口令会让破解者很无奈。

6.2 邮件加密软件

现在我们更多的使用即时聊天软件了, 什么 MSN、QQ、ICQ、Skype 等等, 现在的聊天软件的技术的几乎是公开化的, 编一个简单的 TCP/IP 网络协议的聊天软件是很容易的。

一般的主流 IM 软件在消息传输上做了加密, 但是服务器是可以保存和查看用户的聊天资料的。这点容易被黑客利用。那种可以在即时聊天的软件中使用的加密软件目前还没有看到, 也就只能把文本加密后, 在 IM 联系人发送窗口中复制粘贴密文。哪怕是几个字也生成一大堆的密文。这个领域的软件现在还是一个空白。聊天软件的开发商也依据当地的法律法规来设置技术等级。比如允许政府检查部门对信息核查。这个规定是双刃剑, 既然允许政府部门核查, 保障社会稳定。理论上那也允许黑客进行“核查”, 可以泄露个人的消息信息。

在手机中, 我们几乎非常普遍的使用 SMS(短信)来交流^b, 你可以发现有完善的垃圾短信举报措施。那也就意味着服务器(运营商)完全可以查看到你的短信内容。我们的隐私再次的暴露。其实运营商也无心去管东家长西家短。但是做到对某个号码的记录完全可能, 你可以看一下你的话费清单, 短信清单除了没有消息内容, 其他信息都全了, 做到记录消息内容不是不可以, 一条短信 70 个汉字, 不需要太大的记录文件。做

^a 最大的压缩软件和技术, 访问: <http://www.maximumcompression.com/index.html>

^b 有统计数据称, 亚洲国家的人比欧美更喜欢发短信, 这是由于亚洲人更含蓄。也就只有亚洲区域可以产生 SMS 发送消息量的世界纪录。而男人女人谁喜欢发短信, 这个就难说了, 无论是谁发给谁, 对方一般会回复一条, 在数量上看不出特点, 也许可以不准确的认为女人喜欢发短信。

到手机 SMS 内容加密的软件已经出来了。除了短信的体积变成了 3 条多的体积外, 这个基于公钥算法的方案是很完美的。

我们习惯在 BBS 留言寻求帮助, 会有热心的网友回答你的问题, 各类社区网络无处不在。联系人可以使用电话, 或者是短信。而感觉我们以前用来交流的电子邮件有点落伍了, 它真的过时了吗?

其实在商业领域, 这个技术还是很活跃的。在很多的商业活动中, 电子元件还扮演着很重要的角色。在公司的不同机构交流的时候, 除了开会的办法, 还有电子邮件联系。电子邮件可以包含文字、图片、附件, 并且可以附带字体、版面的格式, 电子邮件是目前远程交流中相对安全的东西。其实, 欧美地区使用电子邮件还是比较多的。

感谢密码技术使这个成为了现实, 下面就开始介绍几个不错的邮件加密软件。一般的邮件几乎使用的都是 PKI (Public Key Infrastructure, 公钥基础设施) 技术的。

6.2.1 The Bat

这个软件^a在北美国家的人人群里很流行, 它不是美国人的软件。公司 10 个人不到, 蝙蝠在中国的文化里面是“福”, 这句话在 The Bat 的 Tip of the day(每日一语), 里面有这么一句: The Chinese word for bat, fu, also means "happiness". Five bats represent the Five Blessings-longevity, wealth, serenity, virtue, and an easy death (汉语中蝙蝠的“蝠”的读音和“福”一样, 表示“幸福”。五只蝙蝠表示五福: 长寿、富贵、康宁、好德、善终。)。这个软件的名称确实粘上点中国文化的味道。

软件加密使用了 OpenPGP 的标准, 当然也可以使用 PGP 软件的引擎, 支持 S/MIME 和 TLS 证书。加密算法支持 RC2-128-bit、3DES-156-bit、IDEA-128-bit、AES-128-bit、AES-256-bit, 还支持最高到 SHA-512 的散列算法。还有本地邮件数据的加密, 并且可以为每一个邮箱设置单独的密码, 软件有完善的保护策略。这个程序已经完全支持中文界面, 而且对中文的显示也不错。

你开始会发现 TheBat!对有些的邮件服务器支持不太好, 主要体现在不能通过服务器发送邮件, 服务器认定邮件是垃圾邮件, 这是服务器反垃圾邮件的功能的作用, 在邮件中使用 X-Mailer 邮件头字段后(当然有时候你也要去掉 X-Mailer 邮件头字段才可以发送), 一些主流的服务商的服务器基本都没有问题。TheBat!的功能还是不错的。

值得一提的是 TheBat!的密钥不能导出 V3 版的密钥格式, 这导致这些 Key 不能被 PGP 识别。

6.2.2 Foxmail

Foxmail^b的开发人是原华中理工大学张小龙, 2005 年 3 月 16 日被腾讯收购。

Foxmail 具备强大的反垃圾邮件功能。数字签名和加密功能在 Foxmail5.0 中得到支持, 可以确保电子邮件的真实性和保密性。通过安全套接层(SSL)协议收发邮件使得在邮件接收和发送过程中, 传输的数据都经过严格的加密, 有效防止黑客窃听, 保证数据安全。其他改进包括: 阅读和发送国际邮件(支持 Unicode)、地址簿同步、通过安全套接层(SSL)协议收发邮件、收取 yahoo.com 邮箱邮件; 提高收发 Hotmail、MSN 电子邮件速度支持名片(vCard)、以嵌入方式显示附件图片、增强本地邮箱邮件搜索功能等等。基本上该有的功能都有了。可以说它是目前国内邮件客户端软件最领先的。

Foxmail 的界面美观, 极强的易用性, 这也就是它吸引中国网民的原因。

当然, 还有一些其它的邮件加密软件, 如: Flexcrypt^c, CenturionMail^a, Comodo SecureEmail^b等等。不推

^a The Bat 的官方网站: <http://www.rttabs.com/>, 该软件收费, 免费使用 30 天。

^b Foxmail 官方网站: <http://www.foxmail.com.cn/>, Foxmail 手机网址: <http://wap.foxmail.com/>

^c Flexcrypt 官方网站: <http://www.flexcrypt.com/>

荐使用一些私有算法的简单的文本加密软件, 因为这类软件的算法可以从程序反编译被找出来, 究竟有多可靠谁也说不清楚, 还是用大家公认的安全技术要放心的多。

6.3 数据安全删除软件

我们在加密所有的文件的时候会遇到一些不安全的因素, 比如我们要将一些文件加密, 加密后变成一个文件包, 这时候我们会删除我们先前的一些文件。如果你简单的放入回收站, 那是绝对不安全的, 任何人可以简单的还原文件, 甚至我们清空回收站, 我们的文件还可以通过专业软件找回来。运气好的话, 我们找回来的文件完好无损。

