可证安全密码学

Shafi Goldwasser[1]，Mihir Bellare[2]

Lanneverlose@yahoo.com.cn

[1] MIT Computer Science and Artificial Intelligence Laboratory, The Stata Center, Building 32, 32 Vassar Street, Cambridge, MA 02139, USA. E-mail: shafi@theory.lcs.mit.edu ; Web page: http://theory.lcs.mit.edu/ shafi

[2] Department of Computer Science and Engineering, Mail Code 0404, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-mail: mihir@cs.ucsd.edu ; Web page: http://www-cse.ucsd.edu/users/ mihir

前　言

这是 Shafi Goldwasser 和 Mihir Bellare 在 MIT 的 1996-2002, 2004, 2005 和 2008 夏季课程中一周的课程注记。密码学是一个很广的课题，这些注记的编辑线索是密码学中可证安全的概念建立和密码协议上的应用。

包括 Shafi Goldwasser 的 MIT 密码和密码分析学课程注记与 Mihir Bellare 在 UCSD 的密码和网络安全课程。另外，Rosario Gennaro (1996 年的课程助教)编辑了整本书的 9.6 节、11.4 节、11.5 节和附录 D，还编辑了附录 E 的一些内容。第二、三、七章中的主要材料是从 MIT 的 Goldwasser 教授的研究生在学习密码和密码分析学课程同时所记录的笔记中整理的。随后，1991 年课程助教 Frank D'Ippolito 进行了编辑。Frank 也对附录中的先进数字理论做了一些工作。第三章的部分内容节选自是 R.Rivest 的计算机科学理论手册。第四、五、六、八、九和十一及 10.5 和 7.4.6 节是出自 UCSD 的 Bellare 和 and Rogaway [23]教授的现代密码学介绍，并且感谢 Phillip Rogaway 允许加入这部分内容。Rosario Gennaro 对这个教案的贡献在 10.6，12.4，12.5，附录 D，并且编辑了多出文章和附录 E 的难题。

目录

第一章 现代密码学导论

密码学是有对手出现的通信。包括：密码、认证、密钥分发和其他的一些概念。现代密码学领域提供理论基础基于需要明确的问题是什么、怎样评估支持和解决问题的协议、怎样构建能够确保安全的协议，这里介绍确保加密难题的基本出发点。

1.1 加密：历史回顾

密码学最古老和最基本的问题是：在不安全的信道上进行安全通信。A 方需要对 B 方传送一个秘密信息，并且这个信息有可能被第三方窃听。传统的解决方法是：使用秘密密钥对消息进行加密。在远程通信前，A 和 B 使用秘密密钥进行协商得到加密和解密算法，另外附加一个信息 S 保证安全。指定 S 作为共享秘密密钥。对手可以知道加密和解密算法但是不知道秘密密钥 S。当 A 和 B 的初始会话结束后，明文消息 m 通过不安全信道传送。A 通过加密算法计算出密文：C=E（S,m），传送 C 给 B。收到之后，B 通过计算解密 C，m=D（S,C）。截获者（或对手）因为不知道 S，所以无法从 C 还原出明文。用代替体制来举例说明该过程。A 和 B 进行协商，使用同一个置换：$f:\sum\rightarrow\sum$ （$\sum$ 是需要替换的消息）。需要加密的消息 $m = m_1\ldots m_n$, $m_i \in \sum$, A 计算 E（f,m）=f（m_1）…f（m_n）,解密过程为：D（f,c）= f^{-1}（c_1）… f^{-1}（c_n）= $m_1\ldots m_n$ 。这个例子中的共享秘密密钥为：函数 f。加密算法和解密算法是特殊的，对手已知。在已知中等数量的密码标志后，可以知道对手非常容易对置换算法进行解密。

基于香侬 1943 年建立的信息论，一个完善保密的严格理论。在这个理论中，对手被赋予了无限的计算能力。香侬证明了适当定义的安全系统存在于秘密信息的大小 S，A 和 B 在远程通信前协商同意的与远程交换的密文大小一致的加密系统。

一个私钥加密的方式只要是才用一次一密的方式都被认为是安全的，甚至设计在攻击者的能力范围内。A 和 B 协商一个比特串：pad = $b_1\ldots b_n$，这里，$b_i \in_R \{0，1\}$（等 pad 从$\{0,1\}^n$ 串，A 计算 E(pad,m)=m $\oplus$ pad（面向比特异或）。解密密本 $C\in\{0，1\}^n$，B 计算 D(pad,m)=m $\oplus$pad = pad $\oplus$ （m$\oplus$pad）= m 。容易证明，$\forall$ m，c，P_{pad}[E（pad,m）=c] = $1/2^n$ 。可看出，C 没有对发送的加密信息进行泄露。（对手通过 C 预测明文 m 不比已知 C 预测 m 容易）。目前，预测 A 要发送附加信息 m’给 B。如果 A 只是简单发送 c=E（pad,m’）,那么 m 和 m’的长度回超过秘钥 K 的长度，根据香侬理论，该密码体制是不安全的。事实上，E(pad,m) $\oplus$ E（pad,m’）= m$\oplus$m’ 。包含了 m 和 m’的信息，而且可以立刻判断出两个信息哪些位相同，哪些位不同。基于这点，在不安全信道上的信息交换应该预先定义长度。

1.2 现代密码学：基于计算复杂度理论

现代密码学抛弃了计算资源无限的假设，采用了计算能力在合理范围内的前提。特别的，在这个注记中，假设攻击者已知在多项式时间内的随机算法；同样，加密和解密算法设计是随机的，并且在多项式时间内运算。

加密算法、解密算法和攻击者的算法可以用一个包含安全参数 k 的函数来度量，k 是在密码系统搭建时确定的。因此，攻击者的算法是在包括参数 k 的多项式时间内运算确定的。

因此，现代密码学中讨论从信息交换中破解密码体制和计算出信息的不可行性，而古典密码中使用的是不可能性。应该注意到，所谓的安全密码体制对于新的攻击者或有强计算能力的攻击方法是不安全的。但是，在远程信息传送前，A、B 方的会晤和协商密钥需用与明文信息相同密码长度确保安全。事实上，在安全通信之前，A、B 不必要提前知道需要使用多少比特密码确保信息安全。会逐步建立起密码体制，确保通信双方在有限的安全密钥共享前提下，在多项式时间内产生足够的密码。怎样建立密码体制促使引入另一个密码学基础理论，命名为密码体制、复杂性或假设前提。

现代密码学是基于加密算法效率和攻击者解密计算不可行性之间的差距的，要求具备密码学最简单基础部分，比如单向函数的基本知识。众所周知的，对单向函数的共识是计算简洁但是求逆困难。另外一些知识，例如随机数发生器和随机数簇，这一部分将稍后再进行定义。基于这些知识，可以容易建立起现代密码学体制。

因此，关键在于这些基础定义是如何得到的。尽管单向函数被普遍认为是存在的，对正在使用的一些单向函数也存在异议，但目前还不能完全用严格的理论证明其存在性。因此，建立了一个密码学假设前提来证明单向函数的存在性。使用单向函数存在的前提贯穿这部注记，需要确定提出的假设导出的结论是完善的。

本书将详细说明各种私钥密码的构置。上世纪七十年代开始的公钥密码要求 A 和 B 必须确立一个共同的私钥进行加密，消息接受方 B 可以向所有人，包括攻击者公开其鉴别码（称为公钥），消息发送方。任意有意向和 B 进行保密通信的人都无须与 B 预先协商就可以通过 B 的公钥与 B 进行通信。加密体制不再仅仅限制在授权范围，只要是拥有 B 的公开密钥都可以与 B 进行安全通信。接收者使用自己的私钥对信息进行解密，这种密码体制称为公钥密码体制。

证明使用陷门函数的安全公钥安全。一个陷门函数是一个单向函数，陷门函数通常定义为一些陷门信息——只有接收者知道，接受者可以对这个函数求逆。公钥密码系统和陷门函数的基础工作是 1976 年 DIFFIE 和 HELLMAN 进行的[71,72]。不久后，的 IDEA 所产生具体实践工作进行了[170，176，143]。

从单向陷门体制所构建的密码体制如下：接受者 B 随机选择一个陷门函数 f 和适当的陷门信息 t，建立 f 描述的公钥和私钥。如果 A 需要发送信息 m 给 B，A 计算 $E(f,m)=f(m)$. 解密 $c=f(m)$，B 计算 $f^1(c)=f^1(f(m))=m$。将证明这种构造不是足够安全的，但是在部分随机变量的前提条件下是安全的。

1.3 部分单向函数简介

如上所述，密码学的许多基础定义建立在单向函数之上，即计算容易而求逆困难（例如公钥体制，必须要有一个陷门）。容易的概念在于可在多项式时间内的函数运算，而计算困难的概念是指在多项式时间计算的小概率性（PPT），即在多项式内求逆函数是困难的。因此，对于所有使用单向函数的潜在人群，所有的函数输入求逆的难度应该是机会均等的。

有若干个满足上述条件的单向函数已经被提出了。

1、因式分解：假设函数 f: (x,y)→xy 是单向函数，最快逼近函数是 DIXON 随机平方算法；这是一个运算时间为 L（n）$\sqrt{2}$，这里，L（n）=$e\sqrt{\log n \log\log n}$ 。数域通过 Lenstra,Lenstra,Manasee,Pollard 算法过滤， 该算法通过 A dlemann，Pomerance 进行优化，在下述时间内可进行因式分解：

$$e^{((c+o(1))(\log n)1/3(\log\log n)2/3)}$$

2、离散对数问题：设 p 是一个素数，乘法群 Z_p^*= ({x<P|(x,p)=1},.mod P)是循环的。因此，对于 g∈Z_p^*, Z_p^*= {g^imod P|1≤i≤P-1}。函数 f:（p,g,x）→(g^xmod p,p,g),p 是素数，g 是 Z_p^* 的一个生成元，假设是单向函数， 使用反复的 SQUARE 函数在多项式时间内计算出 f（p,g,x），然而，已知对离散对数问题的求逆的最快的方法是微分索引算法，运算时间为：L（P）$\sqrt{2}$ 。一个有趣的难题是寻找一个算法产生素数 p 和 Z_p^*的生成元 g。目前还不能确定是否可以在多项式时间内可以找到此类算法。然而，在文献[8]中，E.Bach 表现了怎样产生在（N/2，…，N）随机整数，与快速素性测试结合（例如[126]中的方法），这样可以产生足够的 P 随机数素数（p-1,q_1,…,q_k）。随机选择 g∈Z_p^*,如果（g,p-1）=1，∀ q_i,$g^{p-1/qi}$mod p=1，这样，g 的阶为 p-1（order(g)=|{g^imodp|1≤i≤p-1}|）。可以看出，Z_p^*的生成元是稠密的，因此只需要少量的猜测就可以得到结论，在 Z_p^*上寻找生成元效率的问题是一个公开的课题。

3、子集数量

设 a_i∈{0，1}^{n}，$\bar{a}$=（a_1…,a_n）,S_i∈{0,1}, $\bar{S}$=（S_1…,S_n）,并且，f（$\bar{a}$,$\bar{S}$）→($\bar{a}$, $\sum_{i=1}^{n} S_i a_i$)。在 f 作用下的($\bar{a}$, $\sum_{i=1}^{n} S_i a_i$)的逆是（$\bar{a}$,$\bar{S}$）使得 $\sum_{i=1}^{n} S_i a_i = \sum_{i=1}^{n} S_i' a_i$ 的所有元素对，F 函数是单向函数中的一种。选择元素对难题是一个 NP 完全难题（给定（$\bar{a}$,y）是否存在$\bar{S}$ 使得 $\sum_{i=1}^{n} S_i a_i$ =y?的问题）。当然，子集数量问题是 NP 完全问题并不能为 f_{ss} 的单向性提供证据。另外，子集数量问题对于特殊情况是容易的，例如在低元素密度和隐藏机构的前提下，同样不能提供该提议的弱点。F 是单向函数是基于对随机高密度条件下已知算法求解的失败前提下的。当然，必须承认该候选算法比另外两个候选算法弱一些。

4、使用固定消息的 DES。固定 64 比特消息，定义函数 f（K）=DES_K（M）为 56 比特密钥、64 比特输出。这也可以看做一种单向函数。事实上，这种结构被证明是基于 DES 的一簇随机函数，Luby 和 Rackoff[139]已经给出证明。

5、RSA。这是陷门函数的一种。设 N = pq 是两个素数的乘积。求解 N 是一个困难的事情。函数 f（x）=x^e mod N，这里 e 等于（p-1）(q-1)。陷门在于已知 p，q 求逆函数是容易的。函数 f 是单向函数。目前最好的攻击方法都是计算不可行的。

第二章中，讨论通常意义下的精确定义基于以上的知识构架。

1.4 安全定义

到目前为止，没有严格的定义安全和系统破解的概念。那么这个注记真正的意图是什么呢？明显的，对安全的最低要求应该是：攻击者在已知密文和加密算法的条件下，不能完全还原明文，但是，更多的特点可以被确定：

1、当明文被抽象成随机序列流后，信息是很难还原的。例如，英文转换成为{0，1}，

假设信息空间被攻击者已知。

2、从密文计算部分信息是困难的。

3、当相同的信息被发送两次，从信息发送过程中应该很难发现有用的信息泄露。

4、上述性质应该有很高的随机性。

简而言之，加密过程被描述成信息通过不透明的信封封装后发送。对于合法用户信封是可拆的，而对于非法用户，信封无法打开。必须回答如下问题：

1、 如何用精确的数学定义来定义不透明信封。第六章、第七章描述在攻击者计算范围内的安全精确定义。

2、 不透明替换对数学是否有促进作用？答案是肯定的，描述在不同假设条件下私钥密码体制的替换安全性。

之前描述的基于陷门的公钥加密体制不满足上述特征。稍后会证明，如果假设陷门体制真的存在，简单的随机变化体制是可以满足上述条件的。更特殊的，将证明 RSA 体制的随机性满足下述新的安全要求。假设 RSA 原函数是陷门函数，并且与 RSA 公钥加密体制是在同样的安全水平上的。

1.5 攻击者模型

以下讨论基于假设：攻击者可以窃听在不安全信道上的密文信息，阅读公共信息，产生自己的密码，这些功能都在多项式概率时间内完成。这种模式称为被动攻击。

更强的攻击者可以在发送和接收者替换的信息之间加入自己的信息，而不被通信双发发现，更坏的情况是攻击者可以在多项式时间内完成对密信息的解密，要研究是否有这样一种密码体制可以抗击如此强大的攻击。

在第六章和第七章，假设攻击者已知消息的概率分布，继续讨论更强大的攻击者模型和更安全的体制。

1.6 加密进程

对于简洁的总结，面临的问题是设计私钥和公钥加密体制来满足安全要求，并且加密、解密速度要求尽量快。

在第六章、第七章密码学话题进行深入的研究，由如下部分组成，对私钥密码和公钥密码学而言，将进行如下工作：

1、使用正式的安全定义针对计算能力有限的攻击者。

2、对不同的安全选择定义来讨论和评估目前的加密定义。

3、在承认单向函数、陷门体制、随机数发生器存在的前提条件下，讨论怎样设计安全体制来确保安全性。

4、在适当的改动一批模式的条件下，通过预先离线计算来提高密码体制的效率。

将对一些密码学的前沿课题进行研究，例如选择密文安全，可扩展性，第三方托管协议，网络多用户的解密共享思想。

第二章 单向函数和陷门体制

在密码学的基础定义中，计算容易、求逆困难的单向函数是基本的元素。也许，最简单和最常用的单向函数时 PASSWORD。也就是，一个时间共享的计算机制，对任何口令字对应一个口令表。一个人可以对一个口令 w，存储一个 f（w）。在注册时，口令字可以被容易地检测出是否存在错误，但是系统管理员不能根据口令表获取用户的口令。

在 1、3 节中列出了一些单向函数的简单例子，现在此外对单向陷门函数进行理论上的探讨，仔细对照文献中列出的单向函数例子。将偶尔提及一些数论的概念如同附录 C。

以下将讨论为什么单向函数是密码学的基础理论。

2.1 单向函数：设计动机

在这一章，给出单向函数定义设计的动机。证明了单向函数的存在性是多数密码理论存在的必要前提条件（包括加密体制安全和数字签名）。复杂性理论目前的状态不允许证明单向函数的存在性。使用 P≠NP 的前提条件，证明单向函数的存在性。会对这个假设的安全性进行进一步证明。

在介绍章节的开始，现代密码学是基于有效算法对合法用户的保障与攻击者对被保护信息的获取的实现难度之间的差距而建立的。为了对如下的讨论更清晰，对安全数据通信的密码学任务，命名为密码体制。

在安全加密体制中，合法用户可以用秘密信息进行解密。对于没有秘密信息的攻击者，只对密文进行解密是不可行的。显然的，是否可以破解一个密码算法是取决于在非确定多项式时间是否可行。安全性的定义在于破解是不可行的，或者说在多项式时间内还原算法是不可行的。也就是说，存在这样一种密码体制，可以在非确定的多项式时间进行运算但是不能在确定的多项式时间进行运算。换言之，安全密码体制存在的前提条件在与：BPP 没有包括 NP 难题。（因此，P≠NP）。

然而，上述提到的条件（P≠NP）不时足够的。P≠NP 仅仅证明安全体制在恶劣条件下破解的难度，但是没有排除在各种条件下安全体制被破解的可能。事实上，一个人可以容易地建立一个 NP 难度的加密体制并且存在一个有效的破解算法成功还原 99%的情况。因此，最差情况的难度在于怎样测量安全的弱点。安全性的要求在于大多数条件或至少通常条件下的安全。因此，安全密码体制存在的必要条件是存在普通难度的 NP 安全语言。而且，在普通情况下 NP 难度的语言不能证明 P≠NP 的存在。

在普通难度下存在 NP 难题条件是不够的。为了能够使用这些难题，必须产生一些带有附加信息的例证，使用这些附加信息可以快速的解决这些难题。否则，这些难题对于攻击者和合法用户没有明显的区别来保证安全性。因此，安全加密体制的存在性应该包括难题例证及解决这些难题的附加条件：

（1） 使用附加信息容易解决这些难题；

（2） 当不使用附加信息时，通常很难解决这些问题。

避免明确的描述上述“定义”。仅仅评论了使用随机过程的方法来实现和解决例证的方案。因此，不失一般性的，一个人可以通过随机过程的不可恢复性来满足（2）的条件。上述条件的产生导致了单项函数定义。

2.2 单向函数：定义

在这一节，给出了几个单向函数的定义。在第一版本，以后通常指强单向函数（或仅为单向函数）是最简单的一个定义。也定义了一个弱单向函数，也许容易被发现但是很难构造成为强单向函数和非均衡的单向函数。

2.2.1 强单向函数

最基础和原始的密码学应用是单向函数。非正式的定义这是一个计算简单求逆困难的函数。也就是，任何一个多项式概率时间的计算单向函数的算法是几乎不可能的。在多项式时间尝试求随机过程的逆求定义域中某些元素也是同样困难的。

这种不确定的定义直接与复杂性理论密码学相关。一个简单的计算能够在 PPT 时间完成，一个函数 V:N→R 是可以忽略的，如果它和任何逆多项式函数的可以迅速得到单位元。更抽象的定义为：

定义 2.1、V 是对任意的 c>0 的常数都可忽略的，存在一个整数 K_c，使得任意 $K \geq K_c$, V（k）$<k^{-c}$ 成立，其中，V（k）$=k^{-W(1)}$。

简而言之，首先要关心极小概率。上述定义和讨论，对于输入长度的选择可以根据多项式不同定义域的边界。通过重复多项式时间的算法来产生出一个新的算法也可以得到好的后继概率。换种说法，对多项式概率时间事件进行多项式次数的重复，可忽略的事件仍然是可忽略的。因此，在多项式时间内计算可以与在任意多项式时间段都在小概率范围内的概念等同。如果存在多项式 P，认为函数 V 是可忽略的，使得有足够大的 k 满足 V（k）>1/P(k)。因此，函数是可忽略的，也可以是明显的。

定义 2.2、一个函数：f：$\{0，1\}^* \rightarrow \{0，1\}^*$是单向的，如果：

（1） 存在多项式时间函数，输入为 x,输出为 f(x)。

（2） 对每一个 PPT 时间的算法 A，存在一个可忽略的函数 V_A，能够找到足够大的 K，使得：

$$P[f(z)=y:x \leftarrow \{0,1\}^k;y \leftarrow f(x);z \leftarrow A(1^k,y)] \leq V_A(k)$$

注释 2.3、保障是随机的。攻击者不是不能对函数求逆，但是只能以极低的概率进行攻击，概率分布是以长度 x 为输入以 k 作用输出。明显的，x 可以随即选择，而 y 是 f(x)簇。

注释 2.4、攻击者没有要求还原 x，求取 x 对攻击者来说几乎是不可能的。要求对 y 求逆。自然的，如果函数是一一映射，x 是可以唯一解出的。

注释 2.5、攻击者的函数以 f(x)为输入，安全参数 1^k 可以表示 x 的输入宽度。这表示攻击这在多项式时间内|x|，甚至 f(x)比 x 少得多。因为没有足够的时间对逆函数求输出，所以该函数被认为是单向函数。考虑一个例子，假设 f(x)=y,y 至少有 x 的 logk 签名比特，信息长度为|x|=k。既然|f(x)|=log|x|，在多项式时间内|x|的找到明显的 f（x）的逆函数。注意在一些特殊条件下保留的函数 f，例如|f(x)|=|x|，附加的信息是多余的。

注释 2.6、只要 f（x）是多项式时间可计算方法的，输出 f 的时间在 k 多项式时间内是平凡的。

注释 2.7、从计算复杂度理论的角度给出定义，使用渐近复杂度理论难题的范围扩大。安全性只是要求足够大的输入，或者是 k 尽可能大。对于这个定义，512 比特的输入的函数 f 是可逆的。因此，这类定义与实践并不直接相关，但是在基础理论层面的研究。对于实用密码学的定义必须正视单向函数不仅仅只存在一个，而是存在类似的一簇函数，使用安全参数 k。对于安全参数 k 的每个值，会有一个特殊的函数 f:$\{0，1\}^k \rightarrow \{0，1\}^*$。对于每个参数 k

存在一个函数簇（密码体制）。会在随后的章节中讨论这个问题。

以下的两章将会讨论强单向函数，第一次阅读的作者建议直接阅读 2.2.4 节。

2.2.2 弱单项函数

单向函数可以分为两类：弱单向和强单向函数。上述给出的定义为，指定一个强单向函数。使用以下弱的方式把强函数降成弱函数。

定义 2.8、一个函数 $f:\{0,1\}^*\to\{0,1\}^*$是一个弱单向函数，如果满足：

（1） 存在一个 PPT，使输入为 x，输出为 $f(x)$;

（2） 有一个多项式时间函数 Q，使得每一个 PPT 时间算法，对于足够大的 k，

$$P[f(z)\neq y:\ x\leftarrow\{0,1\}^k;y\leftarrow f(x);z\leftarrow A(1^k,y)]\geq 1/Q(k)$$

两个定义之间的区别是这样的，仅仅要求弱单向函数对一些输入明显的部分求逆困难，而强单向函数要求对所有的几乎明显输入求逆是。更显然，强单向函数更具优越性，但是如果只有弱单向函数存在呢？的理论是，存在弱单向函数就存在强单向函数。而且，给出了使用弱单向函数构造强单向函数的方法。在实际应用中这个方法是重要的。

例 2.9、考虑函数的例子 f：$Z\times Z\to Z$，这里 $f(x,y)=x\cdot y$。这个函数至少有一半的值可以求得逆，所以不是个强单向函数。还有，在第一章提到，当 x,y 是大概是 k 比特多项式分数相同大小的素数时，求逆是困难的。这是定义弱单向函数的依据。可以知道 K 比特大小的整数是素数的概率接近为 1/K，还给出当|x|=|y|=K 时，是素数的概率为 $1/K^2$。因此，对于所有的 K，有 $1-1/K^2$ 的输入是 2K 大小的 f 是相等长度的素数。可以确信对于相等长度的素数 x,y 是求逆是不可能在无可忽略的时间内成功的。基于这种假设，f 是一个弱单向函数（上述条件 2 满足 O（K）=O（K^2））。

定理 2.10、弱单向函数存在当且仅当强单向函数存在。

粗略证明：根据定义，强单向函数一定是弱单向函数。现在假设 f 是弱单向函数，因此在条件 2 下，Q 是在弱单向函数的定义下是多项式时间函数。定义函数：

$F_1(x_1\ldots x_N)=f(x_1)\ldots f(x_N)$,这里，N=2kQ（K）每一个 x_i 的长度是 K。

认为 F_1 为强单向函数。既然 F_1 是一个 N 级 f 串联函数，为了求 F_1 的逆函数，需要对每一个 x_i 求 $f(x_1)$的逆，知道对于攻击者有至少 1/Q（x）的概率求不出 f（x）的逆（概率满足 $x\in\{0,1\}^k$，攻击者满足概率随机）。因此，直观上为了求 F_1 的逆，需要使用计算量 O（kQ(k)）来得到 f 的逆。对于攻击者对至少一种函数的攻击成功概率很高。

正式的证明（此处省略，附录中给出）将以一种归约的形式给出，假设与 F_1 是不是强单向函数矛盾。假设存在对手 A_1 在强单向函数的定义下不满足条件 2，得出 A_1 可以被另外的攻击者 A 使用成为一个子函数，以大于 1-1/Q（|x|）的概率求取 F_1 的逆函数（这里对于输入为 $x\in\{0,1\}^k$ 定义域是 A 是随机的）。

这种证明技术是该书中非常典型的一中方法。只要通过归约的消耗就可以证明是重要的。例如，进行的框架轮廓构造是不够长度保留的，对二次方程函数的输入大小是可扩展的。

2.2.3 非正规的单向函数

对于上述两种单向函数定义的逆函数是多项式时间可计算（PPT），如果满足下述条件：

（1） 计算简单，存在多项式时间算法计算 f 函数

（2） 求逆困难，对于每个多项式时间算法簇 $A=\{M_k\}_{k\in N}$,这里存在一个可忽略的量 V_A,对于足够大的一个数 k,使得：$P[f(z)\neq y:\ x\leftarrow\{0,1\}^k;y\leftarrow f(x);z\leftarrow M_k(y)]\leq V_A(k)$

注意：使用 1^k 作为输入 M_k 的附加条件是重复的。

可以看出，如果 f 是非正规的单向性，那么它也是（强）单向的。多项式时间求逆的证明是非正规的多项式算法簇，而且没有减少后向概率。细节如下：假设 A'为多项式时间（逆）算法。R_k 表示一串随机测试使得 A'的后向概率可以最大化。期望算法 M_k 可以包括 A'的算法代码和序列 R_k（是一个 k 次多项式）。

无法特别确定但是有可能存在的是：强单向函数存在，非正规的强单向函数不一定存在。

2.2.4 单向函数集

考虑一个函数 f：{0，1}*→{0，1}*，容易地得到定义在有限域和有限范围内的函数簇。注意到，然而，可以正式定义单向函数的一些特征。

定义 2.12、假设 I 是指标集并且任意 i∈I，D_i 和 R_i 是有限的。一个强单向函数的集定义为：$F=\{f_i: D_i \to R_i\}_{i\in I}$ 满足下列条件

（1）存在一个 PPT 时间的 S_1，输入是 1^k，输出是 $i\in\{0,1\}^k \cap I$

（2）存在一个 PPT 时间的 S_2，输入是 i∈I，输出是 $x\in D_i$

（3）存在一个 PPT 时间的 A_1，输入 i∈I，$x\in D_i$，$A_1(i,x)=f_i(x)$

（4）对每一个 PPT 时间的算法 A，存在一个可忽略的函数 V_A，能够找到足够大的 K，使得：

$$P[f_i(z)=y:i\leftarrow I;x\leftarrow D_i;y\leftarrow f_i(x);z\leftarrow A(i,y)]\le V_A(k)$$

（对于随机事件 A，选择 i,x 的概率是随机的）

通常的，可以得出一个单向函数的存在性等同于单向函数簇的存在性。

定理 2.13、一个单独的单向函数簇存在当且仅当单向函数存在。

证明：假设 f 是一个单向函数。$F=\{f_i: D_i\to R_i\}_{i\in I}$，当 I={0，1}*，对任意 i∈I，认为 $D_i=R_i=\{0,1\}^{|i|}$并且，$f_i(x)=f(x)$。而且，S_1 均衡地选择输入 1^k，$i\in\{0,1\}^k$，S_2 均衡地选择输入 i，$x\in D_i=\{0,1\}^{|i|}$，并且 $A_i(i,x)=f_i(x)=f(x)$（注意：f 是多项式时间可算的）。定义的条件 4 的单向函数的集与 f 函数的单向性是一致的。

现在假设 $F=\{f_i: D_i\to R_i\}_{i\in I}$ 是一个单向函数集。定义：$f_F(1^k,r_1,r_2)=A_1(S_1(1^k,r_1),S_2(S_1(1^k,r_1),r_2)))$,这里，$A_1$，$S_1$ 和 S_2 是在定义 2.12 中与 F 相配套的。用另外的话来说 f_F 分别随机使用 S_1 和 S_2 的变量 r_1, r_2，产生随机串 $1^k\cdot r_1\cdot r_2$,那么：

（1）当输入 1^k 时，使用随机数 r_1 运算 S_1 产生函数 $f_i\in F$ 索引值 $i=S_1(1^k,r_1)$。

（2）当 S_1 输出是 i 时，找到 r_2 运算 S_2 满足输入 $x=S_2(i,r_2)$。

（3）在 i,x 上计算 A_1 计算 $f_F(1^k,r_1,r_2)=A_1(i,x)=f_i(x)$

注意：对有限制的函数 f_F 已经进行了随机化，既然 A_1 在多项式时间是可计算的，那么，单向函数的条件得到了满足。

在 2.3 节已彻底解决了例子如下：

例 2.14、离散对数的计算难度导致了下列的函数簇的产生。定义 $EXP=\{EXP_{p,g}(i)=g^i \bmod p, EXP_{p,g}: Z_p\to Z_p^*\}_{<p,g>\in I}$,对任意的 I={<p,g>p 是素数，g 是 Z_p^*的生成元}。

2.2.5 陷门函数和集

特别的，陷门函数是一种有特殊性质的单向函数 f。在合法用户所选择的定义域内存在可求 f 的逆函数。应该是在任何定义域的点都容易计算函数 f 的值，但是对没有逆函数知识的攻击者对任意值求逆是困难的。另外，很容易产生 f 的函数对得到通信陷门。一旦陷门对

确定，对 f 公开的信息不能泄露任何关于逆函数的知识。

定义 2.15、陷门函数是一个单向函数 f：{0，1}*→{0，1}*，存在一个多项式 p 和一个多项式时间算法 I，对于每个 k，存在一个 t_k ∈{0，1}*,使得$|t_k|\le p(k)$对于任意的 x∈{0，1}*, I(f(x), t_k)=y, 满足 f(y)=f(x)。

另外一个陷门函数的例子是使用 Rabin 算法对整数分解的困难。设 f(x,n)=x^2 mod n，这里 n=pq，是两个素数的乘积，x∈Z_n^*,Rabin[170]发现如果素数的乘积的合数求取是容易的，那么，分解就是容易的。最有名的陷门函数是 RSA 体制，f(x,n,l)=x^l mod n 当（l,$\phi(n)$）=1。

使用安全参数 k 对陷门函数簇的使用更容易。

定义 2.16、设 I 是一个复数集，i∈I，D_i 是一个有限域。一个强单向陷门函数是 F={f_i:D_i→D_i }$_{i\in I}$ 满足下列条件：

（1） 存在一个多项式 p，一个 PTM S_1 ，输入 1^k ，输出对（i,t_i），这里，i∈I∩{0,1}^{k} 并且| t_i |<p(k),信息 t_i 是陷门输入为 i 的值。

（2） 存在一个 PTM 时间的 S_2，输入是 i∈I，输出是 x∈D_i

（3） 存在一个 PTM 时间的 A_1，输入 i∈I，x∈D_i，A_1（i,x）=f_i(x)

（4） 存在一个 PTM 时间的 A_2，使得 A_2（i,t_i，f_i(x)）= x ,对任意 i∈I，x∈D_i(当 t_i 已知时，求 f_i 的逆是容易的)

（5） 对每一个 PPT 时间的算法 A，存在一个可忽略的函数 V_A, 能够找到足够大的任意 K，使得：P[f_i(z)=y:i←I;x←D_i;y←f_i(x);z←A(i,y)]≤V_A(k)

(对于随机事件 A，选择 i,x 的概率是随机的)

通常的，可以得出一个单向函数的存在性等同于单向函数簇的存在性。具体例子如下：

例 2.17、[RSA 陷门体制]假设 p,q 为素数，n=pq，Z_n^*={1≤x≤n,(x,n)=1},乘法群的集合的势为：$\phi(n)$=（p-1）(q-1), e∈Z_{p-1} ，e 是与$\phi(n)$相关的素数，的复数集为：I={<n,e>使得 n=pq,|p|=|q|，单向陷门对为<n,e>的特别的摘要 d，使得 ed=1mod $\phi(n)$ 。RSA 体制为 RSA={$RSA_{<n,e>}$；Z_n^*→Z_n^*}$_{<n,e>\in I}$ ，这里 $RSA_{<n,e>}$（x）= x^e mod n。

2.3 搜索的例子

数论提供了单向陷门函数的理论基础，让搜索为例脱离纯粹的数论问题，参考附录 C 的数论基础问题。

在 Z_n^*上求逆

考虑集合 Z_n^*={x: 1≤x<p 并且 gcd(x,n)=1}。Z_n^*是在数乘 p 下的群。注意求取任意 x∈Z_n^* 的逆元，是有一个元 y∈Z_n^*使得：yx =1mod p ,可以使用欧几里得算法求逆，找到整数 y,z，使得：yx+zp=1=gcd(x,p)。这样，yx≡1 mod p，这样 y mod p 是所求的逆。

欧拉φ（n）函数

欧拉φ(n)函数的定义为：φ（n）= |{x: 1≤x<p 并且 gcd(x,n)=1}|。以下是关于φ的特点：

（1）对于一个 p 素数，a≥1，φ（p^a）= p^{a-1}(p-1)

（2）对于整数 m,n，gcd(m,n)=1，φ（mn）= φ(m) φ(n)。使用上述规律，通常得到：

$$\varphi(n)=\varphi\left(\prod_{i=1}^{k}p_i^{a_i}\right)$$

$$=\prod_{i=1}^{k}\varphi(p_i^{a_i})$$

$$=\prod_{i=1}^{k}p_i^{a_i-1}(p_i-1)$$

Z_n^*是循环群

一个群 G 是循环的，当切仅当对存在元素 g∈G，任意元素 a∈G，存在一个整数 i，使得 $g^i=a$。称 g 是 G 的生成元，i 是 a 在群中的阶数，记为 $ind_g(a)$。

定理 2.18（猜想）、如果 p 是素数，那么 Z_n^*是阶为 p-1 的循环群。也就是说，存在一个元素，$g\in Z_n^*$,$i<p-1$，使得 $g^{p-1}=1 \bmod p$,并且 $g^i\neq 1 \bmod p$。

从定理 2.18 立刻可以得到如下的事实。

事实 2.19、给定一个素数 p,那么 Z_n^*的一个生成元 $a\in Z_n^*$，只有唯一的一个 i，使得 $a=g^i$, $1\le i\le p-1$。

勒让德符号

公理 2.20、假 p 是一个素数，g 是 Z_n^*的一个生成元，那么

$$G^c=g^ag^b \bmod p \iff c=a+b \bmod p-1$$

从这个理论可以得到存在一个同态 $f: Z_n^* \to Z_{p-1}$,那么 $f(ab)=f(a)+f(b)$。有些情况下，使用 Z_{p-1} 来代替 Z_n^*进行讨论更加方便和简单。例如，决定 Z_n^*有多少个元时隔完全平方问题。（其中的元素是一个模 p 的二次剩余问题。）以下推论告诉模 p 的二次剩余的元素个数为：$|Z_n^*|/2$。

推论 2.21、$a\in Z_n^*$是模 p 的二次剩余当且仅当 $a=g^x \bmod p$,x 满足 $1\le x<p$ 并且是偶数。

证明：设 g 是 Z_n^*的生成元。

（$\Leftarrow$）假设对一些元素 x，一个元 $a=g^{2x}$ ，那么，$a=s^2$,这里，$s=g^x$ 。

（$\Rightarrow$）考虑一个元素的平方，$b=g^y$，$b^2=g^{2x}=g^e \bmod p$,这里 e=2y 是模 p-1 的偶数，因此这些元素可以表示为：g^e,e 是一个偶数，并且是平方数。

因此，模 p 二次剩余的元素个数为 Z_n^*的偶数阶生成元个数。显然，个数为$|Z_n^*|/2$。

勒让得符号 $J_p(x)=\begin{cases}1 & \text{如果}x\text{是}Z_p^*\text{二次剩余}\\ 0 & \gcd(x,p)\neq 1\\ -1 & \text{如果}x\text{不是}Z_p^*\text{二次剩余}\end{cases}$

勒让得符号使用如下定理在多项式时间是可求出的。

定理 2.22[欧拉定理]、 $J_p(x)\equiv x^{(p-q)/2} \bmod p$ 。

使用重复两次的计算指数，可以用 O（$|p^3|$）步计算出 $x^{(p-q)/2}$。当 p 为素数时，$J_p(x)$可以被计算出，通过这种方法，可以对通常意义下的 x,n 来计算 Z_n^*的平方数。

2.3.1 离散对数函数

假设 EXP 为如下定义的函数：EXP（p,g,x）=（p,g,g^x mod p）。特别对 p 是素数时，g 是 Z_n^*生成元的情况关注。一个索引集合：I={（p,g）:p 是素数，g 是 Z_n^*的生成元}。对于（p,g）∈I,根据公理 2.19, EXP（p,g,x）有唯一的逆元，定义 $y\in Z_n^*$,离散对数 DL（p,g,y）=(p,g,x),

$x \in Z_{p-1}, g^x \equiv y \bmod p$，给定 p,g, EXP（p,g,x）可以在多项式时间计算。然而，如果 p-1 是很小的一个因素影响 DL 的逆是否可以在多项式时间内计算[158]。Pohlig 和 Hellman[158]提供了一个非常有效的技术对 p-1 只有一个很小的素因素时。目前被证明是最好的离散对数求取算法是微分索引算法。算法期望的计算时间是 $e^{\sqrt{k \log k}}$,k 是模数 p 的大小。引进数语一个新变量来定义离散对数筛法，可以使用快速算法 $e^{k \log k^{1/3}}$ 。明显地，在有限域 GF（2^k）进行运算比在模数 p 的域上运算更使问题简单化（Coppersmith 和 Odlyzko）。特别的，离散对数计算和整数因子分解有本质的同样的难题，反映了目前算法的状态。

认为 EXP 是一个好的单向算法不错的选择。直接对这个算法进行了假设，基本假设是不存在多项式时间算法，在使用素模时解决离散对数问题。

强离散对数假设（DLA）：对每一个多项式 Q 和每一个 PPT 算法 A，对所有足够大的整数 k，Pr[A（p,g,y）= x 使得，$y \equiv g^x \bmod p$,x 满足 $1 \le x < p] < 1/Q(k)$（对所有素数 p 的概率是$|p| \le k$, Z_n^*的生成元，$x \in Z_n^*$,A 是随机概率事件）

这个假设直接的结果是：

定理 2.23、在强离散对数的假设前提下，存在强单向函数。也就是：以素数为模求幂。一些有用的 EXP 特点和 DL 如下：

注释 2.24、如果 DL（p,g_1,y）对一些生成元 $g_1 \in Z_p^*$容易产生，对于其他的 $g_2 \in Z_p^*$仍然容易产生 DL（p,g_2,y）（Z_p^*有 φ（p-1）个生成元）。这种情况说明了 x_1=DL（p,g_1,y）和 x_2=DL（p,g_2,y），如果 $g_2 = g_1^z \bmod p$,这里，gcd(z,p-1)那么，$y \equiv g_1^{x2z} \bmod p$,因此，$x_2 \equiv Z^{-1} x_1 \bmod p-1$ 。

以下的结论有效计 DL（p,g,y）任意（p,g）∈I,这样可以找到一个多项式时间算法可以计算出 DL（p,g,y），对于一些多项式 Q，至少 1/（Q|p|）可能的部分输入 $y \in Z_p^*$ 。

命题 2.25、设 $\varsigma, \delta \in$（0，1），假设 S 是一个素整数的子集。假设有一个概率随机算法 A，对所有素数 $p \in S$,对所有 Z_p^*的生成元 g 有：Pr[A（p,g,y）=x 并且 $g^x = y \bmod p] > \varsigma$

（A 是随机事件，$y \in Z_p^*$是随机的），A 在多项式时间|p|内运算。存在概率随即事件 A’，概率随机事件 A’以 $\varsigma^{-1}, \delta^{-1}$ 多项式时间运算，对所有的素数 p。对于 Z_p^*的生成元 g，$y \in Z_p^*$。

Pr[A’（p,g,y）= x 使得 $g^x = y \bmod p]$ 〉1- σ 。

（A’是随机事件，概率是随机概率）

证明：选择最小的整数 Z，使得 $1/e^N < \sigma$

考虑算法Ａ′，如下运算输入 p∈Ｓ，g 是 Z_p^*的生成元，$y \in Z_p^*$

重复 ς^{-1} Ｎ次

随机选择 z，使得 $1 \le x < p$

假设 $w = A(p,g,g^z y)$

如果Ａ事件成功，那么，$g^w = g^z y = g^{z+x} \bmod p$，当 $x = DL_{p,g}(y)$, $DL_{p,g}(y) = w - z \bmod p-1$

否则，继续下一个循环

结束循环。

可以预计Ａ′失败的概率。

Pr[A’（p,g,y）失败]＝Ｐｒ［循环Ａ′的单圈Ａ的失败的概率为］ $\varsigma^{-1} N$

$$< (1-\varsigma^{-1}) \varsigma^{-1} N$$

$<(e^{-N})$

$<\sigma$

注意，如果 N=O（$\log(\varsigma^{-1})$）= O(ς^{-1}),A'是一个随机概率算法，在多项式时间内 $\varsigma^{-1},\delta^{-1}$ 和 |p|进行运算。

离散对数算法产生如下的函数集：

假设 I={（p,g）:p 是素数，g 是 Z_p^* 的生成元}，定义：

$EXP=\{EXP_{p,g}:Z_{p-1}\rightarrow Z_p^*$,这里，$EXP_{p,g}(x)=g^x \bmod p\}_{(p,g)\in I}$ 。

那么，在强离散对数的前提假设下，EXP 是一个强单向函数的集。以下证明该结论的正确性。

定理 2.26、在强离散对数假设的前提条件下，强单向函数是存在的。

证明：在 EXP，DLA 条件下确实是一个单向函数集。

对于这个方面，必须满足强单向函数的每个条件。

对于条件 1，定义 S_1 对于 1^k 的输入运算为：

（1） 运算单独的算法（[8]），得到随机整数 n，满足|n|=k,进行因式分解。

（2） 测试 p+1 是否为素数，在 C.9.的素性测试。

（3） 如果是这样，让 P=n+1,给出 p-1 的素因数分解，寻找一个 Z_p^*的生成元：

Ⅰ：随机选择 $g\in Z_p^*$

II：如果 $p-1=\prod_i q_i^{a_i}$ 是 p-1 的素因素分解。对于每个 q_i ,检查 $g^{(p-1)/qi}\neq 1 \bmod p$,如果是这样，g 是 Z_p^*的生成元，输出 p 和 g。

否则，重复步骤 1。

命题 2.27、g 是 Z_p^*的生成元，如果对每个 p-1 的素约数，$g^{(p-1)/q}\neq 1 \bmod p$

证明：g 元是 Z_p^*的生成元，如果 $g^{p-1}\equiv 1 \bmod p$ 并且 $g^i\neq 1 \bmod p$,对任意 j,使得，1≤j<p-1,那么 g 在 Z_p^*中的阶为 p-1。

现在，假设 g 满足 2.27 条件为，假设 m 为 g 在 Z_p^* 的阶。那么 m|p-1,如果 m<p-1，那么存在素数 q,使得 m|(p-1)/q，那么存在一个整数 d，使得 md=（p-1）/q 。因此，

$G^{(p-1)/q} = (g^m)^d \equiv 1 \bmod n$，与假设矛盾。因此，m=p-1, g 是 Z_p^* 的生成元。

Z_p^* 是生成元的个数，φ（p-1），并且满足：$\varphi(k) > \dfrac{k}{6\log\log k}$

因此希望选择 O（loglog p） 个候选元素 g 来得到生成元。因此，S_1 在期望的多项式时间运算。

对于条件 2，单向函数簇的定义为，可以定义 S_2，简单输出为 $x\in Z_{p-1}$ ，随机对 i=(p,q)

既然计算 $g^x \bmod p$ 在条件 3 下是真实的，那么，在多项式时间是可执行的。条件 4 是在强离散对数假设下的定义。

2.3.2 RSA 函数

1977 年，Rivest、Shamie、Adleman 提出了一公钥陷门函数，作为满足 Deffie、Hellman 的公钥密码体制的候选算法。被提出的陷门函数是 $RSA(n,e,x)=x^e \bmod n$, n 是两个大素数 p,q 的乘积，并且，gcd（e,$\phi(n)$）=1,相应的陷门信息是 d, 使得：$d\cdot e\equiv 1 \bmod \phi(n)$。作为一个集合，$RSA = \{RSA_{n,e}: Z_n^* \rightarrow Z_n^*$ 在这里，$RSA_{n,e}(x) = x^e \bmod n\}_{(n,e)\in I}$ 。对于任意的 I={<

n,e >s.t. n=pq, |p|=|q|,(e, ϕ(n))=1}。

RSA 容易计算，求逆的计算难度有多大？如果可以通过中国剩余定理求出 n 的因数分解，就可以求出逆。但是，不能判断求出的逆是否是真实的。求 RSA 逆的一个要素是求 n。目前已经有多种算法来求逆。被充分证实的最快运算时间是 DIXON 随机平方算法是 0（$e^{\sqrt{\log n \log\log n}}$），事实上，考虑到其算法，假设 l=|P|，p 是 n 最小的素因数。椭圆曲线函数才用的时间是 O($e^{\sqrt{2l\log l}}$)。二次筛法算法在 O（$e^{\sqrt{\ln n \ln\ln n}}$）。这个意味着，运算时间的多项式组成的参数是有区别的，有一个素因子比其因子小，就可以考虑椭圆曲线方法，否则就考虑二次筛法。新的数域筛法似乎达到了 O（$e^{1.9(\ln n)1/3(\ln\ln n)2/3}$）运算对一些整数，事实上渐近逼近比二次筛法速度快一些，目前，大家努力去求因子。现在推荐的 n 的位数为 1024 比特。

总而言之，直接假设并证明 RSA 是单向陷门函数簇。

强 RSA 假设：[2] 假设 H_k = {n=pq :p≠q 是素数，|p|=|q|},对每个多项式 Q 每个 PTM 事件 A，存在一个整数 k_0 使得任意 k>k_0 , Pr[A(n,e,$RSA_{n,e}$(x))=x]<1/Q(k)

（这里，对所有 n∈H_k ,e 使得 gcd(e,φ(n))=1,x∈Z_n^* ,A 是随机概率事件）

还需要证明一些附加的条件。

命题 2.28、对于（n,e）∈I, $RSA_{n,e}$ 是 Z_n^* 上的置换。

证明：因为，gcd（e,φ (n)）=1,相应的陷门信息是 d,使得：d·e≡1 mod ϕ(n)，给出 x∈Z_n^* 考虑元素 x^d ∈Z_n^* ,那么，$RSA_{n,e}(x^d)\equiv(x^d)^e \equiv x \bmod n$, $RSA_{n,e}$：$Z_n^* \to Z_n^*$ ，因此，在有限域的|Z_n^* |上的 $RSA_{n,e}$ 是 Z_n^* 上的置换。

评论 2.29、注意，上述利用构建证明了 RSA 有唯一的逆，因此，gcd（e,φ (n)）=1，如果运算扩展欧拉函数，可以找到 d∈Z_n^* ,使得： $RSA_{n,e}^{-1}$（x）=$(x^e \bmod n)^d \bmod n = x^{ed} \bmod n = x \bmod n$

注意，一旦找到了 d，使得 d·e≡1 mod φ (n)，可以求得 $RSA_{n,e}$ 的逆。因为，$RSA_{n,e}(x^d)\equiv(x^d)^e \equiv x \bmod n$。

定理 2.30、在强 RSA 假设下，RSA 是一个强单向陷门置换的簇。

证明：首先注意命题 2.28，$RSA_{n,e}$ 是 Z_n^* 上的置换。必须表示出定义 2.16 下的任何一个条件。条件 1，在输入为 1^k 时，计算 S_1，（n,e）∈I∩{0,1}^{k},得到相应的 d，使得：d·e≡1 mod φ (n)，算法选择两个比特数相同的随机素数，测试它们的素性，并且 n 为它们的乘积，那么，e∈ $Z_{\varphi(n)}$ 是随机选择的，最后，d 可以在多项式时间内计算φ（n）=(p-1)(q-1)，然后使用扩展欧几里得算法。对于条件 2，定义 S_2 在输入为（n,e）时随机产生 x∈Z_n^*，使得 A_1（（n,e）,d,$RSA_{n,e}$(x)）=注意 $RSA_{n,e}$(x)。模 n 求幂的多项式时间，因此条件满足。强 RSA 条件满足，这条件 4 满足。对于条件 5，使得 A_2（（n,e）,d,$RSA_{n,e}$(x)）≡ $RSA_{n,e}(x)^d \equiv x^{ed} \equiv x \bmod n$, 这是一个多项式时间计算。

一个 RSA 函数的特征在于有一个多项式时间算法对输入 x∈Z_n^*，以至少一个多项式时间来求 $RSA_{n,e}$ 逆，此外，对输入 x∈Z_n^*，以多个并行的多项式时间来求 $RSA_{n,e}$ 逆。这个对于给定的 n,e,如果函数难以求逆，而且，对于各个函数点是求逆困难的。

命题 2.31、假设ς,δ ∈（0，1）并且假设 S⊆I，存在一个概率事件 A，使得（n，e）∈S，Pr[A（n，e，$RSA_{n,e}$(x)）]〉 ς

（这是对 $x\in Z_n^*$，随机事件 A 之上的概率）A 在多项式时间|n|上运算。有一个概率事件 A'在多项式时间在 $\varsigma^{-1},\delta^{-1}$ 和|n|运算，对所有的（n,e）∈S，$x\in Z_n^*$，$Pr[A'(n,e,RSA_{n,e}(x))]\rangle 1-\delta$

（这里是随机事件 A'的概率）。

证明：选择满足条件的最小的 N，$1/e^N < \delta$,假设事件 A'使用如下的输入（n,e）∈S 并且 $RSA_{n,e}(x)$。重复 ς^{-1} N 次。

随机选择 $x\in Z_n^*$，假设 $y= A（n,e,RSA_{n,e}(x)\cdot RSA_{n,e}(z)）= A（n,e,RSA_{n,e}(xz)）$

如果 A 成功，那么 y=xz，因此，$x=yz^{-1} \bmod n$，输出 x。

否则继续下一个循环。

结束循环。

可以估计 A'失败的概率：$Pr[A'（n,e,RSA_{n,e}(x)）\neq x]=Pr[一个单圈的\ A'循环失败]\ \varsigma^{-1}{}_N$

$$< (1-\varsigma)\varsigma^{-1}{}_N$$

$$<(e^{-N})$$

$$<\delta$$

注意，既然 N=O（$\log(\delta^{-1})$）=O(δ^{-1}),A'在多项式时间在 $\varsigma^{-1},\delta^{-1}$ 和|n|运算的概率算法。

