IT先锋系列丛书

互联网公钥基础设施概论

INTRODUCTION TO THE PUBLIC KEY INFRASTRUCTURE FOR THE INTERNET

[美] Messaoud Benantar 著
张千里 等 译

PH PTR

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

IT 先锋系列丛书

互联网公钥基础设施概论

[美] Messaoud Benantar 著
张千里 等 译

人民邮电出版社

图书在版编目（CIP）数据

互联网公钥基础设施概论 /（美）贝南塔（Benantar，M.）著；张千里等译. —北京：人民邮电出版社，2003.3
（IT 先锋系列丛书）
ISBN 7-115-11059-X

Ⅰ. 互... Ⅱ. ①贝... ②张... Ⅲ. 因特网—安全技术 Ⅳ. TP393. 48

中国版本图书馆 CIP 数据核字（2003）第 005279 号

IT 先锋系列丛书

互联网公钥基础设施概论

◆ 著 ［美］Messaoud Benantar
译 张千里 等
责任编辑 陈万寿

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www. ptpress.com.cn
读者热线 010-67129258
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：800×1000 1/16
印张：12.5
字数：267 千字 2003 年 3 月第 1 版
印数：1-4 000 册 2003 年 3 月北京第 1 次印刷
著作权合同登记 图字：01-2001-1378 号
ISBN 7-115-11059-X/TN・2028

定价：22.00 元

版 权 声 明

 ISBN 0-13-060927-7

内 容 提 要

本书围绕互联网络公钥基础设施(PKI)的建立进行了广泛的讨论，介绍了密钥密码学和公钥密码学的发展历史和相关知识，并围绕着密钥分发这一问题，探讨了 PKI 引入的必要性。全书重点集中在解答有关 PKI 部署、运行和管理中的一些最重要的问题，这些问题包括：密钥密码学和公钥密码学的基本原理；密钥分发问题以及公钥保障系统的重要地位；用 PKIX 来建设安全互联网系统；了解 PKIX 标记语言、数据编码机制以及拓扑；如何实现有效的 PKI 信任模型；利用 LDAP 作为 PKIX 在互联网中的存储库；证书鉴定、凭证管理，以及密钥更新等。

本书内容深入浅出，比较适合那些希望总体了解公钥基础设施与网络安全的电子商务、电子政务管理人员和技术人员阅读，对于那些已经对公钥基础设施有所了解的信息安全工程技术人员和开发人员来说，本书也不失为一本优秀的综合手册。

译 者 序

自从公钥密码技术产生以来，各种基于公钥的应用层出不穷。随着互联网络的普及，基于公钥的各种服务也迅速流行起来。公钥技术的广泛使用，也就产生了公钥的可信分发问题。在小范围内，公钥的发布还可以以相对简单的方式来进行，但是随着网络规模的扩大，尤其是扩大到像互联网络这样庞大的规模时，简单的方式就无法解决该问题。这样就导致了公钥基础设施的出现。公钥基础设施的核心思想在于通过证书和证书注销列表的介入，使得只要网络中有一个可信点存在，就可以保证整个基础设施的安全，这一方法使得公钥的可靠发布与应用的规模无关。正是因为这一点，在 X.509 证书被采纳为互联网络标准后，互联网络的公钥基础设施就开始飞速发展，并出现了大量的基于它的应用。

本书就是这样一本试图全面介绍互联网络公钥基础设施方面的书籍，它系统地阐述了密码学的历史变迁。正如书中所述，密钥密码学的密钥分发问题导致了公钥密码学的出现，而公钥密码学中的公钥可靠发布的问题，导致了公钥基础设施的出现。在本书后面的章节中，针对互联网络公钥基础设施及其相关应用，进行了更加详细的介绍。

应当说，本书深入浅出地介绍了与公钥基础设施相关的大量问题，并给出了进一步了解时需要阅读的文献，因此比较适合那些希望了解公钥基础设施的读者使用，对于那些已经对公钥基础设施有所了解的人来说，本书也不失为一本优秀的综合手册。

本书主要由张千里翻译，程小梅女士整理了全部书稿并修改了其中的一些翻译错误。在本书的翻译过程中，我们尽量保持了原著的特色，对书中的格式也没有作大的修改。对于一些明显的错误，我们逐一进行了修改，并作了说明。若译文有错误或不妥之处，恳请各方面读者指正。