这里的问题是如果有人拿到了你的硬盘, 或者你的笔记本, 他也可以用这些技术轻易的恢复数据。这时你的数据就暴露无遗, 他也不用去操心如何解密你的加密文件, 所有的原始文件在硬盘上都可以找到。这些恢复技术是基于一些存储介质的性质上的, 下面要介绍一些存储器的常识:

无论是 IDE 接口硬盘、SATA 接口硬盘, 还是 SCSI 接口硬盘, 你看到的有一个像转轮的外壳模型的硬盘就是最普通的硬盘, 采用的都是“温彻思特”技术, 它的特点是:

1. 由磁头, 盘片和马达, 密封结构和控制电路构成, 盘片在马达的带动下旋转。
2. 固定并高速旋转的镀磁盘片表面平整光滑。
3. 磁头沿盘片径向移动。
4. 磁头对盘片接触式启停, 但工作时呈飞行状态不与盘片直接接触。

硬盘是磁性媒介, 就像录音带一样, 每一个记录位只容许一种极性存在, 只要磁头有电流生成, 是会去改变其原值。由于是磁头悬浮的, 所以比较怕震动。它的寿命是比较长的。

闪存 (Flash ROM) 是一种电擦除非易失型存储器, 由浮栅型场效应管构成, 写入时, 利用热电子注入, 使浮栅带电; 擦除时, 则利用高压下的隧道效应, 使浮栅失去电子。

闪存储单元分为两类: SLC (Single Layer Cell 单层单元) 和 MLC (Multi-Level Cell 多层单元)。SLC 的特点是成本高、容量小、速度快, 而 MLC 的特点是容量大成本低, 但是速度慢。MLC 的每个单元是 2bit 的, 相对 SLC 来说整整多了一倍。不过, 由于每个 MLC 存储单元中存放的资料较多, 结构相对复杂, 出错的几率会增加, 必须进行错误修正, 这个动作导致其性能大幅落后于结构简单的 SLC 闪存。此外, SLC 闪存的优点是复写次数高达 10 万次, 比 MLC 闪存高 10 倍。此外, 为了保证 MLC 的寿命, 控制芯片都校验和智能磨损平衡技术算法, 使得每个存储单元的写入次数可以平均分摊, 达到 100 万小时故障间隔时间 (MTBF)。闪存是半导体技术, 内部是相对静态的, 体积小, 抗震性很高, 所以便于携带。加上半导体技术发展很快, 价格下降也很快。新出来的固态硬盘也是这个技术。

所以, 在一些删除算法上面就要区分磁盘和闪存的区别。一般说来磁盘的问题要大一些。

这样的软件太多了, 光是算法相同的就不在少数。下面介绍的一些软件就完全可以解决这些难题, 注意他们是文件安全删除软件^c, 而不是顽固文件删除软件^d。有时候我们更喜欢炒作简易的工具。

^a Centurion 官方网站: <http://centurionsoft.com/centurionmail/>

^b Comodo SecureEmail 官方网站: <http://www.secure-email.comodo.com/index.html>, Comodo 的网络防火墙也是很不错的。

^c 可以擦除记忆体设备上面的数据记录, 使得这些数据很难恢复。

^d 删除一些删不掉的文件, 比如病毒产生的一些只在使用的文件。这些文件会被系统锁定, 无法删除, 专业的软件可以达到这个效果。

6.3.1 O&O Soft SafeErase

O&O 公司^a的磁盘工具软件，可以绝对安全的删除敏感文件和资料，删除后不可恢复。SafeErase 不仅仅表面上删除了文件名，并且会把文件占用的空间全部改写，这样那些黑客和写偷窃数据的“间谍”也只能望盘兴叹。SafeErase 有点特别，不像大部分软件一样有一个运行的界面，它本身没有一个程序供你运行，而是完全集成到资源管理器的右键菜单里面。

如何删除和删除时的安全级别你可以根据你的文件的安全程度来选择，高标准的删除需要更长的时间。越高的安全级别数据被擦除的次数就越多，一共 5 种模式。

- 最低安全等级 (1 次擦除)

5 种方法里面最快的模式，但是也是对抗数据恢复最弱的方法。数据只被一个随机数字改写一遍。

- 低安全等级 (3 次擦除)

这个模式是依据 1995 年 1 月美国国防部 (DoD) 的 (US DoD 5220.22-M)中的'National Industry Security Program Operating Manual (国家产业安全程序操作手册)' 的规定。新的更新 (DoD 5220.22-M E) 提供了 3 次擦除：第一次用固定值改写，验证一遍，再用随机数字改写一遍。速度相对快，安全性一般。

- 中等安全等级 (6 次擦除)

这个中等安全等级是依据德国 BSI 的“BSI IT Baseline Protection Manual (BSI IT 低线保护手册)”中的标准。数据用随机数字改写然后进行值验证，这个过程使用新的随机数值 3 次来回改写。

- 高安全等级 (7 次擦除)

该等级依据 1995 年 1 月 DoD 的'National Industrial Security Program Operating Manual (国家产业安全程序操作手册)'。7 次数据变化(DoD 5220.22-M ECE)，第一步用 DoD 5220.22-M (E)标准改写 3 次，然后用特殊的随机数字改写，最后再使用 DoD 5220.22-M (E)标准改写。

- 最高安全等级 (35 次擦除)

该模式依据 Peter Gutmann 的文章“Secure Deletion of Data from Magnetic and Solid-State Memory^b”的过程。再这个方法中，删除的数据被一系列的 35 个删除方法中的所有算法每个都随机执行一次。删除的时间也是最长的。

可以说这个过程已经相当完善，最高安全等级的时间是在是有点太长了。如果还有不放心的地方你还可以看看其它的软件。

6.3.2 East-Tec DisposeSecure

East-Tec DisposeSecure 可以从计算机硬盘的彻底清除数据和痕迹，根据美国国防部标准，或者设置的用户安全水平进行清除，确保被删除的文件不能被其他任何工具恢复。软件很小巧，能够被从一张软盘，CD 或者 DVD 运行，或者无论文件系统或者操作系统。是否你想阻止特性盗窃，保护你的隐私或者阻止竞争者偷

^a O&O Soft 官方网站: <http://www.oo-software.com/>

^b 该文章可以访问: http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html PDF 格式:
http://www.cs.cornell.edu/people/darkson/secdg/papers.sp06/secure_deletion.pdf