公开难题 2.32、剩余难题是决定如果概率对于在（n,e）∈I 上是否可以继续成立。特别的，如果 ε，δ∈{0，1}并且 A 是 PTM，那么，$Pr[A(n,e,RSA_{n,e}(x))=x]> \varepsilon$

（这里概率是在（n,e）∈I，$x\in Z_n^*$上的，投币事件为 A）是否存在一个 PTM A'在 ε^{-1}，δ^{-1} 的多项式时间内，使得：$Pr[A'(n,e,RSA_{n,e}(x))=x]>1-\delta$

（这里概率是在（n,e）∈I，投币事件为 A'）？

2.3.3 因式分解难题和 RSA 求逆之间的联系

公开问题 2.33、如果存在 n 多项式因子的 PPT 时间算法 A 分解，那么，存在 PPT 时间算法 A'求 $RSA_{n,e}$ 的逆运算。证明显然与 ϕ（n）=（p-1）(q-1)。单向陷门信息 d 使用扩展欧拉算法。因为 $d=e^{-1} \bmod \phi$（n）。

公理 2.34、如果存在一个 PTM 算法 B，对于已知<n,e>找到 d,使得 $ed\equiv 1 \bmod \phi$（n）。存在一个 PTM 算法 B'，完成 n 的因子分解。

公开难题 2.35、剩下的问题是因子分解是否与 RSA 逆运算难度相同。也就是，存在一个 PTM 算法 C，对于已知<n, e>，可以求 RSA 的逆运算，是否存在 PTM 算法 C'，求 n 的因子分解。对于这个问题的答案还不清楚。注意，2.34 的事实并不能暗示答案是肯定的，ε 也许存在一写其他的方法求 RSA 的逆运算，而且不需要找到 d。

2.3.4 Rabin 体制下的平方单向陷门函数

Rabin[164]的单向陷门函数，称为平方函数。平方函数除了 Rabin 体制之外，类似 RSA 体制的求逆计算难度是与因子分解难度相同的。因此，平方函数的逆是一个计算难题，与 RSA 求逆难度相同，概率难度相同。

定义 2.36、假设 I={n=pq:p 和 q 是不同的奇素数。}。对于 $n\in I$,平方函数 $SQUARE_N$ ：$Z_n^* \rightarrow Z_n^*$ 是使用，陷门函数 $n=pq\in I$ 是 $t_n=(p,q)$, 可以表示 Rabin 函数的集为：$Rabin=\{SQUARE_N: Z_n^* \rightarrow Z_n^*\}_{N\in i}$ 。

评注 2.37、当 Rabin 平方函数作为研究对象，RSA 函数使用指数变化，也就是，e 指数（e, ϕ（n））=1,能够保证 RSA 是置换时间函数。另外，Rabin 函数分为第一类和第四类，因此并非只有唯一的逆函数。特别的，使 $n=pq\in I$，使得 $a\in Z_n^*$,在第四节的讨论中，如果 $a\equiv x^2 \bmod p$, 然后，x 和 x^{-1} 构造模 p 的二次平方根，如果 $a\equiv y^2 \bmod q$, 然后，y 和 y^{-1} 构造模 q 的二次平方根。那么，对于 $a\equiv x^2 \bmod n$,有四个不同的解。假设 $c,d\in Z_n$ ，使用附录四中中国剩余定理的理论，化方程组为：

$$c=\begin{cases}1 \bmod p\\ 0 \bmod q\end{cases}$$

$$d=\begin{cases}1 \bmod p\\ 0 \bmod q\end{cases}$$

四个解分别为：cx+dy，-cx+dy，cx-dy，-cx-dy。

Rabin 定理最主要的结论为，它们是一个强单向函数的集，证明依赖于因数分解问题的相关假设。现在描述这个假设：

因式分解假设：假设 $H_k=\{pq: p 和 q 是素数，|p|=|q|=k\}$。那么，对于每个多项式 Q 和每个 PTM 时间事件 A，$\exists K_0$ 使得 $\forall k>k_0$, $Pr[A(n)=p : p|n$ 并且，$p\neq 1]<1/Q(k)$。（对于任意所有 $n\in H_k$ 上的随机概率, A 是随机概率事件。）的终极目标是证明如下的定理：

定理 2.38、在因子分解的假设下，Rabin 函数是一个单向陷门函数的集。

在证明这点之前，将先证明两个其他的推论：推论 2.39 构造了一个多项式时间机 A，计算一个模素数的平方根。推论 2.42 构造了另一个多项式时间机 SQRT，使用陷门函数求 Rabin 函数的逆。特别的，计算了一个使用因数的模合数的多项式平方根，SQRT 直接对应于 A。

推论 2.39、 假设 p 是一个奇素数，a 是模 p 的一个平方根，存在一个概率时间算法 A 在多项式时间内 A（p,a）= x, 这里 $x^2\equiv a \bmod p$ 。

证明：p 是一个奇素数，a 是一个 Z_n^*上的二次剩余，有两个可以考虑的条件为：$p\equiv 1 \bmod 4$，$p\equiv 3 \bmod 4$ 。

条件 1、$p\equiv 3 \bmod 4$，即：$p=4m+3$,m 为整数。

因为 a 是一个平方数，将努力找到一个奇指数 e，$a^e\equiv 1 \bmod p$ 。另外 a 是一个平方数，因此，$1=J_p(a)\equiv a^{(p-1)/2} \bmod p \Rightarrow a^{2m+1} \equiv 1 \bmod p \Rightarrow a^{2m+2} \equiv a \bmod p$,因此，a^{2m+1} 是 a 模 p 的一个平方根。

条件 2、$p\equiv 1 \bmod 4$，即：$p=4m+1$,m 为整数。

在条件 1，找到一个奇指数 e，$a^e\equiv 1 \bmod p$ 。另外，a 是一个平方数，因此，$1=J_p(a)\equiv a^{(p-1)/2} \bmod p \Rightarrow a^{2m} \equiv 1 \bmod p$ 。另外，对于这点，条件 1 上 a 的指数是偶数。注意：$a^{2m} \equiv 1 \bmod p \Rightarrow$

$a^m \equiv \pm 1 \bmod p$ 。如果 $a^m \equiv 1 \bmod p$ 当 m 是偶数时，执行条件 1。

这个建议在于，写 $2m=2^l \cdot r$ 这里，r 是一个奇整数，计算 $a^{2^{l-i}r} \bmod p$,对于 i=1,…,l,并且企图与 $a^r \equiv 1 \bmod p$ 然后按照条件 1 进行操作。但是应该存在一个整数 l’满足条件 $0 \leqslant l' < l$，使得 $a^{2^{l'}r} \equiv 1 \bmod p$ 。如果遇到合适的条件，可以如下进行改进。选择一个二次非剩余 $b \in Z_p^*$,那么，

$-1=J_p(b) \equiv b^{(p-1)/2} \bmod p$ 因此，$a^{2^{l'}r} \cdot b^{2^l r} = a^{2^{l'}r} \cdot b^{(p-1)/2} \equiv 1 \bmod p$。因此，用 $b^{2^l r} \equiv -1 \bmod p$ 乘以上述等式，得到一个新的等式：$(a^r b^{2^{l-l'}r})^{2^{l'}} \equiv 1 \bmod p$ 。继续这个合适的平方根。既然 $l'<l$ ，在 1 步后，得到 $a^r b^{2s} \equiv 1 \bmod p$，这里，S 是一个整数。基于这点，$a^{r+1}b^{2s} \equiv a \bmod p \Rightarrow a^{(r+1)/2}b^s$ 是 a 模 p 的平方根。

从上述的解释，得到一个概率算法 A 作为平方根。算法 A 使用 a,p 运算为 $J_p(a)=1$。

（1） 如果 $p \equiv 4m+3$ ， m 为整数，那么 a^{m+1} 作为模 p 的平方根。

（2） 如果 $p \equiv 4m+1$ ， m 为整数，随即选取 $b \in Z_n^*$,直到找到一个值满足 $J_p(b)=-1$

（1） 初始化 i=2m , j=0

（2） 重复，直到 i 是奇数

I←i/2 , j←j/2

如果 $a^i b^j = -1$ 那么 j←j+2m

输出 $a^{(i+1)/2}b^{j/2}$，是模 p 的平方根。

算法在步骤 2 进行了 O（l）次重复之后（II）a 的指数被 2 整除。注意，既然，Z_n^* 中的一半的元素是二次非剩余的。期待者 2 次反复的将找到适当的 b 的初始值，因此 A 在多项式时间的完成运算，证明了 2.39。

评注 2.40、有一个确定性的 Rene Schoof 算法（[179]），在多项式时间|p|对 a 模 p 的二次剩余计算平方根。并且 a（说明算法要求 $O((a^{1/2+\xi}\log p)^9)$对任意 $\xi>0$），然而，是否确定一个确定的多项式时间|p|。

公开难题 2.41、在多项式时间|p|的 a 模 p 的平方根，存在确定性的计算平方根？

下一个结论要求使用中国剩余定理。这个定理的描述在于在附录 C.4 是一个构造性的证明。另外，中国剩余定理的普通形式在这里表现。

引理 2.42、p,q 为素数，n=pq，a 是模 p 的一个平方数。存在一个概率算法 SQRT，希望在多项式时间内运算。SQRT（p,q,n,a）= x，这里，$x^2 = a \bmod n$ 。

证明：算法 SQRT 将与事件 A 相呼应，推论 2.39 的算法，得到模素数 p,q 的 a 的平方根，然后，合成这些平方根，使用中国剩余定理，得到要求的平方根。

算法 SQRT 如下运算：

（1） 假设 $A\{p,a\}=x_1$, $A(q,a)=x_2$.

（2） 使用中国剩余定理，在多项式时间找到 $y \in Z_n$, 使得，$y \equiv x_1 \bmod p$, $y \equiv x_2 \bmod q$, 输出 q。

SQRT 算法运算正常，因为：$y^2 \equiv \begin{cases} x_1^2 \equiv a \bmod p \\ x_2^2 \equiv a \bmod q \end{cases} \Rightarrow y^2 \equiv a \bmod n$。

另外，如果 n 的因子是已知的，那么计算 n 的平方根的难度是与 n 的因式分解的难度是相当的。证明下一个定理。

推论 2.43、计算模 n 平方根的难度与对 n 进行因子分解的难度是相当的。

证明：假设 I 是一个算法 $n\in H_k$,a 是输出为 y 模 n 的平方根，因此，$a\equiv y^2 \bmod n$, 考虑如下的算法 B，对于输入 n，输出 n 的非平凡因子。

（1） 随机选择 $x\in Z_n^*$

（2） 使 $y=I(n,x^2 \bmod n)$

（3） 检查是否 $x\equiv\pm y \bmod n$,如果不是，gcd(x-y,n)是 n 的非平凡因子。否则，从（1）重复算法 B 可以正确运行，因为 $x^2\equiv y^2 \bmod n \Rightarrow (x+y)(x-y)\equiv 0 \bmod n$ 。因此，n|[(x+y)(x-y)]。但是，n |≡(x-y),因为 x |≡ y mod n ,并且，n ∤ (x-y), 因为 x |≡ y mod n ,因此，gcd（x-y,n）是 n 的一个非平凡的除数。注意，$a\equiv x^2 \bmod n$ 有 0 或 4 个答案。在附录 C.4 上提供了一个结果的证明，因此，如果 I（n,x^2）=y,那么以 1/2 的概率，$x\equiv\pm y \bmod n$ 。因此上述算法在 2 次重复后结束。

现在处于证明主要结论的阶段，定理 2.38。

证明：对于条件 1，定义 S_1 发现 1^k 一个整数 n=pq，这里 p,q 是素数，并且|p|=|q|=k，陷门信息是素因数对（p,q）。

对于条件 2，搜集单向陷门函数集，定义 S_2 是在随机定义的 n 时简单输出 $x\in Z_n^*$。条件 3 是正确的，因为计算 $x^2 \bmod n$ 可以在多项式时间完成。条件 4 满足假设的因素和推论 2.42。

推论 2.43 是可以给出更强的结论，如果推论 2.43 的输入限制在一个更小的范围，仍然可以在多项式时间完成因数分解。建议 2.44:假设 $\varsigma,\delta\in$（0，1），并且假设 $S\subseteq H$,假设存在一个概率算法 I，使得所有 $n\in S$，Pr[I(n,a)=x 使得 $a\equiv x^2 \bmod n$] > ς 。（概率满足 $n\in S, a\in Z_n^*$,是 I 的随机事件）存在一个概率算法 FACTOR，以多项式时间 $\varsigma^{-1},\delta^{-1}$ 和|n|进行运算，对于所有 $n\in S$，Pr[FACTOR（n）=d 使得 d|n 并且 d≠1,n]>1-δ（这里 n 是以概率形式存在，并且，FACTOR 是随机事件）。

证明：选择最小整数 N，$1/e^N < \delta$,

考虑算法 FACTOR，在输入为 $n\in S$ 时，算法 FACTOR 按照如下方式运算：

重复 $2\varsigma^{-1}N$ 次。

随机选择 $x\in Z_n^*$

假设 $y=I(n,x^2 \bmod n)$

检查是否 $x\equiv\pm y \bmod n$,如果不是，那么 gcd（x-y,n）是一个 n 的非平凡除数。

结束循环。

可以估计 FACTOR 失败的概率。注意甚至当 I(n,$x^2 \bmod n$)产生 $x^2 \bmod n$ 的平方根，FACTOR 将使用一半的时间完成。Pr[FACTOR（n）求 n 的因子失败]=Pr[A 单圈 FACTOR 函数失败] $\varsigma^{-1}N$

$$<(1-\varsigma/2)^2\varsigma^{-1}N$$

$$<(e^{-N})$$

$$<\delta$$

既然，N=（$\log(\delta^{-1})$）=O(δ^{-1}),FACTOR 是一个概率算法以多项式时间 $\varsigma^{-1},\delta^{-1}$ 和|n|进行运算。

2.3.5 一个平方置换与求解因数分解难题等同

注意到早期的 Rabin 函数不是置换。如果 n=pq，这里 p,q 是素数，并且 P≡q≡3 mod 4, 那么可以把 Rabin 函数 $SQUARE_N$ 归约为一个置换 g_n , 限制它的定义域在 Z_n^* 的二次剩余的作用域 Q_n 上。这样产生一个单向置换的集，将在定理 2.3.5 提到这一点。这是 Blum 和 Williams 提出的。

定义 2.45、假设 J={pq：p≠q 是奇素数，|p|=|q|,并且，p≡q≡3 mod 4 }。因为，n∈J 假设函数 $g_n : Q_n \rightarrow Q_n$ 定义成为 $g_n(x) \equiv x^2 \bmod n$,使得 BLUM-WILLIAMS = $\{g_n\}_{n\in J}$ 。

将首先证明如下定理。

引理 2.46、每个函数 $g_n \in$ BLUM-WILLIAMS 是一个置换，也就是说，对每个元素 $y\in Q_n$,存在一个唯一的元，$x\in Q_n$ 使得 $x^2=y \bmod n$ 。

证明：假设 $n=p_1p_2 \in J$, 注意使用中国剩余定理，$y\in Q_n$ 当且仅当 $y\in Q_n$ 和 $y\in Q_{p1}$ ，$y\in Q_{p2}$,假设 a_i 和 $-a_i$ 是 y mod p_1 的二次平方根，i=1，2。使用中国剩余定理，构造中国剩余定理的系数 C_1 ， C_2 使得：

$C_1 = \begin{cases} 1 \bmod p_1 \\ 0 \bmod p_2 \end{cases}$ $C_2 = \begin{cases} 1 \bmod p_1 \\ 0 \bmod p_2 \end{cases}$ ，因此四个 y mod n 的平方根是：$W_1 = c_1a_1 + c_2a_2$, $W_2 = c_1a_1 - c_2a_2$, $W_3 = -c_1a_1 + c_2a_2$, $W_4 = -c_1a_1 - c_2a_2$

因为：p≡q≡3 mod 4 ，有整数 m_1 ,m_2，使得：$P_1 = 4m_1+3$, $P_1 = 4m_2+3$ 。因此，因为(p1-1)/2 是奇数，$J_{p1}(W_3)= J_{p1}(-W_1)= J_{p1}(-1)\,J_{p1}(W_1)=(-1)^{(p1-1)/2}\,J_{p1}(W_1)= -J_{p1}(W_1)$, 类似的，$J_{p1}(W_4)= -J_{p1}(W_2)$, $J_{p2}(W_3)= -J_{p2}(W_1)$和 $J_{p2}(W_4)= -J_{p2}(W_2)$, 不失一般性的，可以假设, $J_{p1}(W_1) = J_{p1}(W_2) = 1$（同样 $J_{p1}(W_3) = J_{p1}(W_4) = -1$。

既然，只有 W_1 ， W_2 是模 P_1 的二次剩余，其余的事情就是 W_1 ， W_2 只有一个是模 n 的平方剩余或者等价与模 P_2 。

首先观察，$J_{p2}(W_1)\equiv(W_1)^{(p2-1)/2} \equiv (c_1a_1 + c_2a_2)^{2m2+1} \equiv (a_2)^{2m2+1} \bmod P_2$ 并且 $J_{p2}(W_2)\equiv(W_2)^{(p2-1)/2} \equiv (c_1a_1 - c_2a_2)^{2m2+1} \equiv (-a_2)^{2m2+1} \bmod P_2$ （因为 $C_1 \equiv 0 \bmod P_2$, $C_1 \equiv 1 \bmod P_2$ ）, 因此，$J_{p2}(W_2)= -J_{p2}(W_1)$, 不失一般性，可以假设 $J_{p2}(W_1) = 1$ 并且 $J_{p2}(W_2) = -1$ ， 因此，W_1 是唯一的一个平方根，是 p_1 ， p_2 的平方根。因此，W_1 是 y 在 Q_n 中唯一的平方根。

定理 2.47[WILLIAMS, BLUM]、 BLUM-WILLIAMS 是一个单向陷门函数的集。

从引理2.46 立刻就可以得到该结论，因为每个函数 $g_n \in J$ 是一个置换，陷门信息为 n=pq, 为 $t_n = (p,q)$。

2.4 单向函数的核心难度谓词

记住，*f*(x)不必隐藏任何信息在 x 里，甚至当 *f* 是单向函数时，E.g。如果 *f* 是 RSA 函数，用来保护 Jacobi 函数 x, 如果 *f* 是离散函数 EXP，使用简单的勒让得符号可以容易的求出函数值。还有，至少有 x 中的一比特信息是难于从 *f*(x)中求出的。问题是，是否可以指出哪些比特位是从 *f*(x)中难于求取的，而完全还原 x 是更难的。此外的结论是使用部分 x 的信息泄露是难于还原 *f*(x)的，例如 RSA 机制或离散对数问题，这类问题的数量是已知的，会在随后的章节中讨论该问题。

更一般的，形成了一个关于 x 的谓词的集，从这个集中，对于从 *f*(x)已知的求逆值不会比随机猜测一个 *f* 的核心难度谓词容易。

首先看 Goldreich 和 Levin 的一般的结论，从任何一个单向函数 *f* 和随机判断 B，使得从

f 中猜测 B（x）函数与求 f 的逆函数是同等难度。

历史上的注记：Blum,Goldwasser 和 Micali 提出的单向函数的核心判断，首次出现在 Blum 和 Micali 关于随机数产生器的一篇文章中，表示 EXP 函数（$f_{p,g}(x)=g^x \pmod p$）是难于求逆的，当对于多数的 x，猜测的函数值是同函数值随机产生的概率一至。对于合数的模对二次剩余的假设与二次非剩余的假设是难度一至的，Goldwasser 和 Micali 说明了平方函数也是一个核心难度机制。后来，Yao 给出了一个一般的结论：对于任何给定的单向函数，存在一个谓词 B（x）,对于任意的 f(x),难于从 f(x)中进行猜测，也难于对 f(x)求逆。Goldreich 和 Levin 的模型结论是一个对于 Yao 的早期构造更简单的构造。

2.4.1 一般意义下的单向函数的核心判断

现在介绍一个函数的核心难度的概念，使用直接的构造方法可以使一个强单向函数变化成一个核心难度谓词。

注意：除非特别的提示，这一节内的概率问题就是根据问题中的算法从所有随机概率问题中求出单一的结论。

定义 2.48、一个函数 f 的核心难度谓词：f:$\{0,1\}^*\to\{0,1\}^*$ 是一个布尔函数的谓词，B：$\{0,1\}^*\to\{0,1\}$，使得：

（1）$\exists$ PPT 时间事件 A，使用的任意 x ,A（x）=B(x)

（2） $\forall$ PPT G，$\forall$ 常数 c，$\exists k_0$,s.t. $\forall_{k>k0}$

$$Pr[G(f(x))=B(x)]<1/2 + 1/K^e$$

G 是随机事件的概率，k 是随机选择的 x 的长度。

直观的，给定的定义是 x,B(x)计算有效的，但是如果只给定 f(x),必须给出 B（x）的猜证难度。也就是说，以好于 1/2 的概率猜证 B（x）。

Yao 在[208]中，指出了一些保留长度的置换的陷门函数的存在，这意味着陷门谓词的存在。Goldreich 和 Levin 大大简化了 Yao 的构造，表示任意单向函数可以由如下的随机陷门谓词更改（描述了通常结果的简单版本）。

定理 2.49[94]、假设 f(x)是一个（强）保持长度单向函数，定义 $f'(x\cdot r)=f(x)\cdot r$,这里，|x|=|r|=k,并且 • 是串联函数，那么 $B(x\cdot r)=\sum_{i=1}^{k} x_i r_i \pmod 2$

是 f'的核心谓词。

注意：v·w 表示 v 和 w 的串联，从 f’ 计算 B（x）是非常容易的，因为 f(x)和 r 是非常容易从 f'(x,r)中恢复的，最后注意到 f 是单向函数，那么 f’ 也是单向函数。

对于所有定理的证明，对读者提供[94]。

对一个单向函数集的核心难度谓词，一个单向函数扩展核心难度谓词是容易的。

定义 2.50、 单向函数集 $F=\{f_i : D_i \to R_i\}_{i\in I}$ 是核心难度谓词，是一个布尔谓词，$B=\{B_i : D_i \to R_i\}_{i\in I}$，使得，

（1）$\exists$ PPT 时间事件 A，使用的任意 x ,A（I,x）=B_i(x)

（2）$\forall$ PPT G，$\forall$ 常数 c，$\exists k_0$,s.t. $\forall_{k>k0}$

$$Pr[G(I,f(i))=B_i(x)]<1/2 + 1/K^e$$

G 概率事件是随机的，随机选择 $i\in I\cap\{0,1\}^k$ 和随机数 x 。

2.4.2 离散对数函数的 Bit 安全性

让直接检验 EXP 集的 Bit 安全性，而不是通过 Goldreich 和 Levin 进行普通构造。关注于对模 p 的离散对数 x,y 进行比特签名。

对于（p,q）∈I 并且 $y\in Z_p^*$,使得 $B_{p,q}=\begin{cases} 0 if y = g^x \bmod p 这里 0 \le x < (p-1)/2 \\ 1 if y = g^x \bmod p 这里 (p-1)/2 \le x < p-1 \end{cases}$

希望展示对 P 是素数，g 是 Z_p^* 生成元，$EXP_{p,q}$（x）$\equiv g^x \bmod p$ 是难于求逆的，那么，给出 $y=EXP_{p,q}(x)$,$B_{p,q}(y)$是以很强的判断其计算难度的。也就是说，试图计算 $B_{p,g}$ (y) 时，不会得到比随机计算其结果更好的结论。证明是归约的一中方法，如果可以计算, $B_{p,q}(y)$以 $1/2+\xi$ 的概率在多项式时间概率进行运算，这里，ξ 是不可忽视的量。可以|p|,|q|, ξ 来进行求 $EXP_{p,q}$ 的逆运算。

以下是关于这点的正式的描述。

定理 2.51、假设 S 是素整数的一个子集，假设有一个多项式 Q 和一个 PTM G，使得对所有素数 p∈S 和对于所有 Z_p^* 的生成元 g，$Pr[G(f(p,g,y))=B_{pq}(y)]<1/2 + 1/Q(|p|)$
（这里，$y\in Z_p^*$ 是概率和 G 是随机事件），这里对于每个多项式 P，有一个 PTM I 使得对所有的素数 p∈S 和对所有 Z_p^*的生成元 g，$y\in Z_p^*$。
$Pr[I(p,g,y)=x$ 使得 $y\equiv g^x \bmod p]> 1-1/P(|p|)$
（这里是随机事件 I 上的概率）
指出[40]是上述的一种证明。
作为一种推论，立刻得到如下定义 ：

定义 2.52 、 定义 $MSB_{p,q}$（x）=0，如果 1≤x<（p-1）/2 和 1，否则对于 $x\in Z_{p-1}$,并且 $MSB = \{MSB_{p,q}$ （x）$: Z_{p-1} \to \{0,1\}\}_{(p,q)\in I}$ 对于 I={(p,g) :p 是素数，g 是 Z_p^* 的生成元。

推论 2.53 、在强的 DLA，MSB 的 EXP 核心难度谓词的集。
可以看出，实际上计算 $x\in Z_{p-1}$ 的签名比特隐藏于 $EXP_{pq}(x)$运算量为 O（loglogp），不采取证明而描述下述的结论。

定理 2.54、对于一个 PTM 算法 A，假设

$$\alpha = Pr[A(p,g,g^x,x_{loglogp}x_{loglogp-1}\ldots x_0)=0|x=x_{|p|}\ldots x_0]$$

（这里，$x\in Z_p^*$ 是概率和 A 是随机事件）并且使得

$$\beta = Pr[(p, g, g^x, r_{loglogp}r_{loglogp-1} \ldots r_0)=0|r_i \in_R \{0,1\}]$$

（这里，$x\in Z_p^*$ 是概率和 A 是随机事件, 比特数为 r_i), 那么在离散对数假设的条件下，对于每个多项式 Q 和每个 PTM 事件 A，$\exists k_0$ 使得 $\forall$ $k>k_0$,| α - β |<1/Q(k)。

推论 2.55：那么在离散对数假设的条件下，对于每个多项式 Q 和每个 PTM 事件 A，$\exists k_0$ 使得 $\forall$ $k>k_0$,和 $\forall K_p < \log\log p$

$$Pr[A(p,g,g^x,x_{kp},\ldots x_0)=x_{kp+1}]<1/2+1/Q(k)$$

(概率计算是基于长度为 k 的素数 p, Z_p^* 的生成元为 g,$x\in Z_p^*$是概率和 A 是随机事件。）
对于同时发生，或与离散对数相关联的比特独立安全性的此外信息见[136，112]。

2.4.3 RSA 的比特安全和平方函数

假设$I = \{< n; e > | n = pq\ |p| = |q|; (e;\Phi(n)) = 1\}$，并且$RSA=\{RSA_{<n,e>}:Z_n^* \rightarrow Z_n^*\}_{<n,e>\in I}$ 是在2.17中定义的函数集合。

Alexi, Chor, Goldreich and Schnoor[6]给出了$RSA_{<n,e>}$（x）x的最低位猜测，比随机求RSA的逆难度小一些。

定理2.56[6]、假设$S\in I$，假设c>0，如果存在一个概率多项式时间算法O使得（n,e）$\in$S,

$$Prob\ (O(n,e,x^e \bmod n)= \text{最低n比特}\ \bmod n\ ) \geq 1/2 + 1/k^c$$

（在投币O中随机选择$x\in Z_n^*$）存在一个概率期望多项式时间算法A，使得对任意（n,e）$\in$S, 对所有$x\in Z_n^*$，$A(n,e,x^e \bmod n)= x \bmod n$。

现在定义$LSB=\{LSB_{<n,e>}: Z_n^* \rightarrow Z_n^*\}_{<n,e>\in I}$ 这里$LSB_{<n,e>}$（x）=是x的最低位。由A直接得出的推论为：

推论2.57、在强RSA假设条件下，LSB是RSA的核心谓词假设的一个集合。

一个类似的结果可以由高位直接得到。事实上，x的高位至多有loglog n个同时为x的位。更进一步类似的结果可以从RABIN 和 BLUMWILLIAMS 集合，可以从[6][205]得到更细致的结果。也可以从[84]得到进一步的安全归约。

2.5 单向和陷门谓词

一个单向谓词，最先在[101,102]中提出，是一个使核心难度谓词和单向陷门函数的建立强相关性的定义。对于设计安全加密算法和协议都非常有用。

单向谓词是一个布尔函数 B：$\{0，1\}^* \rightarrow \{0，1\}$，满足：

（1） 概率采样：存在一个 PPT 算法对于输入 1^k，$v\in\{0,1\}$，选择一个随机数使得 B（x）=v，并且$|x|\leq k$。

（2） 猜测难度：对于所有 c>0，对于所有足够大的 k，对于给定的 $x\in\{0，1\}^k$，没有 PPT 时间算法可以以大于 $1/2+1/k^c$ 的概率计算出 B（x ）。（这是中间人攻击随机使用的概率，x 满足$|x|\leq k$）。

一个陷门谓词是一个单向谓词，对于每个 k，陷门信息 t_k 限制在 k 范围的多项式内，并且这些知识能够在多项式时间内计算 B（x），对于所有 x, $|x|\leq k$。）

重复一个单向函数值和陷门谓词是容易的。

定义 2.58、 假设 I 是一个指数集，对于 $i\in I$ 使得 D_i 是有限域。一个单向谓词的集是一个集 $B=\{B_i :D_i \rightarrow \{0,1\}\}_{i\in I}$,满足如下的条件。假设 $D_i^v = \{ x\in D_i , B_i (x)=v\}$。

（1） 存在一个多项式 p 和一个 PTM S_1 ，对于输入 1^k, 找到 $i \in I\cap\{0,1\}^k$ 。

（2） 存在一个 PTM 算法 S_2 ，对于输入 $i \in I , v\in\{0,1\}$,找到 $x \in D_i$，使得 $B_i (x) = v$。

（3） 对于每个 PPT 概率事件 A，存在一个可忽略的量 V_A 使得足够大的 $\forall k$ ，

$$P[z=v:i\leftarrow I\cap\{0,1\}^k ; v \leftarrow \{0,1\}; x \leftarrow D_i^v ; z \leftarrow A(i,x)]\leq 1/2+V_A(k)$$

定义 2.59、 假设 I 是一个指数集，对于 $i\in I$ 使得 D_i 是有限域。一个陷门判断机制的一个集 $B=\{B_i : D_i \rightarrow \{0,1\}\}_{i\in I}$，满足如下的条件。假设 $D_i^v = \{ x\in D_i , B_i (x)=v\}$。

（1） 存在一个多项式 p 和一个 PTM S_1 ，对于输入 1^k,找到 $i \in I\cap\{0,1\}^k$ 并且$|t_i| <p(k),i$ 的陷门是信息 t_i。

（2） 存在一个 PTM 算法 S_2 ，对于输入 $i \in I , v\in\{0,1\}$,找到 $x \in D_i$，使得 $B_i (x) = v$。

（3） 存在一个 PTM 算法 S_2 ，对于输入 $i \in I$,陷门 $t_i , x \in D_i$ $A_1 (i,t_i,x)= B_i (x)$。

（4） 对于每个 PPT 概率事件 A，存在一个可忽略的量 V_A 使得足够大的 $\forall k$ ，

$$P[z=v:i\leftarrow I\cap\{0,1\}^k;\ v\leftarrow\{0,1\};\ x\leftarrow D_i^v;\ z\leftarrow A(i,x)]\le 1/2+V_A(k)$$

注意：定义按时 D_i^0 是粗略的同样的尺寸为 D_i^1 。

2.5.1 陷门集的例子

基于二次剩余假设的陷门谓词集

假设 Q_n 定义模 n 的二次剩余或平方的集，也就是说，$x\in Q_n$，如果存在一个 y 使得 $x\equiv y^2 \bmod n$ 。

回忆 Jacobi 符号（$(J_n(x))$）定义为任意 $x\in Z_n^*$ 并且有一个值{-1，1}。使用互二次规律这个值容易计算，因此 n 的因子分解是未知的。如果 n 是素数，那么，$x\in Z_n^* \Leftrightarrow (J_n(x))=1$，如果 n 是一个合数，$x\in Z_n^* \Rightarrow (J_n(x))=1$。使用 J_n^{+1} 定义一个集$\{x|x\in Z_n^* \wedge (J_n(x))=1\}$，假设 $\overline{Q}_x$ 表示模 n 的随机平方数：这些 J_n^{+1} 的元素，并且不属于 Q_n 。如果 n 是两个素数 p,q 的积，那么：$|\overline{Q}_n|=|Q_n|$，对于任意随机平方数 y，函数 $f_y(x)=y\cdot x$ ，使得一一映射为从 $\overline{Q}_n$ 到 Q_n 上。

二次剩余难题是：给定一个合数 n 并且 $x\in J_n^{+1}$，去决定 x 是否是一个模 n 的平方剩余或非平方剩余。这个问题是一计算困难问题，在定义域的每个地方都很难判断是否是平方数或非平方数。

定义 2.60[101,102]：假设 $S\subset\{ns.t.n=pq,p,q,素数\}$，如果存在一个多项式概率时间算法 O 使得 $n\in S$。

$$Prob(是否存在\ x\in J_n^{+1}\ ，\ 决定\ O(n,x)的正确性)>1/2+\xi \qquad (2.1)$$

在上述式子中，O 是随机选择，在 $x\in J_n^{+1}$ 的选择的概率。对 $x\in J_n^{+1}$ ，那么存在一个概率算法 B 在多项式时间 ξ^{-1},ζ^{-1} 和|n|使得 $n\in S$。

$$Prob(是否存在\ x\in Q_n|x\in J_n^{+1}\ ，\ 决定\ B(n,x)的正确性)>1-\zeta \qquad (2.2)$$

这个概率是在随机假设 B。

也就是，攻击者不在二次剩余困难的条件下的求取 $x\in J_n$ 是为模 n 的平方剩余或者是非平方剩余，在随机猜测的基础上的工作不可以再使一个概率多项式时间的边界更紧（除了比一个多项式时间更小的边界）。

这个建议立刻得到如下的谓词结构，使得：$QR_{n,z}(x)=\begin{cases}0\text{如果}x\text{是一个模}n\text{的平方剩余}\\1\text{如果}x\text{是一个模}n\text{的非平方剩余}\end{cases}$

这里，$QR_{n,z}: J_n^{+1}\to\{0，1\}$并且，$I_k=\{n\#z|n=pq,|p|=|q|=k/2,p$ 和 q 是素数，$(J_n(z))=+1,z$ 是模 n 的非平方剩余$\}$。明显的，$QR=\{QR_{n,z}\}$，是一个陷门谓词的集，陷门信息是与每个<n,z>的因数分解<p,q>索引相关联的。按照如下方法明确的进行检查。

（1） 选择一个随机数对<i,t_i>，首先选择两个素数 p,q，随机大小为|k/2|，决定 n。接下来，继续搜索，直到在 Z_n^* 中使用 Jacobi 函数+1 时，找到一个非平方根。使用的密钥对是（<n,z><p,q>。已经知道了所有运算期望在多项式时间内完成。

（2） 从 $D_{n,z}^v$ 中选择元素找到算法来进行如下运算：

选择 $D_{n,z}^0$ ，使得 $x=y^2 \bmod n$ 这里 y 是 Z_n^* 中随机选择的数。

选择 $D_{n,z}^{1}$ ，使得 $x=zy^2 \bmod n$ 这里 y 是 Z_n^* 中随机选择的数。

（3） 使用<p,q>计算 $QR_{n,z}$ （x）,计算（$J_p(x)$）和(x/q)，如果两个结果都是+1，那么，$QR_{n,z}$（x）是 0，否则是 1。

（4） 如下的结论基于二次剩余假设和上述的定理。

基于 RSA 假设的陷门谓词集

定义 $B_{n,e}(x)= x^d \bmod n$ 的至少比特表达，这里 $ed=1 \bmod \phi(n)$,选择一个 $x \in Z_n^*$ ，使得 $B_{n,e}(x)=v$ ， 简单选择 $y \in Z_n^*$ ，的比特表达是 v，假设 $x=y^e \bmod n$,给定 g，简单计算 $B_{n,e}(x)= x^d \bmod n$ 的至少比特表达。

这个构造的安全性从对 RSA 陷门体制核心难度谓词的集是构造的安全性是非常小的。

第三章 伪随机比特产生器

在这一章，讨论伪随机产生器的概念。直观上来看，PSRG 是一个由短随机序列产生固定的长序列的算法。

PSRG 的概念在如下几个方面得以应用

密码学：对于私钥加密，SHANNON 表示，明文的长度不能超过密钥的长度，也就是说，通信双方必须约定一个非常长的私钥，使用 PSRG 产生器 G，只需产生一个种子密钥，使用如下方法交换信息：G（r）⊕m。

算法设计：一个使用随机比特源的算法，可以使用一个短序列来作为 PSRG 的种子序列。

复杂性理论

给定概率论，一个重要的问题在于是否可以确定的知道该序列。使用可以证明的 PSRG 的定义，假设单向函数是存在的，$BPP \subseteq \cap_{\xi} DTIME(2^{n^{\xi}})$

在这一章，定义好的伪随机数产生器，在单向函数存在的假设下给出的构造。
首先要问是否存在真正的伪随机数序列。

3.0.2 产生一个一次一密序列（对任意密钥）

要求使用自然产生的序列，比如一个掷币实验，一个二进制噪音源，这样的伪随机源可以提供密码系统的初始化密钥。但是，许多自然噪音源被探测到输出存在一些偏差（也就是说输出比特的 0,1 是不平衡的）。或者比特是有相关性的。幸运的是，这些自然产生的噪音源是可以通过一些方式进行弥补的。使用不相关的比特流转化成一比特后，例如，von Neumann 设计的比特对产生器，01 到 0，10 到 1，然后，抛弃 00，11 比特流[206]。结果是一个不存在偏差的不相关的比特流，因为 01,10 是以相同的概率来产生的。Elias[79]产生了输出接近信息源熵的进一步的 IDEA。处理相关比特源更加困难。Blum[42]给出怎样根据已知的 MAKOV 链的产生的相关有偏差的输出怎样产生无偏差的非相关的乱源。

对于一个相关性更复杂的源，Santah 和 Vazirani[182]的模型构造是一个轻型随机源，每个输出比特通过一个掷币协议产生，这里一个攻击者允许选择哪个币产生协议。对于所有的币产生正面朝上的概率是否在δ和 1-δ之间（这里δ是一个很小的正数）。这是一个非常差的概率相关，但是，U.Vazirani[203]证明，如果有两个轻型相关的源 X，Y，那么，一个可以产生几乎独立的δ偏差比特，把输出 X，Y 分割成长度为 k= Ω（$1/\delta^2 \log(1/$ 。 $)\log(1/\xi)$比特的 x,y，每个分组对 x,y 输出比特数为 x·y(是 GF（2）上的内积)，这是一个使用非常确切的定义。Chor 和 Goldreich[54]给出了更一般的结果，怎样从更坏的随机源产生非独立的ξ-偏差，这里，一些输出比特可以被完全确定。

这些结果提供了一个从有缺陷的自然随机序列来产生真实的比特随机序列的有效方

法，这是密码学的本质的需求。

3.0.3 产生伪随机比特或数列

一次一密是非常不现实的因为有大量的密钥数据需要存储。事实上，一个人通常只希望存储短的随机密钥，使用适当的密码算法可以产生长的随机序列。算法是使用短的随机序列 x 产生长的密码序列 y，这样的密码算法叫做伪随机序列产生器。通常这种短的随机序列 x 叫做种子序列，y 叫做伪随机数而不是叫随机数，因为 y 序列并不是所有的输出都是随机的。可能的种子数量对于可能的 y 输出是随机的。虽然如此，目的是为了所有实际 y 与相同长度的真随机序列有所区别。

重要的是注意到使用伪随机序列发生器并不能消除自然随机源的需求。伪随机产生器实际上是一个随机扩展器，必须使用真的随机种子来进行扩展。

为了提高密码的安全水平，当使用一次一密的设计时，伪随机数发生器必须有如下特点：只获取 y 的部分的伪随机输出序列是不能预测其余的输出序列。例如，注意到一个中间人攻击，已知密文 C 可以猜测部分的 y 使用正确的相应的部分信息 M，例如一些标准的格式“your sincerely“等，但是不能象合法用户一样可以准确获得 M 其余的信息。最重要的是，中间人不能从输出 y 的片段推导出有效的种子 x 密钥。一个人怎样构造安全的伪随机发生器。

经典的伪随机发生器已过时了

经典的伪随机序列发生器[125，第三章]是非常有用的，对 Monte Carlo 的有效模拟对密码学的应用已经明显的不适应了。例如，线性反馈移位寄存器[108]对密码学而言明显是不安全的，可以使用少的输出比特来产生反馈。

合适的线性随机数发生器也是不安全的，这些发生器可以使用循环

$$X_{i+1} = aX_i + b \pmod m \qquad (3.1)$$

为了产生输出序列{X_0，X_1，…}使用秘密参数 a,b,m 和起始向量 X_0，通过已知的一些 X_i[163]向量推断秘密参数是可能的，Frieze, Hastad, Kannan, Lagarias, 和 Shamir 得出结论即使只知道非常小的部分连续输出序列，秘密参数未知时，也可以求出初始值 X_0。Lenstra, 和 Lovasz 的算法[87,132]使用基于非平凡格归约（或者 L^3）的方法也可以求出 X_0。

现代的密码学例子是不适合经典的方法的。Kannan, Lenstra, 和 Lovasz[122]的使用 L^3 的算法，任何代数数 y（例如 $\sqrt{5}$ = 10.001111000110111…）的二进制扩展是不安全的，既然一个攻击者可以从足够的比特确定 y 的值，然后可以轻易地推断 y 的扩展。

3.0.4 伪随机的可证安全：概论

本节对现代密码学伪随机比特扩展的历史提供了一个简单的概述。最先提出一部分伪随机序列产生器，从前已知的部分序列可以求出随后的比特信息，从 Shamir[189]的观点，求 RSA 的逆是不可能的。然而，这个设计提供了一个数字序列的产生机制，而不是一个比特序列的产生机制。安全证明表示，从已知道的数字信息中，攻击者是不能预测下一数字的信息的。不是特别的强的指示，当使用一次一密的机制时，每个比特的信息将被妥善的保护。

Blum 和 Micali[44]介绍了一个方法，用来设计可证明伪随机安全的比特序列产生器，

基于单向函数判断机的研究。称伪随机比特安全，如果攻击者不能通过随机猜证并且使用之前的序列信息来求取随后的比特流。Blum 和 Micali 根据离散对数的计算难度提出了一个特别的生成器。Blum, Blum 和 Shub[35] 提供了一个生成器，称做平方生成器，该算法实现简单，基于二次剩余难度的问题是可证安全的。Alexi , Chor, Goldreich 和 Schnorr[6] 给出了假设，基于二次剩余难度问题的难度可以弱一些的难度问题来代替。相关的产生器可以根据 RSA 机制产生。Kalishi 认为怎样给出在这些方法归约出的一些难题，例如基于椭圆曲线难题问题的产生器。提出的技术也可以产生相关一些群[120，121]。Yao[208]证明了基于上述难题的伪随机产生器是完美的。也就是说，不能使用一个随机偏差很小的量在多项式时间进行估计输入 1^k，通过上述难题产生的还是从{0，1}^{k} 选择的真随机数。攻击者可以在某种程度上改写下一比特信息，而通过所有多项式时间的测试。Blum 和 Micali，Blum, Blum 和 Shub 提出的算法，加上 Yao 的证明，代表了可证安全密码系统的一个主要进展。同时，Impagliazzo, Luby, Levin 和 Hastad 的工作说明了单向函数的存在是等同于通过所有多项式时间测试的伪随机比特序列存在。

3.1 定义

定义 3.1、假设 X_n, Y_n，是在{0，1}^{n} 上概率分布的，（也就是说，使用 $t \in X_n$ 意味着 $t \in$ {0，1}^{n} ，根据 X_n 的随机分布选择 t）。假设{X_n}是一个用多项式时间从{ Y_n }中获得的序列，如果∀多项式 Q，∃ n_0 ,s.t ∀PTM A，∀ $n>n_0$ ，

$$|\Pr_{t\in X_n}(A(t)=1)- \Pr_{t\in U_n}(A(t)=1)|<1/Q(n)$$

对于足够长的序列，从多项式时间算法不能推测出 PTM 是 X_n 的采样，还是 Y_n 的采样。

直观的，伪随机分布是与均衡的分布有区别的，定义均衡的概率分布是 U_n 上{0，1}^{n} 的分布。也就是说，对于任意的 $\alpha \in$ {0，1}^{n} , $Pr_{x\in Un}[x=\alpha]=1/2^n$

定义 3.2、称{X_n}是伪随机的，如果它是从 U_n 从多项式时间不可分辨机的。也就是说，∀PTM A，∀多项式 Q，∃ n_0 使得 ∀ $n>n_0$ ，

$$|\Pr_{t\in X_n}(A(t)=1)- \Pr_{t\in U_n}(A(t)=1)|<1/Q(n)$$

评论：算法 A 对如上的定义叫做多项式时间测试。（knuth vol 2 是各种测试例子的建议）。重要的是这种类型的注记不包括单一的序列，即使是从其他的分布中采样出来的。
如果∃A 和 Q 使得定义 2 中的条件是不满足的，说 X_n 对 A 测试失败。

定义 3.3、一个确定的多项式时间程序 G：{0,1}^{k} →{0,1}^{k} 是一个伪随机产生器（PSRG），如果满足如下的条件：

1. $\hat{k} > k$

2. {G$\hat{k}$ } $\hat{k}$ 是一个随机数，这里，G$\hat{k}$ 是 {0，1}$^{\hat{k}}$ 上的分布，并且得到如下的定义：得到 $t \in G\hat{k}$ ，选择 $x \in U_n$,使得：t=G(x)

也就是说，∀PTMA ，∀多项式 Q，∀足够大的数 k，

$$|\Pr_{t\in G_k}(A(t)=1)- \Pr_{t\in G_k}(A(t)=1)|<1/Q(\hat{k})$$

3.2 一个伪随机发生器的存在性

接下来，证明 PSRG's，如果保持长度的单向置换存在，则单向函数存在，但不在这里讨论该问题：

定理 3.4、 假设 f：$\{0，1\}^* \to \{0，1\}^*$ 是一个保持长度的单向置换函数。那么：

1、∃PSRG G: $\{0，1\}^K \to \{0，1\}^{k+1}$, (因此 G 称为一个扩展函数)

2、∀ 多项式 Q，∃PSRG G^Q ：$\{0，1\}^K \to \{0，1\}^{Q(K)}$

证明：（方法 1)：假设 f 如上定义， B 是 f 的核心难度比特谓词，（也就是说 B 是一个布尔的谓词，B：$\{0，1\}^* \to \{0，1\}$，对给定的 x 是计算有效的，但是如果只知 f(x)，在多项式时间内，以可忽略的偏差 ξ 在 1/2 附近进行求 B 是很难的)。回忆上一节已得到的一些结论可以看出每一个 OWF f(x) 可以求逆 f_1(x,r) ，这里，B'（x, r）= $\sum_1^{|x|} x_i r_i \mod 2$ 是一个核心难度谓词。为了简化符号，假设 B 是 f 的核心难度谓词。

定义：G^1 （x）= f (x) • B(x) (• 定义串操作)。将定义 G^1（x）有可以满足的特性。清楚的 G^1 是在多项式时间内可计算的，对于|x|=k，$|G^1$（x）| = k+1，其余的事就是必须证明 $\{G^1_{K+!}\}$是伪随机的。

直观的：事实上，已知 f(x) 应该帮助预测 B（x)。

因为，f 是一个置换，f（U_k）是一个$\{0，1\}^k$ 上的函数簇。任何分辨机 $G_{K+!}$ 和 U_{K+1} 是核心难度的。接下来会证明，任何分割方式会在只知道 f(x)时推出 B(x)，这与从 f 得到一个 B 是相互矛盾的。

使用这个矛盾证明，假设 G 定义不是那么好，那么，∃ 一个统计测试 A，多项式 $Q_{s,t}$

$$\Pr_{t\in G_{K+1}}(A(t)=1) - \Pr_{t\in U_{K+!}}(A(t)=1) < 1/Q(k+1)$$

(注意，在不等式 3.2 中使用的是绝对值，这个可以用 WLOG 来度量，随后可以看到，其方向的不等式是真实存在的。)

直观的，可以翻译成为如果 A 在一串序列中的答案是 1，那么，更大可能的是 t 随后从 $G_{K+!}$ 中获得信息，而如果得到的结果是 0，那么，t 随后的信息将随后从 U_{k+1} 中获得。注意到概率 A（f(x)•b）返回 1，是概率重量的总和，A 返回 1，对有条件的 B（x）=b 并且还有一个条件是 B(x)=1。使用上述的假设，认为 A（f(x)•b）将返回 1，当 b=B（x)，这是非常容易把算法转化为 f(x)的核心难题谓词。

最后，有如下结论：

$$\Pr_{x\in U_k, b\in U_1}[A(f(x)\circ b)=1] = \Pr[A(f(x)\circ b)=1=B(x)]\cdot\Pr[b=B(x)]$$
$$+\Pr[A(f(x)\circ b)=1 \mid b=\overline{B(x)}]\cdot\Pr\{b=\overline{B(x)}\} = 1/2(\alpha+\beta)$$

这里，$\alpha = \Pr[A(f(x)\circ b)=1 \mid b=B(x)] and \beta = \}r[A(f(x)\circ b)=1 \mid b=\overline{B(x)}]$

从假设，可以看出，

$$\Pr_{x\in U_k}[A(f(x)\circ B(x))=1] - \Pr_{x\in U_k}[A(f(x)\circ b)=1] = \alpha - 1/2(\alpha+\beta)$$

$$= 1/2(\alpha+\beta) > 1/Q(k)$$

现在给出一个多项式时间算法 A'，以优于 1/2 的概率对于输入 f(x)计算 B(x)。

A’使用 $f(x)$为输入，以 0，1 为输出。

1、 选择 b∈{0,1}

2、 运算 A（$f(x)\cdot b$）

3、 如果 A（$f(x)\cdot b$）=1 则输出 b,否则输出 $\bar{b}$ 。

注意到，从不等式 3.2 得到的是绝对值的式子，只要对输出情况进行对应的调整，就可以得到另外一种定义。

命题 3.5、 $\Pr[A'(f(x)=B(x))]>1/2+1/Q(k)$

证明：
$$\Pr[A'(f(x)=b]\Pr[A(f(x)\circ b)=1\mid b=B(x)]\Pr[b=B(x)]$$
$$\Pr[A(f(x)\circ b)=0\mid b=\overline{B(x)})\Pr[b=\overline{B(x)}]+\alpha/2+(1-\beta)/2$$

$=1/2+(\alpha-\beta)/2>1/2+1/Q(k)$

这与计算 B（x）的难度是相矛盾的。随后证明 G^1 需要一个 PSRG 跟随。

证明 2：给定一个 PSRG G，把长度为 k 的随机数扩展成为 k+1 的随机数。需要证明的是，∀多项式 Q，∃PSRG G^Q ：{0，1}→{0，1}$^{Q(K)}$ ，在第一次使用 G 使定义 G^Q ，Q（k）时间是如下定义的：

$X \rightarrow G \rightarrow f(x)\cdot B(x)$

$F(x)\cdot B(x) \rightarrow G \rightarrow f(f(x))\cdot B(f(x))$

$F^2(x)\cdot B(f(x)) \rightarrow G \rightarrow f^3(x)\cdot B(f^2(x))$

.

.

.