译者

前　言

现代密钥密码学的基础在于密钥的安全性。这一性质并非刻意所求，而只是客观条件的限制。考虑特定的密码算法是如何进行保密的，首先，可以对算法进行保密：如果不能对算法进行技术分析，那么算法就可以隐藏它的弱点，这样就奉行了那个不受好评的原则——不公开即安全。可是，没有办法能够把一个密码算法的弱点或优点永远隐藏下去，迟早一些人会通过逆向工程得到软件或硬件密码模块中的处理流程。这一结果可能就会宣告这一算法的末日来临。

密钥算法中，密钥需要发布给通信参与者，可是，密钥发布的次数越多，安全性能就越有可能会被损害。长期密钥的发布，违背了密钥密码学（也称作对称密钥密码学）的核心假定。密钥的传输需要建立安全通道，虽然可以由人来亲自传输，但是这不能满足大规模分布计算的需要。而在线发布需要高度安全的秘密通道，这样，就产生了如何启动密钥发布这一难题。

为了缓解密钥分发问题，就自然产生了密钥发布中心（KDC）的概念。这一实体为所有其他实体所信赖，它有两个作用，一方面用来保存长期密钥，另一方面用来发布两个实体通信时所需的短期会话密钥。后者常常还伴随有介绍一个实体给另一个实体的功能，这一般通过使用长期密钥在每个相应的实体和第三方建立的可信通道来实现。尽管这一方法已经成为了最优秀的第三方密钥发布方案，但它缺少在互联网络这一普遍存在的计算模式下应用所需要的的灵活性。

回到我们的主题，来讨论公钥密码学的概念，公钥密码学这一概念的历史要比 KDC 的历史长得多。公钥密码学的基本然而影响深远的概念是，密钥成了相关的一对：公钥和私钥。私钥由所有者安全保护，而公钥则可以自由散播。它的基本假设在于，知道公钥之后计算私钥，在计算上是不可能实现的。用公钥加密的数据只能够用私钥解开。有了这样一个吸引人的性质，公钥密码学看起来最终解决了安全密钥分发问题。有赖于一些密钥交换机制，如 Diffie-Hellman，它确实做到了这一点。而且，公钥密码学不仅能够用于密钥交换协议，它还可以提供各种安全服务，如数字证书、抗抵赖服务以及利用那些著名的公钥算法（如 RSA）进行数据加密。

自由发布公钥的前提，就是信任的建立。基于公钥密码学的安全服务，也要依赖一种信任：即某个特定公钥确实属于它的合法所有者。为了建立信任机制，一个很有前途的方法是利用 X.509 所提供的数字证书，它已被采纳为互联网络标准。本书将试图全面介绍互联网络公钥认证方面的各个主要方面。

作 者 简 介

Messaoud Benantar，博士，高级软件工程师，工作单位是位于美国德克萨斯州奥斯汀的IBM 公司。毕业于美国纽约州 Troy 的轮塞拉尔（Rensselaer）工学院计算机系，并获得博士学位。有超过 10 年的在各个平台上开发安全软件的经验，持有多项关于分布式系统安全的美国专利。研究兴趣包括：系统和网络安全以及所有与互联网络计算相关的课题。Benantar 博士的电子邮件地址是：mbenantar@alum.rpi.edu。

目　录

第 1 章　密钥密码学 …… 1
1.1　概述 …… 1
1.2　背景知识介绍 …… 1
1.3　XOR 基础知识 …… 3
1.4　密钥空间 …… 5
1.5　常见密钥算法 …… 5
1.6　密钥加密法的安全服务 …… 7
1.7　密钥密码学及抗抵赖性 …… 7
1.8　源真实性 …… 7
1.9　数据完整性 …… 8

第 2 章　密钥的发布和管理 …… 11
2.1　概述 …… 11
2.2　共享密钥：拓扑的影响 …… 11
2.3　集中的密钥管理 …… 14
2.4　Needham-Schroeder 方案 …… 15
2.5　有关密钥发布的一点提示 …… 16

第 3 章　公钥密码学 …… 17
3.1　公钥密码学的基础 …… 17
3.2　密钥密码学的归宿 …… 22
3.3　公钥加密服务 …… 22
3.4　公钥的信赖 …… 28