你的机密信息。软件支持多操作系统平台, 也支持很多分区格式。被誉为“Ultimate tool (超级工具)”。最好是一个有比较好的计算机操作基础的的人来进行操作, 不然一些很重要的数据会不小心删除, 这样的话, 神也找不回来了。

这个工具被美国司法部律师办公室使用。也得到很多媒体的赞赏和嘉奖。有兴趣的可以去他们的网站^a看看。

6.3.3 Linux 类系统下的工具

Windows 平台下面的软件很丰富, 但是在 Linux 等系统中, 这样丰富多彩的景象就不存在了。我们必须使用一些算法重新在 Linux 下编译的新程序。当然, 为了高效的使用, 这类程序基本没用什么花哨的外观, Linux 也可以说是一个这样的系统, 尽管现在发展的很不错了。linux 以下就需要一些专门的方法^b。这些程序很像是我们在 DOS 时代的作风。因为 Linux 下的一些资料不好找, 所以干脆找了一篇专业人士的文章。当然, 这些程序中有很多提供免费的源代码, 你可以下载后修改代码重新在 linux 下面编译, 也可以把所有的功能集成到一个程序中, 一旦拥有, 别无所求。

删除文件或者重新格式化磁盘并不会破坏敏感数据, 即使被删除这些数据也可以恢复。彻底删除某些无用数据是好事, 但是如果删除的是财务数据、银行帐号密码或者分类公司信息而无法彻底删除就很危险, 本文将介绍一些能够在 Linux Ubuntu 操作系统中安全删除文件的工具。

Shred 命令

虽然 shred 命令有一些限制, 但是 shred 命令可以很有效地彻底删除文件, 使文件很难或者无法恢复。Shred 是这样进行彻底删除的: 通过使用数据模式对文件进行反复重写以实现最大程度的摧毁, 这样即使使用高灵敏度的数据恢复设备也很难恢复数据了。

使用 rm 命令删除文件实际上并没有毁坏数据, 它只是毁坏列有文件位置的索引, 使文件的数据块还可以重复使用。因此, 使用 rm 命令删除的文件可以很容易使用特殊设备或者命令进行恢复, 前提条件是文件释放的数据库还没有被重复利用。然而, 如果在完全是硬盘驱动(HDD)的动态系统, 释放的空间只能在几分钟或者几秒钟内进行恢复。

Shred 语句: **shred [option(s)] file(s)_or_devices(s)**

选项:

- f, - force - 更改权限允许写入(如有必要)
- n, - iterations=N - 重写 N 次而不是默认的(25)次
- s, - size=N - 将文件粉碎为很多字节(可使用后缀如 K、M、C 等)
- u, - remove - 重写后截短和移除文件
- v, - verbose - 显示进程
- x, - exact - do not round file sizes up to the next full block
- z, - zero - add a final overwrite with zeros to hide shredding
- shred 标准输出
- help - 显示帮助并退出
- version - 输出版本信息并退出

示例:

^a East-Tec 的官方网站: <http://www.east-tec.com>

^b 下文来自邹铮《点评 Ubuntu 下的文件安全删除工具》, 原文访问: <http://linux.chinaitlab.com/safe/775628.html>

- 1) 下列命令能够用于安全删除三个文件夹: file1、file2 和 file3:

```
shred file1 file2 file3
```

- 2) 下列命令将删除第一个 HDD 的第七个分区上的数据:

```
shred /dev/hda7
```

- 3) 可以利用下列命令来擦除你在第一个盘的软盘中创建的文件系统路径, 该命令需要花费 20 分钟来擦除 “1.44MB” (实际为 1440KB)的软盘:

```
shred -verbose /dev/fd0
```

- 4) 要想删除硬盘上所选分区的数据, 可以使用下列命令:

```
shred -verbose /dev/sda5
```

下面介绍在 Ubuntu 系统中的 Nautilus 菜单中添加 shred , 首先在终端使用下列命令安装 nautilus-actions 数据包:

```
sudo aptitude install nautilus-actions
```

现在打开 Nautilus Actions Configuration: System(系统)->Preference(首选项)->Nautilus Actions Configuration 打开后你会看到一个窗口, 点击添加。
现在输入以下信息:

```
Label: Shred
Tooltip: shred utility to securely erase files
Icon: gtk-dialog-warning
Path: shred
Parameters: -f -u -v -z %M
```

点击 Conditions 条目, 勾选 “Appears if selection contains” 上面的 “Only files (只是文件)” (也可以选择文件和文件夹), 勾选 “Appears if selection has multiple files or folders (如果选中多个文件或文件夹时显示)”, 单击确定。

添加后你将看到一个窗口, 点击关闭就可以。

打开终端, 运行下列程序对 nautilus 进行更新:


```
nautilus -q
nautilus
```

这样将打开 `nautilus` 窗口，现在右击单击任何文件，就能在菜单中看到 `shred` 命令选项。

Wipe 命令

`wipe` 是一种小命令，能够安全删除磁性媒介上的文件，该命令能够在各种 `unix` 平台进行汇编，包括 `Linux 2.*`、`(Open+Net+Free)BSD`、`aix 4.1`、`SunOS 5.5.1` 以及 `Solaris 2.6` 等。从磁性媒介上恢复已经被擦除的数据要比人们想象的容易得多，一种称为“`Magnetic Force Microscopy (MFM)`”的技术能够恢复写入磁盘的最后 2 层或者 3 层的数据。而 `wipe` 命令能够向已删除的文件反复写入特殊图案，使用 `fsync()` call 和/或 `O_SYNC` bit 强迫磁盘访问。

在 `Ubuntu` 中安装 `wipe`

```
sudo aptitude install wipe
```

`wipe` 语句:`wipe [options] path1 path2 ... pathn`

示例:

擦除 `/home/berke/plaintext/` 下的每个文件和每个目录(option `-r`)，常规文件将使用 34 次 `passes` 擦除，它们的大小也将骤减几倍。而特殊文件(字符和块设备、`FIFO` 等)则不会这样。所有目录条目(文件、特殊 文件和目录)将被重新命名 10 次，然后用 `unlink` 操作符删除文件。不被允许的操作将进行 `chmod()`处理(option `-c`)。这些都不需要用户确认(option `-f`):

```
wipe -rcf /home/berke/plaintext/
```

假设 `/dev/hda3` 是与主要 IDE 界面的主盘第三区相对应的，则将在快速模式(option `-q`)下被擦除，例如，使用四种任意 `passes`。Inode 不会被重新命名或者 `unlink`，在开始前，系统将要求输入“确认”:

```
wipe -kq /dev/hda3
```