$F^{Q(k)-2}(x)\cdot B(f^{Q(k)-1}(x)) \rightarrow G \rightarrow f^{Q(k)}(x)\cdot B(f^{Q(k)-1}(x))$

$G^Q(x)$的输出为每一串的最后几比特串联的，

$$G^Q(x)=B(x)\cdot B(f(x))\cdot\ldots\cdot B(f^{Q(k)-1}(x)=b_1^G(x)\cdot b_2^G(x)\cdot\ldots\cdot b_{Q(x|)}^G(x)$$

明显的，G^Q 是一个多项式时间函数，并且满足长度要求。需要证明使用多项式时间算法 G^Q，$G^Q(U_k)$ 是从 $U_{Q(k)}$ 是多项式时间不可分辨机的。可以看到矛盾（并且暗示着算法 G 不是 PSRG）。如果 G_{QK} 不是 $U_{Q(k)}$ 多项式时间可分辨机的，∃统计测试 A 和多项式 P，s,t

$$\Pr_{t\in G_Q(k)}(A(t)=1)-\Pr_{t\in U_{Q(k)}(k)}(A(t)=1)>1/P(k)$$

（与上述讨论相同，省略了绝对值）。证明了分布在{0，1}$^{Q(k)}$ 序列 $D_1\,D_2\ldots D_{Q(k)}$, s.t, D_1 是均衡的（序列是随机的），$D_{Q(k)}=G_{Q(k)}$ ，中间的元素是伪随机分布是如下随机串联分布得到的，特别的：

$T\in D_1$ 使 t=s, 这里 $s\in U_{Q(k)}$

$t\in D_2$ 使 $t=s\cdot B(x)$ $s\in U_{Q(k)-1}$, $x\in U_k$

$t\in D_3$ 使 $t=s\cdot B(x)\cdot B(f(x))$ $s\in U_{Q(k)-2}$, $x\in U_k$

•

•

$T\in D_{Q(k)}$ 是使 $t=B(x)\ldots B(f^{Q(k)-1}(x))$ 这里 $x\in U_k$

既然序列从 $D_1=U_{Q(k)}$ 到 $D_{Q(k)}=G_{Q(k)}$ ，并且有一个算法 A 与是可分辨机的 i.e。∃i, $1\le i\le Q(k)$,s.t

$$\Pr_{t\in D_i}(A(t)=1)-\Pr_{t\in D_{+1i}}(A(t)=1) > 1/P(k)Q(k)$$

现在找到了一个与 1/2 有显著偏差的，在 U_{k+1} 和 G_{k+1} 之间的多项式时间算法 A'。这与 G 和 PSGR 相矛盾。

A'使用输入 $\alpha=\alpha_1\alpha_2\cdots\alpha_{k+1}=\alpha'\circ b$

1、随机选择 1≤i≤q(k)

2、假设

$$t=\gamma_1\circ\cdots\circ\gamma_{Q(k)-i-1}\circ b\circ b_1^G(\alpha')\circ b_2^G(\alpha')\circ\cdots\circ b_i^G(\alpha')$$

这里 γ_j 是随机选择的。

（注意 $t\in D_i$,如果 $\alpha'\circ b\in U_{k+1}$，并且，$t\in D_{i+1}$ 如果 $\alpha'\circ b\in G_{k+1}$）

3、现在运行 A（t）,如果得到 1，A'返回 0，如果得到 0，A'返回 1。

（i.e. 如果 A 返回 1，对 D_i 的投票做为一个解释，因此，b≠B(α')并且 $\alpha\in U_{k+1}$,另一方面，如果 A 返回 0，作为 D_{i+1} 的一个解释。因此，对于 b=(α') 并且 $\alpha\in G_{K+!}$ ）。显然可以得到如下结果：

$$\Pr_{\alpha\in D_i}(A(t)=1)-\Pr_{\alpha\in D_{+1i}}(A(t)=1) > 1/P(k)Q^2(k)$$

多余的一个 1/Q（k）因子与 i 的随机选择是相关的。这个与在第一部分证明的 G 是伪随机产生器是相违背的，这是一个矛盾。

3.3 下一比特测试

如果一个伪随机序列具备一个特性：从已知的比特序列不能以多项式时间在可忽略的偏差范围内 ξ 以最精确度来猜测下一比特，就称生成序列比特通过下一比特测试。

定义 3.6、下一比特测试是一个特殊的随机数测试：输入一个前向序列，输出一个下一比特测试。

定义 3.7、 在 S 集上的（离散）概率分布是一个函数，D：S→[0，1]⊂ R，使得 $\sum_{s\in S}D(s)=1$。简而言之，$\{0,1\}^k$ 上的概率分布表示对一个 k 的注解。$X\in X_n$ 意味着 x 满足下列等式，$\forall z\in\{0,1\}^n\Pr[x=z]=X_n(z)$。在这个定义下，$U_n$ 是均匀分布。

复习伪随机序列产生器的定义

定义 3.8、一个伪随机数产生器（PSRG）是一个确定的多项式时间算法，使得：

1、如果|x|=k,那么|G（x）|=$\hat{k}$

2、$\hat{k}$ >k

3、$G_{\hat{K}}$ 是伪随机平方数，这里$G_{\hat{K}}$ 是包括{0，1}^{k} 上的 G 函数的概率分布

定义 3.9、说伪随机序列可以通过 A 的下一比特测试，如果每个多项式 Q 存在一个整数 k_0，$\hat{k}$ >k 并且 p<$\hat{k}$ ，

$$\Pr_{t\in G_{\hat{k}i}}[A(t_1t_2\ldots t_p)=t]>1/2+1/Q(k)$$

定理 3.10、G 通过所有下一比特测试⇔G 通过所有统计测试

⇐ 显然

⇒假设与所求相矛盾，G 通过了所有下一比特测试，但是不满足部分的统计测试。用事件 A 统计测试的事件集合 A’，在概率分布上定义一个运算 Θ，使得[X_n Θ Y_m]=$X_n(z_n)\cdot Y_m(z_m)$，这里，$z=z_n\cdot z_m$，$|z_n|=n$, $|z_m|=m$,(·是一个串联)。对于 j≤$\hat{k}$，假设$G_{j\hat{K}}$ 是由$G_{\hat{K}}$ 在{0，1}^{j} 上导出的概率分布，$G_{j\hat{K}}$（x）=$\sum_{z\in\{0,1\}^k,z\text{扩展}x} G_k(z)$。

定义一个{0，1}$^{\hat{k}}$ 随机分布序列 H_i = $G_{j\hat{K}}$ Θ $U_{\hat{n}}$ 是一个上升伪随机序列。那么，$H_0=U_{\hat{n}}$ 并且$H_{\hat{K}}=G_{\hat{K}}$，因为 G 对于事件 A 是失败的，A 可以在 $U_{\hat{k}}=H_0$ ，$U_{\hat{k}}=H_{\hat{k}}$，也就是说，∃ Q∈Q[x]使得，A（t）=1,当从$U_{\hat{n}}$ 选择了 t（否则，输出 A 的逆）。这样，可以不再讨论不等式的绝对值的情况。那么，∃i,0≤i≤$\hat{k}$ -1，使得 $\Pr_{t\in H_i}[A(t)=1]-\Pr_{t\in H_{+1i}}[A(t)=1]>1/\hat{k}Q(k)$。

下一比特测试 A’的对应选取是：$t_1t_2\ldots t_i$，下一比特猜测值是 t_{i+1}，A’的最初的构造是：

$S_0=t_1t_2\ldots t_i0r_{i+2}r_{i+3}\ldots r_k$

$S_1=t_1t_2\ldots t_i1r_{i+2}^{'}r_{i+3}^{'}\ldots r_{,}^{'}$

这里 r_j 和 $r_j^{'}$是随机比特，对于任意 i+2≤j≤$\hat{k}$。A’然后计算 A（S_0）和 A（S_1）。

如果 A（S_0）= A（S_1），A’输出一个随机比特，

如果 0= A（S_0）=$\overline{A(S_1)}$，那么 A’输出 0，

如果 1= A（S_0）=$\overline{A(S_1)}$，那么 A’输出 1。

命题 3.11、分析了之前进行的研究的相似性，$\Pr[A'(t_1t_2\ldots t_i)=t_{i+1}]>1/2+1/kQ(k)$。这样，得到了一个矛盾，A’是下一比特测试中使 G 失败的事件集，这与的 G 可以通过所有测试的假设相矛盾。

3.4 伪随机数产生器的例子

讨论了一个伪随机数发生器的每一个单向函数。以下列出的是这些函数（包括随后将讨论的 Blum/Blum/Shub 数）以及消耗[44，39，176]。

名称	单向函数	计算单向函数时间	计算第 j 比特生成 n 元的时间

RSA	$x^e \bmod n, n=pqq$	K^3	Jk^3
Rabin	$x^2 \bmod n, n=pq$	K^2	Jk^2
Blum/Micali	EXP(p,g,x)	K^3	Jk^3
Blum/Blum/Shub (如下)		K^2	$Max(k^2 logj, k^3)$

3.4.1 Blum/Blum/Shub 伪随机发生器

Blum/Blum/Shub 伪随机发生器使用了单向函数 $g_n(x)=x^2 \bmod n$ 这里 n=pq，p，q 均为素数，并且 $p \equiv q \equiv 3 \bmod 4$ 。这样，平方映射是 Z_n^* 限制在（Z_n^*）^{2} 上：$x \to x^2$，（如果一个平方数都有唯一的平方根）

命题 3.12、 x 的签名比特应该至少是一个单向函数的难度最强的比特。

Blum/Blum/Shub 的 j^{th} bit 通常由下述方法产生：B（$x^{2^j} \bmod n$ ）= $B(x^\alpha \bmod m)$

这里，$\alpha \equiv 2^j \bmod \phi(n)$。如果 n 在因子分解时是已知的，难么 $\phi(n)$=（p-1）(q-1)可以容易计算出来，这样 α 可以在求幂之前计算出来。$\alpha \equiv 2^j \bmod \phi(n)$可以在 O（$k^2 logj$）时间内计算，并且 x^α 可以在 K^3 时间内计算，因此，计算 B（x^{2^j} ）花费时间为 O（max（$k^2, k^2 logj$）。

Blum/Blum/Shub 序列产生器的一个有趣的特征是：如果 n 的因式分解已知，那么 $2^{\sqrt{n}th}$ 比特可以在多项式时间|n|内产生。接下来产生的问题在于：假设 $G^{BBS}(x,p,q)=B(f^{2^{\sqrt{n}}}(x)) \circ \cdots \circ B(f^{2^{\sqrt{n}}+2k}(x))$，这里，n=pq,|n|=k，假设 G_{2k}^{BBS} 是 G^{BBS} 在$\{0,1\}^{2k}$上的导出的分配。

公开难题 3.13、分布 G_{2k}^{BBS} 是伪随机吗？是否可以证明：

$\forall Q \in Q[x], \forall PTM\ A, \exists k_0, \forall k > k_0 | Pr_{t \in \frac{BBS}{2k}}[A(t)=1] - Pr_{t \in U_{2k}}[A(t)=1]| < 1/Q(2k)$

先前的证明了：G 是伪随机的，在这个条件下，n 的因子分解是种子的一部份。与关于因子分解的难度显然是没有矛盾的。更一般的，有结论：

公开难题 3.14、给定种子 x，伪随机产生器定义了一个有限域链，$g_1^x g_2^x \ldots$,找到了一个伪随机发生器，使得选择决定限制何种多项式时间，可以使得 g_i^x 的比特的子集是伪随机的。对于多项式机制的选择意味着可以把 g_i^x 看做在多项式时间机对于一个 i 的多项式时间（机制必须写成 i 的多项式，把 i 限制在|x|的多项式上）。

第四章 分组密码的运算模式

分组密码是设计秘密共享密钥协议的核心工具。主要用到的是设计的技术，这一章对这些概念进行简述，在构造上进行描述。

重要的是需要强调分组密码是正确的工具。不会只关心怎样为用户的使用方便而进行设计。作为一个有利的工具，一方可以学习去使用一个算法 。如果一个设计安全的分组密码算法都不能保证安全，那需要研究一下的使用方法是否合适。如果使用得当，这确实是一个强大的工具。根据前几章重要的几个概念可以确定怎样正确的使用分组密码算法。不去强调是否去设计或分析一个分组密码算法，这一章主要关注的是典型的分组密码算法是什么，会看两个例子 DES 和 AES，DES 已经逐渐被淘汰了，AES 是逐步代替 DES 的分组密码算法。

4.1 什么是分组密码

分组密码是一个函数，E：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$,使用 2 个输入, 一个 k 比特的密钥 k，和一个 l 比特的明文为 M，输出 l 比特密文 C，C=E（K，M）。密钥长度 K 和分组长度 l 是联合的参数，与分组密码算法本身相关或者从密码到密码变化。对每一个 $k \in \{0,1\}^k$,假设 E_k：$\{0，1\}^l \rightarrow \{0，1\}^l$,是由 $E_k(M)=E(k,M)$定义的。对许多分组密码，任何密钥 k，函数 E_k 是一个$\{0，1\}^n$ 上的置换。这意味着是一个双射，是一个单射并且是$\{0，1\}^l \rightarrow \{0，1\}^l$ 上的满射。根据它有一个逆函数，可以定义为 E_k^{-1} ,这个映射也是$\{0，1\}^l \rightarrow \{0，1\}^l$ 上的映射。因此：$E_k^{-1}(E_k(M))=M$，$E_,(E_k^{-1}(C))=C$，对所有 $M,C \in \{0,1\}^l$， 假设 E^{-1}：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ 被定义为 $E_k^{-1}(C)=E^{-1}$（K,C）,这是一个 E 的逆函数。

分组密码是公开的被详细描述的算法。密码算法 E 和 E^{-1} 容易计算，意味着给定 M，K，可以计算 E^{-1}（K，C）。

一个典型的使用为：选择一个随机密钥 k 并且在使用者间保持秘密。函数 E_k 使用两方去在互相发送前在某种程度上要处理这些数据。攻击者可以看做 E_k 的例子是输入–输出对，记做（M，C），$C=E_k$（M），但是 k 并没有直接给出。这个密码体制的安全依赖于密钥的安全（这是对安全的基本要求，但不一定就可以保证安全）。最起码，可以看到攻击者的目标是通过密文恢复密钥，分组密码算法的设计可以保证求取的计算困难。随后，将提炼这个观点，但是这是最古典的一种方式，会继续讨论。

分组密码是怎样工作的？了解一些基本的看法。

4.2 数据加密标准、数据加密标准

数据加密标准（Data Encryption Standard，DES），是一个分组密码算法的精华。但是现在已经可以感觉它逐渐过时了，另外，但是对所有分组密码算法都不能忽视这种结构。DES 是一个典型的好的算法机制，对密码学的发展有深远的影响。DES 被广泛应用，也许还会再使用几年，每一次你使用 ATM 机，你就使用了 DES 算法。

4.2.1 历史简介

1972 年，NBS（现在的 NIST，国家科学和技术局，以前的国家安全标准局），数据保

护的初始化程序，一个加密算法是可以标准化。他们对这样的算法作出请求。在 1974，IBM 提出了一个基于“lucify ”结构的算法，这个设计最终成为了 DES 算法。

DES 有一个 k=56 的密钥长度，一个分组长度为 l=64，是一个叫 Feistel 的 16 圈的结构。将简单介绍具体结构。

在 NBS 之后，有其他几个机构采用了 DES 作为标准算法，包括美国国家标准机构（the American National Standard Institute, ANSI），和美国银行协会（American Bankers Assiciation, ABA）。

标准会每隔五年来发现是否该算法会被继续使用。虽然有一些算法没有再被确定，但有一些继续被确定使用。直到最近 AES 算法被提出作为 DES 的替代算法。

DES 被证明在当时的计算方法条件下是可保证安全的。从最开始，特别对于穷尽密钥搜索。但是对于给定一个时间长度对，56 比特的密钥长度足够强度来抵抗非常多的资金支持的攻击。

4.2.2 构造

在 FIPS[147]描述了这种结构，接下来如果手边有 FIP 的文件，你可以按照这个文件快速得到指导。

从 87 页开始，你可以看到一个大的图片。输入分组为 64 比特，密钥长度为 56 比特（他们说 64 比特，每八比特有一比特的冗余信息，并且可以从前七比特求取第八比特信息。）注意算法是公共的，你是使用秘密密钥进行运算的，同时算法是隐藏的。

首先的是最先置换或 IP 层的运算，这些提供了比特混乱的位置。也就是说，每个比特都必须移动到不同的位置，怎样做到这一点？使用一个固定和特殊的方法。同样的，恰好在最后，他们提供了同样置换的逆。从现在开始，忽略这些。他们确实不影响安全。（就任何一个人可以告知的结果，他们可以容易的载入芯片）。

DES 的实质是在圈函数，有 16 圈这样的圈函数。每个圈数为 i 的子密钥为 k_i，长度为 48bit，子密钥 $K_1,\ldots, K_{16}$ 从密钥 K 中获得。在文件 FIP 的 95 页中也得到了很好的解释。在每一圈，输入为 32 比特（L_i,F_i）对，这些信息会被传入下一圈的输入对（L_{i+1},F_{i+1}）。通过一个固定的函数 *f*, 依赖于 i 相关的圈密钥 k_i。这个替换函数的结构是非常重要的：叫做 Feistel 结构。

通常，Feistel 结构是这样的。对于一些函数 g 已知部分的函数结构，输入（L,R）并且返回(L’,R’), 这里，L’=R，R’=g(R)⊕L。这个变换的中心特点为：首先，这是一个置换，此外，如果你可以计算 g 那么你可以求变换的逆。事实上，给出(L’,R’)可以求出（L,R），通过计算 R=L’，R=g(R’)⊕L。使用 DES，在 i 圈的 g 函数的角色使用为 $f(k_i,\cdot)$，圈函数可以使用子密钥 K_i ,既然 DES_k （·）是 Feistel 网络的变换，每一个变换都是置换。整个算法都是一个置换。通过密钥 K 的知识可以计算 DES_k^{-1}（·）。

目前这个结构已经非常普及了，许多分组密码算法都使用了这种高水平的结构。Feistel 圈函数的序列。此外的观察为需要看 $f(K,\cdot)$函数是怎样工作的。在 FIPS 的 90 页文件插图。这里 K_i 是 48 比特子密钥，从 56 比特计算出来，在某种程度上依赖于圈数。32 比特 R_i 可以扩展为 48 比特，怎样做到。准确的说，固定的方法，在同样的页码上，可以看到表格的索引，可以说是 E 比特选择表。有 48 比特的输入，这一圈的输入是上一圈的输出。也就是说，输出比特为 32,1,2,3,4,5 然后又是 4,5 圈。不是随机查寻。事实上，不包括 1，32 的交换（看最上面的左面和最底层的右边），看起来是连续的。因为是连续的，有八个 S 盒，首先把八比特变化为六比特。这样，得到 32 比特输出。最后，有一个 P–盒，一个置换使用这些 32 比特去得到另外 32 比特。

什么是 S 盒？每一个是固定的函数表变化，在算法中以代码或硬件的情况。怎样去读？采用 6 比特 $b_1b_2b_3b_4b_5b_6$ 。把最先的两比特是作为行数，在 1 和 4 之间，使用其他的 4 比特作为 1 到 16 之间的列数。这样可以清楚的查表。

即使这样也还没有涉及到具体的细节。上述主要的目标是给一些 DES 结构的 IDEA，当然，主要设计的问题在于：为什么这么设计？什么动机推动这些设计的确定，虽然可以猜测，但是不能断定。

4.2.3 速度

你能多快的计算 DES，为了使 DES 得到更快的速度，进行了硬件实现，因此，DES 的速度是硬件处理速度，将粗略的描述，你可以精确的评估，在不同的地方可以使用不同的结构。可以通过使用 VLSI 来得到 Gbit/s 的速度。最少有 1.6Gbit/s，也许可以得到更快的速度。确实很快。

发现的一些软件指标是：一个 HP9000/887 机器上 12Mbit/s,在一个 DEC ALFA 4000/610 上是 9.8Mbit/s。DES 的软件实现效果并非个人所期望的那样，这也是为什么对 DES 需要替换的原因。

4.3 分组密码的密钥恢复攻击

分组密码算法的实际攻击 E：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$，集中在密钥恢复上。明确表示了此类密码分析的难度，一个 k 比特的密钥 K 是随机选择的，假设 $q \geq 0$ 是一些整参数。

给定：攻击者有一个 E_k 的例子，q 输入–输出的序列，也就是说：

$$（M_1，C_1），\ldots,(M_q,C_q)$$

这里，$C_i=E_k（M_i）$对于 i=1,…,q 并且所有 $M_1,\ldots, M_q$ 是所有方向的 l 比特串。

找到：攻击者必须找到密钥 K。在这种构架下的两种攻击模式。

已知消息攻击：M_1，…，M_q 是清楚的观点，攻击者没有获得任何的先验经验，并且必须说明即将进行的工作是如何完成的。

选择消息攻击：M_1，…，M_q 是攻击者的选择，也许是合适的。也就是说，考虑使用 E_k 函数的随机判断机制的接入。

明显的，一个选择消息攻击给出了更多攻击者的条件，但在实际中是不太现实的。最明显的攻击是穷尽密钥搜索。

穷尽密钥搜索：遍历所有的密钥，$K' \in \{0，1\}^k$,直到发现正确的一个，命名为 K。但是怎样知道找到了正确的密钥 K，如果 $E_{k'}(M_1)=C_1$,假设 K=K'。当然，也可能是错的。尽管错误的机会很小，如果可以做更多的测试，得到的机会也许更少。对于 DES，两次测试足够了。也就是说，这个例子中的攻击算法只需要 q=2，一个很小数量的输入–输出例子。

更仔细地描述这个攻击算法[140]。

4.4 迭代 DES 和 DESX

整合上述讨论的秘钥搜索机制，得出 DES 被破解的结论，DES 的弱点是密钥长度仅仅为 56 比特，不能抗穷举攻击。

人们寻找廉价的方式强化 DES 机制，最简单的方式是使用长一些的密钥，一个范例是

迭代这个算法。

4.4.1 Double DES

假设 K_1，K_2，是 56 比特 DES 密钥，假设 M 是 64 比特明文，假设：

$$2DES(K_1||K_2;M) = DES(K_2; DES(K_1;M))$$

定义一个分组密码 2DES：$\{0，1\}^{112}\times\{0，1\}^{64} \rightarrow \{0，1\}^{64}$，称为 2DES，有 112 比特长度密钥，可以看做是两个 DES 进行的组合。根据分组密码算法的要求，这是可逆的：

$$2DES^{-1}(K_1||K_2;M) = DES^{-1}(K_1; DES^{-1}(K_2;M))$$

对于任意 64 比特 C。

112 比特密钥是足够大的，看来 2DES 有很小的风险，可以抗穷尽攻击，甚至在使用并行和特殊硬件条件下。另外 2DES 可以抗足够的密码算法攻击，也就是差分和线性分析。既然迭代效果增加了 FEISTEL 的圈数，应该指出，2DES 安全是优于 DES 自身的。

然而尽管 2DES 有 112 比特密钥，可以以 2^{57} 用 DES 和 DES^{-1} 进行中间人攻击，假设 $K_1||K_2$ 定义了目标密钥 $C_1=2DES(K_1||K_2;M_1)$ ，攻击者，给定 M_1，C_1，试图找到 $K_1||K_2$，可以观察：

$C_1= DES(K_2; DES(K_1;M)) \Rightarrow DES^{-1}(K_2;M) = DES(K_1;M)$

这导致如下的攻击，对于 $i=1,....,2^{56}$，假设 T_i 是第 i 个 56 比特串，

$MinM_{2DES}(N_1,C_1)$

For $i = 1,....,2^{56}$ do $L[i] \leftarrow DES(T_i,M_1)$
For $j = 1,\ldots,2^{56}$ do $R[j] \leftarrow DES^{-1}(T_j,C_1)$
$S \leftarrow \{(i,j): L[i]=R[j]\}$
Pick some $(l,r) \in S$ 并且 return $T_l || T_r$

对任意$(i,j) \in S$，有：

$$DES(T_i,M_1) = L[i] = R[j] = DES^{-1}(T_j,C_1)$$

作为结果，$DES(Tj, DES(T_i,M_1)) = C_1$。这样用输入输出例子（$M_1$，$C_1$），密钥 $T_i || T_j$ 是一致的：

$\{ T_l || T_r : (l,r) \in D\} = Cons\ \varepsilon\ ((M_1,C_1))$

攻击从 S 和输出 $T_l || T_r$ 使用了相同的对子（l,r），因此，返回一个（M_1，C_1）密钥。

上述的 S 集希望尽可能的大，大概是（2^{56+56}）/$2^{64} = 2^{48}$，意味着攻击者不能返回目标密钥。然而使用一个更多的输入、输出例子，容易在 S 中进行随机选择，希望可以保持目标密钥。

构造攻击 $2^{56}+2^{56}=2^{57}$，计算 DES 或 DES^{-1}，组成了 S 集，可以在线性时间内完成列变化，然后进行使用单向函数。（简单的策略是花销列规模的指数时间）因此运算时间主要是 DES，DES^{-1} 占用。

中间人攻击 2DES 是与攻击一个密码非常不同的，最好的攻击是穷尽搜索。然而这种攻击不太会凑效，甚至设计专用机进行运算。通常这种机器可以并行运算 DES 或 DES^{-1}，而且对于 2^{56} 次运算 2^{64} 输入集合 S 的列 L，R。存储的总数是 $8 .2^{57} \approx 1.15 \times 10^{18}$ 字节，或大约 1.15×10^{6} T 字节。这个数量在实际中是不可行的。

有些策略修改攻击减少上述的存储，加上一些时间消耗。但是攻击仍然是不可行的。

既然 112 比特 2DES 是可以发现 2^{57} 的 DES 或 DES^{-1} 运算，可以认为 2DES 的有效密钥长度是 57 比特。

4.4.2 3-DES

3DES 使用三次 DES 或 DES^{-1} 运算，三次密钥变化为：

$$3DES3（(K_1||K_2||K_3;M) = DES(K_3; DES^{-1}（K_2， DES(K_1;M)）)$$

因此 3DES3 是：$\{0，1\}^{168}\times\{0，1\}^{64} \rightarrow \{0，1\}^{64}$，两个密钥变化为：

$$3DES2(K_1||K_2;M) = DES(K_2; DES^{-1}（K_1， DES(K_2;M)）)$$

所以 3DES2 是：2DES：$\{0，1\}^{112}\times\{0，1\}^{64} \rightarrow \{0，1\}^{64}$，称为 2DES，有 112 比特长度密钥，这个变化是可逆的，是分组密码。指数 3 是指三次使用 DES 或 DES^{-1} ，中间使用 DES^{-1} 的基本原理是因为便于计算：

$$DES(K;M) = 3DES3(K||K||K;M) \qquad (4.1)$$

$$DES(K;M) = 3DES2(K||K;M) \qquad (4.2)$$

与 2DES 算法，这些密码的密钥长度是足够长的，使用穷尽搜索，甚至使用最好概率攻击模型，并且，另外的差分攻击、线性攻击不是特别有效的，因为通过增加 FEISTEL 圈数有效迭代是有效的。

3DES3 面临中间人攻击，使用 DES 或 DES^{-1} 运算 2^{112} 找到 168 比特的密钥，一个有效的密钥长度 112 比特。不想对 3DES2 进行中间人攻击，因此 112 比特是有效攻击。

3DES2 密码在算法中在实际是常用的，使用串行标准的 DES 置换，2DES 尽管有相同的密钥长度，3DES2 至少有足够安全，在实际中不常用，等式（4.2）说明为什么 2DES 比 3DES2 更常用。

4.4.3 DESX

尽管 2DES，3DES3，3DES2 可以提供适当的安全，他们运算速度是比较慢的。第一个是两倍 DES 运算速度，而后两个是三倍运算速度。应该有更好的方式构造基于 DES 的分组密码，密钥长度增加，但是消耗降低。确实有简单的设计实现。假设 K 是 56 比特密钥，K_1，K_2 是 64 比特分组，假设 M 是 64 比特分组：

$$DESX(K||K_1||K_2;M) = K_2 \oplus DES(K; K_1 \oplus M)$$

定义一个分组密码

DESX：$\{0，1\}^{184}\times\{0，1\}^{64} \rightarrow \{0，1\}^{64}$，有 184 比特的密钥，由 56 比特密钥和两个附加密钥组成，每个 64 比特长。注意这是可逆的，称为分组密码：

$$DESX^{-1}(K||K_1||K_2;D) = K_1 \oplus DES^{-1}(K; K_2 \oplus M)$$

长度为 184 比特密钥当然可以抗穷尽攻击。DESX 相对于 DES 并没有增加抗线性分析度和差分分析度。仍然看到还没有实际的攻击。

有一个 DESX 的中间人攻击，找到 184 比特的 DESX 密钥 2^{120}DES 和 DES^{-1} 计算。因此 DESX 的有效密钥长度是 120 比特，当然是安全的。

DESX 的安全强度弱于 2DES，3DES，因为后两者的抗线性和抗差分攻击强度大。DESX 的强度是与 DES 相同，这仍然是足够了。可以看到在实际工作中 DES 的分析弱点仅仅在密

钥长度上，DESX 加强了这点，并且是很廉价的。总而言之，DESX 应该是推荐常用的，因为在提供足够安全的同时，比 2DES 和 3DES 是简单实用的。

4.4.4 为什么设计新密码

DESX 是否是一个好密码是一个争论。否则有一些更重要的原因进行新的密码标准讨论。在 5.8 节中分组密码安全不仅仅体现在密钥长度和分组大小上。一个以 n 为分组长度的密码算法，可以在 n/2 的时间内进行破解。当 n=64，时间仅仅为 2^{32}，尽管 2DES，3DES3，3DES2，DESX，比 DES 有更有效的密钥长度，按照上述的观点，其安全强度并没有增加。看来使分组密码算法的分组足够大，$2^{n/2}$ 就无法在实际中进行运算。这是设计一个新密码的原始初衷。

也许运算速度是更大的动机，希望设计一个软件运算的算法比 DES 运算速度快。

4.5 高级加密标准

2001 年 2 月，一个的标准产生来代替 DES。高级加密标准（Advanced Encyption Standard, AES）是一个叫做 Rijndeal 的分组算法。在[66]中可以看到描述。分组长度为 128 比特，密钥长度由 128 比特、192 比特、256 比特不等。以下是几个运算模式。

假设一个分组密码 E 是固定的，并且假设使用分组密码的通信双方共享密钥。他们只需计算：$E_k(\cdot)$和 $E_k^{-1}(\cdot)$。这些函数可以使用一个输入 l 比特，E_k 的应用过程叫加密，E_k^{-1} 的应用过程叫解密。

典型的分组密码算法分组为 64 比特和 128 比特。在实际使用里，希望处理大的输入长度，也就是说正文的加密。通过一些运算模式进行各种工作，这里有几种模式。将列举三种模式来说明运算工作，分别是电子密本工作模式（Electronic Code Book, ECB）、密文链接工作模式（Cipher Block Chaining,CBC）、计数模式（Counter,CTR）。每一个加密过程是⊕比特串 M，通常叫做明文，返回一个串 C，通常叫做密文。（如果 x 的长度不是 l 的倍数，必须引入一些适当的填充来变为倍数。）一个相应的算法把 C 还原成为 M。

如果 x 是一个 l 的倍数，可以直接把它分解成 l 整数倍的分组，假设 x[i]指明了 i-th 块，i=1,2,…,[x/l]，也就是说，x=x[1]…x[n]，这里 n=|x|/l。

4.5.1 电子密本工作模式

每个明文块分别被加入一个相关的密文块，

算法 $E_k(M[1]\ldots M[n])$	算法 $D_k(C[1]\ldots C[n])$
For i=1,…,n do	For i=1,…,n do
$C[i]\leftarrow E_k(M[i])$	$M[i]\leftarrow E_k^{-1}(C[i])$
Return C[1]…C[n]	Return M[1]…M[n]

4.5.2 密文链接工作模式

CBC 运算过程是基于初始向量 IV 的如下 l 比特串，例如：

算法 E_k(IV，M[1]…M[n])
C[0]←IV
For i=1,…,n do
C[i]←E_k(C[i-1] ⊕ M[i]
Return C[0],C[1]…C[n]

算法 D_k(C[0]，C[1]…C[n])
For i=1,…,n do
M[i] ←E_k^{-1} (C[i] ⊕C[i-1]
Return M[1]…M[n]

与 ECB 加密模式不同的是，加密长度不是保持不变的：输出长度比输入长度多 1 比特。初始向量用以加密，然后是部分的密文，这样接收者不必提前知道这些。不同的说明随机模式，除非有不同的状态，假设在关于运算模式的上述讨论之前，加密方随机选择初始向量，通过加密更新消息 M。一些另外的选择可以考虑，例如假设 IV 是一个计数器，在实用算法中是实用的，这些不同的选择的安全特征随后会进行讨论。

CBC 模式是最普遍的方式，普遍的应用于实际使用里。

4.5.3 计数模式

CTR 模式仍然使用了一个附加的值，一个初始值 IV 是 l 个整数，0，1，…，2^l-1。如下，所有的加法是在 2^l上进行的。并且 NtSl（j）指示 j 的二进制的表示法为 l 比特串。

Algorithm $E_{\cdot}$ (IV,M[1]…M[n])
For i=1,…,n do
C[i] ←E_k(N_tS_l(IV+i)) ⊕M[i]
Return N_tS_l(IV) C[1]…C[n]

Algorithm D_k(N_tS_l(IV)C[1]…C[n])
For i=1,…,n do
M[i] ←E_k(N_tS_l(IV+i)) ⊕C[i]
Return M[1]…M[n]

注意在这个例子中，解密不需要计算 E_k^{-1} ,并且事实上，甚至没有要求 $E_{\cdot}$ 是一个置换。并且注意 CBC 的优势有效性。加密算法是并行处理的，并且，有几个初始向量的选择。可以由信息发送者管理计数器，消息 M 加密后增加 n=|M|/l，每次调用算法时，重新随机选择。

数据加密标准（Data Encryption Standard，DES），是一个分组密码算法的精华。但是现在已经可以感觉它逐渐过时了，另外，但是对所有分组密码算法的都不能忽视这种结构。DES 是一个典型的好的算法机制，对密码学的发展有深远的影响。DES 被广泛的应用，也许还会再使用几年，每一次使用 ATM 机，就使用了 DES 算法。

4.6 基于安全的密钥恢复限制

综上所述，经典的分组密码算法攻击通常是集中在密钥恢复上，对于密码算法 E 的分析的认定在以下几个方面：给定一些输入、输出的例子（M_1，C_1）,…,(M_q,C_q)，这里 K 是一个随机数，密钥未知，C_i=E_k（M_i），对于攻击者恢复 K 的难度有多大？一个分组密码算法被确定为安全的，如果满足：计算灵活，对于实际抗攻击的 q 的取值不是太大。最后，认定这个算法是安全的。

然而，安全的概念在于，抗密钥恢复攻击的安全是非常有限的。一个好的定义是对于使用足够强。这也就是说，分组密码算法是安全的，那么，是值得研究的结构是可以保证数据安全的。但甚至是对分组密码算法匆忙的一瞥也可以说明其是安全的，但不保证使用中更一般的攻击方法。

把 CTR 运算模式作为例子在 4.4 中讨论。假设分组密码算法有如下的弱点：给定 C，F_k（C+1），F_k（C+2），计算 F_k（C+3）。明显加密体制是安全的，因为攻击者恰好知道前两个消息的分组，从密文可以直接计算出第三个分组。（假设攻击者已知前两个分组是完善保密体制。也许，例如公共封装标题信息，或一些已知接收信息的命名）。这就意味着，如果 CTR 模式加密是安全的，分组密码具备特征为：给定 C，F_k（C+1），F_k（C+2），计算 F_k（C+3）是计算不可能的。假设 SP1 的特征，因为一方的安全特征。

当然，任何一个知道密钥的一方是根据 C，F_k（C+1），F_k（C+2），容易计算出 F_k（C+3），但是对于不知道密钥的一方是很难计算的。也就是说，函数 F 的构造抗密钥恢复攻击不能代表 SP1 为真。

继续这个现象的研究，所能看到的密码更多用法，建立起更多的安全特征链 SP1，SP2，…，SP3，…这对基于应用的分组密码是必须的。此外，如果 SP1 为真，CTR 加密运算模式依然比较弱。SP1 不能保证 CTR 运算模式的安全。同样可以简单的提出使用其他的安全特征。这个必要的长清单是没有视为安全的，所需要用的是分组密码一个单独的‘MASTER’特征，使用分组密码本身的算法可以保证安全。一个相信自身好的例子在于抗密钥恢复设计的安全是不够的，确定分组密码算法 E：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$,定义所有的密钥 $K \in \{0，1\}^k$ 和明文 $x \in \{0，1\}^l$ ，计算 F（K，x）得到密钥。也就是说，每个分组密码算法 F_k 是一致的。这是一个好的分组密码算法吗？当然不是。但是，它对密钥恢复攻击算法是安全的。事实上，给定 F_k 的输入，输出，攻击者不能判断出是通过什么密钥产生的。

这也许可以看做是一个特别的例子，许多人会认为：“但是，显然 DES 和 AES 不是这样设计的”。这个观点有些片面，这个例子只是证明仅抗密钥还原攻击的设计不一定是安全的。必须寻找更好的设计来满足安全设计。

4.7 练习和难题

练习 4.1、对于所有的密钥 $K \in \{0，1\}^{56}$ 和 $x \in \{0，1\}^{64}$ ，$DES_k(x)= \overline{DES_{\bar{k}}(\bar{x})}$，这叫做 DES 的密钥取余特点。

练习 4.2、怎样利用 DES 的密钥取余特点来加快密钥穷尽搜索，并对提出的假设进行解释。

练习 4.3、找到一个密钥 K，使得 $DES_k(\cdot)= DES_k^{-1}(\cdot)$,这种密钥称为弱密钥，能找到多少弱密钥？为什么认为是弱密钥？

第五章 伪随机函数

伪随机函数（PRFs）和相关系列伪随机置换(PRPs)，视做协议设计的核心工具。特别是对称密码学。在同一水平上，PRFs 和 PRPs 都可以用做分组密码算法的模型，保证基于分组密码的安全协议分析，因此可以保证安全。在文章的其余部分，PRFs 和 PRPs 是极好的开端。在这一章，介绍 PRFs 和 PRPs，了解的一些基本特征。

5.1 函数簇

一个函数簇是一个映射，F：Keys(F)×Dom(F)→Range(F),这里，Keys(F)是 F 的一个函数簇的子集，Dom(F) 是 F 的一个函数簇的定义域，Range(F) 是 F 的一个函数簇的值域。两个输入的函数 Keys 和 Dom，返回一个值 y，定义为：F（K,x）。对于任何 K∈Keys(F)，定义映射 F_k： Dom(F)→Range(F),表示为 $F_k(x)=F(K,x)$。称 F_k 是 F 函数簇的一个特例。因此，F 说明了一个函数簇，每个函数可由一个密钥确定。这就是为什么称 F 为函数簇。

最常见的是 Keys(F)={0，1}^{k} 并且 Dom(F)={0，1}^{l} 和 Range(F)= {0，1}^{L}，对于 k,l,L≥1 的整数。但是，有些定义域、值域可以使用不同的子集合，包括了不同长度的序列串。对于密钥子集 Keys(F)可以进行随机分配，通常的方法是进行均匀分配，因此，可以随机选择 k 比特作为密钥。指示：K←Keys（F），从 Keys(F)选择一个随机操作 K，那么，*f*←F 代表操作：K←Keys（F），*f*←F_k 。换言之，如果 *f* 函数是函数簇 F_k 中的一个，那么，K 随机选择密钥。对于随机选择的密钥相应输入、输出的函数选择。

一个置换是一个映射，定义域和值域是同一个子集，这个映射是这个子集上保持长度的双射。也就是说，Π：D→D 是一个置换，如果|Π（x）|=|x|对所有 x∈D，Π 是单射也是双射。如果 Dom(F) = Range(F)，F 是一个置换的函数簇，每个 F_k 是一个公共集上的置换。

例 5.1、 一个分组密码是一个置换簇，例如，DES 是一个使用 Keys(DES)={0,1}^{56}, Dom(F)={0，1}^{64} 和 Range(F)= {0，1}^{64} 的置换簇，这里，k=56，l=64，L=64。AES 的函数表示为：Keys(DES)={0,1}^{128}, Dom(F)={0，1}^{128} 和 Range(F)= {0，1}^{128} 的置换簇，这里，k=128，l=128，L=128。

5.2 随机函数和置换

假设 D，R∈{0，1}* 是有限集，l, L≥1 是整数，固定了两个函数簇。一个是 Rand$^{D→R}$ 。所有函数簇是 D 到 R 上的，其中 D 是 Perm^D ,是所有 D 上的置换簇。为了使概念更紧凑，假设 RAND$^{l→L}$ 与 RAND$^{D→R}$ 同构，这里，D={0，1}^{l}, R={0,1}^{L} 。也将在 D{0，1}^{l} 设置 Perm^l 与 Perm^L 同构。

这些簇包括哪些呢？簇 RAND$^{D→R}$ 有定义域 D 和值域 R，Perm^D 有定义域和值域 D，RAND$^{D→R}$ 固定的子集是所有 D 到 R 上的函数簇，Perm^D 的子集代表 D 上所有的置换。对于特殊场合函数的描述是在简单基本条件假设下的描述。例如，定义域的阶 D 是按照 $x_1,x_2,\ldots$，假设对于一个函数 *f* 的密钥是一条链：（$f(x_1),f(x_2),\ldots$）。RAND$^{D→R}$ 上的密钥空间是在均匀分配下，密钥的子集。

对于最感兴趣的事件进行更详细的举例，在 RAND$^{l→L}$ 的函数的密钥是简单的函数输出值的数列，当输入为{0，1}^{l}。因此：

$Keys(RAND^{l\to L}) = \{(y_1,\ldots,y_{2^l}) : y_1,\ldots,y_{2^l} \in \{0,1\}^L\}$是所有长度为 2^l 的序列的子集，每个输入是 l 比特长的序列。对于任意 $x\in\{0,1\}^l$,把 x 转换成为在{1，…，2^l}内的整数并且子集为：

$$RAND^{l\to L}\ ((y_1,\ldots,y_{2^l}),\ x) = y_x$$

注意：密钥空间是非常大的，数据量为 2^{L2^l} 。自然的，有一个密钥对应从 l 比特到 L 比特的函数，数据量就是这种函数的数量。密钥空间可以用均匀分配的组合。这样，$f\leftarrow RAND^{l\to L}$ 是选择一个 l 比特到 比特的函数的运算。

另外，对于 $Perm^l$,密钥空间是 $Keys(Perm^{l\text{-}L}) = \{(y_1,\ldots,y_{2^l}) : y_1,\ldots,y_{2^l} \in \{0,1\}^L$ 并且 $y_1,\ldots,y_{2^l}$ 是都不相同的}，对于任意的 $x\in\{0,1\}^l$ ，把 x 变做一个{1，…，2^l}内的整数并且子集为：

$$Perm^{l\to L}\ ((y_1,\ldots,y_{2^l}),\ x) = y_x$$

密钥空间又进行均匀的设计，这样，$f\leftarrow Perm^{l\to L}$ 是选择$\{0，1\}^l$上的随机置换的操作。换言之，在$\{0，1\}^l$上的所有随机置换是概率相似的。

例 5.2、以 $RAND^{3\to 2}$ 为例，也就是说 l=3,L=2。定义域是$\{0，1\}^3$，而值域为$\{0，1\}^2$。一个簇中函数的例子是：通过输入输出真值表确定函数 *f*：

X	000	001	010	011	100	101	110	111
F(x)	10	11	01	11	10	00	00	10

从这个特殊的函数得出的密钥列为（10，11，01，11，10，00，00，10）。

$RAND^{3\to 2}$ 上的密钥空间是所有这样的数列的集合。所有 8 维空间输出为 2 比特的函数的个数为：$2^{2\cdot 2^3}=2^{16}$=65，536，这是密钥空间的大小。

很少以这种形式来讨论这样的函数簇。事实上，可以更加直观地去讨论该变化，这确实是一个重要的方向。

随后讨论这种设置，假设有一个关于函数 g 的黑盒，这个盒子的输入是 g 的定义域中的一个值，输出就是 g(x)。但是不知道盒子的内容，有一个说明关于黑盒的界面。有一个随机的函数 $g:\{0,1\}^l\to\{0,1\}^L$ 按照下列方式进行相应的变化。每次给黑盒子一个输出，返回一个 L 比特的函数值。唯一的限制是如果给定黑盒子两个相同的输入 x，将会返回两个相同的输出 g(x)。换言之，一个 l 比特到 L 比特的变换何以表示成为一个输入为$\{0,1\}^l$而返回一个随机数，除非可以除去与之前输入相同的值。建议使用一个动态的观点来考虑随机函数。

动态的观点可以使用如下的观点实现。这个流程使用了表格 T 的形式提供了这个函数，这里 T[x]是函数在 x 处的值。最初，表格是空的，程序按照如下的格式处理一个输入 $x\in\{0,1\}^l$ 如下：

如果 T[x]没有定义

随机选择一个串 $y\in\{0,1\}^L$,并且假设 $T[x]\leftarrow y$

返回 T[x]

对于任意一点的答案是随机的，与答案是独立的。

从另外一个角度考虑随机函数是极大、预先知道的表格，输入是形式（x,y），对于每一个 x，随机选择一个 y 并且填入表格的思想是随机的选择点，同时随机选择一个函数。必须记住随机函数的术语是一个误解，可能误导把某些固定的函数当作随机函数而把其他的函数看做不是随机的。（例如，总是返回 0^L 的常数函数不是随机的，但是对于许多不同值的函数是随机的）这是不正确的。函数的随机性是指选择的方式，不是一个选择函数本身的特征。当随机选择一个函数，常数函数和其他函数一样有相同的概率被选择。单独谈论一个特殊的函数是没有意义的。“随机函数”的意义只是随机选择的函数。

例 5.3、假设做简单随机计算与明白一个随机函数。固定 $x\in\{0,1\}^l$ 并且 $y\in\{0,1\}^L$ ，那么：$P[f(x)=y : f\leftarrow Rand^{l-L}]=2^{-L}$ ，注意，该值与 l 无关，同时与 x,y 也无关。假设，$x_1,x_2\in\{0,1\}^l$ 并且 $y\in\{0,1\}^L$ ，那么：

$$P[f(x_1)=f(x_2)=y : f\leftarrow Rand^{l-L}]=\begin{cases} 2^{-2L} & x_1\neq x_2 \\ 2^{-L} & x_1=x_2 \end{cases}$$

这个阐明了随机性，也不依赖于 x,y 的值。

目前，$x_1,x_2\in\{0,1\}^l$ 并且 $y\in\{0,1\}^L$ ，那么：

$$P[f(x_1)\oplus f(x_2)=y : f\leftarrow Rand^{l-L}]=\begin{cases} 2^{-L}, & x_1\neq x_2 \\ 0, & x_1=x_2, y\neq 0^L \\ 1, & x_1=x_2, y=0^L \end{cases}$$

多出两个选项的总数可以以此类推。

5.3 随机函数

随机函数是一个函数簇，对于随机函数是基于计算不可能的输入输出常数。如果只知道黑盒子的函数是指给黑盒子一个输入，然后得到一个输出判断是随机常数函数还是随机函数问题是计算困难的。本节讨论的问题在对这个概念做一个合适的定义，随后，再研究目的和应用。

固定一个函数簇 F：Keys（F）×D→R（对于通常的情况，可以考虑对于整数 k,l,L≥1,Keys（F）={0，1}^{k} ,D={0,1}^{l} ,R={0,1}^{L} ）。假设在一个房间，房间内的计算机可以和外界联系。可以录入以些文件然后向外界发送，随后会有回答发送过来。但录入的内容必须在定义域范围 D 之内，接受到的反馈在值域 R 内。房间之外的计算机负责进行计算，g： D→R,这样，输入 x，反馈为 g(x)。然而，只能通过这个界面接入函数 g，只能看到 g 函数的输入和输出，考虑 g 的两种选择方法，建立两个不同的条件。

条件 0、函数 g 是使用 $Rand^{R-D}$ 算出的值，也就是说通过 $g\leftarrow Rand^{R-D}$,(这样 g 是 D 到 R 的随机函数)。

条件 1、函数 g 是从 F 中获取的，命名为 g←F,（也就是说密钥通过 K←Keys(F)进行选择，g 是 F_k 的一个函数）。

不会知道选择哪些条件，在进入特定空间，开始输入问题之前，选择一个条件，和一个相应的通信函数 g。一旦确定之后，在会话结束之前这些参数都是固定的。的工作就是确定要使用哪个条件，唯一提供给的信息就是当输入是 x 时，输出是 g(x)。当使用了所选择的一些值之后，必须确定所使用的是哪种条件。随机函数簇的性质决定了这个问题的难度，在上述规则下，可以确定使用的值是 0 还是 1。

直观来看，在诸如密码体制的条件中使用函数 g 只是规则模型的一种。很难判断输入输出是否是真随机函数，操作是同样条件下使用伪随机函数 F 还是随机数产生的，随后会看到这项工作是怎样进行的；从现在开始，继续深入这个问题。但是注意到在所有随机函数使用中的伪随机函数不能完全被随机函数取代。对于特殊应用确定这是正确的，必须确认在相关的定义域，正规定义是可以被使用的。

通过一个标识的定义可以确定所使用的 0、1 条件是形式化的。这是一个算法提供一个判断机制接入一个函数 g，试图去确定 g 是随机的还是伪随机的（或者说明 g 是在条件 0 还是在条件 1）。一个分辨机是仅仅使用函数互相影响，给定输入并确定相应输入的输出，并且不能使用任何方法直接检查。把 A 以随机语言机方式接入函数 g 记做 A^g 。直观的，一个函数簇是伪随机的，如果分辨机粗略假设 1 的概率而不管其分辨机是在哪个条件。使用下列的数学概念，在定义之后做进一步解释：

定义 5.4、假设 F：Keys(F) ×D→R 是一个函数簇，假设 A 是一个算法对一个函数的随机判断机 g：D→R，并且返回一比特，考虑两个实验：

Experiment $Exp_{F,A}^{prf-1}$	Experiment $Exp_{F,A}^{prf-0}$
K←Keys(F)	g←$Rand^{D\rightarrow R}$
D←A^{Fk}	d←A^g
Return d	Return d

A 的 prf 优势定义如下：$Adv_{F,A}^{prf} = P[Exp_{F,A}^{prf-1} =1] – P[Exp_{F,A}^{prf-0} =1]$，对于任意 t,q,μ，定义 F 的 prf 优势为：$Adv_F^{prf}(t,q,\mu)= \max_A \{Adv_{F,A}^{prf}\}$。

这是对所有计算难度为 t 的事件 A 求最大值。以所有至多 q 判断机，以 μ 比特的长度序列的总和。

算法 A 模仿假设在房间的一个人，尝试确定或她在哪个条件使用，通过计算函数 g， 输入一条序列。通过形式化的方式，一个人就是一个算法，意味着一段代码。通过事件 A 形式化了一个 g 函数串，输入一个串 x∈D， 输出 g(x)。算法 A 可以决定采用哪个串，也许受到了前向序列的应答后。最终输出一个比特 d 确定它来自哪个条件。

应该这样看，函数 F 是公共的。攻击者 A，或者其他人，知道了这个簇的描述，并且有代码可以去完成。给定值 K，x，计算 F（K，x）。

条件可以得到一个经验值。第一个经验值随机选择函数簇 F 的 F_k 函数，而且使用随机判断机 $g=F_k$ 运行攻击者 A。攻击者 A 使用判断机相互影响，质询然后返回应答，最后输出一个猜测的比特。经验值返回同样的比特。第二个经验值随机选择一个函数 g : D→R，使用这个判断机运算事件 A，然后返回 A 的猜测比特。每次测试都有得到 1 的一个概率。如果事件 A 已经作出了从哪个条件选择的预先工作。因此，对于第一个经验值，随机选择可能产生的事件 A，对密钥 K 的选择，A 是一个随机选择的函数。对于第二次经验值，g 的随机判断机 A 确定的任意随机选择。这两个概率应该是相互独立的，两个经验值是完全确定的。