第 4 章　公钥设施——PKIX …… 31
4.1　概述 …… 31
4.2　背景知识 …… 31
4.3　PKIX 证书和证书注销列表（CRL） …… 32
4.4　PKIX 元素 …… 32
4.5　ASN.1：PKIX 定义语言 …… 32

4.6 PKIX 信息模型 …… 50

第 5 章 X.509 证书和 CRL 扩展 …… 67
5.1 概述 …… 67
5.2 X.509 v3 证书扩展 …… 68
5.3 有关 X.509 证书扩展 …… 82
5.4 X.509 v2 CRL 扩展 …… 83
5.5 原因代码（Reason Code） …… 86
5.6 失效期（Invalidity Date） …… 87
5.7 证书签发者（Certificate Issuer） …… 87
5.8 暂停使用时的指示代码（Hold Instruction Code） …… 88

第 6 章 PKIX 中的信任建立过程 …… 89
6.1 概述 …… 89
6.2 层次化的信任关系 …… 89
6.3 交叉认证（Cross Certification） …… 92
6.4 混合模式 …… 95
6.5 Web 信任模式 …… 95
6.6 证书鉴定 …… 96
6.7 鉴定的输入 …… 97
6.8 鉴定程序 …… 98

第 7 章 PKIX 拓扑和操作协议 …… 108
7.1 概述 …… 108
7.2 基础设施拓扑 …… 108
7.3 PKI 管理操作概述 …… 115
7.4 证书管理协议（Certificate Management Protocol，CMP） …… 119

第 8 章 PKI 的证书和 CRL 库 …… 140
8.1 概述 …… 140
8.2 FTP …… 141
8.3 HTTP …… 142
8.4 电子邮件 …… 143
8.5 DNS …… 143
8.6 LDAP …… 145

第 9 章 PKI 凭证管理 …… 153
9.1 概述 …… 153
9.2 PKCS #8 …… 153
9.3 PKCS #12 …… 154
9.4 PKCS #11 …… 159
9.5 PKCS #15 …… 162

第 10 章 基于 PKI 的安全应用 …… 164
10.1 概述 …… 164
10.2 PKCS #7 …… 164
10.3 内容参数化 …… 165
10.4 PKCS #7 安全服务 …… 172
10.5 CMS …… 172
10.6 CMC …… 177
10.7 CMS 报文的进一步保护 …… 179
10.8 S/MIME v3 …… 180
10.9 SSL/TLS …… 180

参考文献 …… 184

第 1 章　密钥密码学

本章简要介绍了密钥密码学的基本原理。我们先讨论了早期密码系统中的数据编码技术，然后详细阐述了现代密码系统中所使用的基本概念。在描述了目前一些著名的算法后，我们将讨论使用密钥密码学的一些安全服务。

1.1　概　　述

保密性是数据的一个安全属性，它的主要目的是限制数据所含有的信息知识只能在某个能理解这些数据的群体（如一些人，或者是一些可编程的电子系统）之间共享。达到这一限制的过程，有时也被称为保密，就是将数据明文进行编码的过程，编码后的数据是可逆转的、但很可能没有语法，而且通常没有语义。

早在电子系统出现前很久，就已经出现了多种编码转换方法，目前这些方法通常被称为密码学。数据的加解密转换是一个确定性的过程，在这一过程中，明文形式的数据被隐藏为不暴露原始数据的密文。同样，密文可以被特定的接收者用一种确定性的过程逆转换，从而恢复为原始数据。

1.2　背景知识介绍

早期的密码学算法[KAHN67、BECK82]，通常是逐字处理明文输入，然后使用字母替换或换位等方法来进行加密。替换操作，是把输入流中的一个字母替换成字母表中的另一个；而换位操作，则是将输入字符流中的一些字母顺序交换。一个著名的替换操作的例子是凯撒密码，这一密码据说被用于凯撒和他的军队进行联络。这一密码中，每个字符都被扩展字母表中的后三个字母所代替，扩展字母表就是加上空格的字母表。形式上，它的加密算法就是将每个字母的值加 3，然后再对 26 取余，这样就得到了结果。我们给 A～Z 的 26 个字母分别赋值为 0 到 25，然后将每个字母 P 变为：$f(P) = P+3 \bmod 26$，图 1.1 显示了该算法的一个例子："RETURN TO BASE" 被转换为 "UHWXUQ WR EDVH"。