`wipe` 从来不会按照 `symlink` 进行文件删除，除非用户明确要求使用 `symlink`，如果你想要擦除 `/dev/floppy`，而它正是到 `/dev/fd0u1440` 的 `symlink`，则需要明确 `-D` 选项，在开始前，系统将要求输入“确认”:

```
wipe -kqD /dev/floppy
```

在这里，`wipe` 将递归式地(option `-r`)删除 `/var/log` 下的数据，而非 `/var/log` 之外的数据。这将不再是 `chmod()`，而会变得有点复杂(option `-i`)。而且因为 `-f` 选项，系统不会要求用户输入“确认”:

```
wipe -rfi >wipe.log /var/log/*
```

根据操作系统的不同特质,要想获取特定设备可能包含的 **byte** 数量并不是易事(事实上,这个数字是变化的)。这就是为什么有时候需要告诉 **wipe** 擦除 **byte** 数量的原因,也是 **-l** option 的目的。另外,也可以使用 **b**, **K**, **M** 和 **G** 作为乘法器,分别对应 2^9 (512)、 2^{10} (1024, 或是 Kilo)、 2^{20} (Mega)、 2^{30} (Giga)bytes。甚至还可以联合两个乘法器,这样 **1M416K** = 1474560 bytes:

```
wipe -Kq -l 1440k /dev/fd0
```

在 **Ubuntu** 的 **Nautilus** 菜单中添加 **Wipe**, 首先在终端使用下列命令安装 **nautilus-actions** 数据包:

```
sudo aptitude install nautilus-actions
```

现在打开 **Nautilus Actions Configuration: System->Preference->Nautilus Actions Configuration** 输入以下信息:

```
Label: Wipe
Tooltip: Wipe utility to securely erase files
Icon: gtk-dialog-warning
Path: wipe
Parameters: -rf %M
```

点击 **Conditions** 条目,勾选“**Appears if selection contains**”上面的“**both**”,勾选“**Appears if selection has multiple files or folders**”,单击确定。打开终端,运行下列程序对 **nautilus** 进行更新:

```
nautilus -q
nautilus
```

这样将打开 **nautilus** 窗口,现在右击单击任何文件,就能在菜单中看到 **wipe** 命令选项。

Secure-Delete 安全删除工具

该软件包包含很多安全删除工具,这些工具能够安全清除文件数据、释放磁盘空间、**swap** 和 **memory**。**Secure-Delete** 工具利用先进的技术对文件进行永久性删除,是非常实用的工具。想在 **Ubuntu** 中安装 **Secure-Delete** 工具,需要运行下列命令:

```
sudo aptitude install secure-delete
```

Secure-Delete 软件包中包含下列命令:

srm(Secure remove) - 用于删除硬盘上现有的文件或者目录。

smem(Secure memory wiper) - 用于清除计算机内存(RAM)的数据痕迹。

sfill(Secure free space wiper) - 用于清除磁盘可用空间的数据痕迹。

sswap(Secure swap wiper) - 用于删除 swap 分区所有数据痕迹。

srm - Secure remove 安全移除

srm 通过覆盖、重命名和 unlink 前截断数据来删除指定的文件，这可以防止其他人利用命令恢复或复原文件。

srm，像每一个使用 getopt 函数解析句法的程序一样，可以让用户使用一 option 来表明所有句法都是非选项。想要移除当前目录中 '-f' 文件，可以输入“srm -f”或者“srm ./-f”。

srm 语句:srm [OPTION]... FILE...

选项:

- d, -directory - 忽略(与 rm 的兼容性)
- f, -force - 忽略不存在的文件，从不提示
- i, -interactive - 在任何清除操作前的提示
- r, -R, -recursive - 递归地移除目录内容
- s, -simple - 仅使用随机数据的单一 pass 进行覆盖
- m, -medium - 使用 7 US DoD 兼容 passes(0xF6,0×00,0xFF,random,0×00,0xFF,random)重写文件
- z, -zero - 覆盖文件后，文件使用的 zero 块
- n, -nounlink - 覆盖文件，但是并不对文件重新命名或者 unlink
- v, -verbose - 显示正在进行的操作
- help - 显示帮助并退出
- version - 输出版本信息并退出

示例:

使用 srm 删除文件 myfile.txt:

```
srm myfile.txt
```

使用 srm 删除目录

```
srm -r myfiles
```

smem - Secure memory wiper 安全内存擦除器

smem 旨在删除仍以安全方式存在内存的数据，这些数据不能被黑客、执法人员或者其他威胁恢复的。请注意，有了新的 SDRAM 芯片，数据不会消亡而将保持静态，很容易被提取必要的信息!这种擦除机制是基于第六次 Usenix 安全专题讨论会上由 Peter Gutmann 发表的“从磁性和固态内存安全删除数据”文件，Peter Gutmann 是著名的译解密码专家。

smem 语句: **smem [-f] [-l] [-l] [-v]**

选项:

- f - 快速(和不安全模式): no /dev/urandom。
- l - 降低安全性。只写了两个 passes: 0×00 和任意 pass
- l-l 再次降低了安全性: 只写入了 0×00 pass
- v - 详细模式

sfill - secure free space wipe 安全可用空间擦除

sfill 旨在删除仍以安全方式存在于可用空间媒介上的数据, 这些数据不能被黑客、执法人员或者其他威胁恢复的。这种擦除机制是基于第六次 Usenix 安全专题讨论会上由 Peter Gutmann 发表的“从磁性和固态硬盘安全删除数据”文件, Peter Gutmann 是著名的译解密码专家。

Sfill 语句: **sfill [-f] [-i] [-l] [-l] [-l] [-v] [-z] directory/mountpoint**

选项:

- f-快速(和不安全模式): no /dev/urandom, 没有同步模式
- i - 仅清除可用 inode 空间, 而不是可用磁盘空间
- l-仅清除可用磁盘空间, 而不是可用 inode 空间
- l-降低安全性。只写了两个 passes: 0xff 的 pass 和随机值的最后模式
- l-l 再次降低了安全性: 只写入了随机 pass
- v-详细模式
- z - 使用 0 擦除最后的写入, 而不是随机数据

directory/mountpoint 是在文件系统中用户所创建的文件的位置, 应该位于用户想写入的分区。

sswap - Secure swap wiper 安全 swap 擦除器

sswap 旨在删除仍以安全方式存在于 swap 空间的数据, 这些数据不能被黑客、执法人员或者其他威胁恢复的, 这种擦除机制是基于第六次 Usenix 安全专题讨论会上由 Peter Gutmann 发表的“从磁性和固态硬盘安全删除数据”文件, Peter Gutmann 是著名的译解密码专家。