去看事件 A 对所使用条件的确定性，观察概率的独立性使得两个经验值返回 1。如果 A 做到了对自己使用条件的确定，返回 1，第一经验值的概率大于第二经验值的概率，这个区别值可以作为事件 A 的度量。称这种测量为事件 A 的先验优势。把这个优势看做是 A 可以对 F 进行破译的概率，破译传达的概念是基于定义的技术路线。

不同的分辨机有不同的优势，有两个原因确定为为什么一个分辨机的优势会比另一个的大的询问是明智的，而处理输出为回答。另一个可以简化为：询问更多的问题，或花更多的时间去处理回答。事实上，希望可以看到更多的输入输出函数 g 的例子的能力可以确定会到

哪个领域。函数簇 F 的安全性在于可以抗击攻击者测量函数资源。想知道，对所有给定的信息资源的限制，联系一个函数资源 F，一个前向优势的参数资源的输入值是返回一个最大的优势偏差，而攻击者被限制获得这些优势资源。在攻击者被限制使用资源时，考虑作为最大的偏差概率来破译一个密码体制。

资源的选择被认为是可变的，因此会选择 A 为时间复杂性的度量，确定序列 q 的计数，这些序列的总长度为 μ。联系一个优势函数簇，输入一个特定的值然后返回一个最大偏差。换言之，这是在给定资源限制中的“最优势”或“最好”的分辨机限制。F 的优势函数是得到了 F 作为 PRF 的安全性。

更仔细地解释一些在测量之下的潜在的重要的协定的来源。第一个资源是算法 A 的时间复杂性，为了确立这一点，需要建立一个计算模型。固定一个 RAM 模型，考虑一个算法路线，这样可以度量一个可测的时间算法。然而，认定 A 的时间复杂性是不仅仅是指 A 的计算时间，而是指按照定义所计算出的两个值中最大的一个，加一个 A 的码型长度。对于第一个经验值的测定，随机选择一个 K 中的 k 去计算时间，对事件 A 的随机选择序列 x 计算 $F_k(x)$的值是一种随机判断机制。测量第二个经验值时，以动态的方式选择随机判断函数 g，也就是说计算一个真值表（x,g(x)）。表的入口处加上一个 g 构造的序列。随机选择一个新的输入、输出的值。

查询计数是 A 获取一个输入输出的例子。通常，并不是定义域中所有的序列必须有所有的长度，因此可以测量所有序列长度的总和。

每个人都要求的参数并不是只有一个指标。假设 F 有一个密钥函数 K-长度。显然，密钥长度是决定安全性的一个重要原则。密钥长度越大说明安全性越强。然而密钥长度 K 没有明确的函数优势 $Adv_F^{prf}(t,q,\mu)$。为什么是这样呢？优势函数是 k 函数的优势度，没有知道关于 F 的困难是什么。密钥长度是毫不相关的，事实只是密钥长度本身是不相关的：优势偏差可以得到一个分辨机。一个好的分组密码设计是 $Adv_F^{prf}(t,q,\mu)$应该大约是 $t/2^k$。但这只是一个理想值，实际上，不应该假设密码是好的。

这个事实是基于有几种策略可以被分辨机所使用，只限制它的定义。一个分辨机在特定的资源范围内区辩出函数。

希望得到怎样的一个安全的“PRF”，定义 5.4 没有任何清晰的条件和当 F 被确认为安全时的陈述。只有一个 F 函数 prf 优势，直观的，F 是安全的，如果优势函数输入参数是实际值。也就是说，当然不是正规的。可能存在的概念在一个复杂性的理论的构架下使用一个安全 PRF。一个人可以说这是一个显然有多项式时间资源的攻击者所具备的优势。这要求一个可扩展的模型，要求一个安全参数在某种程度上可以进行渐近评估。将随后进行此处的说明，但是现在坚持做一个模型是一个安全 PRF 的失真性测试的概念。原因是反映了实际的模型，事实上，安全不是绝对的布尔体制性质，安全是攻击者发明的一个模型。所有现代密码学体制在理论上是可破的，问题只是需要多长时间。

这是密码学定义的第一个例子，值得花时间去研究或明确，将遇到更多相关概念。作为总结，当看到今后会遇到的问题时，将概述所有的架构。首先，这里有包括一个攻击者的实验。那么，对于攻击者的优势函数，反馈出攻击者可能攻击密码系统的概率。最后，这里有一个与密码协议相伴的优势函数，在所有限制的归约中，包括输入源函数并且攻击者返回优势值最大的一个。这三个组成将在所有定义中出现。最重要的元素是经验值，这里选择一个一个安全体制下的 PIN 码。

5.4 随机置换

一个分组密码算法 F 的置换簇，每个 F_k 的实例是一个置换，介绍一个伪随机置换的概

念来模型化分组密码。对于上述过程进行严密的证明，但是使用 $Perm^D$ 代替 $Rand^{D\to R}$ 。

在这种结构下，可以考虑两种攻击。一种是象以前一样对测试过的函数 g 产生一个判断机。但是，当 g 是一个是攻击者也可以得到的置换，另外，也是 g^{-1} 的判断机制。反过来考虑这个机制：第一个机制是选择明文攻击，第二个机制是选择密文攻击。

5.4.1 在 CPA 下的 PRP

固定一个函数 F 簇，Keys(F)×D→D，（也可以考虑 Keys(F)={0，1}^{k},D={0,1}^{l}，既然这是共识。不要求 F 是一个置换簇，这几乎也是一个共识。）和以前一样，认为一个攻击者 A 在一个空间被代替，它可以使用一种或两种随机判断机制来接近一个函数。

条件 0、函数 g 是随机从 $Perm^D$ 选取的，也就是 g←F(g 是 D 上的随机置换)。

条件 1、函数 g 是随机从 F 中选取的，也就是说 g←F。（意味着密钥通过 K←Keys(F) 选取，然后 g 在 F_k 中给出。）

条件 1 与 PRF 中的设置是一样的，但是条件 0 有一些变化。与以前一样，面临一个攻击者 A 通过函数 g 的输入输出决定其所处的条件。

定义 5.5、假设函数 F：Keys(F)×D→D 是一个函数簇，假设 A 是一个算法，对一个函数 g 设置一个随机判断机制 g : D→D，返回一个比特。考虑两个经验值：

经验值 $Exp_{F,A}^{prp\text{-}cpa\text{-}1}$	经验值 $Exp_{F,A}^{prp\text{-}cpa\text{-}0}$
K←Keys(F)	g←$Perm^D$
D←A^{Fk}	d←A^g
返回 d	返回 d

A 的 prp-cpa-优势定义为：$Adv_{F,A}^{prp\text{-}cpa} = P[Exp_{f,A}^{prp\text{-}cpa\text{-}1} =1]-p[Exp_{F,A}^{prp\text{-}cpa\text{-}0} =1]$

对于任意的 t,q, μ，定义 F 的 prp-cpa-优势：$Adv_F^{prp\text{-}cpa}$（t,q, μ）= $\max_A \{ Adv_{F,A}^{prp\text{-}cpa}\}$

这样，所有具备时间复杂度 t 是 A 上的随机判断机制，这些序列和最多 μ 比特。

对于定义 5.4，直觉是相似的。区别在于有一个完美的目标，被比较的 F 不再是任意函数簇，而是任意随机置换。实验 $Exp_{F,A}^{prp\text{-}cpa\text{-}1}$ 实际上是与 $Adv^{prf\text{-}1}$ 同样的。随机选择密钥 K 的概率并且对于随机事件 A 的随机化，经验值和 A 返回同样的比特。对于经验值 $Exp_{F,A}^{prp\text{-}cpa\text{-}0}$ ，一个置换 g :{0，1}^{l}→{0,1}^{l} 是随机选择的，使用判断机 g 返回的 A 的计算量。概率是基于 G 的选择和任意随机事件 A。与以前一样，对 A 随机性的度量是确实区别了两个条件，可以得到 A 的 prp-cpa-优势，如果与概率不同，返回经验值 1。关于资源的测量同样是和以前相同。非正式的，一个 F 簇是在 CPA 条件下的一个安全 PRP，如果 $Adv_F^{prp\text{-}cpa}$（t,q, μ）对资源参数的实验值。

5.4.2 CCA 条件下的 PRP

假定一个置换簇 F：Keys(F)×D→D（也可以考虑 Keys(F)={0，1}^{k},D={0,1}^{l}，既然这是共识。不要求 F 是一个置换簇，这几乎也是一个共识。）与以前相同，认为一个攻击者被放在一个空间内，但是现在有两个函数的判断逼近，g 和 g^{-1}。G 选择作为 CPA 模式的方式，g 一旦选定，g^{-1} 就会被确定，不必考虑是怎样确定的。

条件 0、函数 g 是使用 $Rand^{R\text{-}D}$ 算出的值，也就是说通过 g←$Rand^{R\text{-}D}$,(这样 g 是 D 到 R

的随机函数）。

条件 1、函数 g 是从 F 中获取的，命名为 g←F,（也就是说密钥通过 K←Keys(F)进行选择，g 是 F_k 的一个函数）。

对于条件 1，$g^{-1} = F_k^{-1}$ 是选择实例的逆，这里条件 0 是选择随机置换的逆。和以前相同，攻击者 A 是基于随机判断机制的输入输出行为，决定了是可以替代的。

定义 5.6、假设 F：Keys(F) ×D→R 是一个函数簇， A 是一个算法对一个函数的随机判断机 g：D→D，并考虑 g^{-1}：D→D，返回一比特，考虑两个实验：

Experiment $Exp_{F,A}^{prf-1}$	Experiment $Exp_{F,A}^{prf-0}$
K←Keys(F)	g←$Perm^{D\to R}$
d←$A^{Fk,Fk-1}$	d←$A^{g,g-1}$
Return d	Return d

A 的 prf 优势定义如下：$Adv_{F,A}^{prp-cca} = P[Exp_{F,A}^{prp-cca-1} =1] - P[Exp_{F,A}^{prp--cca-0} =1]$。对于任意 t,q_e,μ_e，$,q_d,\mu_d$,定义 F 的 prp-cca 优势为：$Adv_F^{prp-cca}(t,q_e, \mu_e, q_d,\mu_d)= \max_A \{Adv_{F,A}^{prp-cca}\}$。

对所有计算难度为 t 的事件 A 求最大值。以所有至多 q_e 序列到 g 判断机，以 μ_e 比特的长度序列的总和，同时对 g^{-1} 也产生至多 q_d 比特的序列, 序列长度的总和是 μ_d。

这个定义与 5.4 直观上是相似的，区别在于攻击者有更强大的能力：并不只是可以询问 g，也可以询问 g^{-1}。这些源测量的协议也同以前的定义类似。因此，加上了一些其他的参数。因为这里有两个判断机，那么单独的计算数字序列，这些序列的总长度为，非正式地，一个 F 簇在 CCA 下是安全 PRP，当 $Adv_F^{prp-cca}(t,q_e, \mu_e, q_d,\mu_d)$对于源参数的实践值是小的。

5.4.3 概念间的关系

如果上述一个攻击者不能询问 g^{-1}, 随后的判断机制可以不太好，攻击者可以有效的进行选择明文攻击，有如下的定理：

命题 5.7、假设 F：Keys(F) ×D→R 是一个函数簇，那么：对于任意的 t,q, μ,有 $Adv_F^{prp-cpa}cpa(t,q, \mu)= Adv_F^{prp-cpa}cpa(t,q, \mu,0,0)$

5.5 分组密码的一些模型

5.5.1 PRFs 和 PRPs 的簇的次序

综上所述，认为函数簇有一个有限的密钥空间，定义域和值域都是有限子集。函数簇的序列是 F^1，F^2，F^3，…，写$\{F^n\}_{n\geq 1}$ 。每个 F^n 是一个输入长度 l(n)函数簇，输出长度是 L(n), 密钥长度是 k(n)，这里 l,L,k 是安全参数 n 的函数，分别叫做序列的输入、输出、密钥长度。

在分组密码算法的模型中，已经考虑了簇的适当的提取。有几个原因，然而，也想考虑序列簇。安全可以被渐近表出，为了更方便的定义，较上述不安全的定义，特别有安全定义更好的说明。同样，当设计的安全是基于数论难题的假定难度，自然得到与原来函数簇相同的序列。叙述一个序列簇的伪随机定义，（省略一个类似置换）

假设 $F=\{F^n\}_{n\geq 1}$ 是一个函数簇的序列，假设 $D=\{D^n\}_{n\geq 1}$ 是一个分辨机的序列， D 的 prf-

优势是函数为 $Adv_{Fn,Dn}^{prf}(\cdot)$，对每个 n 定义为：$Adv_{F,D}^{prf}(D)= Adv_{Fn,Dn}^{prf}$

定义 F 是一个 PRF，如果是多项式时间可计算的，函数 $Adv_{Fn,Dn}^{prf}(\cdot)$是对每个明显的多项式时间鉴别机制 D。

注意这个时间定义强调函数自己是有效计算的。

5.5.2 PRFs 和 PRPs 的用途

讨论这些安全定义的动机。

5.5.3 共享随机函数模型

在对称密码学中，Alice 和 Bob 共享攻击者未知的密钥。想使用这种密钥产生不同的密文。特别的，为了加密和认证数据，互相发送。假设无论允许共享更多的序列，密钥（或应该）是一个短序列。采用的形式是一个 1 比特到 L 比特的随机函数 *f*，对于一些前向说明 l,L。这叫做共享随机函数模型。

共享随机函数模型不能真正的随机实现，因为，就看来，随机函数太大了来做相等的储存。这是一个概念上的模型。为了在这个模型下工作，给定一个多方判断机制来逼近 *f*。可以写做：$x\in\{0,1\}^l$ 并且每一步为 *f*(x)。

可以看出密码学共享随机模型是非常方便的，明确的表示、并且分析了设计。特别的，将看到许多例子，对共享随机模型的设计，并且证明是安全的。对于多种难题是真的，但是最重要的是加密和消息认证。这里的安全性证明是完全的：不对攻击者的计算能力做任何限制。但是能够简单控制攻击者攻击成功的上界。

作为一个例子，讨论一个 4.4.3 节，考虑一个 CTR 模型运算模式。这里考虑初始向量的版本。

用一个随机假设值（invocation）代替每一个 E_k 的假设值（假设 l=L）。在这种情况下，一个运算模式转化成一次一密密码系统。共享随机密钥仅仅是一个随机函数 *f*。如同讨论过的，这是非常著名的强壮的安全体制。因此，在共享随机判断机制模型下，CTR 模式容易看作是好的模式。

到目前为止有哪些进展呢？有一个事前并没有充分判断到的体制，因为它们都是随机函数。也就是说伪随机函数和随后的置换簇。一个 PRF 簇是一个使用小密钥（56 比特或 128 比特）索引出的 F 函数簇。然而，如果有特征，K 是 Alice 和 Bob 共享的，在共享随机函数模型条件下的一些设计中，使用 F_k 代替随机函数 *f*，最终的设计只要在攻击者控制的资源有限时始终是安全的。

换言之，在密钥共享体制中 PRFs 的实例可以用来代替随机函数，PRF 的定义是把这个体制使用在尽可能宽的范围。一个伪随机函数的例子是用便于多方存储的一个小的密钥来说明的。那么，在这个密码体制中使用这个函数来代替随机函数。并且事件应该是可以求出的，应该知道，当随机函数使用时，这个体制应该是安全的。

这是一个很概略的思想。技术上，并不一直是真的：这只是直观印象。伪随机函数不是总可以达到随机函数的性能测试的。也就是说，不能用随后的使用和期望值对随机函数进行代替。但是如果使用正确，可以用大数运算。怎样确定这些事件。必须采取伪随机函数簇的正规定义，基于它的结构安全。将给出证明。

在这个上下文中，强调重点。PRF 的安全依赖于密钥安全，由于攻击者不能得到密钥，因此无法计算出函数。（当然，通过合法的用户使用 F_k ，也许可以得到一些在变化的点上

关于 F_k 的值的信息，当然这是合理的）。换言之，可以用 PRFs 代替共享、秘密随机函数，但不是公共的一方。

随机函数是一个吸引人的概念，一个有利的工具是随后设计的范式。当想要设计一个体制用来加密、认证、或一些目的，在共享随机函数模式中，然后使用一个伪随机函数简单代替随机的一个，那么的体制仍然是安全的。

5.5.4 分组密码的模型建立

主要动机之一是为了伪随机函数（PRFs）的概念并且伪随机置换（PRPs）是分组密码的模型并且因此能够使用分组密码的安全的协议分析。如同 4.6 节中的讨论，经典的 DES 的安全或其分组密码使用看做仅仅密钥恢复。也就是说，分组密码 F 的分析集中在如下的问题：给定输入、输出的例子，

$$(x_1,F_k(x_1)),\ldots,(x_q,F_k(x_q))$$

这里 K 是随机、未知的密钥，发现 K 有多困难？分组密码被认为是安全的，如果用以恢复密钥的资源是有限的。然而，如同所知，甚至对普通分组密码粗略的研究也证明了密钥恢复的难度不能保证安全性。已经想到了使用分组密码的主要特征是分组密码算法提供安全特性的自然使用方法。建议在 CPA 和 CCA 条件下，主要特征是分组密码是一个安全 PRP。

不能证明特殊的分组密码有这个特征，能做的最好的假设是可以做到，并且可以继续用。可以假设说明各种函数的优势函数是数量安全的，例如可以假设推测：

$$Adv_{DES}^{prp\text{-}cpa}(t,q,64q) = C_1 \cdot (t/T_{DES})/2^{55} + C_1 \cdot (q/2^{40})$$

这里，在的固定 RAM 模型，T_{DES} 是做 DES 计算时间，并且 C_1 是一些常数。换言之，猜想既不是穷尽密钥搜索也不是线性攻击。也许应该大胆的假设作为 AES 并且猜测：

$$Adv_{AES}^{prp\text{-}cpa}(t,q,128q) = C_1 \cdot (t/T_{AES})/2^{1128} + C_1 \cdot (q/2^{128})$$

也可以假设在 CPA，CCA 而不是在 CPA 的条件下，可以做分组密码强度的类似的猜测。更有兴趣的是 $Adv_{DES}^{prf}(t,q)$。这里不能做更好的假设：

$$Adv_{DES}^{prp\text{-}cpa}(t,q,64q) = C_1 \cdot (t/T_{DES})/2^{55} + q^2/2^{40}$$

$$Adv_{AES}^{prp\text{-}cpa}(t,q,128q) = C_1 \cdot (t/T_{AES})/2^{1128} + q^2/2^{128}$$

这就是随后讨论的生日攻击。因为分组密码是一个置换簇并且是一个普遍现象，这种攻击是可能的。

强调这里所有的猜测，不包括生日攻击，这是基于这样的假设，对从已知密钥恢复攻击提供对密码主算法的伪随机性最佳的攻击。然而，应该存在更好的攻击是无须还原密钥，作为 PRF 模式来破译密码。到目前为止，还不知道这样的攻击方法，但是密码分析的努力在于目标集中在小范围。当然，假设一个分组密码对于抗密钥恢复攻击是安全的，是较 PRF 的更强的假设。但是，勾画出的动机和结论是更倾向于 PRF 假设的观点在于，如果在 PRF 条件下分组密码算法是可破的，那么，应该是不安全的并可以找到替代品。

5.6 攻击例子

假设距离说明提供攻击者的模式，在这种模式下提供的不同函数簇。

例 5.9、定义一个如下函数簇 F：$\{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 。假设 k=Ll 并且观察一个 k 比特密钥作为详细说明：一个 L 行 l 列的比特矩阵，（具体的说，假设矩阵的第一列指定密

钥 K 的最先的 L 比特，K 的下 L 比特说明了矩阵的第二列），输入的串 x=x[1]…x[l]作为比特串，F（K，x）的值是与矩阵向量的积相应的。也就是说，

$$F_k(x)=\begin{bmatrix} k[1,1] & k[1,2] & \cdots & k[1,l] \\ k[2,1] & k2,2] & \cdots & k[2,l] \\ \cdots & & & \cdots \\ k[L,1] & k[L,2] & \cdots & k[L,l] \end{bmatrix} \bullet \begin{bmatrix} x[1] \\ x[2] \\ \cdots \\ x[l] \end{bmatrix} = \begin{bmatrix} y[1] \\ y[2] \\ \cdots \\ y[L] \end{bmatrix}$$

这里，

$$Y[1]=k[1,1]\cdot x[1]\oplus k[1,2]\cdot x[2] \oplus\ldots\oplus k[1,l]\cdot x[l]$$
$$Y[2]=k[2,1]\cdot x[1]\oplus k[2,2]\cdot x[2] \oplus\ldots\oplus k[2,l]\cdot x[l]\ldots$$
$$Y[L]= k[L,1]\cdot x[1]\oplus k[L,2]\cdot x[2] \oplus\ldots\oplus k[L,l]\cdot x[l]$$

矩阵里的比特数是密钥中的比特，是模二的。的问题是 F 是否是安全 PRF，提出的问题是不。原因在于，一个人可以设计一个攻击算法 A 得到一个接近 1 的最大的优势在两个条件下是可区分的。观察是对任意的密钥 K 有 $F_k(0^l)=0^L$。这是有弱点的因为 l 比特到 L 比特的输入为 0^l, 输出为 0^L 随机函数是不必要的，因此这个事实在于不可区分的攻击者，看看攻击者是怎样工作的。记住，每一个的模型是给定一个随机判断机 $g : \{0,1\}^l \rightarrow \{0,1\}^L$ 并且将输出一比特，的攻击者 D 如下工作：

攻击者 D^g

假设 $y \leftarrow g(0^l)$

如果 $y=0^L$ 返回 1，否则返回 0

攻击者在点 0^l 询问随机判断机，并且用 l 比特表出 y 返回。如果 $y=0^L$ 可以确定 g 是 F 函数簇的特例，如果 $y\neq 0^L$ 则确定 g 是随机函数。攻击者具备怎样的能力。假设，

$$P[Exp_{F,D}^{prf-1} = 1] = 1$$
$$P[Exp_{F,D}^{prf-0} = 1] = 2^{-L}$$

为什么？看看在 5.4 中定义的 $Exp_{F,D}^{prf-1}$ 经验值。这里，对一些 $K,g=F_k$ ，这种编码情况下，$g(0^l)=0^L$ 显然为真，随后将返回 1。另一方面，看看经验值 $Exp_{F,D}^{prf-0}$ 在 5.4 中定义，这里 g 是一个定义域函数，如同在例 5.3 中所见，定义 $g(0^l)=0^L$ 的概率为 2^{-L}。因此 D 的概率值将返回 1。现在每定义 5.4 可以得到：$Adv_{F,D}^{prf} = P[Exp_{F,D}^{prf-1} = 1] - P[Exp_{F,D}^{prf-0} = 1] = 1- 2^{-L}$

假设 D 的时间复杂度是 t，这是 F 的一个置换加法的时间 O（l+L），最多为 O（l^2L）。使用 D 的数据查询仅仅为 1，序列的总长度为 l。因此有：$Adv_F^{prf}(t,1,l) = \max_A \{ Adv_{F,A}^{prf}\} \geq Adv_{F,A}^{prf} = 1-2^{-L}$

第一个不等式成立是因为攻击者 D 是攻击者集合 A 的一个成员，在这上面取得最大值。因此，至少可以得到 D 的最大值的结论，在 PRF 条件下，F 的攻击函数值是高的，甚至在输入参数源值是很低的条件下，也就是说在 PRF 条件下，显然 F 是不安全的。

例 5.10 、假设给定一个安全 PRF F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 。想使用 F 设计一个 PRF G：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^{2L}$ 。G 的输入长度是相同的，不同之处在于 G 的输出函数是 F 的两倍。假设如下的候选结构：对每 k 比特的密钥 K 并且每 l 比特的输入 x：$G_k(x)=F_k(x)||F_k(\bar{x})$

这里||定义为序列串联，$\bar{x}$ 表示串 x 面向比特的取补。要讨论这是否是一个好的结构，“好”的意思是当 F 是 PRF 安全时，G 也是 PRF 安全的。如果不成立，不考虑 F 的性质，结构 G 是不安全的。让来证明这点。

列一个攻击 G 的清单，既然 G 是 l 比特到 2L 比特的映射的特例，攻击者 D 将得到一个 g 函数的随机判断机制是 l 比特到 2L 比特的映射。在条件 1，g 是 G_k 的集合，这里 k 是一个随机 K 比特密钥。攻击者必须判断确定哪一个条件是可以替代的，的攻击工作如下：

攻击者 D^g，

假设 $y_1 \leftarrow g(1^l)$

假设 $y_2 \leftarrow g(0^l)$

把 y_1 分列为 $y_1 = y_{1,1} \| y_{1,2}$ 这里， $|y_{1,1}| = |y_{1,2}| = L$

把 y_2 分列为 $y_2 = y_{21} \| y_{2,2}$ 这里， $|y_{2,1}| = |y_{2,2}| = L$

如果 $y_{1,1} = y_{2,2}$ 那么返回 1，否则返回 0

攻击者在点 1^l 讯问随机判断机，返回 y_1，点 0^l 讯问随机判断机制，返回 y_2,注意 1^l 是面向比特的 0^l 的补。攻击检测只需比较 y_1 的前半部分是否与 y_2 的后半部分相同，如果相同则可以判定属于条件 1。让看看攻击者可以做到什么程度：

$$P[Exp_{G,D}^{prf-1} = 1] = 1$$

$$P[Exp_{G,D}^{prf-0} = 1] = 2^{-L}$$

为什么？看看在 5.4 中定义的 $Exp_{G,D}^{prf-1}$ 经验值。这里，对一些 K,g=G_k

$$G_k(1^l) = F_k(1^l) \| F_k(0^l)$$

$$G_k(0^l) = F_k(1^l) \| F_k(0^l)$$

对于 G 簇的定义，如果比较 $G_k(1^l)$前半部分与 $G_k(0^l)$的后半部分相同，则 D 将返回 1。

与 5.4 中定义相同，从另一方面如定义 5.4 的经验值 $Exp_{G,D}^{prf-0}$。这里，g 是一个随机函数。这样 $g(1^l)$和 $g(0^l)$都是随机并且独立的 2L 比特串。串的前半部分和后半部分相同的概率是多少呢？这个概率就是选择 L 比特串完全相同的概率为 2^{-L}，这是 D 返回 1 的概率。现在每定义 5.4 可以得到：$Adv_{F,D}^{prf} = P[Exp_{F,D}^{prf-1} = 1] - P[Exp_{F,D}^{prf-0} = 1] = 1 - 2^{-L}$

现在假设 D 的时间复杂度是 t，这是 G 的一个置换加法的时间 O（l+L），最多为 O（l+L）再加上四个 F 函数的计算时间。使用 D 的数据查询仅仅为 2，序列的总长度为 2l。因此有：

$$Adv_G^{prf}(t,2,2l) = \max_A \{ Adv_{G,A}^{prf} \} \geq Adv_{G,A}^{prf} = 1 - 2^{-L}$$

的结论为在 PRF 条件下，G 的攻击函数值是高的，甚至在输入参数源值是很低的条件下，也就是说在 PRF 条件下，显然 G 是不安全的。

练习 5.11 假设给定一个安全 PRF F：$\{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 。想使用 F 设计一个 PRF G：$\{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^{2L}$ 。如果 F 是安全的，那么 G 是 PRF 安全的。更简易，可以假设 k=l=L。

5.7 抗密钥恢复攻击安全

已经提到几次，对分组密码的安全概念，抗密钥恢复攻击安全是不充分的。当然，这是必然的：如果密钥恢复攻击是容易的，分组密码算法被称做是不安全的。已经指出，想在 PRF 或 PRP，采用安全的分组密码概念。在这一节，证明这个事实。这样做将保证归约的练习方法。

开始抗密钥恢复攻击的形式化安全。认为攻击者基于输入输出的 F 函数簇的一个例子 F_k，尝试找到密钥 k。这种形式的优点是成功发现密钥的概率是在基于 K 的随机选择之上的，随机选择的权利掌握在攻击者手中。

给出攻击判断机制模拟 F_k ，这样可以得到选择输入输出例子。不去限制攻击者所使用的方法，这里得到如下定义。

定义 5.12 假设函数 F：Keys(F) ×D→R 是一个函数簇，假设 B 是一个算法对一个函数

的随机判断机制 g：D→D，并考虑实验：

Experiment $Exp_{F,B}^{kr}$

K ←Keys(F)

K'←B^{Fk}

如果 K=K'那么返回 1 否则返回 0

B 的 Kr-prf 优势定义如下：

$$Adv_{F,B}^{kr} = P\,[Exp_{F,B}^{kr} =1]$$

对于任意 t,q,μ，定义 F 的 Kr-prf 优势为：

$$Adv_{F,B}^{kr}(t,q,\mu)= \max_{B} \{ Adv_{F,B}^{kr}\}$$

这里 B 上的最大量包括时间复杂度 t，随机判断序列 q, 这些序列取 μ 比特时序列长度总和。

对于各种类型的密钥恢复攻击，定义已经被普及化。密钥穷尽搜索的各种攻击，对攻击者 B 线性分析和差分分析是不同的选择。对密码算法 F_k 的某些输入输出进行密钥还原可以定义在这个框架内。以 DES 为例阐述使用 DES 偏差优势进行古典密钥值恢复攻击。假设测试两种攻击的穷尽密钥搜索总是成功的，那么：$Adv_{F,B}^{kr}$（t,2,2·64）=1

对于计算一次 DES 时间 t 的 2^{55} 次的时间 T_{DES} 。另一方面，线性分析暗示为：$Adv_{F,B}^{kr}$（t,2^{43},2^{43}·64）=1

对于 t 大概是 $2^{43}\cdot T_{DES}$ 。这个给了一对 $Adv_{F,B}^{kr}$ 曲线上的数据，更多具体的例子是例 5.9 簇的密钥恢复优势。

例 5.13、假设 F：$\{0，1\}^k \times\{0,1\}^l \rightarrow\{0,1\}^L$ 是一个例 5.9 的函数簇。可以看到这种模式的 prf-优势是非常高的，现在计算 kr-优势。以下是攻击者 B 恢复密钥。假设 e_j 是 l 比特串在 j 处值为 1，而其处值为 0。假设密钥 K 定义的矩阵是 K 的前 l 比特的模式是从矩阵的第一列开始，接下来 K 的 l 比特从矩阵的第二列取，其他的比特位按照这个方法获得。

假设 K'是一个空序列，

从 j=0,…,l 反复

$y_j \leftarrow F_k(e_j)$

K' ← K' ‖ y_j

结束循环

返回 K'

攻击者 B 调用随机判断机制计算在输入 e_j 时的函数输出，结果 y_j 是密钥 K 条件下的矩阵的 j 列确切值。矩阵输入链接产生 K'，返回如下的密钥。既然攻击者总是找到密钥，有：$Adv_{F,B}^{prf} = 1$

攻击者的时间复杂性为 t= $O(l^2L)$，因为 l=q，所以称为随机语言机制，F_k 的每个计算占用时间为 O(lL)，因此，$Adv_{F,B}^{kr}(t,l,l^2)=1$

参数因此仍然是小的，l 是 64 或 128，对于序列值是小的。因此 F 对于密钥恢复攻击是安全的。注意 F 作为 PRF 并不比抗密钥恢复攻击安全，对于上述足够小的值，作为 PRF 条件下的优势函数是接近 1 的值。这个思想接近下述的描述：对于任意给定的参数值，一个簇的 kr-优势不能比 prf 和 prp-cpa 优势明显。

现在可以宣称如果分组密码 PRF 和 PRP，那么对抗密钥恢复攻击仍然是安全的。从另外一个角度，F 的密钥恢复攻击优势不能比 PRF 的优势大。

命题 5.14、假设 F：$\{0，1\}^k \times\{0,1\}^l \rightarrow\{0,1\}^L$ 是一个函数簇。那么对于任意的 t,q,并且 $q\le 2^l$，有：

$$Adv_F^{kr}(t,q,ql)\le Adv_F^{prf}(t',q+1,(q+1)l)+1/2^L \qquad (5.1)$$

此外，如果 l=L，有：

$$Adv_F^{kr}(t,q,ql) \le Adv_F^{prp\text{-}cpa}(t',q+1,(q+1)l)+1/(2^L-q) \quad (5.2)$$

这里设置 t'为 t 加 F 函数的一个计算时间。

该证明提出了一个归约的重要思想，将展示一个 B→A_B 的一个变换，一个 kr 攻击人 B 到 A_B 的一个对应，使得：$Adv_{F,B}^{kr} \le Adv_{F,AB}^{prf} + 1/2^L$

并且如果 B 可使用的资源包括 t,q,ql，那么 A_B 使用的资源为 t',q+1,(q+1)l。称为非操作，这证明了命题的第一等式。当然，对等式两边取最大，将随后看到得到该问题的等式。

问题是攻击者 A_B 试图确定在哪一个条件可以得到一个随机判断机制 F_k ,并且试图查询序列找到一个 K。简而言之，第一假设是基于 B 不是随机判断机制。目前，当 A_B 趋向 B，可以得到一些密钥 K'。A_B 可以通过检查是否 F（K'，x）满足在某些 x 点的值 g(x)测试 K'。如果这样，可以相信 g 还是属于 F，如果不是，可以假设 g 是随机的。

如果 B 的确是随机判断机制，必须询问怎样使 A_B 跑遍整个 B。随机判断机制是 B 的假设，是不可用的。然而，B 是一个代码，通过一个规定的界面与随机判断进行通信。如果开始运行 B，在一些点将输出一个随机判断询问，也就是说把这些写入指定的记忆存储，停止。等待一个回答，提供另一个记忆存储位置。当回答出现，继续执行操作。提出判断询问时，将返回输出。现在 A_B 跑遍整个 B，该机制可以自己提供 B 的随机判断机制的问讯答案。当 B 停止时已经进行了一些询问，A 将提供规定记忆位置的回答，然后让 B 继续进行操作。B 从搜集的询问返回值中并不能得到仿真随机判断机制和真的随机判断机制之间的区别。B 期望对 x 的回答是 $F_k(x)$的评估。那不是 A_B 所定义的。当 A_B 在条件 1，$g(x)=F_k(x)$,并且 B 在与之适用的普遍的条件，使用等于 kr-优势的一个概率，然后返回密钥 K。然而， A_B 在条件 0，g 是一个随机函数，B 得到的值可以承担部分期望的关系值。这无关紧要。B 是可以完成和产生输出的部分代码，当处于条件 0，没有输出特征的观点。但这是一个 k 比特串，并且 A_B 将进行上述的测试。如果测试点 x 没有被 B 所咨询，那么失败的概率会非常高，并且 A_B 将通过 x 的选择随后得到确定，现在进行实际的证明。

命题 5.14 的证明、先证明第一个问题，然后改变第二个等式证明。

将表示对于给定任意的攻击者 B，所控制的资源包括 t,q,ql，构造一个攻击者 A_B，可控制的资源是 t',q+1,(q+1)l，那么

$$Adv_{FB}^{kr}(t,q,ql) \le Adv_{FAB}^{prf}+1/2^L \quad (5.3)$$

如果这为真，可以建立如下的等式：

$$Adv_F^{kr}(t,q,\mu) = \max_B \{Adv_{F,B}^{kr}\}$$

$$\le \max_B \{ Adv_{F,AB}^{prf} + 2^{-L}\}$$

$$\le \max_B \{ Adv_{F,A}^{prf} + 2^{-L}\}$$

$$= Adv_F^{prf}(t,q+1,(q+1)l)+2^{-L}$$

B 情况下的最大值是对所有具备 t,q,ql 能力的攻击而言的。在第二行，应用等式 5.3。在第三行，对拥有 t,q+1,(q+1)l 计算资源的 A 进行最大化。第三行的不等式是所有型如 A_B 的攻击者的条件。最后一行简单的从定义中得到。这样剩余的问题就是怎样设计 A_B，这样保持等式 5.3。（这是争论的核心，也正是所谓的归约）

在每个定义 5.4，攻击者 A_B 将对函数 g：$\{0,1\}^l \to \{0,1\}^L$ 进行随机判断的归约，并且将试着决定属于哪个条件。为了这样做，将 B 运行为一个子程序。提供如下的描述来进行解释和分析：

攻击者 A_B^g

I←0

运行攻击者 B，应答随机判断提问

当 B 设置一个随机询问 x，做

I←i+1 ; x_i←x

y_i ← g(x_i)

返回 y_i 到 B 作为回答

直到 B 停止并且输出一个密钥 K'

假设 x 是一个 l 比特串不在子集{x_1,…,x_q}

y← g(x)

如果 F（K'，x）=y 那么返回 1，否则返回 0

一个讨论前述证明的问题在于 A_B 在 B 上运算并且通过判断机 g 提供 B 的随机判断机的回答。当 B 运算完成，返回 k 比特串 K'，A_B 使用 F（K'，x）是否与 g(x)符合检测测试。这里 x 是一个 B 的任意的询问中得到的值，并且可以保证这样的值是可以通过询问 $q<2^l$ 进行命题陈述得到确定。现在证明：

$$P[Exp_{F,AB}^{prf-1} = 1] \geq Adv_{F,B}^{kr}$$

$$P[Exp_{F,AB}^{prf-0} = 1] = 2^{-L}$$

将简要证明这些命题的正确性，但是会首先使用该结论。如定义 5.4 的减法，得到：

$$Adv_{F,AB}^{prf} = P[Exp_{F,AB}^{prf-1} =1]-P[Exp_{F,AB}^{prf-0} =1] \geq Adv_{F,B}^{kr}-2^{-L}$$

重新编排等式 5.3，并且证明 5.4 的正确性。

等式 5.4 是正确的，因为 $Exp_{F,AB}^{prf-1}$ ，对于密钥 K 随机判断机 g 是满足的，是 B 期望的随机判断机，因此 B 函数在 Exp_{FB}^{kr} 中。如果 B 成功，意味着密钥 K'的输出等于 K，当然 A_B 返回 1，（当然 A_B 也是可以返回 1 的，甚至当 B 是不成功的。这种情况满足如下条件时发生：K'≠K 但是 F（K'，x）=F(K，x)。基于这个原因，P [$Exp_{F,AB}^{prf-1}$ =1]大于或等于 $Adv_{F,B}^{kr}$ 比等于的因素强。）等式 5.4 是真的，因为在条件 $Exp_{F,AB}^{prf-0}$ 下，g 函数是随机的，因此 x 永不被 B 询问。假设 g(x)是 2^{-L} 的概率击中这个固定点。注意这是真的，无论 B 使 F（K'，x）与 g(x)相同的难度。

对于等式 5.2 的证明，寻找一个归约 B→A_B 具备如下的特征：

$$Adv_{F,B}^{kr} \leq Adv_{F,AB}^{prp-cpa} + 1/(2^L-q) \qquad （5.4）$$

这个归约是与上述相同的，意味着攻击者 A_B 是相同的。对于分析者，有：

$$P[Exp_{F,AB}^{prp-cpa-1} = 1] = Adv_{F,B}^{kr}$$

$$P[Exp_{F,AB}^{prp-cpa-0} = 1] \leq 1/(2^L-q)$$

减法产生：

$$Adv_{F,AB}^{prp-cpa} = P[Exp_{F,AB}^{prp-cpa-1} =1]-P[Exp_{F,AB}^{prp-cpa-0} =1]$$

$$\geq Adv_{FB}^{kr} -1/(2^L-1)$$

并且重新排列后给出等式 5.4，上述第一个等式为真，如果相同条件成立，第二个等式成立是因为 l 比特到 l 比特的随机置换 g 是在条件 0，这样，g(x)假定除去 y_1,…,y_q 的任意随机值，意味着有 2^L-q 种可能，（记住条件 L=l）。

随后的例子说明上述相反的命题是不成立的，一个簇的 kr-优势是可以明显比 prf,prp-cpa 优势小，也就是说意味着一个簇对于密钥恢复攻击是非常安全的，但是对于 prf,prp 是不安全的，因此对于协议的设计是不安全的。

例 5.15、 定义对于 $E_k(x)=x$，分组密码 E：{0，1}^{k} ×{0,1}^{l} →{0,1}^{l}，对于所有 k 比特密钥 K 和所有 l 比特的输入 x，可以证明对于密钥恢复攻击是非常安全的，但是在 CPA 条件下作为 PRP 是不安全的。更详细的，对于所有的 t,q 值，是比较高的：$Adv_E^{kr}(t,q,ql)=2^{-k}$

并且另一方面，$Adv_E^{prp-cpa}(t,1,l) \geq 1-2^{-l}$

对于 t=O(l),简而言之， E_k 给定一个随机判断机制，可以设置任意想用的询问，g：$\{0,1\}^l \rightarrow \{0,1\}^l$，并且使用非常少的时间，可以确定的是否 g 是安全函数 E 或是 l 比特到 l 比特的一个随机函数。为什么总是声称为真呢？既然 E_k 并不依赖密钥，使用随机判断机制的 E_k 的攻击者变为通过询问不能得到信息，因此，对于密钥 K 的猜测仅仅能以 2^{-K} 的概率进行。另一方面，一个攻击者可以测试是否 $g(0^l)=0^l$，并且返回 1，当且仅当攻击成立，得到一个 $1-2^{-l}$ 的优势。

5.8 生日攻击

假设 E：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ 是一个置换簇，意味着是一个分组密码。如果给定一个随机判断机制 g：$\{0,1\}^l \rightarrow \{0,1\}^l$，是一个 E 的模型或随机函数，这是一个简单的测试。在一定距离点的 $x_1,x_2,\ldots x_q$ 判断机制并且给定返回值 $y_1,y_2,\ldots y_q$。已知 g 是一个置换，值 $y_1,y_2,\ldots y_q$ 必须是不同的，同时，如果这些值不同，仅仅是一个置换。

令人意外的，这是非常好的一个辨别机制，将做如下的讨论。粗略的，可以使用 $q=\sqrt{2^l}$ 次询问得到非常接近 1 的优势。生日攻击的矛盾是原因。如果对这点并不熟悉，可以参考 A.1 节，然后得到如下的反馈。

这提供了一个分组密码的模型，以接近 $2^{l/2}$ 的概率做一个输入输出的例子，这对基于协议的分组密码的安全性是非常重要的因素。

命题 5.16、 假设 E：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ 是一个置换簇，假设 q 满足

$$2 \le q \le 2^{(l+1)/2}$$

那么：$Adv_E^{prf}(t,q,ql) \ge 0.3 \cdot (q(q-1))/2^l$，这里 t 是使用 q 计算 E 的时间，加 O(ql)。

命题 5.16 的证明：攻击者 D 使用生日攻击，给定一个随机判断机制 g：$\{0,1\}^l \rightarrow \{0,1\}^l$，按照如下流程工作：

Adversary D^g

For i=1,…,q 做

假设 x_i 是按照字典阶的第 i 个 l 比特串

$Y_i \leftarrow g(x_i)$

结束

如果 $y_1,y_2,\ldots y_q$ 是不同的，返回 1，否则返回 0

则：$Adv_{E,D}^{prf} \ge 0.3 \cdot (q(q-1))/2^l$

命题如下，下面证明这个边界。假设 $N=2^l$，证明：

$$P[Exp_{E,D}^{prp-1}=1]=1 \qquad (5.5)$$

$$P[Exp_{E,D}^{prp-0}=1]=1-C(N,q) \qquad (5.6)$$

这里 C(N,q)的定义见 A.1 节，一些用二进制得到两位或更多数的概率，把 q 种可能划分为 N 种类型的经验值。将简单的证明这点，首先使用这个推断。这样得到：

$$\begin{aligned} Adv_{E,D}^{prf} &= P[Exp_{E,D}^{prf-1}=1]-P[Exp_{E,D}^{prf-0}=1] \\ &= 1-[1-C(N,q)] \\ &= C(N,q) \\ &\ge 0.3 \cdot (q(q-1))/2^l \end{aligned}$$

最后一行是 A.1 的推论，只剩余了等式 5.5，5.6 的正确性。

等式 5.5 是显然的，因为在条件 1，$g=E_k$,因为 E 是一个置换簇，并且 $x_1,x_2,\ldots x_q$ 是不同的，而 $y_1,y_2,\ldots y_q$ 是随机的, 独立分布在$\{0，1\}^l$。来看看生日难题。投 q 个球到 $N=2^l$ 个盒子中，对于没有碰撞的概率，也就是说没有任何一个盒子有两个或两个以上的球在同一个盒子

中的概率。这就是等式 5.6 中 1-C(N,q)的含义。

5.9 PRFs 与 PRPs

当分析基于结构的分组密码，发现了一个重要的二分法。分析非常简单并更自然的假设分组密码是一个 PRF。当然，PRPs 是更自然的分组密码模型。为了沟通这个分歧，对给定的分组密码设置相应的 prf 和 prp-cpa 优势函数，如下简单的说明生日攻击是最好可能的方法，一个特别的置换 E 簇可能比 prp-优势更有 prf-优势的，但是只通过一个数量 $q(q-1)/2^{l+1}$，碰撞概率限制在生日攻击范围内。

命题 5.17 假设 E、$\{0，1\}^k \times\{0,1\}^l \rightarrow\{0,1\}^l$ 是一个置换簇, 那么对于任意的 t,q：$Adv_E^{prf}(t,q,ql) \leq$ （$q(q-1)$）$/2^{l+1} + Adv_E^{prp-cpa}(t,q,ql)$

再次使用归约证明，是非常简单的一种，一个给定的 prf-攻击者 A 是一个到 prp-攻击者的映射，这意味着攻击者并没有改变。相应的，以下给出归约明确的一个确定谈话。

证明：假设 A 是一个攻击者并且感受了一个函数 g 的随机判断机制: $\{0,1\}^l \rightarrow\{0,1\}^l$。那么证明，$Adv_{E,A}^{prf} \leq Adv_{E,A}^{prp-cpa} +$（$q(q-1)$）$/2^{l+1}$ （5.7）

这里 q 是 A 的一个随机判断机，对命题取最大值，这样证明等式（5.7）。

假设 B 表示一个攻击者首先运算 A 得到输出比特 b 然后返回 $\bar{b}$，对 b 取补，那么：

$$
\begin{aligned}
Adv_{E,A}^{prf} &= P[Exp_{E,A}^{prf-1}=1]-P[Exp_{E,A}^{prf-0}=1] \\
&= (1- P[Exp_{E,B}^{prf-1}=1])-(1- P[Exp_{E,B}^{prf-0}=1]) \\
&= P[Exp_{E,B}^{prf-0}=1])- P[Exp_{E,B}^{prf-1}=1] \\
&= P[Exp_{E,B}^{prf-0}=1])- P[Exp_{E,B}^{prf-cpa-0}=1]+ P[Exp_{E,B}^{prf-cpa-0}=1]- P[Exp_{E,B}^{prf-cpa-1}=1] \\
&= P[Exp_{E,B}^{prf-0}=1])- P[Exp_{E,B}^{prf-cpa-0}=1]+ Adv_{E,A}^{prp-cpa}
\end{aligned}
$$

这样满足 $P[Exp_{E,B}^{prf-0}=1])- P[Exp_{E,B}^{prf-cpa-0}=1] \leq$（$q(q-1)$）$/2^{l+1}$ （5.8）

假设 P[·]表示经验值 $Exp_{E,B}^{prf-0}$ ，并且假设 g 表示一个基于经验的随机判断机，假设没有 A 到随机判断机的普通损失研究，所有 A 的随机判断序列是与 B 的某些部分显然相同，假设 D 表示询问这个事件的所有回答不同，假设 $\overline{D}$ 表示整个 D 事件的补，那么：

$$P[Exp_{E,B}^{prf-0}=1]= g[B^g=1] = g[B^g=1|D]\cdot P[D] + g[B^g=1|\overline{D}]\cdot P[\overline{D}]$$

$$\leq g[B^g=1|D] + P[\overline{D}] = P[Exp_{E,B}^{prf-cpa-0}=1] + \ （q(q-1)）/2^{l+1}$$

在最后一步，使用命题 A.1，重新安排(5.8)的期限并且得到证明的推论。

5.10 PRF 簇的构造

哪里可以找到 PRFs？有多种多样的方法，从伪随机比特产生器或单向函数中构建，一个保守的但是数据传送效率低的方法。也有更多的有用的构建，安全基于特殊的数论理论难题。最后，实际上，如上所述，将假设形如 AES 的分组密码所具备的性质。伪随机比特产生器的概念在第三章讨论了。会议多项式时间的计算函数 G，拿走 k 比特种子并且产生 p(k)>k 比特序列，对无论任何有效的测试看起来是随机的。

PRF 簇的首先的构建从 PRBG 的长度是双倍：输出长度是输入长度的两倍。

定理 5.18[97]给定一个双倍长度的随机数比特产生器，可以构造一个序列簇 F 是一个

PRF。构建，叫做如同这样的二差树构造，使用如下的函数 G 导出一个函数 G_z：

定义 G_0（x）·G_1(x)=G(x) 这里，k=|G_0(x)|=|G_1(x)|

定义 $G_{Z\cdot0}$（x）·$G_{z\cdot1}$(x)=G_z(x)这里，k=|$G_{Z\cdot0}$|=|$G_{z\cdot1}$|

那么 f_i(x)是由 G 导出的二分树术语，任意 x，f_i(x)=G_x(i)。现在假设 F={F^k}$_{k\geq1}$ 这里，F^k 是{ f_i: {0，1}^{k}→{0，1}^{k}||i|=k}。也就是说[97]中的结构是安全的。另一个是基于合成构造由 Naor 和 Reingold[150]，这里产生了一个基于构造的 PRBG 是比基于一个二分树更容易并行的。

之前看到，从单向函数[115，111]构造 PRGB，根据上述观点，可以从单向函数构造 PRF 函数簇。另外，看到给定的任意伪随机函数簇，构造单向函数[114]，因此有如下的定理：

定理 5.19 存在一个序列簇是一个分组密码 PRF 当且仅当存在一个单向函数。

这里有一个非常强的假设。单向函数是一个比较弱的原则，一个与很优先的 PRFs 相关的理论。当然，一个可以转化成另一个。构造的不恰当对于实践是不充分的。由 Naor 和 Reingold 已经建议了一个序列簇的构造，F={F^n}$_{n\geq1}$ 。可以证明一个 PRFs 假设 DDH（Diffie-Hellman 决定），问题是非常困难的[151]。对于这种结构，从 F^n 到一个 l(n)比特输入序列的特殊函数的构造是一个模乘和一个模求幂，在一个潜在的组的上面。

5.10.1 扩展定义域大小

假设使用长度 l 开始 PRF 簇。通常需要用扩展定义域（对于这个重要的原因是做一个好的认证码，见定理 8.6）。有各种方法完成，比如分组密码算法的 CBC 模式。

给定一个有限输入输出都是 l 的 PRF 簇，整数 m 是固定的，并且希望构造一个函数映射{0，1}^{lm} 到{0，1}^{l}。相似的构造中旧的密钥和新的密钥是相同的，新函数，给定 M_1，M_2，…，M_t,进行如下的操作：

假设 Y_0=0^l

假设 Y_1=f(M_1⊕Y_0)

假设 Y_2=f(M_2⊕ M_1)

…

假设 Y_M=f(M_M⊕ M_{m-1})

输出 Y_m

假设 $F^{(M)}$ 定义一个函数簇，从一个函数映射{0，1}^{lm} 到{0，1}^{l} 获取密钥，同时使用 f=F_k 给定 CBC 模式。

定理 5.20：假设 l,m≥1，并且 q,t≥0 是整数， F：KeysF×{0,1}^{l}→{0,1}^{l} 是一个函数簇，那么：

$Adv_{F(m)}^{prf}(q,t)\leq Adv_F^{prf}(q',t')+1.5\cdot(q^2m^2/2^{l-1})$ (5.9)

$\leq Adv_F^{prp\text{-}cpa}(q',t')+q^2m^2/2^{l-1}$ (5.10)

这里，q'=mq 并且 t'=t+O(mql)