换位密码通常先把明文分割成不同块，然后使用某种确定程序将不同块间的字符混杂。如图 1.2，需要加密的明文 "RETURN TO BASE" 被分成两块，一块是 "RETURN"，一块是 "TO BASE"，然后将两块字符采用循环的方式进行混杂，这样就形成了密文 "ROTBRS TE UANE"。

再举一个简单换位的例子，把明文写成一个具有固定行列的二维矩阵，然后简单地对矩阵进行换位即可。如图 1.3 所示，明文 "RETURN TO BASE" 被插入一个 2×9 的矩阵中，这样即

形成一列密文。解密这一密文的方法就是其逆操作，只要将密文写入一个9×2的矩阵就可以了。

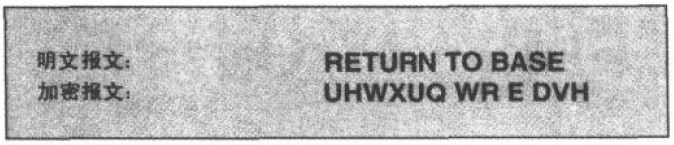
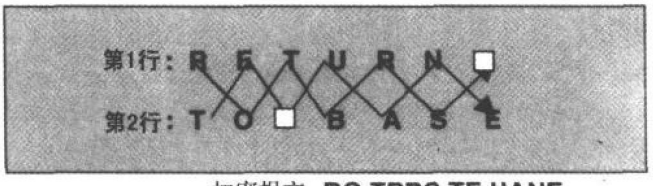


图 1.1　简单的替换密码



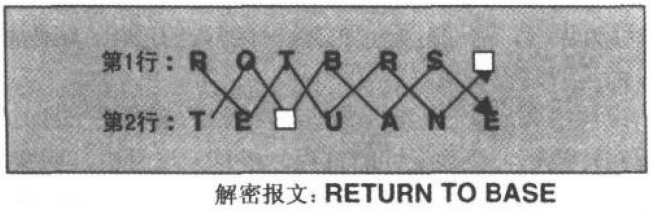
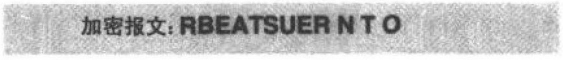
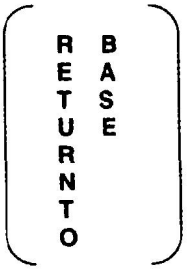


图 1.2　使用换位进行加密解密的例子





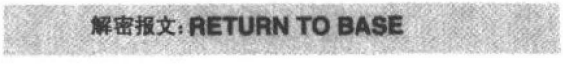


图 1.3　行列变换加密

通常，换位加密容易被破解，不过如果将进行完一次换位的结果，再经过另外一个换位，就会大大增强加密强度。

尽管算法很简单，但是替换方法是密钥密码学的一个基本例子（需要加的数就是其密钥）。对密钥保密，将算法公开，就可以对密文进行穷举破解了，在本例中穷举破解的集合只是由 1 到 26 的数字。因此，在它使用的时代主要通过依赖算法本身的保密性来保证不会被破解。在我们所举的简单例子中，在已知算法后，几乎可以立即破解密文。

随着电子计算机的出现，现代早期加密算法[KONH81、BECK82、DENN83]使用了类似的方法，即利用换位或者替换。主要的不同是，在这些加密算法中，这些转换通常发生于数据为二进制形式的比特层。一个比较常见的例子就是使用 XOR 操作。

1.3　XOR 基础知识

XOR 操作一般用“+”号来表示，是一个比特层的操作，它将{0, 1} X {0, 1}映射到{0, 1}集合，映射方法如下：

0+0=0
0+1=1
1+0=1
1+1=0

如果我们将第二个操作数作为密钥值，XOR 操作就可以看为根据密钥值进行比特层的替换操作。在这一假想下，如果密钥输入为 0，则 XOR 操作返回其本身（0 返回 0 而 1 返回 1），如果密钥输入为 1，则 XOR 操作则进行逆转（1 返回 0 而 0 返回 1）。XOR 具有如下性质：

a+0=a
a+a=0，因此
a+b+b=a

后一性质显示出使用某一定密钥值，XOR 操作就可以用来加密一段明文，使用相同的密钥值进行 XOR 操作，就可以将这一密文解密。这一属性产生了很多弱加密算法变种，这些算法都采用 XOR 操作，因此也容易被破解。