Sswap 语句: **sswap [-f] [-l] [-l] [-v] [-z] swapdevice**

选项:

- f-快速(和不安全模式): no /dev/urandom, 没有同步模式
- l-降低安全性。只写了两个 passes: 0xff 的 pass 和随机值的最后模式
- l-l 再次降低了安全性: 只写入了随机值的 pass
- v - 详细模式
- z -使用 0 擦除最后的写入, 而不是随机数据

示例:

在开始使用 sswap 前, 你必须禁用 swap 分区, 可以使用下列命令来查看安装的 swap 设备:

```
cat /proc/swaps
```

使用下列命令禁用 swap:

```
sudo swapoff /dev/sda3
```

/dev/sda3 - 这是我的 swap 设备

禁用 swap 设备后, 就可以使用下列命令用 `sswap` 擦除它:

```
sudo sswap /dev/sda3
```

完成上述命令后, 需要使用下列命令重新启用 swap:

```
sudo swapon /dev/sda3
```

当然还有一些其它工具:DBAN

Darik 的 Boot 和 Nuke (“DBAN “)是独立的启动盘, 能够安全清除大多数计算机的硬盘数据。DBAN 能够自动完全删除它检查到的任何硬盘数据, 这也使 DBAN 成为删除大量数据或者紧急数据删除的实用工具。

6.4 系统级别加密软件

如果说一些小软件在系统里面小打小闹的话, 那下面的几个就算的上是系统级别的了, 因为他们为系统提供全方位的安全。有的是设计了系统底层驱动, 用最底层控制来加密和解密文件, 这样做的好处是在系统的一系列软件不需要额外的 API 调用就可以访问加密的文件。有的软件甚至将一些新鲜元素引入, 你会看到虚拟加密磁盘, 加密邮件, 加密网络共享等一系列的技术。

6.4.1 MicroSoft EFS

EFS(Encrypting File System, 加密文件系统)是 Windows 2000/XP 所特有的一个实用功能, 对于 NTFS 卷上的文件和数据, 都可以直接加密保存, 在很大程度上提高了数据的安全性。

EFS 加密是基于公钥策略的。在使用 EFS 加密一个文件或文件夹时, 系统首先会生成一个由伪随机数组成的 FEK(File Encryption Key, 文件加密钥匙), 然后将利用 FEK 和数据扩展标准 X 算法创建加密后的文件, 并把它存储到硬盘上, 同时删除未加密的原始文件。随后系统利用你的公钥加密 FEK, 并把加密后的 FEK 存储在同一个加密文件中。而在访问被加密的文件时, 系统首先利用当前用户的私钥解密 FEK, 然后利用 FEK 解密出文件。在首次使用 EFS 时, 如果用户还没有公钥/私钥对(统称为密钥), 则会首先生成密钥, 然后加密数据。如果你登录到了域环境中, 密钥的生成依赖于域控制器, 否则依赖于本地机器。

EFS 加密系统对用户是透明的。这也就是说, 如果你加密了一些数据, 那么你对这些数据的访问将是完全允许的, 并不会受到任何限制。而其他非授权用户试图访问加密过的数据时, 就会收到“访问拒绝”的错误提示。EFS 加密的用户验证过程是在登录 Windows 时进行的, 只要登录到 Windows, 就可以打开任何一个被

授权的加密文件。注意: Windows XP Home 并不支持 EFS 加密。

如果其他人想共享经过 EFS 加密的文件或文件夹, 只需要安装证书。由于重装系统后, SID(安全标示符)的改变会使原来由 EFS 加密的文件无法打开, 所以为了保证别人能共享 EFS 加密文件或者重装系统后可以打开 EFS 加密文件, 必须要进行备份证书。完成证书的导入, 这样就可顺利打开所加密的文件。

EFS 在算法上没有任何破解的办法, 但是为什么你还可以在互联网上找到破解 EFS 文件的工具? 其实这样的软件和方法不少, 软件是利用了操作系统把密钥文件保存在硬盘分区上的这个漏洞, 这个就相当于钥匙和保险箱就放在一个杂乱的房间里了, 找到钥匙就可以了。专业软件会搜索密钥文件, 如果找不到, 显然它的本事就没有那么大了。破解 EFS 的人工方法无非也是一些欺骗操作系统的做法, 过程也需要软件的辅助。

所以, 我们要妥善的备份密钥, 这样在恢复的时候还可以找回文件, 但是也要注意计算机的使用权限问题, 一定不能轻易的让别人动用自己的电脑。这是成本最低的一种加密保护, 因为在微软专业版的操作系统中都集成了这个功能。

6.4.2 MicroSoft BitLocker

BitLocker 驱动器加密是在 Windows Vista 中新增的一种数据保护功能, 主要用于解决一个人们越来越关心的问题: 由计算机设备的物理丢失导致的数据失窃或恶意泄漏。Windows BitLocker 驱动器加密是一种全新的安全功能, 该功能通过加密 Windows 操作系统卷上存储的所有数据可以更好地保护计算机中的数据。BitLocker 使用 TPM 帮助保护 Windows 操作系统和用户数据, 并帮助确保计算机即使在无人参与、丢失或被盗的情况下也不会被篡改。受信任的平台模块(TPM)是一个内置在计算机中的微芯片。它用于存储加密信息, 如加密密钥。存储在 TPM 上的信息会更安全, 避免受到外部软件攻击和物理盗窃。BitLocker 也使用的是 AES 算法。

BitLocker 还可以在沒有 TPM 的情况下使用。若要在计算机上使用 BitLocker 而不使用 TPM, 则必须通过使用组策略更改 BitLocker 安装向导的默认行为, 或通过使用脚本配置 BitLocker。使用 BitLocker 而不使用 TPM 时, 所需加密密钥存储在 USB 闪存驱动器中, 必须提供该驱动器才能解锁存储在卷上的数据。

它的工作原理是通过加密整个 Windows 操作系统卷保护数据。在启动过程中, TPM 芯片将释放密钥, 该密钥仅在将重要操作系统配置值的一个哈希值与一个先前所拍摄的快照进行比较之后解锁加密分区。这将验证 Windows 启动过程的完整性。如果 TPM 检测到 Windows 安装已被篡改, 则不会释放密钥。