强调输入是 nl 比特，并且至多是 nl 比特，也就是说，构造不是安全的。还有其它一些构造，例如，[19]的迭代构造。也就是说，基于 XOR，MAC[11]的思想，在[30]给出了一个保护计数总和的构造。不同的构造对于安全和效率有不同的特征。

类似的，像扩展输出长度。这个变化是容易的，不需要从细节上讨论。

5.11 PRFs 的一些应用

5.11.1 密码的强单向函数

假设 P_1，P_2 是多项式时间的，这样任意的 x，P_1（x）>P_2(x)。定义 $F^{p1p2}=\{f:\{0,1\}^{P1(k)}\rightarrow \{0,1\}^{p1(k)}\}$。希望 HASH 函数对应到地址，这里|name|=P_1（k）并且|address|= P_2（k）。可以使用伪随机函数对 name 进行 HASH 处理，这样，Address=f_i(name)。

命题 5.21 如果存在一个单向函数，对于所有的多项式 P，并且对所有的整数 k 足够大，上一级的单向函数的碰撞量是 O（1/2$\sqrt{address}$ ）+1/P（k）。甚至，对于固定的安排，攻击者选择的名字接入以前的（name,address）对。

5.11.2 判断

一个判断测试 T（1^K）

1. 询问一个判断机 $f\in F_k$，找到（x_1,$f(x_1)$）,…(x_l,$f(x_l)$)
2. 输出一个测试 x，并且
3. 给定一个 y 这样使用概率 1/2，y=f(x)(否则，y 从{0，1}$^{|f(x)|}$-{f(x)}）。
4. 输出 1，如果 y=f(x), 否则输出 0。

F 通过判断测试 T，如果任意 Q∈Q[x],∃k_0, 任意 k>k_0，

Pr[T（1^k）在第三步给定 y 猜测正确]<1/2+1/Q（k）

上述伪随机函数通过所有的判断测试（假设存在一个单向函数）。

5.11.3 学习机制

定义一个空间 S 的概念并且概念 C⊆S。一个学习者给出一对（e_i,±$_i$），这里 e_i∈S 并且 ±$_i$=+ ⇔ e_i∈C。学习者要求确定是否给定一个 e∈S 是一个元素 C。

上述伪随机函数可以看出存在一个单向函数，有一个多项式时间不能学习的概念。（在这种条件下的概念因该是{x,f(x)} ⊆{x,y} ）

5.11.4 朋友或敌人身份识别

考虑在一个空战中两方的作战情况，每个计划希望识别朋友或敌人的潜在的目标。这个可以在如下的随机函数：

1. 一方已知所有计划
2. 为了识别一个目标，如果目标是一个朋友，发送一个随机数 r 并且希望收回 f_i(r)。

那么，虽然敌人希望看见许多形如（x,f(x)）的对，不能对还没出现的 y 计算 f(y)。

5.11.5 私钥加密

假设 A 和 B 私下协商一个 i。然后加密消息，A 产生一个随机序列 r，并且发送（r,f_i(r)⊕m），B 可以计算 f_i(r) ⊕ m⊕ f_i(r)=m，假设存在一个单向函数，体制对于选择密文攻击是安全的，也就是说当攻击者可以通过 r 的集合计算（r,f_i(r)），见第六章关于这部分更多的内容。

5.12 历史注记

伪随机函数的基本定义是由 Goldreich,Goldwasser 和 Micali[97]提出的，特别的这些作者介绍重要的辨别机制的概念，伪随机置换的概念是 Luby 和 Rackoff[138]提出的。这些工作是在复杂性理论或渐近结构下的，这里考虑单个簇，考虑簇序列的性质，对于每个定义 5.8 的安全。目前章节是进一步构造分组密码算法模型的动机及使用，称为“具体安全”，由[13]可以引出。定义 5.4 和 5.5 是从[13]中引出的，得到命题 5.16 和 5.17。

5.13 联系和问题

问题 5.22、假设 a[i]表示二元序列串 i 的第 i 比特，这里 1≤i≤|a|，n 比特序列串 a,b 的内积定义为：

$$<a,b>=a[1]b[1] \oplus a[2]b[2] \oplus \ldots \oplus a[n]b[n]$$

一个函数 F 簇：$\{0,1\}^K \times \{0,1\}^l \rightarrow \{0,1\}^L$ 可以保持内积，对于每个 $K \in \{0,1\}^K$ 并且不同的 $x_1,x_2 \in \{0,1\}^l - \{0^l\}$有：

$$<F(K,x_1),F(K,x_2)>=<x_1,x_2>$$

证明如果 F 可以保持内积，那么 $Adv_F^{prf}(t,2,2l) \geq 1/2 \cdot (1+1/2^L)$

对 $t=q \cdot T_F+O(\mu)$, 这里 T_F 是 F 执行一个计算的时间。在一个句子的说明，当 F 在 PRF 是不安全的，F 可以保持内积。

问题 5.23、假设 E：$\{0,1\}^K \times \{0,1\}^l \rightarrow \{0,1\}^l$ 是一个分组函数，E 的二次迭代是 E^2：$\{0,1\}^{2K} \times \{0,1\}^l \rightarrow \{0,1\}^l$，$E^{(2)}$（$K_1 \| K_2$，x）=E($K_1$,E($K_2$,x))

对于所有 $K_1,K_2 \in \{0,1\}^K$ 并且所有 $x \in \{0,1\}^l$，（这里 ‖ 表示串的并），定义：

$$Adv_{E2}^{prp\text{-}cpa}(t,q,lq) \leq Adv_E^{prp\text{-}cpa}(t,q,lq)$$

对于所有的 t,q，一个句子的解释是如果 E 是 PRP 安全的，那么 E^2 也是安全的。

第六章 私钥加密

私钥值也称对称集，认定共享一个密钥，并使用密钥渗透通信数据，对于使用各种安全特征（但是，共享密钥不是考虑的一部分），主要的安全目标是通信数据的私密性和可认证性，对称密码体制也叫私钥密码体制。对于共享密钥的合法用户可以得到数据安全，这是一个密码学规范的目标。

6.1 对称加密体制

一个对称加密体制说明一个加密体制，通知了发送者怎样用密钥函数处理一个数据的传送过程。同时也定义一个剥离函数通知接收者怎样从传输中恢复数据，以一些概率做证明。最后，有一个密钥产生算法需要共享，如下是正式的描述：

定义 6.1、一个对称加密体制 SE=（K，E，D）由如下三个算法组成：

1、 密钥产生算法 K 是随机函数，返回一个串 K，假设 Keys(SE)表示所有串序列的 K 输出的非零概率，所有这些元素的组成称为密钥。对于 K 运算的操作，记 K←k，K 定义为密钥的返回。

2、 加密算法 E，得到密钥 K∈Keys（SE）并且有一个明文表 M∈{0，1}*，返回一个密文 C∈{0，1}*∪{⊥}，算法是一个随机的和描述的，记做：C←E_k（M）。

3、 确定的解密算法 D 占用一些密钥 K∈Keys(SE)和一个明文 M∈{0,1}*,返回 M∈{0，1}*∪{⊥}，记：M←D_k(C)。

要求对任意的 K∈Keys（SE）和一些消息 M∈{0，1}*，如果 E_k（M）返回一个明文 C≠⊥那么 D_k（C）=M。

密钥产生算法，作为定义的标识，使用这些去选择密钥。典型的，密钥仅仅是一些长度的随机串，也就是说这是密钥表的长度。当双方想知道所使用的密表，假定通过占用 K 的资源产生 k。当然需要使用这种方式对手是否可以产生密钥是关心的问题，这是使用前提。

一旦拥有共享密钥，就可以加密消息进行传输了。为了加密明文 M，使用密钥的发送方或加密方和使用输入的 M 得到串序列称为密文。

加密算法也许即是随机的也是正式的。如果是随机的，则服从随机过程规律，使用这些规律，输入 K，M 计算出输出，每次调用算法，重新投掷货币。在输入相同的条件下，特别调用两次算法也许不会产生两个相同的输出。如果加密算法是正规的，依赖一个计数的全局变量进行操作，也许每个加密算法的假设是可升级的。因此，加密算法进行 K，M 的调用，计算一个基于 K，M 的密文和当前的计数值。然后更新计数值，储存新的计数值。（接受者不需要保存一个计数值，特别解密不需要使用任何的全局变量或召集各方的任意同步。）

当没有这样的计数器和全局变量，这个体制是不稳定的。在这个状态设计下加密算法不会是中间随机的。（如果仍然成立，则正好是随机算法，仅仅是给定随机数产生器。）在不固定的设计下，正如看到的，随机是安全的本质。

一旦密文表示可计算，这样就传送给接受者。随后可以用相同的密钥进行解密算法对密文进行恢复。也就是通过 M←D_k(C),解密算法既不是随机的也不是固定的。

许多加密算法限制了序列串，并且可以用来加密。（例如，也许一个算法可以仅仅使用分组长度 l 的正整数倍，并且仅用于标准长度到最大长度的明文加密）。这些归约机制被使用加密算法返回符号⊥当传送一个信息到归约机制中。在一个固定的体制下，存在一个典型的串序列，称为明文空间，例如 E_k（M）≠⊥，对于所有的 K 和所有的明文 M 空间。在固

定的体制下，是否 E_k（M）返回⊥并不仅依靠于 M 和可能的固定变量。例如，当使用计数⊕，典型的存在一个加密运算的数字限制，当计数器达到加密算法的一个固定值，返回⊥并且联合消息。

6.2 一些加密体制

开始使用一些例子说明。

设计 6.2、一次一密加密体制（也叫做完善密码体制）是 SE=（K，E，D）固定的并且是确定的。密钥产生算法简单的返回一个随机 kbit 串 K，当密钥长度 k 是一个体制的参数，这样密钥空间是 Keys（SE）={0，1}^{k}。加密者会保持一个计数 ctr 的零初始向量，加密和解密算法操作如下：

Algorithm E_k(M)
Let n=|M|
如果 ctr+n>k 那么返回⊥
For i=1 to n do
C[i]←K[ctr+i]⊕M[i]
For 循环结束
Ctr←ctr+n
C←C[1]…C[n]
Return(ctr,C)

Algorithm D_k((C,ctr))
Let n←|M|
For i=1 to n do
M[i]←K[ctr+i]⊕C[i]
For 循环结束
M←M[1]…M[n]
返回 M

这里，X[i]定义成二元比特串的第 *i* 比特。加密算法对消息与密钥进行 XOR 运算，使用当前计数器的值，以信息长度对计数器进行累加。Key 比特并不再使用，如果没有足够的密钥比特对消息进行加密，加密算法返回⊥。注意密文返回值包括了计数。这是为了在加密中可以使用。（回忆解密算法，如同定义 6.1 所述，必须是不稳定和确定的，这样不必考虑计数器）。

如下的机制几乎不依赖于置换簇（例如分组密码）或一个函数簇，使用消息的长度加密是一个对簇的分组长度整数倍的加密。相应的，如果不是这个状态加密算法返回⊥。事实上，可以首先适当加密消息，这样填补消息使得分组长度有一个整数倍，使用加密算法进行消息填充。填充函数是单射并且容易求逆。

设计 6.3、假设 E：$\{0,1\}^K\times\{0,1\}^l\to\{0,1\}^l$ 是一个分组函数，运算 ECB 模式产生一个不稳定的对称加密体制，SE=（K，E，D），密钥加密算法简单的返回一个分组密码的随机密钥。也就是说，选取一个随机 k 比特串密钥并返回，这样密钥空间是{0，1}^{k}。加密和解密算法如下：

Altorithm E_K(M)
If|M|<l 返回⊥
If|M| mod l≠0 返回⊥
把 M 分解为 M[1]…M[n]
For i=1,…,n do
C[i]←E_k(M[i])
EndFor
C←C[1]…C[n]
返回 C

Algorithm $D_{k©}$(C)
If|C|<l 返回⊥
If|C| mod l≠0 返回⊥
把 C 分解为 C=C[1]…C[n]
For i=1,…,n do
M[i]←E_k^{-1}(M[i])
EndFor
M←M[1]…M[n]
返回 M

分解 M 意味着分解成 l 比特的分组，假设 M[i]表示第 i 个这样的分组，对 C 是类似的。注意存在加密算法不做任何的随机选择，这并不意味这不把这部分称为随机算法。这只是简

单的一个随机算法碰巧的选择不是随机的。

CBC 模式是最普遍使用的模式

机制 6.4、假设 E：$\{0,1\}^K\times\{0,1\}^l\to\{0,1\}^l$ 是一个分组函数，使用初始向量运算 CBC 模式，产生不稳定的对称密码体制，SE=（K，E，D）。密钥发生器仅仅返回分组密码的随机密钥并返回。这样密钥空间为$\{0，1\}^k$。加密解密的算法如下：

Altorithm $E_K(M)$
If|M|<l 返回⊥
If|M| mod l≠0 返回⊥
把 M 分解为 M[1]…M[n]
C[0]←$\{0，1\}^l$
For i=1,…,n do
C[i]←F_k(C[i-1]⊕M[i])
EndFor
C←C[1]…C[n]
返回 C

Algorithm $D_{k©}(C)$
If|C|<2l 返回⊥
If|C| mod l≠0 返回⊥
把 C 分解为 C[0]C[1]…C[n]

For i=1,…,n do
M[i]←E_k^{-1}(C[i] ⊕C[i-1])
EndFor
M←M[1]…M[n]
返回 M

分解 M 意味着分解成 l 比特的分组，假设 M[i]表示第 i 个这样的分组，分解 C 意味着分解成 l 比特，但是分组计数从 0 开始。初始向量是 C[0]，用加密算法随机选择。算法调用是每次独立的选择。

以下的介绍对于随后的概念有一些作用，若果 l≥1 并且 i 是整数并且满足 $0\le i\le 2^l-1$ 然后假设 $NtS_l(i)$（读做数转化为序列）表示 l 比特串是 *i* 的二进制表示。如果 S 是一个串 StN(s)（读做序列转化为数）。定义二进制表示为 S 的非负整数。

CTR（计数模式）可以进行如下的使用，对于知识的最后的认识，也许是更大的错误。随后可以看到有很好的密码学性质。与 CBC 相反，加密和解密过程是平行处理的。在目前的装置处理水平上可以拓展处理速度。有两个模式的转化，一个随机，另一个指定，随后可以看到，的密码学性质不同。

设计 6.5、假设 F：$\{0,1\}^K\times\{0,1\}^l\to\{0,1\}^l$ 是一个函数簇，（不必是一个置换簇）使用 CTR 模式随机更新在一个不稳定的对称密码体制条件下的密钥产生。SE=（K，E，D），称为 R-CTR 模式或 R-CTR 对称密码体制。密钥产生算法仅仅对 F 返回一个随机密钥，也就是说发现一个随机 k 比特串并且返回。这样密钥空间是$\{0,1\}^K$，加密和解密算法如下：

Algorithm $E_k(M)$
If |M|<L 返回 ⊥
If |M| mod L≠0 返回 ⊥
分解 M 为 M[1]…M[n]
R←$\{0，1，…2^l-1\}$
For i=1,…,n do
C[i]←$F_k(NtS_l(R+i))$ ⊕M[i]
EndFor
C[0] ← $NtS_l(R)$
C ←C[0]C[1]…C[n]

Algorithm $D_k(C)$
If|C|<l+L 返回 ⊥
If(|C|-l) mod L≠0 返回 ⊥
假设 C[0]是 C 的前 l 比特
分解剩余的 C 为 C[1]…C[n]
R←StN(C[0])
For i=1,…,n do
M[i]←$F_k(NtS_l(R+i))$ ⊕C[i]
EndFor
M←M[1]…M[n]

Return C　　Return M

分解 M 意味着把 M 分解成 L 比特的分组，假设 M[i]定义为第 i 分组，对于 C 的解密算法最前端的 l 比特，然后把剩余分解为 L 比特分组。使用加密算法随机选择的随机值是一个 0 到 2^L-1。定义一个序列的值，F_k 应用于随机一次一密，使用 XOR 的数据，随机值包括密文为了能够解密。

设计 6.6、假设 F：$\{0,1\}^K \times \{0,1\}^l \rightarrow \{0,1\}^l$ 是一个函数簇，（不必是一个置换簇）使用 CTR 模式使用随机更新在一个不稳定的对称密码体制的进行密钥的产生。SE=（K，E，D），称为 R-CTR 模式或 R-CTR 对称密码体制。密钥产生算法仅仅对 F 返回一个随机密钥，也就是说发现一个随机 k 比特串并且返回。这样密钥空间是$\{0,1\}^K$，加密使用一个初始为 0 的计数 CTR，加密和解密算法如下：

Algorithm $E_k(M)$
　If|M|<L 返回 ⊥
　If |M| mod L≠0 返回 ⊥
　分解 M 为 M[1]…M[n]
　If ctr+n≥2^l 返回 ⊥
　For i=1,…,n do
　　C[i]←$F_k(NtS_l(R+i))$ ⊕M[i]
　EndFor
　C[0] ← $NtS_l(ctr)$
　C ←C[0]C[1]…C[n]
　ctr ← ctr+n
　Return C

Algorithm $D_k(C)$
　If|C|<l+L 返回 ⊥
If(|C|-l) mod L≠0 返回 ⊥
　　假设 C[0]是 C 的前 l 比特
分解剩余的 C 为 C[1]…C[n]
　ctr←StN(C[0])
For i=1,…,n do
　　M[i]←$F_k(NtS_l(ctr+i))$ ⊕C[i]
EndFor
　M←M[1]…M[n]
　Return M

分解 M 意味着把 M 分解成 L 比特的分组，假设 M[i]定义为第 i 分组，对于 C 的解密算法的最前端的 l 比特，然后把剩余分解为 L 比特分组。计数器不允许隐藏圈数：如果这种情况发生返回⊥，计数器包括在密文中为了能够解密，每次运算加密算法使用计数器累加，在下一次算法的使用中做为初始值。

在适当的拓展一些概念后，对上述体制进行安全性分析。

6.3 私密性分析

6.3.1 安全性分析

固定一个特别的对称密码体制 SE=（K，E，D），两方共享一个密钥 K，这个密钥产生为：K←k，攻击者没有先验知识，现在希望探究这个机制的安全程度。

攻击者假设可以从两方之间的信道俘获一些密文，这样可以搜集密文并且从中获得一些有用的信息。问题是：这些有用信息的确切定义是什么？主要是攻击者完成怎样确定体制的不安全性。因此，相对的，攻击者不能完成这些，是否可以认为这个体制是安全的。

认定不安全比安全更容易一些，认为鉴定攻击者的行为确实暗示体制是不安全的。例如，攻击者可以从一些密文里获取优先的密钥 K，可以随后从不安全的体制中解密出一些容易恢复的密钥，当然，一个不安全的密钥恢复机制是不够证明安全的，也许攻击者可以用其他的

方法进行恢复。

一个人也许可以讲一些诸如此类的事情：给定 C，攻击者没有对 M 的知识，即使这些不是真的，也可以称为先验信息。通常，一些关于信息的知识是已知的。例如，也许是一个没有帧头的数据包，也许是一个英文单词。这样攻击者或其他人在解密之前有一些先验知识。

一个人也许会说假设是这样的：给定密文 C，攻击者不能轻易的恢复出明文，但实际上，这也不足够好。原因是攻击者也许可以计算出部分 M 的信息。例如，即使不能恢复出 M，但是攻击者也许可以通过给定的 C 还原出 M 前半部分的明文，或者 M 的总的比特数。不够好的意思是承载一些有用的信息。

作为具体的例子，从前向的存储信息获得和经纪人之间沟通是否买卖，也就是说，有确切 1 到 l 的存储，如果第 i 比特信息为 1，否则为 0。消息通过加密发送，但是前几比特缺损，攻击者知道是否想买或卖为 1，也许并没有可以公开的信息。如果总的比特数不完善，攻击者知道买了多少东西存储。

也就是说，如果数据处于不同的格式下，这也许不是一个困难。然而，假设或要求多少用户进行格式化，或怎样对于安全协议设计进行恶劣和逼近攻击。对于无论何种格式的数据这个进展是一个重要的原则。用户不必担心数据格式化的问题：这与安全无关。

换言之，作为安全协议的设计者，不用对数据内容和格式进行假设。协议必须保护无论任何格式的数据，把这个工作当成协议设计者安全部分，并且希望设计在自然假设的概率条件下是安全的。

这样最好的希望是什么？对于思考虑的经验值是有用的，怎么是完美理想加密？也就是说，从另外一个角度去处理从发送方到接受方的信息 M，从一些特殊的方法。攻击者根本什么都看不到，直观的，目标是接近最好的概率。也可以用完善加密体制去加密，特别的，没有部分信息泄漏。

作为一个例子，考虑例 6.3 的 ECB 加密机制。给定密文，攻击者是否可以窃听消息。如果 F 是一个好的分组密码，在未知密钥时，求 E_k 的逆是时间难度的。但是，这不是一个好的机制。仅仅考虑一个单独的分组消息表示为 n=1，假设仅有两个消息 0^l 表示买，1^l 表示卖。保持发送数据，但总是两个中的一个。发生了什么？攻击者是相同的，也就是说，前两个条件等同与第三个条件。

在一个安全加密体制中，不应该与密文相关的信息通过这个渠道进行泄漏。

也就是说在某种程度上是一种相当的暗示，如果不是，是否同样的信息必须发送两次。每次加密必须使用新的随机过程，或者说一个计数器，不过一个特别信息的加密算法每次是不同的。某种程度上意味着 E 是一个概率或确定的算法，为什么定义上述的随机加密算法，允许这种加密的类型。

这种结论的理由是使用许多方法对加密的历史和流行的定义而产生的。加密是作为明文到密文的编码和固定映射，这不再是真的了。一个单独的明文可以有许多可能的密文，（依赖于加密的随机选择的算法）。也就是说应该可能会被解密，怎样实现呢？上述已有了几个例子。

现在开始用更正规的方法观察私密性。将使用香侬介绍的信息论中的完善私密的定义，分析一次一密的体制，然而，完善安全要求密钥长度与数据加密相同，这是不现实的。来看一个计算安全的概念，安全不仅是限制在计算能力范围内。如果攻击者进行了足够的工作，可以计算出特征不是特别明显的一些明文。这是分析上述提出的概念安全性一个重要手段。

6.3.2 信息论安全

讨论信息论的称为完善保密的安全定义，这样可以获得一次一密完善保密的体制。

固定一个特别的对称加密体制，SE=（K，E，D），这个体制两方共享一个密钥，攻击者假设获得两方之间通信的任何密文。获得密文之后，试图搜集关于明文的信息。

以一次一密体制并且假设一个单独的k比特消息是加密和传送的，这里k是密钥的长度。根据一个密钥的随机选择，看起来确实非常安全。应该承认攻击者是在已知密文的基础无法还原消息，但是这个判断无法确定真伪。攻击者应该总是可以猜测部分消息，或者可以通过一些好的信息还原部分关系。例如，可以估计到对于所包头部分会包括发送者的 IP 地址。

这样不能真正说攻击者不能在给定密文时无法进行攻击。相应的，有安全的比较测量。关心的是对于给定密文攻击者可以知道的消息是已知密文前的情况是不同的。完善保密体制是基于如果“攻击者”的对信息的最好猜测在已知密文和未知密文的情况下是相同的。换言之，密文对于计算出新的消息是无关紧要的。是这样获得信息的，假设一个单独的消息将被加密，仅关注这个加密的安全机制。有一些消息的明文空间，Plaintext$\subseteq\{0,1\}^*$，加密机将进行加密。（例如，使用一次一密体制，如果密钥长度是 k 比特，那么 Plaintext=$\{0,1\}^k$。注意这对于体制的固定是有效的。

模仿一个先验的信息（攻击者的信息已分布在消息中）。在一个概率消息集合里的消息分布为。正式的，明文的消息认证是一个函数 D：Plaintext→[0,1],例如：

$$\sum_{M\in Pla\,\mathrm{int}\,exts} D(M)=1$$

并且对于 D（M）>0，所有 M∈Plaintext，例如，这里有四个消息的概率，00，01，10，11 分别为：D（00）=1/6，D（01）=1/3，D（10）=1/4，D（11）=1/4 考虑到发送者根据 D 随机选取一个消息，一个特殊的消息是 M∈Plaintext 根据概率 D（M）进行选择，例如，发送者以 1/6 概率进行选择 00。

攻击者已知消息的分布，发送者根据这点进行选择。在密文传送之前，攻击者从 D 中选择发送的固定消息的状态。也就是说，以 1/6 的概率发送 00 是在 D 中规定了的。

加密机制是完善保密的，如果密文的泄漏不超过攻击者对 D 已知的先验概率。这样建立起了密码体制。当发送者根据 D 传输消息后，密钥 K 随之被选定，根据密钥产生算法，意味着 K←k，消息通过加密得到密文，C←E_k（M），攻击者已知 C。攻击者被询问：当已知密文产生的 C 时，对每个消息的概率值，当选择特殊的密文时的概率？如果攻击者不知道 M 所选择的是 D（M）之外更多的信息，意味着密文的占有率并没有比已知的知识更多。这就是完善保密体制。

更正式的叙述这个关系，首先假设：S=Keys(SE)×Plaintexts×$\{0,1\}^r$表示经验值下的样本空间。这里 r 是加密算法的随机事件数量，（如果加密算法是确定的就是 0，如同一次一密的定义是相同的）。假设介绍如下的随机变量为：

定义（K，M，R）→K　　K：S→Keys(SE)

定义（K，M，R）→M　　M：S→Plaintexts

定义（K，M，R）→E_k（M；R）　　C：S→{0，1}*

因此当 M 返回消息值时，K 简单返回选择密钥的值。最后的随机变量返回用 K 加密的消息加密和随机值 R。在样本空间的概率分布定义为 $P_{D,\,SE}[\cdot]$并且对于每个 k 的密钥 K 的选择，每个 D 中 M 的选择，R 的随机选择，都是独立分布。

定义 6.7、假设 SE=（K，E，D）使用消息空间的明文是对称密钥体制，假设 D，明文 D→[0，1]是一个明文上的消息分布。也就是说 SE 对应于 D 的一个完善保密体制，对于每个 M∈Plaintexts，并且对于每个概率的密文是：

$$P_{D,\,SE}[M=m|C=c]=D（m）\qquad (6.1)$$

SE=（K，E，D）是完善保密的如果对于每个明文上的分布的消息是完全安全的。

这里“M=m”是这样的事件：M 是发送者的选择，并且“C=c”是发送方计算的，攻击

者接受的是 c，消息是 M，密文是 C 时考虑条件概率的定义，也就是说消息 M 的严格的先验概率定义为 D（M）。

考虑一次一密加密体制，因为仅有单独的消息进行加密，省略密文的部分计数器。因此密文是一个 k 比特串，k 是密钥和消息的长度。注意在这个体制中 r=0，因为加密算法不是随机的。

例 6.8、假设 SE=（K，E，D）是一次一密加密体制，密钥长度（明文长度和密文长度）是 2，消息空间是 Plain={0，1}^{K}，假设 D 是明文 Plaintext 的消息分布概率，定义 D（00）=1/6，D（01）=1/3，D（10）=1/4，D（11）=1/4。对于每个概率加密 D∈{0，1}^{k}，图 6.1 的第一个真值表是 $P_{D,\ SE}$[M=m|C=c]，特殊的密文，使用一次一密体制加密明文。如同真值表所示，概率总是 0.25，为什么？固定 M，所有不同的 k 比特对于密文 M⊕K 是相同的，K 的范围是{0，1}^{k}。这样不用考虑 M 的值，对于密文，所有不同的 k 比特值是相等的。相应的通常的描述在于引理 6.9 叙述了。$P_{D,\ SE}$[M=m|C=c]的第二个真值表，对于攻击者发现明文 C 的消息 M 的概率。注意这总是等于先验概率 D（M）。

如下的引理是获取一次一密的基本的安全特征：无论是什么信息，因为密钥的随机选择，每个概率的 k 比特密文由概率为 2^{-k} 密文。

引理 6.9、假设 k≥1 是一个整数，假设 SE=（K，E，D）是体制 6.2 的一次一密加密体制，密钥长度置为 k 比特，消息空间为 Plain={0，1}^{K}，假设 D 是一个明文的消息分布。那么：$P_{D,\ SE}$[C=Y|M=X]=2^{-k} 对于任意的 X∈Plaintext 并且任意 Y∈{0，1}^{k}。

引理 6.9：如果 X 是固定并已知的，看到 Y 的概率是什么呢？既然对于一次一密体制 Y=K⊕X，如果 K=Y⊕X，K 是特别的串的概率是 2^{-k}，因为 K 是随机选择的比特串。
这点满足一次一密的上述考虑过的完善保密体制。

	C	00	01	10	11
D（M）	M				
1/6	00	0.25	0.25	0.25	0.25
1/3	01	0.25	0.25	0.25	0.25
1/4	10	0.25	0.25	0.25	0.25
1/4	01	0.25	0.25	0.25	0.25

	C	00	01	10	11
D（M）	M				
1/6	00	1/6	1/6	1/6	1/6
1/3	01	1/3	1/3	1/3	1/3
1/4	10	1/4	1/4	1/4	1/4
1/4	01	1/4	1/4	1/4	1/4

图 6.1:对于第一个真值表，相应的通信是 M 排和 C 列表示了值 $P_{D,\ SE}$[C=c|M=m]。对于例 6.8 的一次一密体制。密钥和消息的长度都是 2。对相同的体制第二个真值表相应的通信是 M 排和 C 列 $P_{D,\ SE}$[M=m|C=c]。

定理 6.10、假设 M∈Plaintexts 是一个消息，假设 C∈{0，1}^{k} 是一个概率密文，需要证明下述 6.1 是正确的，这样有：

$P_{D,\ SE}$[M=m|C=c]= $P_{D,\ SE}$[C=c|M=m]·$P_{D,\ SE}$[M=m]/ $P_{D,\ SE}$[C=c]=2^{-k}·$P_{D,\ SE}$[M=m]/ $P_{D,\ SE}$[C=c]

Bayes 规则第一个等式，第二个引理 6.9 的申请是 X=M，Y=C，定义：$P_{D,\ SE}$[M=m]=D（M）是 M 的前向概率。对最后一个期限：

$P_{D,SE}[C=c]=\sum_X P_{D,SE}[M=X]\cdot P_{D,SE}[C=c|M=X]$

$=\sum_X D(X)\cdot 2^{-k}$

$=2^{-k}\cdot\sum_X D(X)$

$=2^{-k}\cdot 1$

对于所有的消息 X∈Plaintexts，并且引理 6.9。堵塞上述所有的漏洞：$P_{D,SE}[M=m|C=c]= 2^{-k}$ D（M）/ 2^{-k} = D（M）是一个期望值。

一次一密体制不仅仅是完善保密体制，也是最简单和最自然的一种。

6.4 选择明文攻击的分辨机

完善保密体制与信息加密体制的长度相同是不现实的（不必证明这点）。希望找到一个安全的概念，与上述的概念不相同，但是与实践符合。概率上的，主要的不同是考虑攻击者的计算能力是有限制的。有密文的信息，如果不能计算密文不能告知任何信息。在某种程度上，这是现代密码学的开始。相对而言，最相关的不是上面的描述，而是如下的研究。

已经讨论了 6.3 的版本，将过滤关于安全的信息。

6.4.1 定义

讨论了上述的例子，发送者传递了两个消息中的一个。这种情况是对于安全的最坏条件，考虑一个攻击者（不持有秘密密钥）已知两个消息有相同的长度（实际上，是可以选择的）。那么，一个是加密并且密文给定攻击者。这个体制是安全的，如果攻击者区分哪个加密机制是有时间难度的。

事实上考虑两种情况的加密，但是的整个序列，这个思想必须扩展。有一些密码对（$M_{1,0}$，$M_{1,1}$），…，（$M_{q,0}$，$M_{q,1}$），这里对于每个密码对的两个消息有相同的长度。序列对攻击者是已知的，目前一个“挑战”的比特值是随机选择的，产生一个密文的序列 C_1，…C_q，这里，C_i←E_k（$M_{i,b}$）。注意在这些加密中，加密算法使用了新的随机事件，必须成功猜测 b 比特。换言之，攻击者试图确定发送者是否发送了消息 $M_{1,0}$，…，$M_{q,0}$，或…，$M_{1,1}$，$M_{q,1}$ 。对攻击者此外的授权是通过选择明文攻击允许选择消息序列，也就是说选择第一个对，然后接受 C_1，然后选择第二个对，然后继续。现在对上述过程正规化，固定一个特别的加密体制 SE=（K，E，D），（可以是无状态的，也可以是固定的）。考虑一个攻击者 A，是一个体制接近一个随机判断机，可以用相同长度的两个输入（M_0，M_1）。随机判断机返回密文，将考虑用随机判断机制计算的可能方式密文，对相应的两个攻击者存在的条件。为了做这些，最先计算左或右的加密判断机 E_k(LR(.,.b))，如下：

Oracle E_k(LR(M_0,M_1,b)) //b∈{0,1}, M_0，M_1 ∈ $\{0,1\}^*$

C←E_k(M_b)

Return C

随机加密一个消息，根据比特 b 的选择，现在两个条件如下：

World 0: 随机判断机制提供给攻击者 E_k(LR(.,.0))。这样，无论攻击者使用一个序列（M_0，M_1）作为随机判断机制，最后计算 C←E_k(M_0)，并且返回 C 作为回答。

World 1: 随机判断机制提供给攻击者 E_k(LR(.,.1))。这样，无论攻击者使用一个序列（M_0，

M_1）作为随机判断机制，最后计算 $C \leftarrow E_k(M_1)$，并且返回 C 作为回答。

使用第一个条件，或判断机制，“左”条件或判断机，第二个“右”的条件或判断机。对攻击者的问题在于，有时对随机判断机的谈论，告诉哪两个随机判断机。当 PIN 码下载，怎样使用随机判断，假设进一步的确切的阐述。

作为 A 接入的一个子程序，可以做一判断机（M_0，M_1）在特殊状态写为（M_0，M_1），特别固定在内存中，并且一步，返回的是回答。没有控制在怎样计算的问题上，或者 A 可以看到随机判断机的工作。将依赖 A 所给定的秘密信息，A 仅仅是子程序的中间界面。对返回回答的机制是一个黑盒子。

首先的假设是给定对称加密体制 SE 是固定的。随机判断机制，在另一条件，是概率的，因此叫做加密算法。回忆算法是概率的，如上所述，当讲到 $C \leftarrow E_k$（M_b），按时 E 选取自己任意的随机算法，用来计算 C。

加密算法的随机选择是某种程度上的阀下信道，应该被忘记。这对于深远的概念是重要的，并且机制是安全的。

如果对于固定状态的判断机制，给定对称加密体制 SE，换言之，也变得稳定。（考虑一个子程序，保持全局变量在各个子程序可以使用），在条件 b，根据说明的机制，随机判断机使用一个初始状态值，例如，对于计数器模式 CTR，一个计数器 ctr 置为 0。目前每次调用判断机，计算 E_k（M_b），根据算法 E 的说明，将使用新的计数器值。

阐明条件的选择，只做一次选择，一个预先机制，然后攻击者执行。在条件 0，所有发送给判断机制的对使用随机加密消息对的右边，在条件 1，对右边消息使用随机加密机制的所有消息对。从判断机到判断机不进行跳、停的选择，一旦作出选择，对所有消息保持一致。如果一个理智的攻击者不能获得明显的优势，考虑一个加密机“抗选择明文攻击安全”的一个加密体制，给定判断机，对于 b=0,b=1 的区辩机接入，可以理智的反应资源的使用。技术概念在选择明文攻击，定义为 IND-CPA，叫做不可分辨机。

在展示之前，需要讨论一个细节。攻击者可以制定 Ir-加密随机机制，通过询问确定隐藏的比特 b，（决定在哪个条件）设计者认为是不合理的。一方询问消息的随机判断机不同长度的（M_0，M_1）对。不必要求加密隐藏明文的长度，确实是有共同的机制反应这点因为密文的长度依靠于明文的长度。一个攻击者可以容易成功的实用这样的序列，另外，攻击者少数明显的攻击得到一个长度相等的 M_0，M_1 对。这样 E_k（M_0）$\neq \perp$，并且 E_k（M_1）$= \perp$，（如果体制是不固定的，意味着 M_0 是明文空间，而 M_1 不是）。对于一些体制，对攻击者容易发现这些问题。然而，Ir-加密判断机制的反应是放弃比特 b，攻击者从获得的序列中简单选择处理这些问题。也就是说，攻击者是不合法的，如果获得两个不同长度的 Ir-加密判断机制，或者安排 Ir-加密 M_0，M_1 对，对一些值 C，E_k（M_C）$= \perp$ 有一个正的概率值。如果攻击者不是不合法的，那么就是合法的。

这个合法的结果可以讨论，对于忘记的，既然在归约中，将仅有一个合法攻击者，确实在定义中是会处理的。

定义 6.11、假设 SE=（K，E，D）是一个对称密码体制，假设 $b \in \{0, 1\}$，假设 A 是一个算法可以接近判断机制，输入一个串对然后返回一个串。考虑如下的实验：

Experiment $Exp_{SE,A}^{inf\text{-}cpa\text{-}b}$

$K \leftarrow k$

$D \leftarrow A^{Ek(LR(.,.,b))}$

Return d

如果 A 是合法的，A 的 ind-cpa-优势定义如下：$Adv_{SE,A}^{inf\text{-}cpa} = P[Exp_{SE,A}^{ind\text{-}cpa\text{-}1} =1] - P[Exp_{SE,A}^{ind\text{-}cpa\text{-}0} =1]$。否则，为 0。对任意 t,q,μ 定义 SE 的 ind-cpa-优势为：$Adv_{SE}^{ind\text{-}cpa}(t,q,\mu) = \max_{A} \{ Adv_{SE,A}^{inf\text{-}cpa} \}$

这里最大值是具有时间复杂度的所有合法 A 上的，判断机制最多是 q 序列，这些序列的总长度是 μ。

讨论一些重要的协议，关于上述的时间复杂度是最坏情况，经验值的所有执行时间，加上攻击者的编码尺寸，固定在一些 RAM 计算模型中。强调经验值总的执行时间是比攻击者运行的时间多，在经验值中包括所有的执行时间，密钥产生时间和随机序列的计算复杂度。测量时间复杂度的协议与这个注记的其部分是相同的。

另外的协议是，串 M_0，M_1 到左右随机判断机的序列长度定义为$|M_0|$（假设$|M_1|$是相等的，因此攻击者有合法的时间）。这个协议用来测量参数 μ。

如果 $Adv_{SE,A}^{inf-cpa}$ 足够小，意味着 A 输出 1，通常在条件 0，条件 1，没有做足够好的工作区分在哪个条件。如果数量比较大，（意味着接近 1），那么攻击者 A 做得比较好。意味着体制 SE 是不安全的。

对于对称加密体制 SE 是抗选择明文攻击是安全的，无论使用什么策略，攻击者的 ind-cpa-优势必须足够小。然而，希望攻击者对处理能力进行更多的投入获得更好的优势。定义一个上述优势函数进行数据获取：$Adv_{SE}^{ind-cpa}$，就是说对对称加密体制 SE 的函数有三个：首先，时间复杂度，根据上述定义所做的测量；第二，随机判断序列的数字，或消息对的数字，攻击者要求使用这个随机判断机制。这些消息有不同的长度，第三个参数是所有这些长度的总数，表示 μ，根据上述协议测量。ind-cpa-优势函数体制的测量最大概率，体制 SE 的安全性，可以用安全体制 SE 最大概率是用攻击者指定的协议。

6.4.2 优势解释的选择

为什么 $Adv_{SE,A}^{inf-cpa}$ 称为攻击者的“优势”，可以观察攻击者的任务的观点，尝试猜测在哪个条件。对攻击者一个琐碎的猜测返回一个随机比特。在那种情况下，有 1/2 的概率是在右边。明显的，在这种条件下没有任何的损失。攻击者的优势测量是怎样优于处于某个条件的概率。也就是说超过 1/2 的攻击者猜测正确的概率。在这一节，将看到上述的定义怎样选择上述的条件。对定义一些另外的直观判断在以后的定义使用中仍然有用。

通常，固定一个对称加密体制：SE=（K，E，D），考虑如下的博弈或经验值。

Experiment $Adv_{SE,A}^{ind-cpa-cg}$

随机选择一比特 b

假设 K←k

$G \leftarrow A^{Ek(LR(.,.,b))}$

如果 b=g 返回 1　否则返回 0

这里，A 选择一个 b 条件运行一个随机判断机制。最后输出一个比特 g，猜测 b 的一个值。如果 A 的猜测是准确的，返回 1。因此，A 对条件猜测正确的概率是：$P[Exp_{SE,A}^{ind-cpa-cg}=1]$（是在给定比特 b，密钥 K，任意随机选择 $E_{,}(\cdot)$的的初始化选择条件的概率是，A 的随机事件）。如下的命题中描述到，A 正确猜测所处的条件的优势超过随机概率。

命题 6.12、假设 P[·]是概率事件，“·”的经验值是 $Exp_{SE,A}^{ind-cpa-cg}$，在这个经验值数量下讨论。对如下命题进行直接计算：

$P[Exp_{SE,A}^{ind-cpa-cg}=1]$

$=P[b=g]$

$=P[b=g|b=1]\cdot P[b=1]+P[b=g|b=0]\cdot P[b=0]$

$=P[b=g|b=1]\cdot 1/2+P[b=g|b=0]\cdot 1/2$

$=P[g=1|b=1]\cdot 1/2+P[g=0|b=0]\cdot 1/2$

$=P[g=1|b=1]\cdot 1/2+(1-P[g=0|b=0])\cdot 1/2$

$=1/2+1/2(P[g=1|b=1] - P[g=0|b=0])$

$=1/2+1/2([Exp_{SE,A}^{ind\text{-}cpa\text{-}1}=1]- [Exp_{SE,A}^{ind\text{-}cpa\text{-}0}=1]$

$=1/2+1/2Adv_{SE,A}^{inf\text{-}cpa}$

通过标准的训练，开始扩展利益的数量。因为 b 的选择，形成第三行的 1/2 的条件。在第四行，给出已知 b=1 时，要求是否 b=g，同样的给定 b=1 询问是否 g=1，分析 b=0。在第五行和第六行仅仅使用概率和放大进行操作。第二行非常重要，分别在真实和随机的博弈条件在这个问题中观察概率条件确实是后续概率。意味这如期望恢复了优势。

6.5 选择明文攻击的例子

举例说明对 ECB 提供发祥攻击的模型，对通常的确定和不固定体制的攻击。

6.5.1 ECB 攻击

假设固定分组密码 E：$\{0,1\}^K\times\{0,1\}^l\rightarrow\{0,1\}^l$。ECB 对称加密算法体制 SE=（K，E，D）是 6.3 设计，假设攻击者一个密文 $C=E_k$（M）相应对于一些未知的明文 M，在未知攻击者的密钥 K 条件下进行。攻击者是否可以恢复明文 M，不容易，如果 E 是好的分组密码？例如如果 E 是 AES，看来相当的不切实际。然而，已经讨论了怎样从密文恢复明文是不可行的，不是一个安全的指示。ECB 有一些弱点，注意如果两个明文 M 和 M'在第一个分组是相同的，这样进行相应的密文。因此攻击者，给定密文，通知是否与相应的明文一致。这样丢失了明文的部分信息，对安全加密体制是不允许的。

这是定义的一个测试看见定义获得弱点并且发现体制是不安全的。确实，使用一个小的资源通过一个高的 ind-cpa-优势，攻击者可以被发现。如同下面的推论所述：

命题 6.13、假设 E：$\{0,1\}^K\times\{0,1\}^l\rightarrow\{0,1\}^l$ 是一个分组密码，并且 SE=（K，E，D），相应的 ECB 对称加密体制在设计 6.3 中描述，那么：$Adv_{SE,A}^{ind\text{-}cpa}(t,2,2l)=1$
对于 t=O（l）加上分组密码的两个运算时间。

尽管难于应用任何资源，攻击者的优势为 1，仅仅是一个询问，并且是不复杂的询问，明显的，一个指示这个体制是不安全的。

命题 6.13 证明：描述一个攻击者 A，具有时间复杂度 t，对随机判断询问回复 1，长度 2l 的询问为：$Adv_{SE,A}^{ind\text{-}cpa}=1$。命题如下。

记住攻击者 A 给定 Ir 加密判断机制为 E_k（LR（.,.,b））输入一对消息，返回一对加密中的左边或右边的加密，依赖于 b 的值，A 的目标决定 b 的值。攻击者的工作如下：

Adversary $A^{Ek(LR(.,.,b))}$

$M_1 \leftarrow 0^{2l}$; $M_0 \leftarrow 0^l \| 1^l$

$C[1]C[2] \leftarrow E_k(LR(M_0,M_1,b))$

如果 C[1]=C[2] 那么返回 1 否则返回 0

攻击者单独的随机询问是消息对 M_0，M_1 。既然每个部分是两个分组的长度，因此根据 ECB 计算密文。现在可以认为：

$$P[Exp_{SE,A}^{ind\text{-}cpa\text{-}1}=1]=1$$

$$P[Exp_{SE,A}^{ind\text{-}cpa\text{-}0}=1]=0$$

因此，$Adv_{SE,A}^{ind\text{-}cpa}=1=0=1$

A 仅仅使用随机判断机制产生达到这个优势，每个分组的长度是 2l 比特，这样

$$Adv_{SE,A}^{ind\text{-}cpa}(t,2,2l)=1$$

为什么上述两个问题为真？必须返回问题的定义数量，通过这里定义的经验值的踪迹。在假设条件 1，当 b=1 时，随机判断机制返回 C[1]C[2]=E_k（0^l）‖ E_k（0^l）,这样，C[1]=C[2]并且 A 返回 1。在假设条件 0，意味着 b=0，随机判断机返回 C[1]C[2]=E_k（0^l）E_k（1^l）。既然 E_k 是一个置换，C[1]≠C[2]。在这种条件下，A 返回 0。

作为练习，作为攻击者试这分析同样的攻击对 CBC 或 CTR 模式，不用高的优势条件确信自己。攻击必须确定的一个重要的条件需要得到重视。也就是说，ECB 不是一个安全的加密体制，分组密码算法 E 是安全的。弱点不是工具的使用，但是在习惯上使用。ECB 机制是错误的，一个好的工具也需要正确的使用。

这是一种设计的缺陷，应该能够确认和消除。目标是确定对称加密体制是安全的，当分组密码是安全的前提下。换言之，体制没有内在的缺点。只要使用正确的成分，配方产生好的膳食。如果不能使用好的成分，那是的问题了。

6.5.2 稳定、固定的机制是不安全的

ECB 模式是确定的和没有状态的，因此同样的消息加密两次，返回相同的密文。这个性质是改变，通常的，一个安全体制的结果是，提供一个更为什么 ECB 模式失败的理解。更细致的描述更一般的事实。

命题 6.14、假设 SE=（K，E，D）是一个确定的，不固定的加密体制。假设有一个整数 m，因此体制的明文空间包括两个长度为 m 的不同的串。那么：$Adv_{SE,A}^{ind\text{-}cpa}(t,2,2m)=1$
是时间 t=O(l)加上两个加密时间。

必要的条件是构造的消息空间是最小限度的，典型的机制是包括从小到大的不同长度的串。注意，命题应用 E C B 模式，并且指出后者是不安全的。但是命题 6.13 只有一个询问比两个询问的机制稍微强一些。

命题 6 .14 的证明、将介绍一个攻击算法 A，具有时间复杂度 t，制定两个随机判断机的询问，每个询问的长度是 m，并且有：$Adv_{SE,A}^{ind\text{-}cpa}=1$
命题如下。

记住攻击者 A 是给定的一个 Ir 加密随机判断机制，E_k（LR（.,.,b)），获取消息输入对。返回左或右的消息的加密对，依赖 b 的值。A 的目标是确定 b 的值。攻击者的工作如下：
Adversary $A^{Ek(LR(.,.,b))}$

假设 X，Y 是明文空间的一个独特的 m 比特串，

C_1←E_k（LR（X，Y，b））

C_2←E_k（LR（X，Y，b））

如果 $C_1=C_2$ 那么返回 1 否则返回 0

现在，声称：

$$P[Exp_{SE,A}^{ind\text{-}cpa\text{-}1}=1]=1$$
$$P[Exp_{SE,A}^{ind\text{-}cpa\text{-}0}=1]=0$$

因此，$Adv_{SE,A}^{ind\text{-}cpa}=1\text{-}0=1$。并且 A 达到两个随机判断机制的优势，每个长度是每个协议仅仅是随机判断的优势，是 m 比特。因此，$Adv_{SE}^{ind\text{-}cpa}(t,2,2m)=1$
为什么上述的两个问题是真的？在条件 1，意味着当 b=1 时，随机判断机制返回 $C_1=E_k(Y)$ 并且 $C_2=E_k$（Y）,既然加密函数是确定并且不固定，$C_1=C_2$，因此，A 返回 1。在条件 0，意味着当 b=0 时，随机判断机制返回 $C_1=E_k(X)$并且 $C_2= E_k(Y)$，因此要求解密可以还原消息，并且 $C_1≠C_2$。这样 A 返回 0。

6.6 抗明文恢复攻击的安全性

在 6.3 节，注意数字的安全特征是必要的安全保证，但不时充分的。例如，对一个攻击者，从一些明密对恢复密文成为明文是计算不可行的，定义的测试暗示这些特点，从某中程度上，在体制上抗明文恢复攻击或密钥恢复攻击是安全的。

在 PRFS 的条件下，情况是类似的。安全 PRF 抗密钥恢复是安全的。为了有一写变化，这次选择一个不同的特征，叫做明文恢复攻击。把这部分形式化，那么有一个攻击者 B 有能力从给定的密文中恢复出明文。这样可以构造一个攻击者 A 在 IND-CPA 条件下，计算出处于两个条件。但是如果体制在 IND-CPA 条件下，后者的攻击不能存在。因此，前者也不存在。

争论阐述的思想，怎样确信上述的定义是好的，获得选择明文攻击的所有特征。得到一些特征认为是应该具备的安全体制：从一些明密对恢复密钥的不可能，预测 XOR 明文的不可能。考虑有一个攻击者 B 可以完成这种攻击。声称可以构造一个攻击模型 A 分析左右的形式。更详细的，可以使用体制不安全函数限制攻击者 B 成功的概率。假设不安全函数在特定参数值的条件下是小的这就是攻击者成功的机会。

现在通过细节上明文恢复的例子，面临攻击者的任务是解密一个密文，使用随机加密成形选择一些长度 m 的挑战消息。在处理过程中，赋予攻击者发现明密对的能力，并且逼近一个加密机制。上述加密机制不是 Ir 加密判断的，并且通过接近一个加密机制：相应的，简单的输入一个单独的消息 M 并且返回一个密文 C←E_k（M）用加密 M 进行计算。为了得到一个攻击者对密文挑战的前提条件，介绍另一种没有输入的随机判断机制，调用一个随机 M 比特明文串，计算 C←E_k（M），返回 C。攻击者仅仅允许一个随机判断机制的简单的询问，如果可以挑战随机判断机制的密文 C 还原出相应的明文成功，定义挑战判断机制为 E_k（$\$^m$）。这里，$\$^m$ 意味对 E_k 输入的随机 m 比特串的选择。

更简单的，假设加密体制是不固定的，因此，{0，1}^{m} 是一个与明文空间相关联的子集。通常条件下，调用加密体制进行加密或挑战随机判断机制，简言之，不妨碍加密函数的随机性，意味着随机事件对判断机制每个挑战。