假设一个固定长度的密钥 K 用于对一明文块进行 XOR 操作加密。在知道了一个明文块 P 之后，通过将明文和相应的密文进行 XOR 操作的办法，可以直接得到 K。

C=P+K

$$P+C=P+P+K$$
$$=K$$

类似的，知道了两个明文块的相应的密文后，可以通过 XOR 操作，得到这两段明文块的 XOR 值：

$$C1+C2=P1+K+P2+K$$
$$=P1+P2$$

检查 P1+P2 的比特模式可以很容易地恢复其中的一段明文。明文和相应的密文做 XOR 操作之后就可以得到密钥值。

每次只转换一个比特的特点使得 XOR 操作可以被划到流加密（stream cipher）的算法类中。块加密（block cipher）是另外一类，该算法中把明文分割成具有相同长度的若干块，通常每块长度等于或者大于 64 比特，然后每次对每块进行相同的加密变换。流加密算法适用于内存缓冲区有限或者是字符单独传输的情况下，比如说在某传输媒介的终点上。由于它们各比特单独进行变换，因此当发生传输异常时，不会产生更大影响。

一次性 XOR 密码本

尽管 XOR 操作简单，在固定密钥长度时强度不够，有一种办法，可以只使用这一种操作就建立完美的加密机制。如果密钥流数字随机产生，且只使用一次，那么所产生的密文就要强壮得多。这样加密方法可被证明对于只具有一套密文的解密者来说是安全的。图 1.4 就是简单的基于一次性 XOR 密码本加密方法的图形表述。

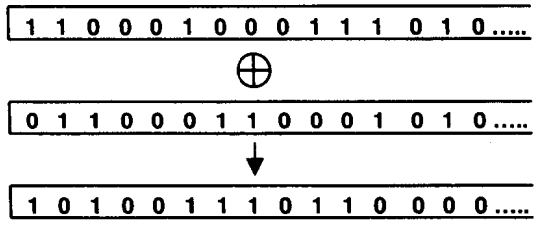


图 1.4　基于 XOR 操作的一个简单的一次性密码本

一次性 XOR 密码本的安全性来源于，试图猜出密钥流的难度和直接猜测明文的难度相同。注意一次性 XOR 模板的密钥流长度和要加密的明文的长度相同。这样的属性使得散发和维护这么长的密钥流的工作变得十分困难，因此也就必须需要一种流加密算法，通过这样的算法，密钥流从一个可管理的密钥中以伪随机数的方式产生。

比特层的一次性密码本在字符层也工作得很好。其实在历史上，当 Joseph Mauborgne 和 Gilbert Vernam 最早发明这一方法时，本来是用在字符上的。每个模板中的字符操作于一个明

文中的字符，接收方使用相同的密码本进行解密，然后就将密钥毁去。

1.4　密 钥 空 间

密钥，也称作对称密钥，加密函数 E 通过用于加密和解密相同的密钥 K，将明文 P 转换为密文 C。类似地，加密转换可以通过解密函数 D 逆转换成明文。如图 1.5 所示。

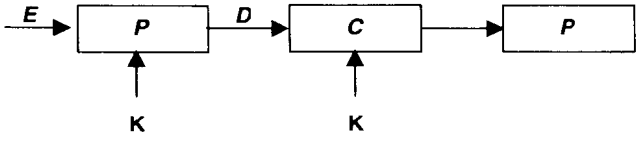


图 1.5　对称密钥加密和解密

现代密钥加密方法的强度一般只依赖于加密密钥的安全性，而不依赖于对加密算法的保密。因此，可以使用密钥空间中的穷举法对这样的密码系统进行破解。密钥空间就是对于某个加密算法中有效的所有可能的密钥值的集合。举例来说，对于英文字母的任意换位组合有 26!种，因此这种加密算法的密钥空间由 26！种组合构成。如果加上一些限制条件，如每个字母只是转化为字母表中其后面固定个数的字母，且每次只转化一个字母，这时密钥空间就要小得多了，只含有从 1 到 26 的整数。复杂一些的，如规定块长度为 3，也就是说将每个(p_1, p_2, p_3)块转换为(e_1, e_2, e_3)，其中每个位置都有自己的规则，这时密钥空间的大小就是（26!）^{3}。