默认情况下, BitLocker 安装向导配置为与 TPM 无缝使用。管理员可以使用组策略或脚本启用其他功能和选项。为了增强安全性, 可以将 TPM 与用户输入的 PIN 或存储在 USB 闪存驱动器上的启动密钥组合使用。在不带有兼容 TPM 的计算机上, BitLocker 可以提供加密, 而不提供使用 TPM 锁定密钥的其他安全。在这种情况下, 用户需要创建一个存储在 USB 闪存驱动器上的启动密钥。

特别要注意 TPM 这个微芯片, 它设计用于提供基本安全性相关功能, 主要涉及加密密钥。TPM 通常安装在台式计算机或者便携式计算机的主板上, 通过硬件总线与系统其余部分通信。

合并了 TPM 的计算机能够创建加密密钥并对其进行加密, 以便只可以由 TPM 解密。此过程通常称作“覆盖”或“绑定”密钥, 可以帮助避免泄露密钥。每个 TPM 有一个主覆盖密钥, 称为“存储根密钥(SRK)”, 它存储在 TPM 的内部。在 TPM 中创建的密钥的隐私部分从不暴露给其他组件、软件、进程或者人员。

合并了 TPM 的计算机还可以创建一个密钥, 该密钥不仅被覆盖, 而且还被连接到特定硬件或软件条件。这称为“密封”密钥。首次创建密封密钥时, TPM 将记录配置值和文件哈希的快照。仅在这些当前系统值与快照中的值相匹配时才“解封”或释放密封密钥。BitLocker 使用密封密钥检测对 Windows 操作系统完整性的攻击。

使用 TPM, 密钥对的隐私部分在操作系统控制的内存之外单独保存。因为 TPM 使用自身的内部固件和逻辑电路来处理指令, 所以它不依赖于操作系统, 也不会受外部软件漏洞的影响。

BitLocker 主要有两种工作模式: TPM 模式和 U 盘模式, 为了实现更高层次的安全, 我们还可以同时启用这两种模式。

- TPM 模式, 要求计算机中必须具备不低于 1.2 版 TPM 芯片, 这种芯片是通过硬件提供的, 一般只出现在对安全性要求较高的商用电脑或工作站上, 家用电脑或普通的商用电脑通常不会提供。
- U 盘模式, 则需要电脑上有 USB 接口, 计算机的 BIOS 支持在开机的时候访问 USB 设备 (能够流畅运行 Windows Vista 的计算机基本上都应该具备这样的功能), 并且需要有一个专用的 U 盘 (U 盘只是用于保存密钥文件, 容量不用太大, 但是质量一定要好)。使用 U 盘模式后, 用于解密系统盘的密钥文件会被保存在 U 盘上, 每次重新启动系统的时候必须在开机之前将 U 盘连接到计算机上。

选择一个受信任的平台模块是实现 TPM 模式 BitLocker 的前提条件。在安装 SP1 之后的 Vista 企业版和旗舰版中, 我们可以使用三种模式的 BitLocker:

- 纯 TPM 模式, 要求系统中具有 TPM 芯片, 这样用于解密的密钥以及用于验证引导文件完整性的相关文件都会保存在 TPM 芯片中。
- 纯 U 盘模式, 要求系统符合上文中提到的和 USB 设备有关的条件, 这样用于解密的密钥会被保存在 U 盘中。
- 混合模式, 可以使用 TPM+U 盘、TPM+PIN, 以及 TPM+U 盘+PIN 的形式进一步增强系统安全。

当然, 这个技术看上去好些很完美, 其实还是有一些弊端。这个技术发挥最大效率的前提是主板上带有 TPM 芯片, 否则你也会遇到密钥文件被盗, 非法拷贝后使用欺骗的方法, 解密数据。这样也会造成用户的数据损失。

6.4.3 PGP

PGP(Pretty Good Privacy)^a是目前最优秀, 最安全的加密方式之一。PGP 也是全世界最流行的文件夹加密软件。它的源代码是公开的, 经受住了成千上万顶尖黑客的破解挑战, 事实证明 PGP 是目前世界上最安全的加密软件。

PGP 有丰富的产品线, 服务器和网关级产品有 PGP Universal Server(通用服务器)、PGP Universal Gateway Email(通用网关邮件)。个人桌面级产品有 PGP Desktop Email(桌面电子邮件版)、PGP NetShare(网络共享版)、PGP Whole Disk Encryption(完整磁盘加密版)、PGP Desktop Professional(桌面专业版)、PGP Desktop Storage(桌面存储版)、PGP Desktop Corporate(桌面企业版)、PGP Mobile (手机版)。这些软件应用的操作系统平台很广, 几乎任何一个平台都可以找到, 用户可以按照自己的需求使用了。前面的章节已经有太多的介绍, 在这里就不读说了。

6.4.4 TrueCrypt

TrueCrypt^b是一款免费、开源软件, 支持 Windows Vista/XP/2000 和 Linux, 是一个虚拟加密磁盘工具, 可以在硬盘上创建一个或多个虚拟磁盘, 所有虚拟磁盘上的文件都被自动加密, 需要通过密码来进行访问。同样它还

^a PGP 官方网站: <http://www.pgp.com/>, PGP 中文版: <http://www.pgp.com.cn/>

^b TrueCrypt 官方网站: <http://www.truecrypt.org/>

支持加密单个分区或整个硬盘。加密 windows 系统所在的分区时, 启动 windows 前需要密码, 这个和 PGP 的技术差不多。并且提供两级方案, 以应对被强迫说出密码的情况。可以隐藏分区 (使用覆盖式密码术, steganography)、隐藏操作系统。也无法探测到 TrueCrypt 加密分区, 这些加密数据会被其它软件认为是随机数据。

它的官方网站提供很多信息, 包括软件的在线说明^a、算法原理等。真的时非常详细, 最主要的是它是一个开源软件。并且支持中文语言界面。

6.4.5 Utimaco SafeGuard

Utimaco SafeGuard^b提供全硬盘加密防护措施, 以个人用户的版本 SafeGuardEasy 来说, 和 PGP 一样有开机前认证(PBA)技术, 让木马程序也无法取得开机密码与加密密钥, 可搭配使用智能卡或是 USB Token 来加强身份认证管理。

SafeGuard Easy 完全不改变使用者使用电脑的习惯, 还支持服务器进行远程安装, 提供管理员从远程协助密码重设、安装删除加密软件的紧急还原, 即使是跨国企业也可以轻易的大量部署。SafeGuard Easy 通过 Common Criteria EAL3, FIPS 140_2 国际安全认证, 也连续多年被 SC Magazine 评选为五星的最佳产品。