定义 6.15、假设 SE=（K，E，D）是一个不固定的对称加密体制，明文空间在{0，1}'，假设 B 是一个可以接近两个随机判断机的算法。考虑如下的经验值：

Experiment $Exp_{SE,B}^{pr}$

K←k

M←$B^{Ek(.)E,(\$m)}$

If D_k(C)=M，这里 C 是 B 对 E_k($\$^m$)询问应答

返回 1

否则返回 0

B 的先验优势定义为：$Adv_{SE,B}^{pr}$=P[$Exp_{SE,A}^{pr}$=1]=1

对于任意的 t,q,μ,定义 SE 的先验优势：$Adv_{SE,B}^{pr}$（t,q,μ）= $\max_{B}$ {$Adv_{SE,B}^{pr}$}

在上述的经验值，B 使用两个判断机制进行操作。回忆允许严谨的询问挑战随机判断机制，定义密文返回 C 询问的应答。（返回挑战随机应答没有输入）。如果可以对 C 进行正确的解密，攻击者胜利，在这种情况下，经验值返回 1。在处理过程中，攻击者可以按照自己的思想进行询问。

如下的建议是攻击者成功的从密文的挑战还原明文不能超过体制的 ind-cpa 优势。（使用资源的参数的是攻击者）加上简单猜测的明文的机会。换言之，在 IND-CPA 的安全体制下

抗明文恢复攻击是安全的。

命题 6.16：假设 SE=（K，E，D）是一个不固定的对称加密体制，明文空间在{0，1}*，对任意的 t,q,μ,有：Adv_{SE}^{pr}（t,q,μ）≤$Adv_{SE}^{ind-cpa}$(t,q+1, μ+m)+$1/2^m$

原因是真的并且简单的。如果攻击者 B 是描述密文攻击的能力，可以容易的构造 A_B，把 B 作为一个子集合，可以判断出条件 0 或条件 1，换言之，是一个归约。命题 6.16 的证明：给定一个攻击者 B，对 t,q,μ 进行限制，构造一个攻击者 A_B，使用资源 t,q+1, μ+m，因此：

$$Adv_{SE,B}^{pr} \le Adv_{SEAB}^{ind-cpa} + 1/2^m \qquad (6.2)$$

根据通常的最大化处理建议。

对于定义 6.1，攻击者 A_B 将提供一个 Ir 加密判断机制，尝试确定是哪个假设条件。为了进行这个工作，将运行攻击者 B 作为子集。提供一个如下解释和分析的描述。

Adversary $A_B^{Ek(LR(.,.,b))}$

运行攻击 B，回答随机判断询问

当 B 确定一个加密随机判断询问 X，运算

Y←E_k(LR(X,X,b))

返回 Y 到 B 作为回答

当 B 进行随机判断询问时，做

M_0←{0，1}^{m}；M_1←{0，1}^{m}

C←E_k(LR(M_0，M_1，b))

返回 C 到 B 作为回答

直到 B 停止，输出一个明文 M

如果 M=M_1 那么返回 1，否则返回 0

合理 A_B 运算 B 并且提供 B'对判断询问的的回答。当 B 提供一个加密判断询问 X，攻击者 A_B 需要返回 E_k（X），调 Ir 加密判断机制，使用密钥对到 X 的消息。因此，不考虑比特 b 的值，密文返回一个 B 所希望的 X 的加密值。当 B 做一个询问挑战判断机，A_B 选择两个随机消息，长度都为 m，并且调用 Ir 加密判断机并返回一个密文 C。现在 B 返回消息 M 假设是对 C 的解密值。攻击者 A_B 测试是否 M=M_1，如果这样，猜测是否是条件 1，另外，测试是否是 0。现在，可以证明：

$$P[Exp_{SE,AB,\ 1}^{ind-cpa-1}=1] \ge Adv_{SEAB}^{pr}$$

$$P[Exp_{SE,AB,\ 1}^{ind-cpa-0}=1] \le 2^{-m}$$

对这个结论进行简单的证明，但是假设使用结论。对定义 6.1 所减法，

$$Adv_{SE,AB}^{ind-cpa} = P[Exp_{SE,A}^{ind-cpa-1}=1] - P[Exp_{SE,A}^{ind-cpa-0}=1] \ge Adv_{SE,B}^{pr} - 2^{-m}$$

给定等式 6.2 重新安排条件，证明等式 6.3 和 6.3。

攻击者 B 将使用最小概率 $Adv_{SE,B}^{pr}$ 返回 M=D_k(C)，在条件 1，密文 C 是 M_1 的一个加密，这样意味着以最小概率 $Adv_{SE,B}^{pr}$ 得到 M=M_1，因此等式（6.3）是真的，现在假设 A_B 在假设条件 0，在那种条件下，A_B 将返回 1 当且仅当 B 返回 M=M_1，但是 B 返回关于 M_1 的没有信息，既然 C 是 M_0 的一个加密并且是与 M_0 没有关系的 M_1 是随机加密。对 B 的简单可能是以大于 2^{-m} 的输出 M，因此等式 6.3 是真的。

类似的争论可以确定一个从定义出发的对称加密体制的其安全的特征设计。例如，攻击者 B 是可能的，给定一些明密对并且一个密文 C 的挑战，可以计算 XOR 的 M=D_k(C)比特？或者这些比特的总数？或者至少 M 个比特？做这些的概率不超过 1/2 的概率，因为是这样，设计一个攻击者 A 使用资源 B 的比较可以获得左或右的猜测的胜利。作为一个公式表示或其命题延续 6.6 诸如此类的例子。

当然，一个人不能尽力列举所有的期望的安全特征。对左右的定义可以覆盖所有安全的

自然特征在选择明文攻击条件下应该是可信任的。确实，如果任何事情在保守的一边是错误的。在实际中有一些攻击，如果上述的一些式子是不满足的，定义称为的体制是不安全。这完全是正确的，这对一些保守算法没有任何的损害。更重要的是在真实的假设条件下，如果有攻击可以被认为是损耗，这个体制在左右测试时将失败，正式的概念也声称是不安全的。

6.7 抗明文攻击的 CTR 安全

假设 F：$\{0,1\}^K\times\{0,1\}^l\rightarrow\{0,1\}^L$ 是一个函数族。在两个变量下，CTR 对称加密体制：随机（无状态）体制 6.5 并且基于计数的一个体制 6.6。抗明文攻击是都是安全的，但是，有趣的，计数版本比随机版本更安全。首先关于体制的法则，讨论并证明，对于计数版本有：

定理 6.17、假设 F：$\{0,1\}^K\times\{0,1\}^l\rightarrow\{0,1\}^L$ 是一个函数族。假设 SE=（K，E，D）是相应的 C-CTR 对称加密体制并且描述体制同 6.6，对于任何 t,q,μ,满足 $\mu<L2^l$，有：

$$Adv_{SE}^{ind\text{-}cpa}(t,q,\mu)\leq 2\cdot Adv_F^{prf}(t,q',lq')$$

这里 q’=μ/L。

对于随机版本有：

定理 6.18、假设 F：$\{0,1\}^K\times\{0,1\}^l\rightarrow\{0,1\}^L$ 是一个函数族。假设 SE=（K，E，D）是相应的 R-CTR 对称加密体制并且描述体制 6.5。对于任何 t,q,μ,满足 $\mu<L2^l$，有：$Adv_{SE}^{ind\text{-}cpa}(t,q,\mu)\leq 2\cdot Adv_F^{prf}(t,q',lq')+（\mu（q-1）/L2^l）$。这里，q’=μ/L。

这种结论是上述的改进。也就是，提供高水平的密码结构设计进行可能的安全保障，（前提条件是对称加密体制）。基于假设，一些分组构造是安全的（也就是说针对分组密码算法的 PRF 条件）。这是最早构造的碰撞例子。这样中止，试着明白这些定理并且什么是实现。

如果对一些加密机制想委托数据，需要确认的是加密机制可以保证安全。如果是不安全的设计，也许不是。这种情况发生在 ECB 模式，甚至使用了一个安全的分组密码，ECB 模式使用的缺陷也会使加密体制不安全。

当最先的印象中 CTR 模式没有明显的缺陷，但是也许缺陷是存在的。如果不能对所有可能的攻击方法或搜索空间进行穷尽，问题是很难使别人确定这种缺陷确实存在。在上述定理原则条件下是安全的。声称 CTR 模式没有弱点，也就是说，只要使用一个好的密码算法就可以保证没人能够破解的密码算法。一个人不能要求得更多，如果无法使用好的密码算法，也就不能保证使用的安全。因此可以确信，所有攻击不会成功，即使不能准确的使用这些攻击手段。这就是进展的能力。

现在，一个人可以赏识这种能力，获得工作的有力的支持。也就是说，必须使用这种工作拓展定义：对于正式的安全概念是意味深长的。对于对定义没有预先概念的读者，至少必须已知，这种努力是值得的。只是花一点时间去体会，但是回馈是巨大的：证明确实有能力保证安全性。

然而，特别的，定理是描述这些的吗？上述的讨论是在数量上推动了基本概念，这是结论的重要部分。如果研究一个具体例子会更有帮助。

例子 6.19、假设 F 是 AES，这样密钥长度是 128 比特，分组程度是 l=L=128。假设希望加密 $1=2^{40}$，每个消息有 $128*2^3$ 比特长，因此加密总数为 $\mu=2^{50}$，能够使用 CTR 模式保证安全吗？攻击者有多少的机会计算出数据。适当的，如果攻击者有 $t=2^{60}$ 计算周期，那么使用定义的机会在于 $Adv_{SE}^{ind\text{-}cpa}(t,q,\mu)$，使用定理是无关紧要的，这只是一种定义，这是使用可用的信息资源破解这个密码体制的最大概率。使用选择参数降低了优势值 $Adv_{SE}^{ind\text{-}cpa}(t,q,\mu)$，这个体制是否安全的。之前，没有任何先验经验。但是现在，声称定理 6.17 的优势至多是 $2Adv_F^{prf}(t,q',128q')$，这里 q’是定理中定义的。也就是说，$q'=\mu/L=2^{50}/128=2^{43}$。这样问题是，$Adv_F^{prf}(t,q',128q')$的值中 t’，q’的值。

因此，从加密体制对还原估计的问题是对 AES 的伪随机测试。对每节的 5.6.2，也许可以推测为：$Adv_F^{prf}(t,q',128q')=C_1(t/T_{AES})/2^{128} + (q')^2/2^{128}$

这里 T_{AES} 是对于固定的 RAM 运算模式进行一圈的 AES 计算。现在加上 $t=2^{60}$ 并且 $q'=2^{43}$ 并且认为上述的计算，得到：

$$\begin{aligned} Adv_{SE}^{ind\text{-}cpa}(t,q,\mu) &\le 2Adv_F^{prf}(t,q',128q') \\ &\le 2\ C_1(t/T_{AES})/2^{128} + 2(q')^2/2^{128} \\ &= 2^{61}/2^{128}\cdot C_1/\ T_{AES} + 2^{43*2+1}/2^{128} \\ &= 1/2^{67}\cdot C_1/\ T_{AES} + 1/2^{41} \\ &\le 1/2^{41} \end{aligned}$$

在最后一步，做一个非常有利的假设，t/T_{AES} 最多是 2^{26}。因此，攻击者的机会以 2^{-41} 的概率得到任何关于加密消息的数据，甚至可以攻击者时间为 2^{60}，并且加密 2^{50} 比特数据。这是非常小的机会，能够使用这种方法生存。认为这种机制是安全的。

例 6.20、鼓励使用同样的方式继续工作，不必假设 F 是 AES，但是同样假设比 PRF 更好。同样有 L=l=k=128，但是假设这不是一个置换，因此不存在生日攻击，特别的，假设：

$$Adv_F^{prf}(t,q',128q')= C_1(t/T_{AES})/2^{128} + C_1q'/2^{128}$$

现在，考虑基于计数的 CTR 模式和随机的一种。在法则上，不同的是 μ（q-1）/$L2^l$ 条件，试着看看有什么不同，对于每个机制，在满足最低安全条件下，t,q,μ 的最大值可以取到多少。使用哪种体制，可以得到更好的结果。哪个机制因此是更安全的？

这些例子说明了怎样使用规则计算出从 CTR 的一些安全加密模式应用。

6.7.1 定理 6.17 的证明

使用的范例在许多方面是通用的，并且将再次使用，不仅仅是加密体制，但是对于其他的体制是基于伪随机函数的。

Algorithm $E_g(M)$
- If |M|<L 返回⊥
- 如果|M|modL≠0 返回⊥
- 分解 M 为 M[1]…M[n]
- If ctr+n≥2^l 返回⊥
- For i=1,…,n do
  - C[i] ←g(StN_l(ctr+i))⊕M[i]
- EndFor
- C[0] ←StN_l(ctr)
- C←C[0]C[1]…C[n]
- ctr←ctr+n
- Return C

Algorithm $D_f(C)$
- If|C|<l+L 返回⊥
- 如果(|C|-l) modL≠0 返回⊥
- 假设 C[0]是 C 的最先的 l 比特
- 分解剩余的 C 为 C[1]…C[n]
- ctr←StN(C[0])
- For i=1,…,n do
  - M[i] ←g(StN_l(ctr+i))⊕C[i]
- EndFor
- M←M[0]M[1]…M[n]
- Return M

图 6.2：版本 C-CTR 的体制 SE（G）=（K，E，D）并行于 G 函数簇。

一个重要的观察报告在于 CTR 体制，加密解密体制不需要直接访问密钥 K，但是仅仅接近子程序，或一个随机判断机制，也就是执行函数 F_K。用一些小的函数代替 F_K 时，重要的是会发生什么？为了确认这种替换，再规定这种体制。介绍一个体制作为一个参数给定一个 G 函数簇，定义域为{0，1}^{l}，值域为{0，1}^{L}。随后将要看到 G=F 是价值，并且 G=$Rand^{l,L}$。首先描述这个参数机制，在剩余的证明中，SE（G）=（K，E，D）表示对称加密体制，定义如下：密钥产生算法仅仅返回一个 G 的随机数，意味着随机选择一个函数 g←G，把 g 作

为一个密钥。加密器保持计数 ctr，初始化为零，加密和解密算法如图 6.2。在图中，分解 M 意味着分为 L 分组，（不是 l 分组），假设 M[i]定义 i 个这样的分组。对于 C 这样的解密算法首先选出前 l 比特，然后把剩余的部分做 L 分组。加密算法把计数器每次进行升级，这个值作为下次加密的初始值。作为描述的说明，体制确实是严密的 C-CTR，除了函数 g 代替 F_k，表面上的平滑处理观点是很有用的，正如所见到的。

观察有兴趣的机制，并且根据上述的规则，简单的 SE[F]，这里 F 是每个规则给定的函数簇。现在，现在证明可以分成两个部分。第一步把 F 从图中移开，并且代替研究理想化的版本。也就是说，考虑体制 $SE[Rand^{l\to L}]$。这里，一个 l 比特到 L 比特的随机函数 g 可以用于最初的体制 F_k，这样攻击理想化体制进行逼近。这个概率几乎为零，这是分析中主要的推论。

这一步是彻底的经验值，不是真实的运算可以使用一个随机函数代替 F_k，作为记忆总数过高的函数。这个理想体制的分析是授予 CTR 模式可能的弱点，作为在分组密码算法下对弱点的分析的提高。可以认为这个理想体制是安全的，也就是说这个模型本身是安全的。

现在就看怎样把这个模型用于一个现实事件了，宁愿使用给定的 F 簇进行 SE[F]体制的安全评估。这里开拓一个伪随机函数 F 的破解概率。

推论 6.21、假设 A 是任何的 ind-cpa 攻击者攻击 $SE[Rand^{l\to L}]$。那么：$Adv_{SE[Randl-L].A}^{ind-cpa}=0$ 只要 A 的随机判断机制序列的总长度为 $L2^l$。

推论说明为：绝对的攻击者。假设攻击者有时间复杂性 t，产生 q 序列到 Ir 加密判断机，总数是 μ 比特。推论并不关心 t,q 的值，但是强调 μ 的总数是 $L2^l$，很难强调实际使用里的一个约束。典型的 l 长度是 64 比特，事实上，信息论中 t 指示的结果：不必理会攻击者有多少的计算时间。

当然，推论提到理想化的设计，也就是说，g 函数可以随机的用做加密函数。记住在这个框架下 ECB 是不安全的，（攻击者提供 ECB 模式，当 E 是 $Perm^l$，所有置换簇），这样声明是不能随便满足的，也就是说一些事对 CTR 是意味深长和重要的，对其模式是不重要的。

推迟推论的证明。也就是说对于定理证明这样首先使用，这里的证明是一般和简单的。这个论点是简单和普通的。

G 是一个随机函数时，推论 C-CTR 加密模式是安全的。但是对于 g 是否是一个 F 簇的函数非常有兴趣，因此担心实际的体制是 SE[F]是安全的，理想体制 $SE[Rand^{l\to L}]$是安全的。换言之，担心实际上的体制 SE[F]是不安全的，攻击者攻击 SE[F]时有大的 ind-cpa 优势，甚至已知攻击 $SE[Rand^{l\to L}]$的优势度为 0，但是如果 F 是 PRF 安全的，这种假设是不可能的。直观上的，这样的攻击者存表示 F 不接近 $Rand^{l\to L}$，因此，有一些可洞察的事件，也就是攻击者的后继概率使用确定的实验，当 F 在使用时，以高概率事件发生，当 $Rand^{l\to L}$ 使用时以低概率发生。为了具体使用这种直觉判断，假设 A 是 ind-cpa 的一个攻击。假设 A 是一个分辨机，D_A 对给定的随机机制逼近一个函数 $g: \{0,1\}^l\to\{0,1\}^L$，试着确定所处的假设条件，在假设条件 0，g 是一个 $Rand^{l\to L}$ 的随机机制，在假设条件 1，g 是一个 F 的一个随机实例。对分辨机使用如下的建议，运行 A，使用这种方法回答 A 的询问：A 在 D_A 的假设条件 0 是攻击者 $SE[Rand^{l\to L}]$，A 在 D_A 的假设条件 1 是攻击者 $SE[Rand^{l\to L}]$。可以运算图 6.2 的 $E_g(\cdot)$ 的加密算法执行是 D_A 的攻击概率，这个简单的询问就可以接近函数 g，如果攻击者 A 是胜利者，意味着加密体制 D_A 假设 g 是 F 的一个例子，否则，D_A 假设 g 是 $Rand^{l\to L}$ 的例子。

强调密钥构造证明的工作，C-CTR 体制的加密函数仅仅使用简单的随机判断机进行调用 F_k 函数，如果有密钥 K 的直接的使用，上述的范例不能实施。完整的证明如下：

定理 6.17 的证明、假设 A 是任意的 ind-cpa 攻击者攻击 SE=（K，E，D），假设 A 作为 q 随机判断询问总计为 μ 比特，并且具备时间复杂度 t，设计一个分辨机 D_A，因此，

$$Adv_{SE,A}^{ind-cpa}\le 2\cdot Adv_{F,DA}^{prf} \quad (6.3)$$

更多的，D_A 将假设 μ/L 的随机判断询问并且有时间复杂度 t，目前，定理 6.17 进行如下通常的最大化的使用。因此，在等式 6.3 中提供给分辨机的主要事情是真的，分辨机使用 A 作为一个子程序。

记住 D_A 作为一个随机函数 g：{0，1}^{l}→{0，1}^{L}，这个判断机从 F 或 Rand$^{l→L}$ 中随机选择，D_A 并不知道先验值，为了发现这个值，D_A 将使用 A，记住 A 也获得一个随机判断机制，名义上一个 Ir 加密判断。从 A 的观点，判断机是一个简单的子程序：A 可以写，在一些地点，一对消息，返回一个称为判断机的实体。当 D_A 运算一个 A 作为子程序，D_A 将模拟一个 Ir 加密 A 的判断机制，意味着 D_A 将提供 A 进行的任何判断询问应答。以下是 D_A 的描述：D_A^g 的分辨机

B←{0,1}

运行攻击者 A，应答如下的随机判断询问

当 A 做了一个随机判断询问（M_0 ，M_1）做

C←$E_g(M_b)$

对 A 返回 C 作为回答

知道 A 停止输出比特 d

如果 d=b 返回 1，否则返回 0

这里 E_g（·）定义了产生 C-CTR 模式的加密函数，在 6.2 中进行了定义。关键的事实是，使用 g 的给定的判断机可以进行实现。使用分辨机 D_A 发现比特 b 的挑战随机数对 A 的假设条件的选择，那么看看 A 是否在猜测比特值的后继。如果是，可以假设 g 是 F 的一个代替，否则是 Rand^{l-L} 的代替。为了便于分析，假设：

$$P[Exp_{F,DA}^{prf-1}=1]=1/2+1/2\cdot Adv_{SE[F],A}^{ind-cpa} \quad (6.4)$$

$$P[Exp_{F,DA}^{prf-0}=1]=1/2+1/2\cdot Adv_{SE[Randl-L],A}^{ind-cpa} \quad (6.5)$$

简单的介绍这些命题，首先得到结论。对定义 5.4 做减法，得到：

$$Adv_{F,DA}^{prf} = P[Exp_{F,DA}^{prf-1}=1]-p[Exp_{F,DA}^{prf-0}=1]$$
$$= 1/2\cdot Adv_{SE[F],A}^{ind-cpa}-1/2\cdot Adv_{SE[Randl-L],A}^{ind-cpa}$$
$$= 1/2\cdot Adv_{SE[F],A}^{ind-cpa} \quad (6.6)$$

最后的不等式从引理 6.21 是得到了，告诉大家期限是 $Adv_{SE[Randl-L],A}^{ind-cpa}$ 是简单的 0，重新安排期限给定等式 6.3，现在假设检查资源的使用方法。每个计算 $E_g(M_b)$要求 g 的应用 $|M_b|/L$，因此询问总数是 D_A 到判断机制 g 是 μ/L。D_A 等式的时间复杂度是 A 一旦计算出协议时间复杂度是指全部实验的时间。接下来是证明等式（6.4）还是（6.5）。

b=d 的分辨机 D_A 是返回 1，意味着 ind-cpa 攻击 A 正确的指出 b 所处的假设条件，或者 6.5.2 的语言，进行“正确性验证”。D_A 的假设条件是简单的改变真的加密体制，当 D_A 处于假设条件 1，加密体制，从 A 的角度，是 SE[F]，当 D_A 在假设条件 0，加密体制，从 A 的角度，是 SE[Rand^{l-L}]。因此，使用 6.5.2 节的概念，有：

$$P[Exp_{F,DA}^{prf-1}=1]=P[Exp_{SE[F],A}^{ind-cpa-cg}=1]$$
$$P[Exp_{F,DA}^{prf-0}=1]= P[Exp_{SE[Randl→L],A}^{ind-cpa-cg}=1]$$

为了得到等式 6.4 和 6.5，现在使用命题 6.12。

对没有使用基于上述看似复杂的安全性证明的 PRF，但是下述的思想实际上是非常简单的，可以反复进行研究。一个可以进行 ind-cpa 攻击者对密码体制中 g 的使用的攻击使用，如果 g 是 F 的一个特例攻击者可以获得比 Rand^{l-L} 更好的效果，然后在 F 和 Rand^{l-L} 之间进行测试。现在证明关于理想 C-CTR 体制的安全。

引理 6.21 的证明、直观上是简单的，当 g 是一个随机函数，连续的计数值产生一次一密体制，一个真实的随机数和不可测的比特序列。只要数据比特的加密数不超过 L2^l，仅仅

在所有的加密过程中调用 g。既然数据使用 XOR 与序列结合，攻击者不给定关于序列信息。现在，必须在所有的流程中确信直觉。推论陈述涉及了安全的概念，因此必须使用 6.5.1 节的配置。攻击者 A 产生接近 Ir 加密判断机制。因此，这个体制考虑是 $SE[Rand^{l-L}]$，随机判断机是 E_g（LR（.,.,b））,这里函数由图 6.2 定义了。G 是 $Rand^{l-L}$ 一个任意的值，意味着一个随机函数。

攻击者对判断机制的一些数 q。假设（$M_{i,0}$,$M_{i,1}$）是第 i 个咨询，假设 n_i 是 $M_{i,0}$ 的分组数。（这是与 $M_{i,1}$ 相同的分组数）。假设 $M_{i,c}[j]$是第 j 个 L 比特分组，对于 c∈{0,1}。假设 C_i 是随机判断机制（$M_{i,0}$,$M_{i,1}$）询问。由 n_i+1 个分组，第一个分组是计数 l 比特的二进制表示 $E_g[M_{i,b}]$。这里 b 是 E_g[LR（.,.,b）]的判断机制，其他的分组表示为 $C_i[1]…C_i[n_i]$。如图：

$M_{1,b} = M_{1,b}[0]\ M_{1,b}[1]…M_{1,b}[n_1]$
$C_1 = NtS_l(0)C_1[1]…C_1[n_1]$
$M_{1,b} = M_{2,b}[0]\ M_{2b}[1]…M_{2b}[n_2]$
$C_2 = NtS_l(n_i\ )C_2[1]…C_2[n_2]$
…
$M_{q,b} = M_{q,b}[0]\ M_{qb}[1]…M_{qb}[n_q]$
$C_q = NtS_l(n_1+…+n_{q-1}\ )C_q[1]…C_q[n_q]$

A 有怎样的配置才会接受输出，主张 n_1+…+ n_q 的值是 $C_i[j]$，（i=1,…q 并且 j=1,…n_i）是随机并且独立分布。不仅仅是相互，但是询问信息和比特 b，而且，这在两个假设条件下都是成立的。为什么？这里使用一个关键的 CTR 模式特征，也就是，XOR 数据是计数器中 g 的值。根据配置观察：

$$C_i[j] = g(NtS_l(n_1+…+n_{i-1}+j))\oplus\begin{cases}M_{i,1}[j]在环境1\\ M_{i,0}[j]在环境0\end{cases}$$

目前，可以看到这是真正的中心。

G 应用的值是确定是明显的，因此 g 的输出是随机和独立的。无关紧要，那么，XOR 这些输出，回来是随机的。

这样随机判断机制对两个假设条件任何给定的输出尺寸是相等，既然攻击者决定基于输出序列的输出，返回 1 的可能性在下述两种假设条件是相同的：

$$P[Exp_{SE[Randl\to L],A}^{ind\text{-}cpa\text{-}1}=1] = P[Exp_{SE[Randl\to L],A}^{ind\text{-}cpa\text{-}0}=1]$$

A 的 ind-cpa 优势是零。

Algorithm E_g(M)
If|M|<L 那么返回⊥
如果|M|modL≠0 那么返回⊥
分解 M 为 M[1]…M[n]
R←{0，1，…，2^l-1}
For i=1,…,n 做
C[i]←g(NtS_l(R+i)⊕M[i]
EndFor
C[0]←NtS_l(R)
C←C[0]C[1]…C[n]
Return C

Algorithm D_f(C)
If|C|<l+L 那么返回⊥
If(|C|-l)mod L≠0 那么返回⊥
假设 C[0]是 C 的前 l 比特
分解剩余的 C 为 C[1]…C[n]
R←StN(C[0])
For i=1,…,n 做
M[i]←g(NtS_l(R+i)⊕C[i]
EndFor
M←M[1]…M[n]
Return M

图 6.3、版本 SE[G]=（K，E，D）是使用参数函数簇 G 的 R-CTR 体制。

6.7.2 定理 6.18 的证明

6.18 证明重新使用了许多在 6.17 中使用的方法。首先看 g 是随机函数，然后使用给定的 F 簇推论定理。如前所述，一个 G 函数的簇是$\{0，1\}^l$和值域$\{0，1\}^L$，一个 R-CTR 模式的参数版本是 SE（G）=（K，E，G），密钥产生算法是仅仅返回 G 的随机例子，意味着随机从 G 簇中选取一个函数 g←G，作为一个密钥观察 g，并且加密和解密算法按照 6.3 指示。这是主要的推论。

推论 6.22、假设 A 是任意的 ind-cpa 攻击者攻击 $SE[Rand^{l\to L}]$，那么：

$$Adv_{SE[Randl\to L],A}^{ind\text{-}cpa} \le \mu(q-1)/L2^l$$

这里 q 是使用 A 随机序列的数字并且 $\mu<L2^l$ 是这些序列的总长度。

定理 6.18 的证明是给定推论在这点上是容易的，因为上述几乎所有的定理 6.17 的证明。这样首先完成，继续证明 6.22。

定理 6.18 的证明、假设 A 是任意 ind-cpa 攻击者的攻击是 SE=（K，E，D）。假设 A 构造 q 随机判断机制总数是 μ 比特，并且有时间复杂度 t。将设计一个分辨机 D_A，因此：

$$Adv_{se,a}^{ind\text{-}cpa} \le 2\cdot Adv_{F,DA}^{prf} + \mu(q-1)/L2^l$$

此外，D_A 将使用 μ/L 判断机制并且有时间复杂度，然而注意如下的算法 E_g（·）变化了，现在开始图 6.3 此图 6.2。因为分析，仅仅的变化术语是：$Adv_{SE[Randl\to L],A}^{ind\text{-}cpa}$ 在等式 6.6，比零更多，对推论 6.22 是上限，因此，

$$Adv_{F,DA}^{prf} \ge 1/2\cdot Adv_{SE[F],A}^{ind\text{-}cpa} - 1/2\cdot\mu(q-1)/L2^l \qquad (6.7)$$

其余是事情就如前。

上述说明，有多普通是“仿真的”论点，定理 6.17 的证明，真正的，不仅仅是采纳容易随机版本体制，但是在许多体制下，一些体制是伪随机函数，对于消息认证的任务，密钥指出工作是 g 作为随机判断机制。

在推论 6.22 中的证明是分析确定的概率策略。单独的问题纯粹是概率上的，对加密甚至是密码学是不起作用的。

推论 6.23、假设 q,n,l 是正整数，假设 $n_1,n_2,\ldots,n_q<2^l$ 也是正整数。从{0，1，…，2^l-1}假设选取 q 个整数，$r_1,\ldots r_q$,随机单一讨论。考虑如下 $n_1+n_2+\ldots+n_q$ 数据，

$R_1+1,r_1+2,\ldots,r_1+n_1$

$R_2+1,r_2+2,\ldots,r_2+n_2$

…

$R_q+1,r_q+2,\ldots,r_q+n_q$

这里加是模 2^l 的加法，如果上述成员产生碰撞的机会是相等的，那么：

$$P[Col]\le(q-1)(n_1+n_2+\ldots+n_q)/2^l$$

这里 Col 表示碰撞产生的事件。

推论 6.23 证明、在这个领域的许多概率设置，这个问题是关于弹球放置与盒子数量、在 A.1 节中的生日攻击也是相关的。事实上，一个读者可以从附录中发现有用的研究。考虑有 2^l 个柜子，计数为 0，1，…，2^l-1。有 q 个球，计数为 1，…，q，对每一个球随机选择一个盒子 r_i，一个一个的选择盒子，这样首先选择 r_1，然后选择 r_2，…，这样继续下去。当安第一个球，定义上述阵列的第一个元素，也就是，r_1+1，…，r_1+n_1，这样选取 n_2，这样定义一个 r_2+1，…，r_2+n_2，在第一行的第一列的产生一个碰撞的值。对于第 q 个值，每次定义表格的一个行。在处理过程中的一个碰撞，为了提高边界值，可以用这个方法进行一步一步

概率的分析，也就是说，从某个角度去观察，固定的使用一些球，是否投超过一个的球会产生碰撞。最后，假设 Col_i 定义一个事件，表的前 i 个行可能产生一个碰撞，对于 i=1，…，q。假设 $NoCol_i$ 也定义一个事件，对于 i=1，…，q。那么，在这个条件下：

$$P[Col]=P[Col_q]$$
$$=P[Col_{q-1}]+P[Col_q|\ NoCol_{q-1}]\cdot P[NoCol_{q-1}]$$
$$\le P[Col_{q-1}]+P[Col_q|\ NoCol_{q-1}]$$
$$\le\ldots$$
$$\le P[Col_1]+\sum_{i=2}^{q} P[Col_u|\ NoCol_{i-1}]$$
$$=\sum_{i=2}^{q} P[Col_u|\ NoCol_{i-1}]$$

因此需要使用在前 i-1 个球没有产生碰撞，而投第 i 个球产生碰撞的上界，这样可以对数量获得更多的边界。

对于任意 i=2,…,q,有

$$P[Col_i|\ NoCol_{i-1}]\le(i-1)n_i + n_{i-1} +\ldots+n_1 \qquad (6.8)$$

首先看到为什么推论的证明是然后返回证明是正确的。对于上述的等式(6.8)，有：

$$P[Col]\le\sum_{i=2}^{q} P[Col_u|\ NoCol_{i-1}]$$
$$\le\sum_{i=2}^{q} (i-1)n_i + n_{i-1} +\ldots+n_1$$
$$=(q-1)(\ n_1+n_2+\ldots+n_q)/2^1$$

对于最后一个等式怎样处理呢？条件 n_i 产生第 i 个总数的重量 i-1，那么，对 j=i+1,…,q 在第 j 个重量为 1。

剩余是证明等式 6.8，为了得到一写直观的结论，从 i=1，2 开始证明。当投第一个球，产生碰撞的概率是 0，因为没有与之产生碰撞的前一列，这是显然的。第二列产生碰撞的概率是多少？问题转化为第二个球编号为 r_2+1，…，r_2+n_2 与编号为 r_1+1，…，r_1+n_1 的第一个球相等的概率是多少？观察 r_1 是固定的，当且仅当 r_1- n_2+1≤r_2 ≤r_1+ n_1-1。这样，（r_1+ n_1-1）-（r_1- n_2+1）+1，= n_1 + n_2 -1 选择 r_2 可以产生碰撞，也就是说，$P[Col_2|NoCol_1] \le(n_1 + n_2 -1)/2^1$。

当投更多的球时，需要扩展这些争论。这样假设 i-1 个球是已投入了，这里 2≤i≤q，假设在表的前 i-1 行没有碰撞产生，投入第 i 个球，并且想知道产生碰撞的概率。在固定的前提下，观察表 i-1 行，问题仅仅是某一个定义为 r_i 的数等于表的前 i-1 行的数的概率。一个小的观点是不好的，当存在 i-1 行是好的扩散的。现在合理使用碰撞的上界，除非有 i-1 的不同间隔是比只有一行困难。第 i 行可以与第一行交换使用，也可以是第二行、或其行直到 i-1 行。这样，得到：

$$P[Col_i|NoCol_{i-1}] \le((n_i+n_1-1)+(\ n_i+n_2-1)+(\ n_i+n_{i-1}-1))/2^1$$
$$=(\ (i-1)n_i + n_{i-1} + \ldots + n_1 -(i-1))\ /2^1$$

等式 6.8 是上述不等式的取反。

现在扩展推论 6.21 的证明去推 6.22。

推论 6.22 的证明：推论 6.21 的证明的思想是当 g 是一个随机函数，评价连续的计数值

产生一次一密。只要 g 产生不同的值，在计数的条件下，g 的输入总是不同的。在随机的条件下，也许是不同的。这个时间思想是明显不同的，也就是说攻击者没有优势，这不经常发生。现在可以介绍更多的细节。

攻击者安排一些随机判断询问 q，假设（$M_{i,0}$，$M_{i,1}$）是第 i 个询问，假设 n_i 是 $M_{i,0}$ 分组中一个数据，（这在 $M_{i,1}$ 是相同的数据分组），假设 $M_{i,c}[j]$是 $M_{i,c}$ 分组的第 i 个 L 比特的值，c∈{0，1}。假设 C_i 是随机判断询问（$M_{i,0}$，$M_{i,1}$）的应答值，有 n_i+1 个分组，第一个分组是 l 比特整数 r_i ∈{0，1，…，2^l-1}二进制表示，选择 E_g（$M_{i,b}$），这里 b 是 E_g（LR（.,.b））的随机判断挑战，另外的分组表示 $C_i[1],\ldots, C_i[n_i]$，描述为：

$$M_{1,b} = M_{1,b}[1]\ M_{1,b}[2]\ldots M_{1,b}[n_1]$$
$$C_1 = NtS_l(r_1)C_1[1]\ldots C_1[n_1]$$
$$M_{2,b} = M_{2,b}[1]\ M_{2,b}[2]\ldots M_{2,b}[n_2]$$
$$C_1 = NtS_l(r_2)C_2[1]\ldots C_2[n_2]$$
$$\ldots\ldots$$
$$M_{q,b} = M_{q,b}[1]\ M_{q,b}[2]\ldots M_{q,b}[n_q]$$
$$C_q = NtS_l(r_1)C_q[1]\ldots C_q[n_q]$$

如下假设时间 NoCol 的 $n_1+n_2+\ldots+n_q$ 是不同的，

$R_1+1,r_1+2,\ldots,r_1+n_1$

$R_2+1,r_2+2,\ldots,r_2+n_2$

$\ldots R_q+1,r_q+2,\ldots,r_q+n_q$

假设 Col 是 NoCol 的补，也就是说上述表格至少包括两个相同的值。为了分析方便，简记为：

$P_0[\cdot]$= 事件 • 在假设条件 0 的概率

$P_1[\cdot]$= 事件 • 在假设条件 1 的概率

使用如下的三个假设，随后会被证实。首先上述表并不依赖于处于哪一个假设条件的概率。

命题 1、$P_1[Col] = P_0[Col]$

第二个命题是 A 在左右选择的博弈时，表中没有碰撞产生。也就是，输出 1 的概率是同样的，在两种假设条件下，表的值的范围内没有碰撞产生的概率。

命题 2、$P_0[A=1|NoCol]=P_1[A=1|NoCol]$

也就是说，如果上表产生碰撞，A 的优势是无关仅要的。也就是说，也许是较大的优势。然而，仍然把碰撞概率认为是小的。已知在命题 1 的两个假设条件下碰撞的概率是相同的。

命题 3、$P_0[Col]\le(\mu(q-1))/L2^l$

而且看到完成命题证明是怎样进行的，然后返回并且证明。

引理的证明在于，是一个简单的条件争论：

$$Adv_{SE[Randl\rightarrow L],A}^{ind\text{-}cpa} = P_1[A=1]-P_0[A=1]$$
$$= P_1[A=1|Col]\cdot P_1[Col]+P_1[A=1|NoCol]\cdot P_1[NoCol]$$
$$- P_0[A=1|Col]\cdot P_0[Col]-P_0[A=1|NoCol]\cdot P_0[NoCol]$$

使用命题 1，命题 2，上述的等式为：

$$(P_1[A=1|Col]-P_0[A=1|Col])\cdot P_0[Col]$$
$$\le P_0[Col]$$

在最后一步，仅仅是表示为 1 的参数的边界。现在证明命题 3。

剩余就是证明三个命题了。

命题 1 的证明：事件 NoCol 仅仅依赖于选择加密算法 $E_g(\cdot)$随机值 $r_1\ldots r_q$ 。这些选择，

然而，在两个条件下严格的方法。两个条件下的不同是消息怎样加密，随机值是怎样选择的。

命题 2 的证明：给定事件 NoCol，在另一个博弈中，在新的点调用 g 函数进行评估。（这里使用假设 μ< $L2^l$，否则，对每个询问被唯一的分布）。因此，输出是随机的，并且在{0，1}^{L} 上分布。独立于其事件，给定推论 6.21，也就意味着基于体制的计数，也就是说，根据

体制观察：$C_i[j] = g(NtS_l(r_i + j)) \oplus \begin{cases} M_{i,1}[j]\text{如果处于条件1} \\ M_{i,0}[j]\text{如果处于条件2} \end{cases}$

因此每个分组密码是一个消息分组 XOR 一个随机值。这个的结果是每个分组密码是一个分布，独立于任何一个密码分组和消息。

命题 3 的证明：从推论 6.23 可得，简单的记为：$n_1+n_2+\ldots+n_q = \mu/L$
这就是证明的结论。

6.8 抗选择明文攻击的 CBC 安全

CBC 加密，在体制 6.4 中出现，是最普遍的模型。首先看到 CTR 模型因为容易分析。事实上，这里 CBC 模式加密的分析没有呈现，但是将描述结论。证明在[20]可以找到。

定理 6.24[20]假设 F：{0，1}^{k} × {0，1}^{l} → {0，1}^{L} 是一个函数簇，假设 SE=（K，E，D），是相应的 CBC 加密模式，在体制 6.4，将描述结论。那么对于任何 t,q,μ，有：

$$Adv_{SE}^{ind\text{-}cpa}(t,q,\mu) \le 2 \cdot Adv_F^{prf}(t,q',lq) + 2\mu^2/(l^2 2^l)$$

这里，q’=μ/L

注意消息都是 n 分组的，那么 μ=nql，因此上述附加的条件是 O（$n^2q^2/2^l$）。

这样试着提高计数器的使用，也可以用 CBC 模式做计数模式。但这种用法是不恰当的，这是一个好的方法去进行训练。注意这是真实的，当计数器随消息分组的增加而增加比仅从一个消息的增加更有优势。

关于计数器怎样加密？这些攻击几乎都消失了，但是生日攻击又在进行了。这样并不真正获得 IV 表，事实上，两种攻击是几乎相同的。F(c)中 c 计数确实是一个随机数。（并不相同，c 在明文中仍然可以使用）

分析这点，怎样通过已经构造的模型对这些体制。把 *f* 作为随机函数考虑，在这种条件下，随机 CBC 和加密计数模式是类似的。

6.9 对选择密文攻击的不可分辨机制

对于上述选择明文攻击考虑机密性，有时当攻击者有能力采用一中更强的攻击手段，称做选择密文攻击时，需要考虑机密性。在这种类型的攻击中，一个攻击者仍然可以接入一个加密机制。可以输入这个加密机制一个密文，返回一个明文。

这种情况是怎样发生的呢？假设一个攻击者在某些点可以得到暂时的访问权限，可以执行解密过程。可以输入这个加密机制一个密文，看到明文消息是怎样混合的。

如果攻击者可以接入加密随机判断机制，第一印象似乎是安全的，因此可以破解所有想具备的密码。为了产生一个安全深远的概念，使用解密判断机制是一个限制，为了看到是这些是什么，更近的看形式化问题。选择密文攻击的条件，考虑两个条件：

条件 0、攻击者提供判断机制 $E_k(LR(.,.,0))$与判断机制 $D_k(\cdot)$相同。

条件 1、攻击者提供判断机制 $E_k(LR(.,.,1))$与判断机制 $D_k(\cdot)$相同。

攻击者的目标是与选择明文攻击类似的，希望计算出处于哪个条件。没有简单的办法处理这个问题，也就是，使用两个清晰、长度相等的消息 M_0，M_1，返回密文 C，现在称做 C 上的解密判断机制。如果解密返回的消息是 M_0，那么攻击者在条件 0，如果解密返回的消息是 M_1，那么攻击者在条件 1，加以的约束是简单的称为解密判断是不允许的。更一般的，如果 C 预先返回 Ir 加密判断，称为非法的解密机制的一个询问。那么，仅仅是合法的用户才会被授权。在以下的形式化过程中，如果攻击者做一个非法询问，则经验值返回为 0。（在询问解密判断之后，如果 C 被 Ir 加密返回，这样很清楚询问 C 是合法的。当 C 是 Ir 加密判断说明 C 是非法的）

这个限制将给攻击者一些权利。特别的，引用一个 Ir 加密体制，采用一个加密 C，一个成功的选择密文攻击收益。更改为一个相应的密码 C’，用 C’做一个解密机制询问。用这个方法，攻击者寻找 C’询问。攻击者询问 C’产生在 M 下的消息区分攻击者，将在以下节 6.11 上阐明。

考虑这个模型也许是非常显然的。如果攻击者接入解密机制，怎样防止在确定的消息机制下在一个特定时间段，可以阻碍这个加密消息的访问。在不能接入这个加密体制后，有一些密文，并且俘获一些密文的安全。此外的动机是模型融合，怎样使用协议的加密体制。将看到一个认证密钥交换加密体制，攻击者对于讨论的攻击者有效的使用选择密文攻击方法。目前，假设提供定义并且进行练习。

定义 6.25 假设 SE=（K，E，D），是一个对称加密体制，假设 b∈{0,1}，假设 A 是一个算法可以访问两个模型，返回一比特，考虑如下的经验：

Experiment $Exp_{SE,A}^{ind\text{-}cpa\text{-}b}$

K ← k

D ← $A^{E,(LR(.,.b)),D,(.)}$

如果 A 对前密文询问 D_k（·）返回 $E_k(LR(.,.,b))$

那么返回 0

否则返回 d

A 的 ind-cca 优势定义为：

$$Adv_{SE,A}^{ind\text{-}cpa\text{-}b} = P[Exp_{SE,A}^{ind\text{-}cpa\text{-}b}=1]-P[Exp_{SE,A}^{ind\text{-}cpa\text{-}0}=1]$$

对于任意的 t,q_e,μ_e,q_d,μ_d ，通过如下的定义：

$$Adv_{SE,A}^{ind\text{-}cpa}（t,q_e,\mu_e,q_d,\mu_d）= \max_{A}\{Adv_{SE,A}^{ind\text{-}cpa}\}$$

这里 A 都具有时间复杂度 t 的最大值，长度至多是 μ_e 的询问 q_e 的总和是 Ir 加密判断机制。并且决定加密体制是长度至多是 μ_d 的询问 q_d 的总和是 Ir 加密判断机制。

特别的，在选择明文攻击的条件下，考虑协议的资源是相同的。一个询问 M_0，M_1，对于 Ir 加密判断机制是至多 q_d 询问，定义长度 M_0，时间复杂度是运算时间加攻击者时间的总和。考虑一个加密体制是抗选择密文攻击是安全的，如果一个合理的攻击者在区分 b=0,b=1 在给定判断机制访问权限后，不能获得明显的优势，但是可以合理使用资源。这个技术概念称为选择密文攻击条件下不可分辨，记为：IND-CCA。

6.10 选择密文攻击例子

选择密文攻击是足够有效去分析所有的运算模式，例如 CBC、CTR 模式（随后可以看到）对于选择明文攻击是安全的。一次一密体制对选择密文攻击也是不安全的，完善安全体制仅仅考虑选择明文攻击，现在讨论一些选择密文攻击的例子。

6.10.1 CTR 攻击

假设 F：$\{0，1\}^k \times \{0，1\}^l \rightarrow \{0，1\}^L$，是一个函数簇，假设 SE=（K，E，D），是 R—CTR 对称加密体制可以在体制 6.5 中描述。这个体制的弱点是容易使如下的选择密文攻击易受影响。也就是说，C[0]C[1]是 L 比特消息 M 的密文，因此，翻转比特 C[1]的 i 比特，得到一个新的密文 C[0]C'[1]。假设 M'是解密新密文得到的消息，那么 M'与 M 在 i 比特翻转后是相同的。（应该在 6.5 体制上检查和明确）因此，做为解密 C[0]C'[1]的判断询问，当已知道 M'，就得到 M。如下，可以知道怎样利用自己的模型去攻击对方的体制。计算出攻击者处于怎样的条件。

命题 6.26：假设 F：$\{0，1\}^k \times \{0，1\}^l \rightarrow \{0，1\}^L$，是一个函数簇，假设 SE=（K，E，D），是 R—CTR 对称加密体制可以在体制 6.5 中描述。那么：

$$Adv_{SE}^{ind\text{-}cca}(t,1,L,1,l+L) =1$$

T =O(l+L)加上一个 F 运算的时间。

攻击者在无法使用任何的资源条件下的优势是 1：对于每个判断只有一次询问，也就是表明这个体制是不安全的。

命题 6.26 的证明：将介绍一个攻击算法 A，具备时间复杂度 t，对 Ir 加密机制进行询问得到 1，这个询问的长度是 L，询问解密机制得到 1，是 l+L 长度的询问。并且有：

$$Adv_{SE,A}^{ind\text{-}cca} =1$$

命题如下。

记住 Ir 加密机制是 E_k（LR（.,.,b））输入一个消息对，返回一个对子里左或右的加密，依赖于 b 的值，A 的目标是确定 b 的值，攻击者的工作如下：

Adversary $A^{Ek(LR(.,.b)),Dk(.)}$

$M_0 \leftarrow 0^L$ ； $M_1 \leftarrow 1^L$

$C[0]C[1] \leftarrow Ek(LR(M_0,M_1，b))$

$C'[1] \leftarrow C[1] \oplus 1^L$ ； $C' \leftarrow C[0]C'[1]$

$M \leftarrow D_k(C')$

If $M=M_0$ 那么返回 1，否则返回 0

攻击者单独的 Ir 加密判断询问，是不同消息 M_0，M_1 对，每个分组的长度。返回密文 C[0]C[1]。翻转 C[1]的比特得到 C'[1]，并且返回密文 C[0]C'[1]到解密判断。如果返回 M_0，假设在条件 1，否则选择条件 0。这是很重要的，C[0]C'[1]≠C[0]C[1]，这样解密判断机制是合法的，现在，假设：

$$P[Exp_{SE,A}^{ind\text{-}cca\text{-}1} =1]= 1$$
$$P[Exp_{SE,A}^{ind\text{-}cca\text{-}0} =1]= 0$$

因此，$Adv_{SE,A}^{ind\text{-}cca}$ =1-0 =1。并且 A 达到这个优势，正好是 Ir 加密判断询问，谁的长度，每一个协议仅仅是 M_0 的长度，是 L 比特，并且是一个解密判断询问，长度是 l+L 比特。这样，$Adv_{SE,A}^{ind\text{-}cca}$（t,1,L,1,l+L）=1

为什么上述两个问题是真的？不得不返回一个问题定义的数量，与描述体制本身相同，并且全程处理。在条件 1，意味着当 b=1，假设 C[0]C[1]表示密文，返回 Ir 加密机制，假设 R=StN(C[0])，那么：$C[1]=F_k（NtS_l(R+1) \oplus M_1 = F_k（NtS_l(R+1) \oplus 1^L$

现在注意：

$M=D_k(C[0]C'[1])$

$= F_k（NtS_l(R+1) \oplus C'[1]$

$= F_k（NtS_l(R+1) \oplus C[1] \oplus 1^L$

$= F_k（NtS_I(R+1)）\oplus (F_k（NtS_I(R+1)\oplus 1^L）\oplus 1^L$

$= 0^L$

$= M_0$

因此，解密判断机制将返回 M_0 ，并且因此 A 将返回 1。在环境 0，意味着当 b=0，假设 C[0]C[1]表示密文返回 Ir 判断机制，并且假设 R= StN(C[0])，那么：

$$C[1]= F_k（NtS_I(R+1)\oplus M_0 = (F_k（NtS_I(R+1)\oplus 1^L）\oplus 0^L$$

现在注意：

$M=D_k(C[0]C'[1])$

$= F_k（NtS_I(R+1)\oplus C'[1]$

$= F_k（NtS_I(R+1)\oplus C[1] \oplus 1^L$

$= F_k（NtS_I(R+1)）\oplus (F_k（NtS_I(R+1)\oplus 0^L）\oplus 1^L$

$= 1^L$

$= M_1$

因此，解密机制将返回 M_1，因此 A 将返回 0，表示将用概率 0 返回 1。

一个对于 C-CTR 的攻击（体制 6.6）是类似的，并且告诉读者。

6.10.2 对 CBC 的攻击

假设 E：$\{0，1\}^k\times\{0，1\}^l \rightarrow \{0，1\}^l$ 是一个分组密码，假设 SE=（K，E，D）是与在体制 6.4 中描述的对称加密体制 CBC 是相关联的，体制的弱点是如下的选择密文攻击是易受影响的。也就是说 C[0]C[1]是一个 l 比特消息 M，对初始向量 IV 的 C[0]求反，得到一个新的密文组 C'[0]C[1]。假设 M'是解密新的密文得到的消息，那么 M'的第 i 比特取反后与 M 相同。（在配制 6.4 中可以得到原理）因此，发展解密机制 C'[0]C[1]的询问可以得到 M'，随后可以得到 M。随后，可以看到这个思想怎样用来破解模型的体制，计算出攻击者处于哪个条件。

命题 6.27、将介绍一个攻击算法 A，具备时间复杂度 t，对 Ir 加密机制进行询问得到 1，这个询问的长度是 l，询问解密机制得到 1，是 2l 长度的询问。并且有：$Adv_{SE,A}^{ind\text{-}cca}=1$