大多数密钥密码系统中都使用固定长度随机产生的密钥。这些系统一般都会面临密钥空间的穷举性攻击。对于密码系统来说，要想安全的一个必要但不充分条件是密钥空间大到足以避免穷举性攻击。具有讽刺意义的是，加密算法快也有利于进行快速的穷举破解。

1.5　常见密钥算法

所有广为流传的密钥块算法都具有块加密算法的密码学特性。首先，密文中的每一比特都依赖于明文块中的所有比特。改变密钥中的任何比特，都会导致有 50%的可能性要改变所有密文比特。而且，在明文和密文间无法推断出统计联系。最常见的密钥算法的详细描述请见参考资料[SCHN96、MENE96]。

1. DES

现代密钥密码系统通过 DES 算法而受到广泛注意。DES 是一种对称密钥算法，该算法中，加密和解密使用相同的密钥。DES 算法产生于 20 世纪 70 年代早期的 IBM 公司，从 1976 年起，它成为了用于保护敏感但未列入密级的电子信息的政府标准。这一算法是一种块算法，它将 64 比特的输入块转化为相应的 64 比特的输出密文。它使用 56 比特长的密钥，但通常写

为 64 比特，其中每个字节具有一位奇偶校验位。图 1.6 是 DES 算法的高层抽象。

DES 标准中，数据经过 16 轮处理，每一轮中都根据密钥数，使用了包括换位、替换，以及像 XOR 的标准算术或逻辑操作等在内的各种操作。

最近由于计算能力的大幅度提高，DES 算法正在面临强力破解式的攻击，并显示出该算法易于受到穷举法攻击[WIEN94]。TripleDES 是一种利用两个或三个密钥对明文进行三次 DES 加密的算法。使用两个密钥时，triple-DES 先用第一个密钥进行加密，然后再用第二个密钥进行解密，然后再用第一个密钥加密，从而得到密文。

使用三个密钥的 triple-DES 对于这三步使用不同的密钥。在 triple-DES 中可能的密钥数目是 2^{112} 个，而在 DES 算法中密钥空间只有 2^{56} 个。

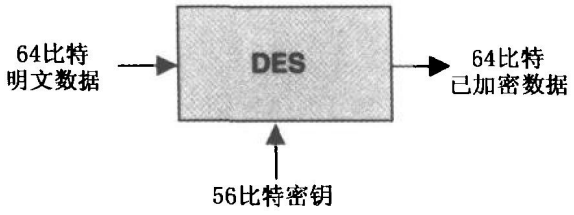


图 1.6　DES 算法的抽象表示

2. IDEA

尽管不如 DES 算法那样出名，IDEA 算法被一些当代密码学专家认为是最安全可靠的块加密算法[LAI91、ETH92]。和 DES 算法相似，IDEA 也是以 64 比特为单位加密，并得到其输出相应 64 比特密文块。它使用相同的算法进行加密或解密，只是在加密解密的期间密钥调度有所不同。与 DES 算法不同的是，IDEA 使用 128 比特的密钥，并主要使用三类操作：XOR、对 2^{16} 模加，以及对 $2^{16}+1$ 的模乘。这些操作组合起来进行相似的 8 轮计算组合，然后经过输出转换得到最后的密文。

3. AES

Advanced Encryption Standard(AES) [NIST01]，是美国政府准备用来替代 DES 的标准。最近被指定为 AES 的一个候选算法，Rijndael 算法，是一种迭代块密码算法，它的密钥长度和块长度都是可变的，可以是 128、192 或 256 比特。Rijndael 算法简单精巧的设计使得它在现代处理器上运转快速高效。而且它只在单或者是 4 字节的字（word）中使用简单的全字节操作，该算法的实现中需要的内存也相对比较少。这一算法适合于在各种处理器上实现，包括 8 比特低能耗、存储空间有限的像智能卡之类的硬件。该算法也适用于并行处理或者多计算逻辑设备（ALU）的处理器中。Rijndael 算法与传统的 Feistel 加密法不同。通常这种算法中其中间状态的某些部分比特是不变的。Rijndael 算法没有采用这一传统的结构，相反，每一轮变换由 3 个不同的不可逆转的变化组成，每一个变化都以统一类似的方式来对中间状态