SafeGuard Easy 支持很多分区, 包括的加密分区有 FAT-12、FAT-16、FAT-32、HPFS、NTFS、NTFS5。支持的算法是我见过的最多的, 包括^cAES-256 32 bytes (256 bits)、AES-128 16 bytes (128 bits)、Rijndael-256 32 bytes (256 bits)、DES 7 bytes (56 bits)、3DES 21 bytes (168 bits)、IDEA 16 bytes (128 bits)、Blowfish-8 32 bytes (256 bits)、Blowfish-16 32 bytes (256 bits)、STEALTH-40 5 bytes (40 bits)、XOR 8 bytes (64 bits)。

SafeGuard Easy 也是全球唯一通过 IBM/Lenovo 认可, 指定使用在 ThinkPad/ ThinkCentre 的资料加密产品, 可见它的安全性确实很不错。它的安装 CD 是可启动光盘, 包含了安装程序、PDF 说明文件、小工具, PDF 说明文件很详细, 基本都可以看懂。初学者如果英文过关, 照着做的话, 这个设置一定不难。

6.4.6 The GNU Privacy Guard

GnuPG^d(The GNU Privacy Guard)是 GNU 计划的完成项目, 基于 RFC4880 中的 OpenPGP 标准。GnuPG 可以加密和签名数据, 拥有多样的密钥管理系统和各种公钥目录的连接模型。GnuPG 也像 GPG 一样出名, 它是一个整合了其它应用程序的命令行工具。它提供了一系列的前端应用程序和库文件。GnuPG 2 中提供了 S/MIME 的支持。

GnuPG 是一个免费软件, 意味着你可以免费使用、修改, 可以在 GNU 公共授权证书分发软件, 这点和收费的 PGP 软件是不同的。GnuPG 的出名来自他的 2 个版本: 1.4.9 是最出名的便携独立版本, 2.0.11 是增强版, 而且也是一个比较复杂的一个编译版本。

Gpg4win 项目提供了 Windows 版本的 GnuPG。全都整合到了一个带有用户说明文件的安装包内。Aegypten 项目发开了 GnuPG 2 中的 S/MIME 功能。

GnuPG 自身是一个没有用户界面的命令行工具。它确实是一个真正命令行的密码引擎, 可以被其它程序的脚本代码运行。所以, 也可以认为它是一个其它程序的后台程序。然而, 除了使用命令行工具提供所有的功能外, 也包含了交互式选项系统。这些命令行工具通常是被其它的前端程序使用的。当然, 你使用的时候可能会发现有户界面程序, 那只是外壳程序。调用了命令而已。

在功能上, 它可能完全代替 PGP。它比 PGP 更好的函数扩展性, 可以解密和验证 PGP 5、6 和 7 的加密消

^a TrueCrypt 在线说明: <http://www.truecrypt.org/docs/>, 并且提供包括源代码在内的下载: <http://www.truecrypt.org/downloads>

^b Utimaco 官方网站: <http://www.utimaco.com/>, 对于企业用户有 SAFEGUARD ENTERPRISE, 对于个人用户有 SafeGuardEasy。

^c 后面的按照 算法名 密钥大小 (密钥大小用 bit 单位表示法) 的排列。

^d The GNU Privacy Guard 官方网站: <http://www.gnupg.org/index.en.html>

息.支持的算法有 ElGamal、 DSA、 RSA、 AES、 3DES、 Blowfish、 Twofish、 CAST5、 MD5、 SHA-1、 RIPE-MD-160 和 TIGER。同样支持 HKP 密钥服务器系统 (www.keys.pgp.net)。最关键的它不使用任何专利算法,意味着没有任何的法律条款制约你。

这些平台中,价格比较高的是 PGP,所以你如果不怕麻烦的话可以考虑 GnuPG,官方的技术资料也是很全的。有时候我们可以多个技术一起使用,比如 EFS+PGP 的做法。当然解密就要复杂一点了,这也起到了很好的保密效果。你可以现象一下,当一个数据小偷好不容易使用证书欺骗拿到你的文件后却发现,数据需要 PGP 密钥解密,他一定会疯掉。当然你也可以像鸡蛋一样把数据一层一层的包裹住。它的坚固性取决于外壳的硬度。其实也没有那种必要,一般的加密技术已经很成熟了。没有什么理由去做加壳。

到这里,软件介绍就完了,还有其它什么好软件,你也可以告诉我,如果不错,下一版的书中会有新软件的介绍。

7 关于 PGP 的开创者

7.1 背景

Philip R. Zimmermann^a是 Pretty Good Privacy, 一个邮件加密软件的开创者. PGP 于 1991 年在 Internet 上免费的发布, 它的最初设计目标是一个保护人权的工具. 由于 PGP 在世界范围的传播违反了美国政府关于加密软件的出口限制, Zimmermann 受到了为期三年刑事调查. 尽管缺少资金, 缺少任何付酬员工, 缺少在后面支持的一个公司, 还有着政府的烦扰, PGP 仍然成为了世界上使用最为广泛的邮件加密软件. 1996 年初, 在政府撤手这个案子^b之后, Zimmermann 创立了 PGP 公司. 1997 年 12 月, 公司被 Network Associates Inc(NAI)收购, 在那里他做了三年的高级职员. 2002 年 8 月, PGP 被一家名叫 PGP 有限公司的新公司从 NAI 购得, Zimmermann 在那里担任特殊顾问和咨询. Zimmermann 目前就密码学方面为一些公司和工业组织提供 咨询, 同时还是斯坦福法学院 Internet 与社会研究中心(Stanford Law School's Center for Internet and Society)的一员.

在创办 PGP 公司之前, Zimmermann 已经是一个有 20 多年经验的软件工程师, 特别是在密码学和数据加密, 数据通信, 和实时的嵌入式系统方面. 他对密码学在政治方面应用的兴趣由他在 军方政策方面的工作背景而来.

Zimmermann 因为在密码学方面的先驱性贡献而获得了多项技术和人道主义者的奖项. 2003 年, 他被列入 Heinz Nixdorf MuseumsForum Wall of Fame, 2001 年被列入 CRN Industry Hall of Fame. 在 2000 年时, 信息世界(InfoWorld)提名他为电子商务中的 十佳创新者. 1991 年他荣获国际隐私保护组织(Privacy International)的 Louis Brandeis 奖, 1998 年获可靠计算杂志(Secure Computing Magazine)终身成就奖, 并在 1996 年因有促进负责应用技术的社会责任感中计算机行业的 Norbert Wiener 奖. 1995 年因 设计创新获 Chrysler 奖, Electronic Frontier Foundation 的先锋奖, 1996 PC 周刊 IT 杰出奖, 以及 1996 因为"最安全产品"而获最佳连接网络计算奖. 1994 年, PGP 被信息周刊选为十大最重要产品. 1995 年, Newsweek 提名 Zimmermann 为"网络 50 杰", 50 位 Internet 上最有影响力的人.

除了 Zimmermann 开创公司前开发的版本赢得了如此多的奖项之外, 公司的技术小组改进的后来版本继续每年赢得 众多的工业界奖项.

Zimmermann 于 1978 年在佛罗里达大西洋综合大学获计算机学士学位. 目前, 他是国际密码学研究会, 机器计算协会以及程序自由联盟的成员. 他供职于 科学通信与国家交流小组(Roundtable on Scientific Communication and National Security), 一个国家科学院与战略和国际学习中心的协作计划. 他还是 OpenPGP 联盟的主席, 担任 为社会负责的计算机工作者组织(Computer Professionals for Social Responsibility)的董事, 并在 圣克拉克大学计算机工程系, Anonymizer.com, Hush Communications, Encenuate, and Qualys 咨询部任职.

7.2 PGP 的起源

在开发 PGP 前几年, Zimmermann 从事一些与地域有关的政治活动. 上世纪 80 年代在科罗拉多的 Boulder, Zimmermann 以一个全职军事政策分析员的身份在核武器冻结计划中工作, 同时还保持着他在白天的软件工程师的工作.

那时的世界局势与现在不同. 里根还在白宫, 勃列日涅夫在克里姆林宫, FEMA(联邦应急管理局)告诉市民做好撤退的准备, 无数的人们恐惧着世界将陷入残酷的核战争. 百万的民众走到中央公园为和平而游行.

在这样的政治环境下, 1984 年, Zimmermann 看到了开发后来成为 PGP 的软件的需要, 不仅为了保护海外

^a 个人站点 <http://www.philzimmermann.com/ZH/background/index.html>

^b http://www.philzimmermann.com/EN/news/PRZ_case_dropped.html

的人权, 也为了保护国内政治组织的根基. 那时起 Zimmermann 就开始了 PGP 的早期设计, 但由于参与和平运动的事务繁忙, 大部分的开发工作一直推到几年后.

Zimmermann 在 科学家联合会(Union of Concerned Scientists)的演讲人办公室工作, 公开与政府官员辩论, 在几轮的总统大选和参议院换届中担任防御政策顾问, 训练一些说客, 并帮着组织环绕 Rocky Flats 核武器实验田的运动. 并且组织了核武器冻结运动政治活动委员会的冻结投票的数据库软件的开发. Zimmermann 因为非暴力不合作在内华达州核武器试验场被捕. 那时最值得回忆的事情之一就是在监狱中见到了 Carl Sagan, Martin Sheen, 和 Daniel Ellsberg.

整个 80 年代中期, Zimmermann 教授一门名为在军备竞赛中聪明起来(Get Smart on the Arms Race)的课程. 课程覆盖了军备竞赛的历史, 北大西洋公约学说, 有限限制还是扩大限制, 相互确定性毁灭(MAD), 限制军备谈判, 条约认证, 核扩张, 主动防御策略, 美苏力量结构, 命令与控制系统, 预警, 摧毁发射井的反导弹策略, 超远程发射的对比等内容.

后记

终于看到这页了, 现在你也对密码学有了一个简单的形状, 密码学确实的是一门科学。普通人不需要去了解密码技术的核心内容是什么。我们只要知道这个技术暂时是无法破解的就可以了。当然, 理论上任何密码就可以破解的。我们不需要“终极加密工具”, 要知道我们现在的技术可以让破译时间超过我们的生命周期就已经足够了。如果我们非要这样做的话, 也许未来的历史学家就无法解密我们的字符(有必要让历史学家也学习密码学)。当然, 这些是后话。

这些深层次的密码技术也需要我们去研究。不仅仅是破译和否定别人的算法。有时候我们自己也要考虑使用自己的算法, 也许, 我是说也许, 别人的技术也许更高明, 他自己就可以完全破解自己的算法, 却仍然还有一群糊涂蛋在高兴的使用这些算法。你敢现象一个拿着超级密钥的人, 站在一大堆密文中间时他的表情吗?

这里要特别批评中国的密码软件, 我联系了很多拥有密码技术的软件厂商, 居然没有一个对我的疑问有回复, 在官方网站上也没有任何的技术白皮书, 最多的就是写有“在国家密码局备案”的文字, 这让我很失望。而且我的很多网友也遇到这样的客服“闭门羹”。原来中国人更喜欢关起门来做研究, 中国的软件厂商对于一般用户的“低级问题”不会去答复。这样对中国的密码技术发展是很不利的, 我希望国内有像 RSA 一样的竞争过程, 算法放在互联网上由全世界的人来检测, 毕竟我们也不笨。也希望中国的软件业走上国际化的道路。所以你在文章中会看到有很多的英文软件介绍和推荐, 这是在国外一个竞争环境下的产物, 并不是我崇洋媚外。如果有不错的中国算法出现, 我也会介绍。如果中国的密码技术软件厂商看到这段文字, 希望他们把密码技术公开化, 这样大家不会遇到如同国内 DVD 高清标准狗咬狗的恶性局面。这点在很多密码学专家看来也会同意我的看法。难道王小云教授破解 MD5 算法是为了帮助外国人提高技术? 技术是全球化的, 中庸的学术氛围不会有多大的技术提高。

密码学中对于别人的密码算法都会有怀疑的态度, 因为这个是事关重大的技术。也不要坚信自己的私有算法是独一无二的, 密码学发展到今天, 简单的密码算法是很容易破解的。你可以去参阅大量的其它密码学资料, 包括本书的脚注里面的信息, 都对你很有帮助。

这本书算是最终版了, 以后不再更新了, 很多东西已经完全涵盖, 你只需要在搜索引擎中搜索一下就可以得到答案。也希望你对于这本书提出宝贵的意见, 我会虚心的接受。本书在《中华人民共和国著作权法》的保护之下。当然, 并不意味着你要掏钱才可以看, 阅读本书籍是免费的。也就是说, 你可以在互联网的任意位置下载这本书(仅 PDF 格式), 甚至你可以任意拷贝、打印这本书。如果发现带任何广告信息或者特殊水印, 请立刻联系我。我没有授权任何商业用途的目的和行为, 任何这样的行为都构成侵权。如果你有意, 可以和我联系(估计在中国也没有人会和我联系了)。未经授权的任何修改本文(即 PDF 文件)的行为都是违法的!

杨新

2010.1.27