命题如下。

记住 Ir 加密判断机制 $E_k（LR（.,.b)）$得到一个消息对，返回一个左或者右的加密对，依赖于 b 的值。A 的目标是确定 b 的值，攻击者工作如下：

Adversary $A^{Ek(LR(.,.b)),D,(.)}$

$M_0 \leftarrow 0^l$ ； $M_1 \leftarrow 1^l$

$C[0]C[1] \leftarrow E_k(LR(M_0, M_1,b))$

$C'[0] \leftarrow C[0] \oplus 1^L$ ； $C' \leftarrow C'[0]C[1]$

$M \leftarrow D_k（C'）$

如果 $M=M_0$ 那么返回 1，否则返回 0

攻击者的单一 Ir 加密判断询问，使不同消息 M_0，M_1 对，每一个分组长度。返回密文 C[0]C[1]，取初始向量 IVC[0]的反 IVC'[0]，反馈密文 C'[0]C[1]去分解判断机制。如果返回 M_0，假设条件是 1，否则假设条件是 0。重要的是 C'[0]C[1]≠C[0]C[1]，这样解密判断机制是合法的。现在假设：

$$P[Exp_{SE,A}^{ind\text{-}cca\text{-}1}]=1$$

$$P[Exp_{SE,A}^{ind\text{-}cca\text{-}0}]=0$$

因此，$Adv_{SE,A}^{ind\text{-}cca}=1-0=1$。并且 A 通过仅仅使用一个 Ir 加密判断询问，长度是，协议是仅仅 M_0 的长度，是 l 比特，仅是一个解密判断询问。长度是 2l 比特，因此，$Adv_{SE,A}^{ind\text{-}cca}$

（t,1,l,1,2l）=1。

为什么上述两个等式是真的？不得不返回询问的定义，与描述体制本身是相同的。条件1 中，Ir 加密体制返回 C[0]C[1]，C[1]= E_k（C[0]⊕M_0） = E_k(C[0]⊕ 1^l）

现在注意：

M = D_k（C'[0]C[1]）
= E_k^{-1}（C[1]）⊕C'[0]
= E_k^{-1}（E_k（C[0] ⊕1^l））⊕C'[0]
= (C[0] ⊕1^l) ⊕C'[0]
=(C[0] ⊕1^l) ⊕(C[0] ⊕1^l)
= 0^l
= M_0

因此，解密判断机将返回 M_0，因此 A 将返回 1。在条件 0，意味这当 b=0，Ir 加密体制返回 C[0]C[1]，C[1]=E_k（C[0] ⊕ M_0）=E_k(C[0] ⊕0^l)

现在注意：

M = D_k(C'[0]C[1])
= E_k^{-1}(C[1]) ⊕C'[0])
= E_k^{-1}（E_k（C[0] ⊕0^l））⊕C'[0]
= (C[0] ⊕0^l) ⊕C'[0]
= (C[0] ⊕0^l) ⊕(C[0] ⊕1^l)
= 1^l
= M_1

因此解密判断机制将返回 M_1，因此 A 将返回 0，也就是说，以概率 0 返回 1。

6.11 对称加密的其它方法

6.11.1 伪随机函数的普通加密算法

存在普通的加密方法产生伪随机函数。假设有 m 比特消息需要加密，（m 足够大）。假设有一个伪随机函数簇 F，每个密钥 k 是说明了一个 l 比特到 m 比特的函数 F_k，对于一些固定的足够大的值 l。那么对于随机数 r，通过 E_k（M）=（r,F_k(r)⊕M）。通过计算 M=F_k（r）⊕C。这就是[97]中的方法。

定理 6.28[97]、假设 F 是一个输出长度为 m 的伪随机函数蔟。那么，配置（E，D）是上述的私钥，加密 m 比特的消息。

在后面 CBC，XOR 方式之间的方法，仅仅需要一个 l 比特到 l 比特的 PRF 映射，l 是固定的，与分组大小无关。得到一个 PRF 是使用 DES 或一些其他的分组密码。因此，CBC 和 XOR 是有效加密的方法。为了使用这种普通体制，对于大的 m 需要构造 PRF 上一个 l 比特到 m 比特的映射。

有几个构造大的 PRFs 进展是，依赖一个有效的算法思想，一个什么样的假设。在第五章已看到，伪随机函数簇可以构造一个给定的单向函数。因此可以使用这个方法，但是十分无效的。换言之，可以试着构造这些长度的扩展 PRFs，给定 PRFs 的长度，详见 5.11 节。

6.11.2 随机比特产生器的加密

一个伪随机比特的产生器是一个确定的函数 G，获得 k 比特种子密钥，产生一个看起来伪随机的 P(k)>k 比特序列。这些将在第三章中定义和研究。回忆所有的条件，是无效算法，可以区别一个随机 P(k)串和随机密钥 K 的串 G（K）。

回忆一次一密加密算法，仅仅使用 XOR 方式对密钥流进行运算。困难是很快就把密钥使用完，伪随机比特产生器提供所有自然产生序列的方法来得到这些。如果 G 是一个伪随机比特产生器，K 是比特共享密钥，多方共享长序列 G（K）。现在使用 G（k）比特 XOR 消息比特。永远不要重复使用 G（K）一比特。既然，可以延长一些多项式时间，这样有足够的比特进行加密。

更详细的，多方保持一个计数 N，初始化为 0。假设 G_i（K）表示 G（K）的第 i 比特输出。假设 M 是需要加密的消息，假设 M_i 是第 i 比特，并且假设 n 是长度。发送者计算 $C_i=G_{N+!}(K)\oplus M_i$ ，对于 i=1,…,n 并且假设 $C=C_1…C_n$ 是密文。这是传输到接受者的。现在，多方通过 N←N+n。比特总数加密是产生器 P（k）比特输出。使用 PRBGS 的定义工作如下：

定理 6.29、如果 G 是一个安全伪随机比特产生器，那么上述是一个安全体制。

一旦使用 PRBG 的优势是多方必须保持一致，同步计数，因此双方需要知道在序列 G（k）哪个位置。（注意，上述讨论的体制应该讨论这个问题，尽管上述的一些体制可以随意的使用一个计数器代替一个随机值，计数器不是同步的：发送方发送一个计数器，但是接收方没有也不理会发送方的计数器。为了得到这些，也许有发送者对每个消息发送当前的计数值 N，如果使用身份认证，值 N 应该被认证。

更多主要的优点在于，伪随机序列 G（K）应该不能随机访问。为了提出第 i 比特，一方可以从开始并且产生所有的比特。（这就意味着加密 M 的时间依赖过去加密消息的数量和大小，不是一个合适的特征）。作为选择，序列 G（K）是可以预先计算和存储的，但是使用了大量的存储，是否这个缺点存在或不依赖于 PRGB G 的选择。

因此，怎样得到伪随机比特产生器？在第三章会看到一些数论构造。这里有一些基于分组密码算法的较少的效率，但是是基于更优越的不同假设，更重要的，然而，这些构造是有缺点的，随机访问是不可能的。作为选择，一方可以构造一个基于有限域 PRFS 的伪随机比特发生器，从这一点可以看出随机访问是可以达到的，然而，加密结果体制从构造上与 XOR 体制是没有太大区别的。作为计数器的一点，还不清楚是否值得单独进行研究。

6.11.3 使用单向函数加密

在第三章，如果单向函数[113]存在则伪随机比特产生器存在，也知道，给定任何私钥加密体制，一人可以构造一个单向函数[114]。因此，有如下的定理：

定理 6.30、存在一个安全私钥加密体制当且仅当存在一个单向函数。存在安全公钥加密体制要求不同的假设种类，也就是存在简单的陷门。

6.12 安全注记

加密理论上的先驱工作是 Goldwasser 和 Micali[102]，在[148，96]中得以进一步的细化。这个工作的主体是无论在非对称体制的框架还是使用多项式时间攻击和可忽略的概率。对称加密的处理使用是引用 BellareRo 的对称密码学。对特别的定义 6.1，从[20]中获得具体的安

全体制。分析 CTR 加密体制，从[20]中给出定理 6.17 和 6.18。进一步可以在[23]中讨论 CBC 的安全。

6.13 练习和困难

困难 6.31、正规化一个安全的概念，并且证明命题 6.16 的类似情况。

困难 6.32、假设 l≥1，m≥2 是整数。假设 SE=（K，E，D），是一个给定的加密体制，的明文空间是$\{0,1\}^l$，意味着可以加密长度 l 的消息。为了可以加密更长一些的消息，定义了一个新的加密体制 $SE^{(m)}=(K,E^{(m)},D^{(m)})$有相同的密钥产生算法 SE，明文空间$\{0,1\}^{lm}$，并且加密和解密算法如下：

```
算法 E_K^(m)(M)                    算法 D_k^(m)(C)
    分解 M 为 M[1]…M[m]                分解 C 为（C[1]…C[m]）
    For i=1 to m do                For i=1 to m do
        C[i]←E_k(M[i])                 M[i] ←D_k(C[i])
    Endfor                             If M[i]=⊥那么返回⊥
    C←（C[1]…C[m]）                    EndFor
    Return C                       M←M[1]…M[m]
                                   Returen M
```

这里，M 是 lm 比特长。为了加密，M 分解为分组 M=M[1]…M[m]，每个分组 l 比特长。为了解密，C 分解为 m 串的序列，每组 c 比特长，这里 c 代表在体制 SE 下的一个密文分组长度。如果任何组成密文 C[i]是不完整的，（意味着 D_k 返回⊥）那么所有的密文 C[1]…C[m]是不完整的：

（a） 因此：

$$Adv_{SE(m)}^{ind\text{-}cca}(t,1,lm.1,cm)=1$$

对于一些小的 t，必须详细说明。

（b） 因此：$Adv_{SE9m0}^{ind\text{-}cpa} \le Adv_{SE}^{ind\text{-}cpa}(t,mq,lmq)$

对于任意的 t,q。

（c） 部分描述了 $SE^{(m)}$ 对选择密文攻击是不安全的，注意这是真正的无识 SE 的安全特征，也许抗选择密文攻击是安全的。（b）部分描述如果 SE 抗选择密文攻击是安全的，那么，$SE^{(m)}$ 是抗选择密文攻击是安全的。

难题 6.33、假设 E：$\{0,1\}^k\times\{0,1\}^l\to\{0,1\}^l$ 是一个分组密码，使用 IV 作为一个计数产生一个固定的对称加密体制，SE=（K，E，D）做 CBC 模式运算。密钥产生算法简单的算法，返回一个密钥空间$\{0,1\}^k$。加密和解密算法为：

```
算法 E_k(M)                        算法 D_k(C)
    如果|M|<l 那么返回⊥                如果|C|<2l 那么返回⊥
    如果|M|mod l≠0 那么返回⊥           如果|C|mod l≠0 那么返回⊥
    分解 M 为 M[1]…M[m]                分解 C 为（C[1]…C[m]）
    C[0]←NtS_l(ctr)                    For i=1,…,n do
    For i=1,…,n do                         M[i]←E_K^-1(C[i]⊕C[i-1]
        C[i]←F_K(C[i-1]⊕M[i])          EndFor
    EndFor                             M←M[1]…M[m]
    C←C[0]C[1]…C[n]                    Return M
```

ctr ← ctr+n

Return C

M 分解为每个长为 l 比特的分组，M[i]为第 i 个分组。对 C 分解，为 l 比特长度，这次分组计数从 0 开始。初始向量为 C[0]，选择整数 ctr 的 l 比特二进制表示。计数器用加密算法作为说明进行累加。

也就是说 SE 是抗选择明文攻击是不安全的，也就是说，提出一个小边界的 $Adv_{SE}^{ind\text{-}cpa}(t,q,\mu)$对于确定的特别的，t,q,μ 小的值，将详细、使用一个特定的大的值的低边界，证明代替和分析一个相应的攻击者。

攻击可以假设初始向量使用 LR 加密判断机制的计数使用是零。（这符合设置一个攻击者从目前使用体制加密）。一旦解决安置的困难，然而，试着找到未知初始 LR 加密判断机制的计数向量攻击者（这符合更多现实的设置，攻击者输入一个图片，这个体制有时会被使用）。

难题 6.34、假设 P：$\{0，1\}^k\times\{0,1\}^{2l} \rightarrow \{0,1\}^{2l}$ 是一个置换簇，定义一个对称加密体制，SE =（K，E，D），密钥产生算法对 P 简单返回一个随机密钥，意味着选取一个 K 比特随机密钥串并且返回，因此，密钥空间是$\{0，1\}^k$。消息空间是$\{0，1\}^{ln}$，这里，给定整数 n>1 有些位是固定的。加密和解密算法如下：

Algorithm $E_k(M)$

如果|M|≠nl 那么返回⊥

分解 M 为 M[1]…M[n]

Fori=1,…,n do

$R[i]\leftarrow\{0,1\}^l$;$C[i]\leftarrow P_k(R[i] \| M[i])$

Return C[1]…C[n]

Algorithm $D_k(C)$

如果|C|≠2nl 那么返回⊥

分解 C 为 C[1]…C[n]

For i=1,…,n do

$R[i] \| M[i]\leftarrow P_k^{-1}(C[i])$

Ruturn M[1]…M[n]

也就是说，只要 P 是安全 PRP，那么这个体制就是抗选择明文攻击是安全的。更仔细的，有：$Adv_{SE}^{ind\text{-}cpa}(t,q,lnq)\leq 2\cdot Adv_p^{prp\text{-}cpa}(t',q')+(n^2q^2)/2^l$ 当必须说明作为 t,q,l,n 函数的 t',q'值。提示：继续分析上述 CTR 加密模式。首先分析体制，使用在 2l 比特的随机置换，代替 P 的 $Perm^{2l}$ 簇。返回使用给定的 PRP P 模式的体制。

难题 6.35、 假设 l≥64 是一个整数，假设 P：$\{0，1\}^k\times\{0,1\}^l \rightarrow \{0,1\}^l$ 是一个伪随机置换。定义如下一个对称加密体制，密钥是一个随机选择 k 比特串 K，意味着对 PRP 体制的一个密钥。加密和解密算法如下：

Algorithm $E_k(x_1\ldots x_n)$

$R\leftarrow\{0,1,\ldots,2^l-1\}$

For i=1,…,n do

$Y_i\leftarrow P_K(<r+i>\oplus x_i)$

End For

Return$<r>y_1\ldots y_n$

Algorithm $D_k(<r>y_1\ldots y_n)$

For i=1,…,n do

$x_i\leftarrow P_k^{-1}(y_i)\oplus<r+i>$

End For

Return $x_1\ldots x_n$

这里加密算法是输入长度是 1 的整数倍，观察分解为 1 比特分组，$x = x_1\ldots x_n$，返回长度为 l(n+1)的串 y，解密算法是使用 y 返回 x。这里“+”定义为模 2^l 加，并且<j>定义为整数 j 的 1 比特二进制表示。

因此，这个体制是不安全的，更细致的，表示为：$Adv_{SE}^{ind\text{-}cap}(t,q,\mu)\geq 1/3$
这里 t.q.μ 是必须说明的值，应该尽可能小。

第七章 公钥加密

1976 年 DIFFIE 和 HELLMAN 在的文章[72]提出了公钥加密体制。在没有任何预先的公共密钥的基础上，革命性的思想可以保障发送和接收者之间的安全信息交换。提出了陷门函数的概念以及怎样使用陷门函数构造一个密码体制。简而言之，在 Rivest,Shamir,Adelman 提出了第一个后续的陷门函数 RSA。接下来是现代密码学的详细进展。

公钥体制的建立是用户 $u_1,\ldots,u_n$,网络比单独的用户对。每个用户 u 在网络中有一个密钥对$<P_u,S_u>$与之关联。在用户公钥 P_u，在每个人都可以访问公共目录下的用户名单下。私钥 P_u，是只有用户 u 知道。密钥对通过运算密钥加密算法产生。为了发送一个秘密信息 m 到 u 的每一个用户，使用相同的严格的算法，包括查找 Pu，计算 E（Pu，m），E 是一个公钥加密算法，结果是发送密文 c 到 u。在接收密文 C 之上，用户 u 可以用查找私钥 Su 进行解密，计算 D（Su，C），这里 D 是公钥加密算法。明显的，对于这个工作需要 D（Su，E（Pu,m））=m。一个特别的 PKC 是因此定义为一个公钥算法的三段论（G，E，D），密钥产生、加密和解密算法。

7.1 公钥加密的定义

目前形式上定义一个公钥算法，到目前为止，定义将不表示任何意思，意味着体制的安全。（这是随后部分的许多讨论的主题）。

定义 7.1、一个公钥加密体制是三段论，（G，E，D），一个概率多项式时间算法满足：

（1） 密钥产生算法：一个多项式时间算法 G 的概率期望，也就是，安全输入 1^k 产生一个对（e,d）,e 是公钥，d 是私钥。（注意：（e,d）$\in G(1^k)$）。也可以指一个加/解密对（e,d）。

（2） 一个加密算法：一个概率多项式时间算法 E， 输入一个安全参数 1^k，一个公钥 e 范围为 G（1^k）,串 m$\in\{0，1\}^k$，称为消息。作为输出串 C$\in\{0，1\}^*$，成为密文。（使用定义 C$\in$E（1^k，e，m）定义 C 作为一个加密算法，使用安全参数 k 的密钥 e。为了更清晰，简要记为：C$\in E_e$(m)或者 C$\in$E(m)。）

（3） 一个加密算法：一个概率多项式时间算法 D，作为一个安全输入 1^k，一个私钥 d 的范围为 G（1^k）。一个密文 C 的范围为 E（1^k，e，m）,作为输出串的 m’ $\in$D(1^k，e，m),产生输出串为 m’ $\in\{0，1\}^*$。因此在范围 G(1^k)对每一个对（e,d），对每一个 m，对每个 C$\in$D(1^k，e，m),概率 P（D(1^k，d，c)$\neq$c’）是可忽略的。

（4） 此外，系统是安全的。（见定义 7.3）

怎样使用这个定义。为了用安全参数 1^k 使用一个公钥加密体制（G，E，D），用户 A 运算 G（1^k）得到一对（e,d）加/解密密钥。用户 A 在文件中公开 e，并且保存私钥 d。如果任何人想发送给 A 一个消息，需要查找 e 并且计算 E（1^k，e，m），在接收 C$\in$E（1^k，e，m），一个计算消息 m= D(1^k，e，m)。

定义评论

评论 0、注意本质上私钥加密体制和公钥加密体制的没有任何区别，定义了一个私钥的加密体制以私钥 e 加密，公钥 d 解密。两个定义的不同在于安全定义的建立中，在一个公钥加密体制中，攻击者或“攻击算法”是给定 e，作为附加的输出；这里攻击者没有私钥体

制 e。（不失一般性的，假设 e=d）。

评论 1、在这个阶段，加密使用一个长度为 k 的密钥，仅仅定义长度为 k 的消息。通常延期为定义 7.1。

评论 2、注意算法 G 是多项式时间，输出（e,d）的长度为在 k 多项式时间是有限制的。另外，既然 k 也作为安全参数，反映了在 k 中的多项式和在 d 中的多项式。

评论 3、7.7 和 7.1 的定义也许是不严格的，因此不等式是以可忽略的概率实现的。简单的，选择更保守的估计采用。

评论 4、允许上述概率定义的加密算法，也就是说，对于相同的消息，有许多可能的密文。对于一个基于陷门体制的简单的公钥加密例子轮廓介绍，每个消息有一个相应的密文。这个太限制了，例如，如果 E 是确定的，相同的输入总能产生相同的输出，一个不需要的特征。

评论 5、D 是一个概率算法，也许相信考虑加密体制可以提供更高的安全保证[46]。从而，可以放松要求，对任意的 m，D（E（m））=m 可以保持高概率。

关于定义的协议

消息的长度不等于 k，并且把分成长度为 k 的分组后再进行加密，对最后的分组进行填充。扩展这个概念因此：$E_e(a_1 \ldots a_l a_{l+1}) = E_e(a_1) \ldots E_e(a_l)\, E_e(a_{l+1}p)$，这里$|a_1| = \ldots = |a_l| = k$，$|a_{l+1}| \le k$，并且 p 是一些标准的填充码 $k-|a_{l+1}|$。

上述规定可以用两种方法解释。首先，极端的波动在于加密体制仅仅用于加密长度与密钥长度相等的条件下。第二，允许减少加密许多信息的安全，使用相同的密钥加密单独的信息。第二个协定介绍了一个安全的分枝，是关于加密体制的。然而，随后看到的关于明文长度的一些信息是泄露了一个安全体制。加密算法映射的消息是相同长度的密码体制。

7.2 PKC 体制的简单例子：陷门函数模型

一个陷门函数，讨论在这一章单向函数和陷门函数的长度，定义为：$F=\{f_i : D_i \to D_i\}_{i \in I}$，I 是复数集，回忆任意 i，$f_i$ 容易被计算并且难于求逆；并且任意 i，存在 t_i 这样给定 t_i 并且给定 $f_i(x)$，$f_i(x)$在给定多项式时间是可逆的。

Diffie 和 Hellman 建议使用假设存在的陷门函数去完成如下的公钥加密体制：

（1） 作为安全参数的 1^k 产生器 G 输出是一个陷门函数对（f,t_f）并且 t_f 是与陷门信息相互关联的

（2） 对每个消息 m∈M，E（f,M）=f(m)

（3） 给定 c∈E(f,m)并且 t_f，D（t_f ，c）= $f^1(c) = f^1(f(m))=m$

7.2.1 陷门函数模型的难度

有几个直接的难题，对公钥加密使用陷门函数模型

简单的总结几个重要的问题，在下几节里描述：

（1） 特殊消息空间：F 是陷门函数的事实是不存在简单的逆 f（x），当 x 是特别难的。也就是，假设消息集，可以发送一个大的消息空间，例如一个英文语言集，或者简单

的 M={0，1}。也许容易求 f(m)逆。事实上，在 f(0)和 f(1)之间是容易分辨的。

（2） 部分信息：事实上 f 是单向函数和陷门函数，不必要暗示 f(x)所有关于 x 的信息。甚至一比特的泄露也许对一些应用太多了。例如，对于单向函数 $f(p,g,x)=g^x \bmod p$，这里，p 是素数并且 g 是产生器。X 的最低位总是容易从 F（x）计算的。对于 RSA 函数 $f(n,l,x)=x^l \bmod n$。Jacobi 符号 $J_n(x)=J_n(x^l \bmod n)$。也就是说，x 的 Jacobi 符号容易从 f(n,l,x)计算的，这从 Lipton[130]中是可以观察得到的。对于 Shamir Rivest 和 Aldman[185]的智力协议。见下述条件，此外，事实上，对于任意单向函数，诸如信息“f(m)的奇偶”性，m 总是容易从 f(m)中计算的。如下：

（3） 清晰的加密消息之间，可以发送消息，与通信的环节是相关的。一些例子是：发送相同的保密消息给几个相同的收件人，或几次发送同一个消息。因此实质上是值得和保持秘密从属关系的。在陷门函数模型中，发送两次相同的消息是几乎可测的。更严肃的问题被几个研究人员发现了，最值得注意的是 Hastad 提到的，如果 RSA 在指数 l 是可用的，相同的消息可以发送给 l 个收件员，那么消息可以被攻击者计算。

7.2.2 使用确定加密的普通困难

上述困难在于使用公钥密码体制是事实上共享的，加密算法是确定的。

对于上述困难 1 和 3 是显然的。容易如下表示困难 3：假设 E 是任意确定加密算法，使用如下相同的谓词，可以精确的计算出部分信息。P（x）= $\begin{cases}1, E(x) is even \\ 0, E(x) is odd\end{cases}$

明显可以计算出所有谓词，必须记录 E（x）低几比特，如果不是 E（x）。除非 E（x）总是偶数或是奇数，E（x）的低几比特不包括任何信息。但是 E（x）的其他比特必须包括一些信息，否则，消息空间仅仅能包括一个消息，在这个消息空间有所有的信息。那么，简单使用比特代替低比特，可以得到一部分简单的谓词信息。

7.2.3 RSA 加密体制

1977 年，Shamir,Rivest,Adelman 提出一个陷门函数的执行，RSA[176]体制。在第二章提到了，特别在 2.2.5 章和 2.17 章，彻底处理 RSA 陷门体制。

这里假设使用 RSA 陷门体制，对于 Diffie 和 Hellman 直接加密体制，将看到如果不满意目前的期望的安全体制，随后将看到随着工作的变化而产生变化。

回忆 RSA 陷门体制的 2.17 函数，假设 p,q 表示素数，n=pq，Z_n^*={1≤x≤n，（x,n）=1}，乘法群的势为 φ（n）=(p-1)(q-1)，并且 e∈Z_{p-1}，是与 φ（n）相关的素数。复数的集是 I={<n,e>因此，n=pq，|p|=|q|}。并且陷门与特别的摘要<n,e>是 $t_{<n,e>}$=d，这样，ed=1mod φ（n）。假设 RSA={$RSA_{<n,e>}$: $Z_n^* \rightarrow Z_n^*$}$_{<n,e>\in I}$ ，这里，$RSA_{<n,e>}(x)=x^e \bmod n$。

稀疏消息空间：也就是说，RSA 函数必须有一些不错的性质，看起来对于使用 PKC 是特别好。例如，给定一个对<n,e>，对任意 x，x∈Z^*，容易求 $RSA_{<n,e>}$的逆。这意味着 RSA 体制是难于破解的，对几乎所有的因子分解是困难的？回答否定。

假设消息空间 M 是研究的英文，假设 $M_k = \{0,1\}^k$，这里，m∈M_k 是一个英文句子。与整个空间相比较，英文句子是很小的一部分。例如：$|M_k|/|Z_n^*| \le 1/2^{\sqrt{n}}$ 。因此，对于所有 x∈M_k ，$f_{n,e}$(x)容易求逆。因数分解是困难的，换言之，英文判断是不同的结构。对所有这

一类的输入，函数是容易求逆的。明显的，可以最终将保证所有消息空间加密体制的安全。包括英文空间。

RSA 的部分信息：关于 x 的部分信息可以从 $RSA_{<n,e>}(x)$中计算。在单向和陷门函数，需要一些签名 RSA 的低比特和高比特隐藏得非常好，这是好消息。

一些情况下，对部分的信息泄露是敏感的，可以破解所有的加密方式，当 RSA 发明后，Lipton 提出灵活的信息。一个例子：智力游戏（SRA'76）：智力游戏是两方的协议，任何一方都不信任其他方，没有其他的欺骗方式能够应用。从 A 到 B 的一个协议栈包括：

（1） A 和 B 协商一个子集 $X=\{x_1,\ldots,x_{52}\}$,$x_i \in Z_n^*$是随机数集，这里 n=pq，p,q 是素数，并且 A 和 B 都已知。这些数表现一个协议栈，x_i 代表栈里的第 i 个卡。

（2） 一个 S 的挖掘是（$s,\varphi(n)$）=1，秘密选择 t，使得 $st\equiv 1(\mathrm{mod}\ \varphi(n))$。B 有相同的 e,f，（Ie，$ef\equiv 1(\mathrm{mod}\ \varphi(n))$）

（3） 对 i=1,2,…,52，计算 $x_i^s \bmod n$,混乱数据，发送到 B

（4） 对 i=1,2,…,52，一个计算$(x_i^s \bmod n)^e$ ，混乱数据，发送到 A

（5） 对 i=1,2,…,52，一个考虑$((x_i^s \bmod n)^e \bmod n)^t \bmod x_i^s \bmod n$,然后随机选择一个卡发送到 B

（6） B 然后计算$(x_i^s \bmod n)^d \bmod n=x_j$,这个 B 卡已被处理

为什么会工作：注意只要没有部分信息可以从 RSA 陷门体制可以获取的，既不是 A 也不是 B 可以影响 B 得到任何给定的卡的概率。A 不能给出 B 的坏卡，这个从加密卡，如同 B 不能处理自己的好卡一样。如下的加密事实是购置相同的密码锁类似的。只要卡所在盒子里，其他的参与者使用了这个锁，没有关于这个事情可以通告并且和其上锁的盒子是有区别的。当 B 从第三部得到一副牌，没有任何关于这副牌的信息，因此不能影响这副牌的出法。然而，A 可以告知禁止进入这个盒子。为了禁止 A 影响卡片，B 把自己的卡片放在盒子里并且进行混合。目前 A 不能区分卡片，因此禁止进行选择，禁止处理随机卡片。因此，只要两个参与者互相不信任才能玩游戏。

怎样从 RSA 获取部分信息：协议失败，然而，因为可能从 RSA 函数中获取信息，因此确定卡片影响移动的精确度，做这件事的另一种方法是计算 Jacobi 符号，S 是奇数则 $J_n(x_i))=(J_n(x_i^s))$。因此，x_i 平均有一半为 1 的概率，既然在 Z_n^* 上是随机数，可以粗略的从每一个卡片抽取 1 比特信息。为了个人习惯的影响，简单的使用 Jacobi 符号为 1 或-1 进行对这个游戏，然后再抽取卡片集。

一方对这点的反映是，当然，简单的修改协议的第 1 部， Jacobi 符号是 1 的数会被选择。通过计算 Jacobi 符号不能得到任何信息。然而，没有别的什么保证判断，不存在获取任何信息的方法。对于自然存在的一些陷门，确实这样的函数必须存在。

低指数攻击：假设指数为 e=3，可以看到任何与$\varphi(N)$相关的素数是可行，可以容易的选择 N=pq，这样 3 是与$\varphi(N)$=（p-1）(q-1)相关的素数。因为运行的原因，这是一个普通的选择。目前加密是快速的，RSA 在这个条件下仍然是单向的。

这样 m 的加密是 $m^3 \bmod N$，有一个 Coppersmith, Franklin, Patarin 和 Reiter 说明 RSA 体制弱点的有趣的攻击，假设加密 m 然后是 m+1。声称可以还原 m，有密文：

$C_1 = m^3$

$C_2 = (m+1)^3 = m^3 +3m + 3m^2 +1 = c_1 + 3m + 3m^2 +1$

现在试着假设解 m。也许首先通过源头，有 m 的二次方程。但是得到二次根是困难的，还不知道是怎样解这个问题。可以从以下工作中得到反映：

$(C_2+2C_1-1)/(C_2-C_1+2)=((m+1)^3+2m^3-1)/((m+1)^3-m^3+2)=(3m^3+3m^2+3m)/(3m^2+3m+3) = m$

这只是没有显著特点的。首先，可以产生一个消息 m 并且$\alpha m+\beta$对于已知的α，β。第二，

对比 3 大的指数进行工作。攻击可以在时间 O（e^2）内进行运行，这对小的指数是灵活的。最后，可以对高次数多项式相关的 k 消息进行攻击。这里有需要避免的几种相关攻击。

7.2.4 Rabin 公钥体制

回忆第二章描述的 Rabin 公钥体制 $F_n(m)\equiv m^2 \bmod n$

N 是两个大素数 p,q 的乘积，同样，这个函数可以产生另一个公钥陷门的例子，不同的是 f_n 不是一个置换，而是一个四选一函数。一个 $f_m(m)$的逆是：$F_n^{-1}(m^2) = x$ 。这样， $x^2 = m^2 \bmod n$

然而实践上，当求 Rabin 函数的逆，不仅仅简单考虑任何加密消息的二次根。但是四个中正确的一个意味着发送者正确的发送，并且可以容易的倾向于接受。因此，需要限制唯一的根 x 的识别。必须用在 $f_n(x^2)$上的解密函数的输出，因此找到一个 x 使得 $x^2 = m^2 \bmod n$，并且 $x\in S$，这里 S 是一个具备特征，这是非常不同的，存在两个根 $m,x \in S$ 。S 可以是什么呢？如果消息空间 M 在 Z_n^*是稀疏的（可以是相同的），那么 S 可以简化 M。在这种条件下，存在 $m\neq m' \in M$ ，因此，$m'^2 = m^2 \bmod n$，（如果 M 是不稀疏的，S 也许是所有 x，最后的 20 个整数是 r，最后 r 的随机值。然后秘密的发送 m，（$f_n(m')=f_n(2^{20} m+r),r$）需要发送。

回想，比这一部分更早的是，可以看到求 Rabin 函数的逆是与因数分解难度相同的。名义上，可以看到 Rabin 函数的逆，对 ε ，$m^2 \in M$ 是因数的采样。证明如下：

假设存在一个黑盒子，输入一个 x^2 ，有一个相应的 y，使 $x^2 = y^2 \bmod n$，那么，对于因数 n，从 Z_n^*中随机的选择一个 i，对黑盒子给出一个输入 $i^2 \bmod n$。如果黑盒子回应一个 y，这样 $y\neq\pm i$，那么通过计算 gcd(i±y，n)，因此，$y=\pm i$，不能得到任何多余的信息，然后重复。如果消息空间 M 是稀疏的，那么消息回答是否定的。为什么？对于上述使用的黑盒子，需要使用一个 $f_n(i)$反应，存在一个 y，$y\in M$ 并且 $y\neq i$。y 存在的概率是$|M|/|Z_n^*|$，也许是一个小的指数。

如果消息空间是不稀疏的，运算难题。Rabin 体制是不安全的，目前存在一个具备进行选择密文攻击能力的攻击者，很容易看到再次使用上述证明，求 Rabin 体制的逆函数的难度是与因数分解的难度是相当的。临时访问一个解码算法对 M 的 Rabin 公钥加密体制，与上述访问黑盒子的类似。攻击者随机选择 i，反馈解密算法 $f_n(i)$。如果攻击者反回 y，使得 $y^2=i^2 \bmod n$，（并且，$i\neq\pm y$），因数 n，得到秘密密钥。如果 M 不是稀疏的，试着用 *i* 的多项式分解，从这里开始，攻击者可以通过秘密密钥的帮助解密任何密文，而不需要黑盒子。因此，Rabin 体制与因数分解是不相同的，当在稀疏消息空间上求逆，或选择密文攻击是不安全的。

7.2.5 背包体制

有提议说一个公钥加密体制的数量是背包（更简单的说是子集问题）困难：给定一个整数向量 $a=(a_1,a_2,\ldots,a_n)$和一个目标值 C，决定是否存在一个 n 长度的零向量 x 使得 $a\cdot x=C$，这个困难是 NP 完全难题[91]。

使用背包难题作为公钥加密体制的基础，用产生背包向量 a 的方法产生一个公钥，然后公开作为的公钥。某人发送待加密的消息 M（M 是 n 长度的比特向量），把内积 $C=M\cdot a$ 的值发送给，解密密文是一个背包难度的例子。为了使这个问题更容易，还需要在背包体制中构造一个隐密的结构（陷门），这样加密运算成为一对一的，因此，可以解密一个收到的密文。然而，看来解决包括陷门的背包体制不是 NP 完全难题。因此解这种背包难题不再是

P=NP 的问题。

事实上，历史上还没有这样一种背包，大多数都被特别分析过了，使用 L^3 次方的格运算[132]。详见[143，186，188，2，190，128，48，157]。

一些背包和类似背包问题还没有被破解，Chor-Rivest 体制[60]和多重背包是例子，McEliece 有一个基于纠错码的背包公钥体制[143]。这些体制还没有被破解，是在加密过程中使用随机化的一个例子[142]。

随后将介绍一个安全公钥体制的加密体制。

7.3 安全定义

意味着安全的头脑风暴几个值得介绍的特征，开始介绍和构造一个最小的条件。

首先从一个已知的最初的一个公钥是不容易还原的，第二，对于任何消息空间的高可能性，消息不能完全从加密表格和公共文件中计算出消息。第三，可以从加密表格没有还原消息的可使用的信息。第四，对于消息的传送，攻击者无法获得任何有用的消息。例如包括两个消息发送并且确认是发送了的，也不希望两个连续解密的可能增大，如果传递的时间和以前加密的消息是已知的。

必须回答几个问题：在一个严格细致的数学定义下，怎样获得不透明封装？是否不透明封装是可进行数学抽象的？提出了几个获取安全不透明封装的分析定义，所有定义的提出是等价的，描述两个等价定义。

7.3.1 安全定义：多项式不可分辨性

非正式的，加密体制是在多项式时间不可分辨的，如果没有攻击者可以发现相等的两个消息，加密的消息是可区分的。如果回想封装分析，这个调度不能说明可以分解两个封装。

定义 7.2、公钥体制（G，E，D），是多项式时间不可区分的，如果每个 PPT 时间的 M，A 对于每个多项式 Q，任意足够大的 k 是

$$\Pr(A(1^k,e,m_0,m_1,c)=m|(e,d)\leftarrow F(1^k);\{m_0,m_1\}\leftarrow M(1^k);m\leftarrow\{m_0,m_1\};c\leftarrow E(e,m))<1/2+1/Q(k) \quad (7.1)$$

换言之，在多项式时间发现两个消息 m_0,m_1 ，使得多项式时间在 k 内是可能的，一个多项式时间算法在 $c\in E(e,m_0)$和 $c\in E(e,m_1)$。

关于定义的评注

（1）评注一个更强的安全机制可能是：上述隐藏的信息对于任意 m_0 ，m_1 ，不仅仅是这些在多项式时间的运算 M（1^k)。这种安全是在选择密钥前、非均匀的模式下发现的，因此不能包括一些关于密钥自己的信息。

（2）在私钥加密的条件下，定义发生轻微的变化，算法 A 不包括加密密钥 e。

（3）注意加密配置在加密算法 E 通过安全论证可以确定立刻失效，（例如给定 f, m_0 ，m_1 并且 $c\in f$（m_0 ，m_1）是可忽略的，这里 $c=f(m_0)$或 $c=f(m_1)$。

（4）注意如果攻击者知道消息一半加密，还不能区分一个消息同另一个消息的分类。

7.3.2 另一个定义：语义加密

考虑如下两个游戏，假设 h：M→{0，1}*，这里 M 是一个消息空间，简单的在多项式时间，或相等的，一个多项式时间算法 M 的概率是作为输入 1^k，并且产生一个消息 m∈{0，1}^{k}，并且 h(m)=1，如果 m 包含字母 e，那么 V={0，1}。

Game 1、告诉攻击者，将要选择消息 m∈M(1^k)，并且要求猜测 h(m)

Game 2、告诉攻击者 α∈E(m),对于一些 m∈M(1^k),并且要求猜测 h(m)

对于上述两个案例，假设攻击者已知消息空间算法 M 和公钥 P。

在第一个游戏中，攻击者仅仅知道将要选择的消息 m，对于另外的事实，Game 2 看到实际上的密文本身。对于所有类型的消息空间，语意安全是本质上要求攻击者赢得 Game1 的概率会和赢得 Game2 的概率大致相同。名义上，攻击者不能赢得任何优势或者任何形式的消息，从看到密文结果。

不同的是，定义将要求所有消息空间的所有概率分布，从密文不能计算出任何的信息。这要求香侬完全保密定义的记忆，以应对攻击者的计算边界。

定义 7.3、也就是说加密体制（G，E，D）是语意安全的，对于 PPT 时间算法 M 和 A，函数 h，多项式时间 Q，有一个 PPT 算法 B 对于足够大的 k，

$$\Pr(A(1^k,c,e)=h(m)|(e,d)\leftarrow G(1^k);m\leftarrow M(1^k);c\leftarrow E(e,m)) \leq \Pr(B(1^k)=h(m)|m\leftarrow M(1^k))+1/Q(k) \quad (7.2)$$

这里，游戏 1 由 PTM B 代表，游戏 2 由 PTM A 代表，然而，当加密算法的概率为 1 时，这里选择任何一个消息的编码是一种概率，否则，如果 E 是确定的，并且 M={0，1}，那么任何攻击者有 100%的机会正确猜测 h(m)，对于 m∈M，简单的测试是 E（0）=c 还是 E（1）=c。

定理 7.4、一个公钥加密体制通过不可分辨的安全测试，当且仅当可以通过语意安全测试。

7.4 公钥加密的概率

现在说明怎样实际建立一个公钥是多项式时间不可区分的。

为了这样进行，必须抛弃陷门函数的 PKC 模型，确定一起的加密算法，以满足加密算法的概率。构造的加密算法的概率是假设陷门函数，并且使用进行简单的分组。构造密钥手下首先回答一个简单的问题：怎样安全的加密单比特，有两种方法回答这个问题，首先是基于陷门预测在 2.5 节进行讨论，第二种基于核心谓词判断体制在 2.4 节进行讨论。

7.4.1 加密单比特：陷门判断

为了加密单比特，单向的概念和陷门谓词在[102]中进行了介绍。随后发现了对协议设计的完全使用，指出了 2.5 节的读者，对于这个主题的通常研究。这里，看一看加密的使用。思想：简言之，一个单向谓词是一个布尔函数，在很强的条件下仍然是难以计算的。名义上，一个攻击者不能计算出比随机化更好的判断值。还有，有可能对判断函数定义领进行采样，对于判断评估为 0 和 1，一个陷门函数支配额外的特征，存在一些陷门信息是判断计算。可以构造陷门判断集合的因子分解难题，RSA 求逆运算，从非剩余的二次剩余有区分的难题。

现在，给定一个存在的判断陷门，保证陷门函数的安全。在陷门判断机制中随机选择一

个元素，发送给定一比特信息 m 到 A。为了解密，A 使用陷门信息计算定义域上收到的判断值。注意，这里有一个概率加密算法，密文中 0 和 1 的概率是相等的，那么本质上，一个攻击者不可以区分一个编码 0 和一个编码 1。

回想，2.59 中一个陷门函数的定义

假设 I 是一个复数集合，对于 $i\in I$，假设 D_i 是一个有限域。一个陷门判断的集合是一个集合 $B=\{B_i : D_i \to \{0,1\}\}_{i\in I}$ 满足如下的条件。假设 $D_i^v = \{ x\in D_i , B_i(x)=v\}$。

1、 存在一个多项式 P 和一个 PTM S_1 ，输入 1^k 找到一个对子（I,t_i），这里 $i\in I\cap\{0,1\}^k$ 并且$|t_i|<p(k)$。信息 t_i 是指 i 的陷门。

2、 存在一个 PTM S_2，输入为 $i\in I$，并且 $v\in\{0，1\}$，输出 $x\in D_i$，这样，$B_i(x)=v$。

3、 存在一个 PTM A_1，使得对于 $i\in I$，并且陷门函数 t_i，$x\in D_i$，A_1（I,t_i,x）$=B_i(x)$。

4、 对于每个 PPT 时间存在一个可忽略的 ν_A 使得对于任意足够大的 k，

$$P[z\neq v: i\leftarrow I\cap\{0,1\}^k;v\leftarrow\{0,1\};x\leftarrow D_i^v;z\leftarrow A(I,x)]\leq \nu_A(k)$$

定义 7.5 假设 B 是一个陷门判断的集合，现在定义一个公钥体制（G，E，D）$_B$ 按照如下要求发送单比特：

密钥产生算法：G（1^k）选择(I,t_i)，运算算法 S_1 是可行的。

加密算法：假设 $m\in\{0，1\}$是消息。加密算法 E（I,e）选择 $x\in D_u^n$，因此，运行算法 S_2 是可行的。

解密算法：D（c,t_i）计算 $B_i(C)$。也就是给定陷门函数，使用 A_1 是可行的。

从定义一个陷门函数集合的定义角度是很清晰的，也就是说对于上述运算在期望的多项式时间并且那个消息可以使用这个方式发送。从陷门判断函数的定义在限制比特信息空间是在多项式时间是不可分辨的。

7.4.2 加密单比特：内核判断

作为选择，可以采取如下简单的接近，直接开始陷门函数并且使用内核判断。对于陷门函数细节的讨论，详见 2.59 节。这里的讨论假设已知有如下的先验：

复习一下单向陷门置换函数的集合：$F=\{D_i\to D_i\}_{i\in I}$ ，使得：

1. $S_1(1^k)$采样（I,t_i）,这里，$i\in I$, $|i|=k$ 并且$|t_i|<p(k)$对于一些多项式 P
2. S_2（i）采样 $x\in D_i$
3. $\exists$PTM A_1 使得 A_1（I,x）$=f_i(x)$
4. $Pr[A(I,f_i(x))\in f_i^{-1}(f_i(x))] < 1/Q(k), \forall$ PTM A, $\forall$ Q, $\forall k > k_0$
5. $\exists$PTM A_2 使得 A_2 （$I,t_i,f_i(x)$）$=x$, $\forall x\in D_i, i\in I$

此外，假设 B_i 是 $f_i(x)$的核心判断。之前说过 F 的存在意味着 F’的存在是一个核心判断。因此，对于符号的简单假设为 F=F’。也就是说 RSA 陷门函数的集合，LSB 是一个 LSB 核心判断的集合。

定义 7.6、给定一个核心预言的集合 F，定义公钥密码体制（G，E，D）$_B$ ，按照如下的流程发送单比特：

密钥产生算法：G（1^k）运行 S_1（1^k）,选择对<I,t_i>。（对于 RSA，G（1^k）选择<n,e>，d，这样 n 是一个 RSA 模，并且 $ed=1 \bmod \phi(n)$。）

加密算法：E（I,m）随机选择一个 $x\in D_i$ ，因此 $B_i(x)=m$，并且作为一个密文 $f_i(x)$输出，使用一个核心判断的 Goldreich Levin 构造，简单的选择 x,r，使得 x,r 的内积是 m，并且输出 $f(x)\cdot r$。（对于 RSA，加密比特 m，随机选择一个 $x\in Z_n^*$，使得 $LSB_{<n,e>}(x)=m$，输出

一个密文 $RSA_{<n,e>}(x)$。

解密算法：解密 $c=f_i(x)$，给定 I, t_i，解密算法 D（t_i,c）计算 $B_i(f_i^{-1}©)=B_i(x)=m$。使用 Goldreich Levin 构造总数的记录是 $c=f_i(x)\cdot r$ 计算 x,r 的内积。（对于 RSA，解密 c，给定 n,e,d,计算 LSB（（ $RSA_{<n,e>}(x)^d$）=x 的低比特）。

7.4.3 普通概率加密算法

是否可以加密任意长度的消息。

第一个回答是使用上述方法中的一种，简单的加密个别比特。同上述情况相同，在从一个有效的观点中考虑是否合适，需要讨论确实产生一个加密体制是否是多项式时间可分辨的。这个需要反馈，甚至个别比特是安全的。可以是这种情况，在所有比特条件下，一些比特上的预言计算是容易的，例如唯一比特或多个比特。如果正巧在这个条件外，则需要证明。现在提供构造和证明。

定义 7.7、在陷门集 F 上定义一个概率加密，使用一个核心比特 B 构造 PE=（G，E，D），这里：G（1^k）运行 S_1（1^k）, 选择对<I,t_i>（公钥是 i，私钥是 t_i）

假设 $m=m_1\ldots m_k$，这里 $m_j\in\{0,1\}$，是消息 E（I,m）按照如下流程加密：

选择 $x_i\in_R D_i$ ，使得 B_i（x_j）=m_j 对于 j=1,…,k

输出 $c=f_i(x_1)\ldots f_i(x_k)$

假设 $c=y_1\ldots y_k$ 这里，$y_i\in D_i$ 是密文，

D（t_i ,c）截面密文如下：

计算 $m_j = B_i(f_i^{-1}(y_i))$对于 j=1,…,k

输出 $m=m_1\ldots m_k$

命题 7.8 如果 F 是一个陷门置换的集合，那么概率加密算法 PE=（G，E，D）是不可识别安全的。

证明：假设（G，E，D）是不可识别安全的，那么有一个多项式 Q，一个 PTM A 和一个消息空间算法 M，使得对于任何的 k，$\exists m_0,m_1\in M(1^k)$ 满足：

$$Pr[A(1^k,I, m_0,m_1,c)=j|m_j\in\{ m_0,m_1\},c\in E(I,m_j)]]>1/2 + 1/Q(k)$$

这里，概率是使用随机投币概率事件 A，（I,t_i）∈G（1^k）,E 是投币集合，$m_j\in\{ m_0,m_1\}$。换言之，A 选择 0 更多一些，当 c 是 m_0 的一个加密算法，如果选择 1 更多一些，c 是 m_1 的一个加密算法。

定义一个分布 $D_j=E(j,s_j)$，对于 j=0,…,k。这里 $k=|m_0|=|m_1|$,并且 $S_0=m_0,S_1=m_k$ ， S_j 与 S_{j+1} 的区别正好是 1 比特。

假设 $P_j = Pr[A(1^k,i,m_0,m_1,c)=1|c\in D_j=E(j,s_j)]$

那么，1/2+1/Q（k）<Pr[A 正确的选择 j]=$(1-P_0)(1/2)+P_k(1/2)$

因此，$P_k-P_0>2/Q(k)$，并且既然 $\sum_{j=0}^{k-1}$（$P_{j+1}-P_j$）=P_k-P_0，$\exists$j 使得 $P_{j+1}-P_j>2/Q$（k）k

现在考虑如下的算法 B，产生输入 i,$f_i(y)$并且输出 0，1，假设 S_j,S_{j+1} 是在第 l 比特不相同的，也就是说，$S_{j,l}\neq S_{j+1,l}$,或者记为：$S_{j+1}=\overline{S_j}$ 。

B 按照如下的输入 i,$f_i(y)$进行运算：

（1） 选择 $y_1\ldots y_k$ 使得 B_i（y_r）= $s_{j,r}$ 对于 j=1,…,k

（2） 假设 $c=f_i(y_1)\ldots f_i(y)\ldots f_i(y_k)$，这里，$f_i(y)$在第 l 分组代替为 $f_i(y_l)$

（3） 如果 A（1^k,i,m_0,m_1,c)=0 那么输出 $S_{j.l}$

如果 A（1^k,i,m_0,m_1,c)=0 那么输出 S_{j+1} = $\overline{S_j}$

注意：如果 $B_i(y)= S_{j.1}$,那么 $c\in E(j,s_j)$，如果 $B_i(y)= S_{j+1.1}$,那么 $c\in E(j,s_{j+1})$。因此，在算法 B 的第三步，输出 $S_{j.1}$ 符合 A，预测 c 是 S_j 的编码，换言之，c 是最靠近 m_0 的编码链。

命题：$Pr[B(i,f_i(y))=B_i(y)]>1/2+1/Q(k)k$

证明：$Pr[B(i,f_i(y))=B_i(y)]=Pr[A(1^k,i,m_0,m_1,c)=0|c\in E(i,s_j)]Pr[c\in E(i,s_j)]$

$+Pr[A(1^k,i,m_0,m_1,c)=1|\ c\in E(i,s_{j+1})]$

$\geq(1-P_j)(1/2)+(P_{j+1})(1/2)$

$=1/2+1/2(P_{j+1}-P_j)$

$>1/2 + 1/Q(k)k$

因此，B 将预测 B_i（y）是 $f_i(y)$的核心谓词判断。

概率加密 PE=（G，E，D）可分辨安全的。

事实上，概率加密 PE=（G，E，D）是语义安全的。这是基于语义安全和可分辨的安全是等价的。

7.4.4 有效概率加密

概率配置怎样才有效？在目前的计划中，根据参数设置，密文比明文要多一些。然而，正使用随后的伪随机数据产生器产生陷门的证明方法[39，43]，仅仅附加的一个函数，构造多项式安全的概率体制使用的密文长度比明文长度大。最有效的概率加密体制归于 Blum ,Goldwasser[43]，根据数据和扩展速度，比得上 RSA 确定加密体制。回复私钥加密看来更有效。事实上，实践的公钥加密体制，在从不会面的两方，通常用于传送会话密钥。随后秘密会话密钥使用私钥加密方法。

首先描述一个概率加密公钥加密体制 PE，对于一个陷门置换的集合 $F=\{f_i:D_i\rightarrow D_i\}$使用核心判断 $B=\{B_i: D_i\rightarrow\{0,1\}\}$,这一节，$D_i\subseteq\{0，1\}^k$，这里 k=|i|。

当 EPE=（G，E，D）是基于 F 的 PKC：

密钥产生 G（1^k）=$S_1(1^k)$=(i,t_i)，公钥是 i，私钥是 t_i。

加密算法：加密 m，E（i,m）运算如下，这里 l=|m|:

(1) 随机选择 $r\in D_i$

(2) 计算 $f_i(r), f_i^2(r),\ldots, f_i^l(r)$

(3) 假设 $P=B_i(r)\ B_i(f_i(r))\ B_i(f_i^2(r))\ldots B_i(f_i^{l-1}(r))$

(4) 输出密文 $c=(p\oplus m, f_i^l(r))$

解密算法：解密一个算法 c=(m’,a),D(t_i,c)做如下运算，这里 l=|m’|:

(1) 计算 r，使得 $f_i^l(r)=a$ 。这样可以使陷门信息求 f_i 的逆，t_i，既然 f_i 是一个置换，那么这个 r 是唯一的。

(2) 计算上述的底码并且加密 $P=B_i(r)\ B_i(f_i(r))\ B_i(f_i^2(r))\ldots B_i(f_i^{l-1}(r))$。

(3) 输出解密信息 m=m’ $\oplus$p。

为了考虑这个信息的有效性，注意信道的带宽为|c|=|m|+k，这里 k 是上述定义的安全参数。在|m|·k 有明显的进步，在前一章，在这个配置的前提条件下，对于安全进行最小限度的带宽加大。

如果 C_{i1} 是计算 f_i 的消耗，并且 C_{i2} 是给定 t_i 计算 f_i^{-1} 的消耗，那么加密的消耗是|m| C_{i1}，解密的消耗是|m| C_{i2}。假设计算 B_i 的消耗是可忽略的。

其余的兴趣在于，对所有目前与陷门相关的函数是，甚至与 t_i 相关，计算 f_i^{-1} 比计算 f_i

更容易。也就是说，$C_{i1} < C_{i2}$，尽管如此，两个都是 k=|i|的多项式。因此在 EPE，对计算 f_i^{-1} 是更有效和可行的，同 f_i^{-1} 的成分相同，那么计算 r= f_i^{-1}(a)，计算 f_i(r)，f_i^2(r)，…，f_i^{l-1}(r)可以减少全部解密的消耗。如下的运行证明了这一点。

7.4.5 一个 EPE 的运行与 RSA 的消耗是相同的

这一节，考虑一个特殊的等同与 RSA 的运行。这是在第五节中，Rabin 陷门函数的一个子集 F 的一个使用方法。如果只考虑 Blum 素数，在二次剩余集合的定义域上可以还原 Rabin 函数成为置换。事实上，将讨论限制在形如 p≡7 mod 8 的素数。

假设 N={n|n=pq，|p|=|q|；p,q≡7 mod 8}。假设 F={f_n:D_n→D_n}$_{n∈N}$，这里，f_n（x）≡x^2mod n，并且 D_n=Q_n={y|y≡x^2mod n}。因为 p,q≡3 mod 4。有 f_n 是一个在 D_n 上的置换，B_n(x)是 x 的（LSB）至少 n 比特，例如是一个因式分解问题的难度。从第 5 节的观点看是成立的。（这在实际上是成立的，但是没有正式成立）。考虑 EPE（G，E，D），有：

密钥产生 G（1^k）=(n,(p,q)),这里 pq=n∈N，并且|n|=k。因此，n 是公钥，并且（p,q）是私钥。

加密：E（n,m），这里 l=|m|

(1) 随机选择 r∈Q_n

(2) 计算 r^2,r^4,r^8,…,r^{2l}(mod n)

(3) 假设 p=LSB(r)LSB(r^2)…LSB(r^{2l-1})

(4) 输出 c=(m⊕p，r^{2l} mod n)

因此加密的消耗是 O（k^2·l）

解密：D((p,q),c)，这里，c = (m’,a)，l=|m’|

(1) 计算 r，使得 r^{2l} ≡ a mod n

(2) 计算 p= LSB(r)LSB(r^2)…LSB(r^{2l-1})

(3) 输出 m=m’ ⊕p

既然，p,q≡7 mod 8，因此 p = 8t + 7 并且 q = 8s+7 对于整数 s,t。从第三节可得 p 是素数，Legendre symbol $J_p(a)=a^{(p-1)/2}$ ≡1 mod p 当且仅当 a∈Q_p，既然 a∈Q_n，同样有 a∈Q_p，因此计算：

$$a \equiv a \cdot a^{(p-1)/2} \equiv a^{1+4t+3} \equiv (a^{2t+2})^2 \pmod p$$

产生，$\sqrt{a}$ =a^{2t+2} mod p。此外，a^{2t+2} = $a^{(t+1)2}$ ∈Q_p，因此，可以重复使用这种方法找到：r_p ≡$\sqrt[2^l]{a}$ ≡$a^{(2t+2)^l}$mod p。（为什么要求 p≡7 mod 8）。类似的，可以找到 r_p ≡$\sqrt[2^l]{a}$ ≡$a^{(2t+2)^l}$mod q，使用中国剩余定理，可以找到一个 r≡$\sqrt[2^l]{a}$ mod n。解密的代价在于 O（k^3·l）。

然而，可以直接计算 a^u mod p 并且 a^v mod q，使用中国剩余定理，使用给定的 r 计算 p，然后加密。这个运算量是 O(k^3+k^2·l)=O(k^3)如果 l=O(k)。

EPE 通过不可分辨的安全

希望看到EPE可以通过不可分辨的安全，为了做到这一点。使用伪随机数产生器（PSRG）的概念，注意 PSRG（r,i）=f_i^l(r)·B_i(r) B_i(f_i(r)) B_i(f_i^2(r))…B_i(f_i^{l-1}(r))=a·p，这里 a,p 是消息加密时产生的，是一个伪随机数产生器。事实上，这是给定单向置换，使用 PSRGS 的存在性的构造。当然，如果 P 的底码完全随机，就不可能解密消息，既然 m’=m⊕p 映射到对于 m 产生的一个随机串。既然 p 是伪随机的，如果没有此外的信息，对于任何 PTM 看起来是随机的，

陷门 t_i。然而，攻击者确实已知 $a=f_i^1(r)$，必须得出使用这部分无法计算出 p。

更细致的，如果这里存在一个 PTM 算法 A，当 R 是完全随机串{0，1}^{1}，可以区分（m⊕p）·a 和（m⊕R）·a。这样就可以区分 p·a 和 R·a。可以这样使用，做一个概率测试证明是否 PRG 产生的输出序列是满足这个概率的。如果 PRSG 是随机矛盾的，就可以声称 f_i 是单向函数，并且不能使用这个计算出 p。

7.4.6 基于加密的实践 RSA：OAEP

考虑一个发送方，有一个 kbit 到 kbit 的陷门置换函数 *f*，希望传输一个消息到一个接收者有一个置换的逆函数 f^1。需要加密的文件长度精确的是 k；文件 x 的长度 n 是接近于 k，这种条件的例子是[179，117]。

不幸的是，这是一个有启发式的体制。一个可证安全的机制是可取的，已经看到可证安全效果较好的公钥体制的例子。最有效的是 Blum-Goldwasser 体制[43]，但是，不幸的是还是不能与启发式的体制相称。因此，实践方继续启发式的构造。

这一节称这个体制为 OAEP（最佳非对称加密体制），可以完全使用这个隔阂。可以使用 Bellare 和 Rogaway[25]作为设计。这迎合了实际构造，同时也可以有效的满足合理的安全要求。假设一些单向函数是理想的，这个体制可以证明是完善的。正式的，单向函数作为随机预言机制的模型。在执行上，单向函数是起源欲密码学单向函数。

这个随机预言机制表示了一个概念的折衷，在这个条件下可以进行有效而合理的安全保障。见[15]对这点进展的讨论。

RSA-OAEP 包括目前几个标准和标准草案，在不同的几个体制下执行。特别的，这是一个 RSA PKCS#1 V2 加密标准并且也同样是 IEEE P1363 /P1363a 标准草案。这同样也使用在 Visa 和 Mastar 卡的 SET（安全电子传输）协议上。

简单的嵌入配置和 OAEP 特征

启发式的体制总是使用下述条件：一个概率求逆的 x 嵌入一个串 r_x 并且 x 加密为 $f(r_x)^2$，称为简单嵌入体制的一个处理过程。作为一个目标构造一个可能好的简单的嵌入体制，假设 n 接近于 k。

简单嵌入体制的例子是 RSA PKCS#1 标准，这个作为 AD HOC 标准，标准假设为陷门置换，这里 RSA 并不是简单的体制安全。事实上，这个体制满足于选择密文攻击[34]。如下讨论的 OAEP 体制是与 RSA PKCS#1 的体制，但是拒绝这种攻击。然而，这种抵抗是安全证明的。RSA PKCS#1 标准的新版本是，命名为 V2，使用 OAEP。

OAEP 是简单的比特优化嵌入体制，（x 串长度可以用 $f(r_x)$加密为长度 k）。这种证明在单向函数是理想的前提下，可以看到是 RSA-OAEP 达到语义安全（作为[102]的定义），可以在[89]中看到，同时到达一个概念叫做已知明文攻击，在[25，24]中定义。随后的定义中，特别在[21]定义语义安全，对于在理想单向函数模型下的选择语义安全和非延展性。

现在简单描述基本定义和特征。对于提交读者的完全描述和[25，91]证明是安全的。

体制：

回忆 k 是一个简单的参数，*f* 是一个 k 比特到 k 比特的单向陷门置换函数。假设选择 k_0，因此攻击者的运算时间比 2^{k0} 小。固定消息的长度加密为 $n=k-k_0-k_1$，（短消息可以使用相应的填充码进行长度的适应）。体制使用一个产生器 G：{0，1}^{k0}→{0，1}$^{n+k}$，一个 HASH 函数{0，1}$^{n+k}$→{0，1}^{k0}。选择一个随机 k_0 比特加密{0，1}^{n}，并且设置：

$$E^{G,H}(x) = f(x0^{k1} \oplus G(r) \| r \oplus H(x0^{k1} \oplus G(r)))$$

解密 $D^{G,H}$ 定义如下，使用 f^{-1} 进行解密，找到一个形如 a ‖ b 的串，$|a|=k-k_0$，$|b|= k_0$。计算 $r=H(a) \oplus b$，并且 $y=G(r) \oplus a$。如果最后 y 的 k_1 比特不全为 0 那么拒绝，另外输出 y 的前 n 比特作为明文。冗余的使用（0^{k1} 条件是解密）为了提供已知明文。

效果

函数 *f* 能够对任意的诸如 RSA 的陷门置换选择集合或者有标志的平方[164，35]，在这种计算 G 和 H 的时间对于计算 f,f^{-1} 的时间是可忽略的。根据这种体制，仅仅要求一种简单的 *f* 的加密应用计算，一个简单的 f^{-1} 的解密计算。密文的长度为 k（与 $k \geq n+k_0+k_1$）。理想单向函数范例：

作为上述的指示，当证明安全可以随机的选择 G，H，当考虑一个具体的体制，G，H 是起源与单向函数的指示。在这个条件下，根据范例[15]争论在一个理想单向函数并不提供计算模型的可证安全，假设一个理想随机单向函数和比单纯的 ad hoc 协议设计提供更大的安全保障。指出阅读这篇文章此外深远、有目的理想单向逼近的考虑。

细致的考虑

实践需要此外的结果。特别的，这就意味着应该可以谈论更深远的安全机制安全参数说明值（例如 k=512），需求不仅仅是回避渐近和严密的地址安全，这样保证安全归约尽量有效。因此，提供了基本体制办法的安全，潜在攻击者可能的概率是：运算时间 t，开始 q_{gen} 的 G 询问，并且 q_{Hash} 是 H 询问。假设攻击者可以用 ε 优势。然后提供一个算法 M 和数字 t’，ε’，因此，在时间 t’与 M 的逆在陷门置换 *f*，加强的结果是在 t’，ε’是 q_{ge} ，q_{Hash} 的函数 $k,k_0,n(k=k_0+n)$。现在一个用户使用特别的 f 的强度能够得到一些观点破解加密体制，见更详细的[25]。这样完成了已知明文，因此抗选择明文攻击是安全的。当 *f* 是 RSA 体制时，见[89]。

7.4.7 进一步讨论

Johnson 和 Matyas[118]对于 OAEP 的增加，用做冗余，代替上述 0^{k1}，一个与密钥相关的信息 HASH，OAEP 的版本在 ANSI X9.44 草案标准。

7.5 探测活动的攻击者

到目前为止，对活动的攻击者已经做了很多的研究。如果攻击者是活动的，会发生什么？这点给出了比语义安全更强的概念，例如非扩展[75]概念。抗选择密文攻击的安全，并获得明文[25，21]。见[21]对于这些概念的划分，在其中进行相关的讨论。

特别的，考虑抗选择密文攻击。在这个模型下，假设攻击者有临时的解码设备访问。可以使用选择解密一些密文。攻击者可以直截获得密文而不需要合法访问。注意，这样产生简单的消息和密文对，作为攻击者总是简单的使用加密消息。在这种条件下，攻击者得到了密文的选择，从解密设备得到相应的消息。

在前一节中，这样的攻击者可以完全的破解 Rabin 体制。还是不知道是否其体制在目前的攻击中是把 PKC 作为安全的前提讨论的。然而，可能的防护尝试是抗这样的攻击。

一个思想是检查解密设备解密失败，攻击者仍然可以获得一些知识。这样 α_1 和 α_2 可以加密相同的消息。例如，上一节提出的概率加密体制，一个攻击者希望学习核心知识比特 B_i（y）在一些 y 是未知的条件下，这里可以直截获得 $f_i(y)$。给定具有保护的解密设备进行上述的描述，攻击者可以发现如下的这个比特：

（1） 选择 $m \in M(1^l)$，消息空间，假设 b 是消息 m 的最后一比特

（2） 随机和独立的选择 $\alpha_1 \in E$（i,m）

（3） 回忆 $\alpha_1 =（f_i(x_1)，f_i(x_2), \ldots, f_i(x_l)）$，从 D_i 中随机选择 x_l ，对于 j=1,2,…,l。假设 $\alpha_2 =（f_i(x_1),\ldots,f_i(x_{l-1}),f_i(y)）$

（4） 对 $c=(\alpha_1, \alpha_2)$使用解密设备，如果回答是 m，那么 $B_i(y)=b$。如果不需要解密 c，那么，$B_i（y）=\bar{b}$

做了什么代替使用非交互零知识证明（NIZK）[45，153]，思想在于任何一个人可以检查 NIZK 看是否正确，但是没有可以从中提取的知识。如果陷门函数存在，Shamir,Lapidot 证明那么 NIZK 存在。这样密文由三个部分组成：两个不同的消息编码 α_1 ，α_2 ，并且一个 NISK 的 α_1 ，α_2 加密同一个消息。那么一个解密机制在对方没有给定任何新的知识条件下拒绝解密，因为已知解密无效。

选择密文攻击的重要实践是最近引起广泛关注的 Bleichenbacher 对 RSA PKCS#1 的加密标准。Bleichenbacher[38]展示怎样在选择密文攻击条件下破解一个加密体制，一个人可以在条件 7.4.6 下记录对这种攻击免疫 OAEP。

第八章 HASH 函数

一个 HASH 函数通常是一个压缩函数，也就是输出小于输入。通常这样的函数是任意长度的输入，固定长度的输出，比如 160 比特。HASH 函数使用在密码学的许多部分，有许多不同类型的函数，他们具备不同类型的密码学特征。在这章进行讨论。

8.1 HASH 函数 SHA1

是一个简单但是特殊的函数，固定输出为 160 比特。函数在 1995 年完成，作为联邦信息处理标准 FIPS 成为 NIST 的 SHA1。

假设$\{0，1\}^{<l}$ 定义为长度小于 l 的串的集合，函数 SHA1：$\{0，1\}^{<2^{(64)}} \rightarrow \{0,1\}^{160}$ 在图 8.1 中可见。(因为 2^{64} 是一个很大的长度，那么认为 SHA1 是一个允许任意长度输入的 HASH 函数)。开始先通过 shapad 进行消息填充，然后进行消息填充迭代得到输出。图 8.1 中进行的算法运算在图 8.2 中进行了描述。(首先输出的 SHF1 编码是 128 比特串写成 32 比特的四个字，每个字由 8 个十六进制组成，在 SHF1 中还包括初始变量 V 的编码)

SHA1 是 Ron Rivest 在 1990 年设计的 MD4 的衍生设计，SHA1 背后关键的设计思想在 MD4 中已体现了。另外 SHA1 更象是 MD4 和 MD5 的孩子，真正意义上仍然是 Rivest 的设计思想体现。MD4，MD5 和 SHA1 有相同的结构，前两个算法产生一个 128 比特的输出，然后进行 512+128 比特到 128 比特的链式函数结构变换，而 SHA1 是重长度为 512+160 比特到 160 比特链结构产生 160 比特的输出。

那么 SHA1 希望做些什么？首先需要确定没有人可以找到不同的串 M 和 M'，使得 SHA1（M）=SHA1（M'）。这个特点叫做抗碰撞特征。

考虑抗碰撞特征，这种特点是很令人吃惊的。函数 SHA1 把任意长度串映射到 160 比特。因此即使限制定义域为短的串，比如是 256 比特，那么有大量的串 M 和 M' 可以 HASH 为相同的值。从鸽笼原理就可以知道：如果 2^{256}（256 比特消息）只鸽子放入 2^{160}（160 比特 HASH 值）只笼子，那么一定有两只鸽子（两个消息）放入同一个笼子（HASH 值）。难度在没人可以找到不同的自变量得到相同的值，就是找到在一个笼子中的两只鸽子。

algorithm SHA1(*M*) // *jMj* < 264
V ← SHF1(5A827999||6ED9EBA1||8F1BBCDC||CA62C1D6 *; M*)||
return *V*
algorithm SHF1(*K;M*) // *jKj* = 128 and *jMj* < 264
y ← shapad(*M*)
Parse *y* as *M*1||*M*2|| ¢ ¢ ¢||*Mn* where *jMij* = 512 (1 · *i* · *n*)
V ← 67452301||EFCDAB89||98BADCFE||10325476||C3D2E1F0
for *i* = 1*; : : : ; n* do
V ←shf1(*K;Mi*||*V*)
return *V*
algorithm shapad(*M*) // *jMj* < 264
d ← (447 ¡ *jMj*) mod 512
Let `be the 64-bit binary representation of *jMj*
y← *M*||1||0*d*||` // |*y*| is a multiple of 512
return *y*

algorithm shf1($K;B||V$) // jKj = 128, jBj = 512 and $jV\,j$ = 160
Parse B as $W0||W1||\,¢\,¢\,¢||W15$ where $jWij$ = 32 $(0 \cdot i \cdot 15)$
Parse V as $V0||V1||\,¢\,¢\,¢||V4$ where $jVij$ = 32 $(0 \cdot i \cdot 4)$
Parse K as $K0||K1||K2||K3$ where $jKij$ = 32 $(0 \cdot i \cdot 3)$
for t = 16 to 79 do
$Wt \leftarrow$ ROTL1(Wt_i3 ©Wt_i8 ©Wt_i14 ©Wt_i16)
$A \leftarrow V0$; $B \leftarrow V1$; $C \leftarrow V2$; $D \leftarrow V3$; $E \leftarrow V4$
for t = 0 to 19 do
$Lt \leftarrow K0$; Lt+20 $\leftarrow K1$; Lt+40 $\leftarrow K2$; Lt+60 $\leftarrow K3$
for t = 0 to 79 do
if $(0 \cdot t \cdot 19)$ then $f \leftarrow (B \wedge C)$ _ ((:B) ^ D)
if $(20 \cdot t \cdot 39$ OR $60 \cdot t \cdot 79)$ then $f \leftarrow B$ © C © D
if $(40 \cdot t \cdot 59)$ then $f \leftarrow (B \wedge C)$ _ $(B \wedge D)$ _ $(C \wedge D)$
$temp \leftarrow$ ROTL5(A) + f + E +Wt + Lt
$E \leftarrow D$; $D \leftarrow C$; $C \leftarrow$ ROTL30(B) ; $B \leftarrow A$; $A \leftarrow temp$
$V0 \leftarrow V0 + A$; $V1 \leftarrow V1 + B$; $V2 \leftarrow V2 + C$; $V3 \leftarrow V3 + D$; $V4 \leftarrow V4 + E$
$V \leftarrow V0||V1||V2||V3||V4$
return V

图 8.1: The SHA1 hash function and the underlying SHF1 family.

为了定义 SHA1 的抗碰撞攻击，立刻进行基础难题构造。进行无法实现的计算找到不同的串 M，M’，得到 SHA1 的碰撞。但是在什么条件下是不可实现的呢？有一个程序——是一个非常小而且简单的程序，仅仅有两个打印描述，输出是一个碰撞。输出一个碰撞不是一个计算难题，也不应该是一个计算难题。难度应该人类还不知道怎样描述这个问题。

看起来非常难，怎样进行一个数学描述一个怎样获得 SHA1 的碰撞描述，为了得到一个细致的数学描述，可以得到一个很自然的 HASH 函数的定义。也就是，一个单独的函数，这是一个函数簇。这在某种程度上是不太幸运的，因为这是具体例如 SHA1 的 HASH 函数，但是不改变是众所周知的。

$X \wedge Y$ bitwise AND of X and Y
$X \vee Y$ bitwise OR of X and Y
$X \oplus Y$ bitwise XOR of X and Y
:X bitwise complement of X
$X + Y$ integer sum modulo 2^{32} of X and Y
ROTL$_l$(X) circular left shift of bits of X by l positions (0 l 31)

图 8.2: Operations on 32-bit words used in sha1.

8.2 抗碰撞 HASH 函数

一个 HASH 函 H：K×D →R，这里 D 是 H 的定义域，R 是 H 的定义域，通常，如果 K∈ k 是一个特殊的密钥，那么 H_k ：D →R，定义对所有的 M ∈ D，通过 H_K（M）=H（K，

M），这里通过 k 的定义作为 H 的例子：

一个例子是 SHF1：$\{0，1\}^{128} \times\{0，1\}^{<2^{\wedge}64} \rightarrow \{0，1\}^{160}$ 在图 8.1 中进行定义，HASH 函数采用了一个 128 比特密钥和一个小于 2^{64} 分组，返回一个 160 比特的输出。SHA1 是一个这种函数簇的例子。也就是一个并联密钥：5A827999||6ED9EBA1||8F1BBCDC||CA62C1D6。

假设 H: K×D →R 是一个单向函数，在这一章使用一些定义，对任意的密钥 k 和 y ∈ R，假设：H_K^{-1} （y）=$\{x \in D : H_k(x) = y\}$

定义在 H_K 下的 y 的预先图，假设：

$$Image(H_K) = \{ H_k(x) : x \in D\}$$

定义为 H_K 的图。

对于函数 h : D →R 的碰撞，是一个对 $x_1,x_2 \in D$，使得（1）$H_K（x_1）= H_K（x_2）$并且（2）$x_1 \neq x_2$ 。HASH 函数的最基本特征是抗碰撞，用来度量攻击者从函数簇里发现碰撞的能力。有抗碰撞的不同的概念，在攻击者发现碰撞的空间变化。

为了介绍不同的概念，假设一个游戏，用一个整数 s∈ {0,1,2}和一个攻击者 A，组成一个先验密钥攻击阶段，然后是密钥选择阶段，最后是后密钥攻击阶段。攻击者试图发现 H_K 的碰撞，攻击者试图发现一个 H_K 的碰撞，这里，K 是从密钥选择阶段从 K 里的随机选择。由对 x_1 ，x_2 在 D 中的点，攻击者试图找到一个 H_K 的一个碰撞。攻击者在预先密钥分发阶段，在得到关于密钥消息前，要求 2-s 的说明。一旦攻击者说明这些点，攻击者给定密钥，选择剩余的点作为密钥的函数，在后密钥攻击阶段，如果对于 2=（2-s）+s 从 H_k 中选择得到就算是成功。

$Exp_H^{cr2\text{-}kk}(A)$ K ← K; (x_1 ， x_2) ← A(K) If ($H_K(x_1) = H_K(x_2)$ 并且 $x_1 \neq x_2$ 并且 x_1 ， x_2 ∈ D)那么返回 1，否则返回 0
$Exp_H^{cr1\text{-}kk}(A)$ （x1,st）← A()，K ← K; x2 ← A(K,st) K ← K; (x_1 ， x_2) ← A(K) If ($H_K(x_1) = H_K(x_2)$ 并且 $x_1 \neq x_2$ 并且 x_1 ， x_2 ∈ D)那么返回 1，否则返回 0
$Exp_H^{cr0}(A)$ （x_1 ， x_2）← A()，K ← K K ← K; (x_1 ， x_2) ← A(K) If ($H_K(x_1) = H_K(x_2)$ 并且 $x_1 \neq x_2$ 并且 x_1 ， x_2 ∈ D)那么返回 1，否则返回 0

图 8.4 在已知密钥条件下，经验定义三种抗碰撞分析的安全定义

图 8.3 概括了这种网络。参数 s 的三种选择给出了安全的三种概念，s 的值越大，攻击者的能力越强，因此安全的概念越严谨。图 8.4 提供了在上述框架下更细致的描述，用 st 表示攻击者希望维持跨越攻击阶段。在预先密钥攻击阶段输出这个信息，并且在后密钥攻击阶段开始时提供这个信息。

在 8.6 节的考虑的这个模型的变量里，攻击者在后密钥攻击阶段没有给定密钥 K，但是给定了一个 H_K 的随机预言机制。为了消除二意性，目前的定义是在已知密钥攻击下的抗碰撞攻击获取。在图 8.4 和定义 8.1 反馈的经验值是通过使用的“KK”，出去 CR0，已知符合

隐密钥攻击，因此仅仅讨论 cr0。

考虑 HASH 函数的三种概念在学术上的定义在图 8.5 中已有讨论。

定义 8.1：假设 H : D →R 是一种 HASH 函数，假设 A 是一种算法，假设：

$$Exp_H^{cr2\text{-}kk}(A) = Pr[Exp_H^{cr2\text{-}kk}(A)=1]$$

$$Exp_H^{cr1\text{-}kk}(A) = Pr[Exp_H^{cr1\text{-}kk}(A)=1]$$

$$Exp_H^{cr0}（A) = Pr[Exp_H^{cr0}A)=1]$$

为了度量一个攻击者使用的资源，使用普通的习俗。可以总是考虑硬件到一个最好的点 x_1 ， x_2 ，意味着这样的选择：

$$Pr[H_K(x_1) = H_K(x_2): K \leftarrow K]$$

$$= \max_{y1 \neq y2} [\, H_K(y_1) = H_K(y_2) : K \leftarrow K]$$

上述的值等于 Adv_H^{cr0}（A)是可获得的最大优势。

类型	学术定义
CR2-KK	自由碰撞，抗碰撞，相互碰撞
CR1-KK	普通单向[152]（aka，抗目标碰撞[27]）
CR0	普通，几乎普通

图 8.5 HASH 函数的类型，在框架条件下，在学术和相应条件

明显的，一个 CR2HASH 函数是 CR1，CR1 的 HASH 函数是 CR0。正式进行相应的描述，证明是繁琐的并且是可以忽略的。

推论 8.2：假设 H：K×D → R 是一个 HASH 函数，那么对于任意的攻击者 A_0 ，存在一个攻击者 A_1，与 A_0 有相同的运算时间。

$$Adv_H^{cr0}（A_0) \leq Adv_H^{cr1\text{-}kk}(A_1)$$

对于任意的攻击者 A_1，存在一个攻击者 A_2 ，与 A_1 有相同的运算时间

$$Adv_H^{cr1\text{-}kk}(A_1) \leq Adv_H^{cr2\text{-}kk}(A_2)$$

相信 SHF1 是 CR2，意味着没有实际的算法 A 对于每个 $Adv_H^{cr2\text{-}kk}$(A)是有一点大的。也就是，然而，简单一些，基于目前没有能力发现这样的算法。也许，可以浮现出一个。

对任意整数 n，得到 $SHF1^n : \{0,1\}^n \rightarrow \{0,1\}^{160}$，定义了 SHF1 的定义域到{0，1}^{n}的定义域。注意一个碰撞 $SHF1^n$ 对于一些固定的 n 比 SHF1 自身。

8.3 碰撞发现攻击

假设目标瞄准 CR2，随后看到的应用是非常重要的特征。考虑 CR2 上不同类型的碰撞，H：K× D →R，这里 D，R 是有限集。假设簇可以完成一些合理的压缩，也就是说：|D|≥2|R|，规范的例子保持为 $H=SHF1^n$ 对于 n≥161 并且 shf1，SHF1 的压缩函数。

抗碰撞是并不意味着发现一个碰撞是不可能的。分析一个单向的原因，有一个显然的发现碰撞的理由。假设 D 的元素，因此 D={D_1，D_2，，D_d}，这里 d=|D|。如下攻击者可以完成一个穷举搜索，发现碰撞攻击：

攻击者 A（K）

$X_1 \leftarrow D$; $y \leftarrow H_k(x_1)$
For i = 1,...,q, do
If ($H_k(D_i) = y$ 并且 $x_1 \neq D_i$ ，那么返回 x_1 ， D_i
返回失败
For i = 1,...,q, do
$X_i \leftarrow D$; $y \leftarrow H_k(x_i)$
如果（有 j<i，使得 $y_i = y_j$ 但是 $x_i \neq x_j$ //发现碰撞
返回 x_i ， x_j
返回失败 //没发现碰撞

图 8.6 对 HASH 函数的生日攻击

称 q 是实验次数，每次试验包括 H_K 的一次计算，因此，计算次数是对碰撞攻击的次数统计。为了成功发现碰撞，攻击者 H_K^{-1} 测试两次。如果 $|D| \geq 2|R|$。然而，仍然希望可以使用 q = |D|去找到一个碰撞，为了避免这种情况发生，D 通常是很大的。例如，对于 F= shf1，定义域的大小是 2^{672} ，足够大了。对于 $SHF1^n$ ，希望 n 可以足够小，但是 $n \geq 161$ 测量碰撞产生的概率，但是使用 2^{161} 进行碰撞的测试是不现实的。

现在这里有另外一些观点，随机选择一些点，希望在 H_K 的初始目标值下是映射相等的。称这种随机现象为碰撞发现攻击。可以如下过程完成：

攻击者 A（K）
$X_1 \leftarrow D$; $y \leftarrow H_k(x_1)$
For i = 1,...,q, do
$X_2 \leftarrow D$
如果（$H_k(x_1) = y$，但是 $x_i \neq x_j$ ） 返回 x_i ， x_j
返回失败

一个特殊的试验次数是在|R|里概率为 1 的碰撞，这样希望可以在|R|=q 次里发现碰撞。这对最初的|D|次试验碰撞发现有更好的。特别对 SHF1 的碰撞发现是在 2^{160}，而不是在 2^{672}。但是这仍然是远远无法实现的。结论是只要 HASH 的范围足够大，攻击就不足以产生威胁。

现在考虑另外一个策略，称为生日攻击。比上述的攻击方法更好。在图 8.6 中得到证明，从定义域中随机选择 q 点，把 H_k 分布在定义域上。如果找到两个不同的点产生相同的输出值，就证明在 H_k 上发现碰撞。问题是多大的 q 可以产生碰撞，回答起初看来是另人吃惊的，也就是仅仅 $q=O(\sqrt{|R|})$。

随后将进行判断，但首先注意碰撞。考虑 $SHA1^n$，$n \geq 161$。正如描述的对随机输入，进行 2^{160} 后回发现碰撞。生日攻击大概只需要 $\sqrt{2^{160}}=2^{80}$，这远远比 2^{160} 小。类似生日攻击在 shf1 里发现碰撞次数在 2^{80} 左右，但是随机发现碰撞必须有 2^{160} 次试验。

来看看生日攻击是否是如上所述。假设有 q 个球，把他们做上记号 1，...，q，也有 N 个盒子，这里 $N \geq q$。从球 1 开始，随机把球一个接一个的扔入盒子，对每个球是概率独立的。如果一个盒子装入两个球则攻击成功，对 C（N，q）表示关注，是一个概率碰撞。如同在附件中所述：

$C(N, q) \approx q^{2/}2N$ (8.1)

对于 $1 \le q \le \sqrt{2N}$。因此，当 $q \approx \sqrt{2N}$ 时，C（N，q）≈1。

生日攻击类似于相同生日的人在相同房间的概率。考虑到每个人的生日分布在 365 天的某一天内，这类似于把一个人放入 365 个房间。这里第 i 个房间代表第 i 天生日，预测说明在 $q \approx \sqrt{2 \cdot 365} \approx 27$。也就是大约 27 人就会有相同的生日发生。27 这个数字远远比想象的要小，这称为生日攻击范例。

来看看图 8.6 的生日攻击是怎样产生的，假设点为 R_1，...，R_N，这里，|R|=N，每个点定义为一个盒子。把 x_i 描述为一个球，y_i 是一个盒子，把球扔进盒子。这里，$y_i = H_k(x_i)$。因此球的碰撞可以细致的描述为两个值 x_i，x_j 产生相同的输出 H_k。对 q 的函数发生的概率感兴趣（可以忽略 x_i，x_j 相等的概率，只计算 $H_k(x_i)= H_k(x_j)$的概率。可以讨论的是既然 D 大于 R，x_i，x_j 相等的概率是小到可以忽略的）

但是不能直接应用生日攻击分析，因为每个球在每个盒子的概率是相等的。通常对攻击是真实的。假设 P（R_j）定义概率球是放入每个盒子的，这个条件对攻击是没有意义的。假设 P（R_j）定义概率 R_j ，也就是概率是 $H_k(x)= R_j$，从 D 中随机产生一个选择 x，那么：

$$P（Y）= | H_K^{-1}(R_j）|/|D|$$

为了对于 $P（R_1）=P（R_2）=...=P（R_N）$ 为真，要求应用生日分析，必须产生：

$$| H_K^{-1}(R_1）|= | H_K^{-1}(R_2）|=...= | H_K^{-1}(R_N）|$$

一个函数是 H_K 具备这种规律，对每个 K，如果 H_K 是规律的，H 称为是规律的。结论是如果 H 称为规律的，那么攻击成功的概率是 C（N，q），因此在上述条件下，如果 H 不规律，使攻击成功更快也提醒设计者在 HASH 设计时尽量接近规律。

总而言之，有一个 $2^{l/2}$ 或在任意 HASH 函数输出为 l 比特时，发现碰撞更好的时间。这导致设计者选择更大的 l，使得 l/2 被否定。在条件 SHF1 和 shf1 里，选择 l=160 是因为 2^{80} 是被认为不可执行的试验次数。这些函数不能认为是不能经受生日攻击的。

更进一步的确保，对生日攻击是不脆弱的，并且可以保证抗碰撞攻击。考虑簇 $H：K\times\{0，1\}^{161} \rightarrow \{0,1\}^{160}$ 定义如下。对任意 K 和任意 x，函数 $H_k(x)$返回 x 的前 160 比特，使用 2^{80} 生日攻击是不可行的，但还是容易发现碰撞。也就是，在输入 K 时，攻击者可以仅仅选择 160 比特 y 输出 y_0，y_1，这告诉设计者对抗碰撞攻击仅有足够长的输出是不够的。这里没有更短的捷径找到碰撞，意味着在快速寻找碰撞的过程中发现结构的弱点。

相信在这种观点下，shf1 是好的设计。还没有人声称可以有少于 2^{80} 条件下的试验次数。甚至在某种程度上，甚至在某种程度上，一些更强的攻击，例如在 2^{65} 的攻击条件下是可以找到碰撞的，这对攻击者是很大的试验次数，因此，可以认为 shf1 是抗碰撞攻击的。

如果相信 shf1 是抗碰撞攻击的，定理 8.8 告诉设计者 SHF1，与 SHFn 是相同安全程度的，对任意 n 认为是抗碰撞攻击的。

8.4 抗碰撞的单向 HASH 函数

直观的，一个函数簇 H 是单向的，如果给定 H_K 计算 $y= H_K (x)$，x 从定义域中随便选择，在 H_K 下找到 y 的前向图，是计算不可能的。既然定义隐藏了密钥的版本，指出在下述已知密钥条件下的版本。

定义 8.3：假设 $H ： K \times D \rightarrow D$ 是一个函数簇，假设 A 是一个算法，考虑如下的过程：

$Exp_H^{ow\text{-}kk}(A)$

$K \leftarrow K; x \leftarrow D; y \leftarrow H_K (x); x' \leftarrow A(K,y)$

如果（$H_K(x') = y$，并且 $x' \in D$），那么返回 1，否则返回 0

假设：

$$Adv_H^{ow\text{-}kk}(A) = Pr[Exp_H^{ow\text{-}kk}(A)=1]$$

现在的问题是是否抗碰撞就是单向的，容易看到，然而，在没有抗碰撞的附加假设的条件下，答案是否定的。例如，假设 H 是一个函数簇，在不同情况下的恒等式。那么 H 是具有高碰撞次数的，但不是单向的。

希望真正的单向函数，意味着对数据进行非平凡的压缩是单向的。为真，但希望细致的量化（范围大于定义域）是单向的，为了明白这点，也许自然的会对抗碰撞的攻击进行讨论。

假设有一个攻击者 A 有一个攻击单向函数明显的优势，希望试着使用 A 通过如下的策略找到一个碰撞。在密钥预先分发过程中，从 D 中随机选择返回一个点 x_1，在后密钥处理过程中，受到了密钥 K，计算 $y= H_K(x_1)$，并且给定 K，y 到 A，随后返回一些 x_2，但是，如果成功，$H_K(x_2) = y$，因此，$H_K(x_2) = H_K(x_1)$，这样就有了一个碰撞。

获得 $x_1 \neq x_2$，有一个碰撞，在如下数量下是可以发生的：

$PreIm_H(1) = Pr[|H_K^{-1}(y)|=1：K \leftarrow K；x \leftarrow D；y \leftarrow H_K(x)]$

这是可能的，y 的预处理图概率是 1，使用 y 进行改写。如下的推论是抗碰撞攻击 H 是单向的，当且仅当 $PreIm_H(1)$是足够小的。

推论 8.4：假设 $H : K \times D \rightarrow R$ 是一个 HASH 函数。那么对于任意 A，存在一个 B，使得：

$$Adv_H^{ow\text{-}kk}(A) \leq 2\, Adv_H^{cow\text{-}cr1}(B) + PreIm_H(1)$$

更进一步 B 的运算时间是 A 加上时间作为一个定义域的点，计算一次 H。

这是关于抗碰撞的 CR1 型的结果。然而推论 8.2 暗示对 CR2 也是相同的。

关于上述推论的一个更宽的结果是抗碰撞攻击意味着单向函数，只要 HASH 的定义域是显然超过边界的。量化为：

$$Adv_H^{ow\text{-}kk}(A) \leq 2\, Adv_H^{cr1\text{-}kk}(B) + |R|/|D|$$

更进一步的运算时间是，A 加上一个定义点的抽象时间，计算 H 一次。

推论 8.5 定义：对于任意密钥，H_K 范围内的点的数量包括前向图最多为|R|，这意味着：

$$PreIm_H(1) \leq |R|/|D|$$

推论与 8.4 相同。

推论 8.5 说明，如果 H 是抗碰撞的，完成足够的压缩，|R|远远小于|D|，那么仍然是单向的。为什么？假设 A 是攻击 H 的单向函数的一个实际的攻击者，那么 B 也是可以实际攻击的，既然 H 是抗碰撞的，知道 $Adv_H^{cr1\text{-}kk}(B)$是低的。等式（8.2），那么只要|R|/|D|很小，$Adv_H^{ow\text{-}kk}(A)$是低的，意味着 H 是单向函数。

作为一个例子，如果 H 是压缩函数 shf1，当$|R|=2^{160}$，$|D|=2^{672}$，$|R|/|D|=2^{512}$，是非常小的。相信 shf1 是抗碰撞攻击，上述因此也是单向函数。有一些自然的单向函数，对推论 8.5 是不成立的。考虑一个单向函数 H，要求是二到一，那么定义域是值域的两倍。因此等式右边的 8.5 推论的结果是 1。意味着边界是空的，然而这样一个函数是特殊的，考虑如下的推论。

推论 8.6：假设 d>r 暗示着 $PreIm_H(1) =0$，再应用推论 8.4。

现在回过来证明 8.4。

推论 8.4 的证明：这里 B 是怎样工作的：

密钥预先分发过程　　后密钥分发过程

攻击者 B（）　　攻击者 B（K，st）

$x_1 \leftarrow D$; $st \leftarrow x_1$　　从 st 找回 x_1

返回（x_1; st）　　$y \leftarrow H_K(x_1)$; $x_2 \leftarrow B(K, y)$

返回 x_2

假设 Pr[.]定义时间“.”的概率在经验值 $Adv_H^{cr1\text{-}kk}(B)$，对任意 $K \in k$，假设

$$S_K = \{x \in D: |H_K^{-1}(H_K(x))| = 1\}$$

$$Adv_H^{cr1\text{-}kk}(B) \quad (8.2)$$

$$= Pr[H_K(x_2) = y \wedge x_1 \neq x_2] \quad (8.3)$$

$$\geq Pr[H_K(x_2) = y \wedge x_1 \neq x_2 \wedge x_1 \notin S_K] \quad (8.4)$$

$$= Pr[x_1 \neq x_2 \mid H_K(x_2) = y \wedge x_1 \notin S_K]. Pr[H_K(x_2) = y \wedge x_1 \notin S_K] \quad (8.5)$$

$$\geq 1/2 [H_K(x_2) = y \wedge x_1 \notin S_K] \quad (8.6)$$

$$\geq 1/2 [H_K(x_2) = y] - Pr[x_1 \in S_K] \quad (8.7)$$

$$= 1/2 (Adv_H^{ow\text{-}kk}(A) - PreIm_H(1) \quad (8.8)$$

重新安排等式条件（8.2），调整上述的步骤，等式（8.3）由 $Adv_H^{cr1\text{-}kk}(B)$和 B 进行定义。等式（8.4）为真，因为 $Pr[E] \geq Pr[E \wedge F]$对任意事件 E，F。等式 8.5 使用标准等式 $Pr[E \wedge F]=Pr[E|F]Pr[F]$。等式（8.6）可以如下调整，攻击者 A 没有关于 x_1 的信息。在 $H_K^{-1}(y)$ 是一个随机点，然而如果 $x_1 \notin S_K$，那么$|H_K^{-1}(y)| \geq 2$。因此，$x_1 \neq x_2$ 的概率是 1/2。等式（8.7）是采用另外标准概率不等式，也就是 $Pr[E \wedge \overline{F}] \geq Pr[E] - Pr[F]$，等式 8.8 是是使用包括数量的定义。

8.5 MD 变换

可以看到上述 SHF1 使用压缩函数 shf1 进行压缩应用。随后，在任何密钥下，压缩 672 比特到 160 比特。SHF1 是使用 shf1 进行输入为 512 比特。

迭代方式被认真的选取，当 SHF1 保证是抗碰撞的，那么 shf1 保证是抗攻击的。换言之，设计抗碰撞攻击单向函数是更难的任务，采用长的可变的输入减低了设计难度，设计一个抗碰撞攻击压缩函数仅仅采取一些固定长度的输入。

有明显的利益，需要寻找 SHF1 的攻击。为了使之成规模，并且确定是抗碰撞攻击的，仅仅需要对 shf1 进行具体运做，然后是抗碰撞的。

这是一个重要的 HASH 设计原则称为 MD 范例[145，67]。这个范式给出了怎样由一个压缩函数设计出单向函数，前者的抗碰撞攻击导致后者抗碰撞攻击。现在对这个范式进行更进一步的研究。

假设 b 是一个整参数，称为分组长度，并且 v 是另外一个整参数，称为链式变化长度。假设 $h : K \times \{0,1\}^{b+v} \rightarrow \{0,1\}^v$，是函数簇，称为压缩函数，并且假设是抗碰撞的。

假设 B 定义为所有串的集，长度是 b 比特正数次数，D 是$\{0,1\}^{<2b}$

定义 8.7：一个函数底码，$D \rightarrow B$ 称为一个 MD 底码适应函数，如果对所有 M，M_1，$M_2 \in D$ 有如下的特征。

（1） M 是 pad(M)的前缀

（2） 如果$|M_1|=|M_2|$那么$|pad\ M_1|= |pad\ M_2|$

（3） 如果 $M_1 \neq M_2$，那么 pad（M_1）的最后一个分组是与 pad（M_2）的最后一个分组不同

上述一个分组由 b 比特组成，回忆在 B 中的输出底码，意味着是 b 比特分组串。定义的条件（3）说明如果两个消息不同，那么当使用底码，在最后的分组中是不同的串。

一个 MD 底码适应函数的例子是 shapad，然而，这里有其他的例子。

现在假设 IV 是 v 比特值，称为初始向量，从 h 和 pad 中构建一个簇 H：$K\times D \rightarrow \{0,1\}^v$ 在图 8.7 中已描述了。注意 SHF1 是这样一个簇，从 h=shf1 和 pad=shapad，主要的过程是这个方法如下：

$H(K;M)$
$y \leftarrow \text{pad}(M)$
Parse y as $M_1 ||M_2 || ... ||M_n$ where $|M_i| = b$ $(1 \cdot i \cdot n)$
$V \leftarrow$ IV
for $i = 1; : : : ; n$ do
$V \leftarrow h(K;M_i ||V)$
Return V

Adversary $A_h(K)$
Run $A_H(K)$ to get its output $(x_1; x_2)$
$y_1 \leftarrow \text{pad}(x_1)$; $y_2 \leftarrow \text{pad}(x_2)$
Parse y_1 as $M_{1;1} ||M_{1;2} ||...||M_{1;n[1]}$ where $|M_{1;i}| = b$ $(1 \cdot i \cdot n[1])$
Parse y_2 as $M_{2;1} ||M_{2;2} ||...||M_{2;n[2]}$ where $|M_{2;i}| = b$ $(1 \cdot i \cdot n[2])$
$V_{1;0} \leftarrow$ IV ; $V_{2;0} \leftarrow$ IV
for $i = 1; : : : ; n[1]$ do $V1;i \leftarrow h(K;M_{1;i} ||V_{1;i_i1})$
for $i = 1; : : : ; n[2]$ do $V2;i \leftarrow h(K;M_{2;i} ||V_{2;i_i1})$
if $(V_{1;n[1]} \neq V_{2;n[2]}$ OR $x_1 = x_2)$ return FAIL
if $|x_1| \neq |x_2|$ then return $(M_{1;n[1]} ||V_{1;n[1]i}1;M_{2;n[2]} ||V_{2;n[2]_i1})$
$n \leftarrow n[1]$ // $n = n[1] = n[2]$ since $|x_1| = |x_2|$
for $i = n$ downto 1 do
if $M_{1;i} ||V_{1;i_i}1 \neq M_{2;i} ||V_{2;i_i}1$ then return $(M_{1;i} ||V_{1;i_i1};M_{2;i} ||V_{2;i_i1})$

图 8.7：Hash 函数 H 根据 MD 范式从压缩函数 F 构造，攻击者 A_h 在定理 8.8 中证明

定理 8.8：假设 h: $K\times\{0,1\}^{b+v} \rightarrow \{0,1\}^v$ 是一个函数簇，假设从 h 按照上述构造 $K\times D \rightarrow \{0,1\}^v$ 。假设给定一个攻击者 A_H 试图在 H 中找到碰撞，那么可以构造一个攻击者 A_h 并且试图在 h 中找到碰撞：

$$\text{Adv}_H^{\text{cr2-kk}}(A_H) \leq \text{Adv}_h^{\text{cr2-kk}}(A_h) \qquad (8.9)$$

更进一步，A_h 的运算时间是 A_H 加上执行时间 $(|\text{pad}(x_1)| + |\text{pad}(x_2)|)=b$，h 的计算值，这里（$x_1; x_2$）是 A_H 的碰撞输出。

这个定理是讲如果 h 是抗碰撞的，那么 H 也是抗碰撞的。为什么？假设 A_H 是一个实际的攻击者攻击 H，那么 A_h 也是可实现的，因为运算时间是 A_H 是加上另外计算 h 的时间，但是既然 h 是抗碰撞的，可以知道 $\text{Adv}_h^{\text{cr2-kk}}(A_h)$是低的。等式(8.9)告诉公众 $\text{Adv}_H^{\text{cr2-kk}}(A_H)$ 是低的，意味着 H 同样抗碰撞。

定理 8.8 的证明：攻击者 A_h 是输入一个密钥 $K \in K$，在图 8.7 中可以进行描述。A_H 是在 D 里得到消息对（x_1, x_2），可以知道 x_1, x_2 是 H_k 的碰撞，那么 A_h 是返回一个 h_k 的一个碰撞。

攻击者 A_h 计算 $V_{1,\,n[1]} = H_k(x_1)$，并且 $V_{2,\,n[2]} = H_k(x_2)$，如果 x_1, x_2 是 H_k 的碰撞，那么知道 $V_{1,\,n[1]} = V_{2,\,n[2]}$ 。如果假设成立，观察产生这些输出的 h_k 的应用，如果这些输入是不

同的，则组成不同的碰撞。

在这个问题里的输入是：$M_{1,\,n[1]}\|V_{1,\,n[1]-1}$ ，$M_{2,\,n[2]}\|V_{2,\,n[2]-1}$ 。现在考虑两个条件，第一个条件 x_1, x_2 是两个不同长度的输入，由定义 8.7 的条件（3）可知：$M_{1,\,n[1]} \neq M_{2,\,n[2]}$，这意味着 $M_{1,\,n[1]}\|V_{1,\,n[1]-1} \neq M_{2,\,n[2]}\|V_{2,\,n[2]-1}$ ，因此这两个点构成一个 h_k 的碰撞，可以避免进行 H_k 的输入。

第二个条件 x_1，x_2 是两个相同的输入，由定义 8.7 的条件（2）可知道：y_1,y_2 是同样有相同的长度，可以知道这个长度是 b 的正级数，既然底码的范围是 B 的定义域。假设 n 是 b 比特分组，组成 y_1,y_2 。

$Exp_H^{cr2\text{-}hk}(A)$

K← K , Run $A^{HK(.)}$ ()

如果存在 x_1, x_2 ，使得：

$x_1 \neq x_2$ ，并且 $x_1, x_2 \in D$

随机询问机制有 A 产生的 x_1, x_2

返回的随机询问相同的回答

那么返回 1，否则返回 0

$Exp_H^{cr1\text{-}hk}(A)$

（x_1,st）← A()，K← K , Run $A^{HK(.)}$ (st)

如果存在 x_2 ，使得：

$x_1 \neq x_2$ ，并且 $x_1, x_2 \in D$

随机询问机制有 A 产生的 x_1, x_2

返回的随机询问相同的回答

那么返回 1，否则返回 0

图 8.8：定义安全概念的经验值是两种在隐密钥攻击的抗碰撞单向函数。

假设 V_n 定义 $V_{1,n}$，假设与 $V_{2,n}$ 相同，在 h_K 产生 V_n ，比较输入 $M_{1,\,n[1]}\|V_{1,\,n[1]-1}$ ，$M_{2,\,n[2]}\|V_{2,\,n[2]-1}$ ，如果不同，构成了一个返回 h_K 的碰撞。如果相同，那么 $V_{1,\,n[1]-1} = V_{2,\,n[2]-1}$。用 V_{n-1} 定义这个值，在 h_K 产生 V_{n-1} ，比较输入 $M_{1,\,n[1]}\|V_{1,\,n[1]-1}$ ，$M_{2,\,n[2]}\|V_{2,\,n[2]-1}$ ，讨论是自己重复的，如果这些输入是不同的，对 h_K 产生一个碰撞，那么可以停止，再返回一次。

是否可以继续返回但是没有发现碰撞？这是不可能的，因为 $y_1 \neq y_2$ ，为什么继续可以为真？既然知道 $x_1 \neq x_2$ ，但是定义 8.7 的条件（1）是 x_1 是 y_1 的前缀，x_2 是 y_2 的前缀。因此，$y_1 \neq y_2$ 。

讨论过对任意输入 K，如果 A_H 在 H_K 中发现一个碰撞，那么攻击者 A_h 在 h_K 中发现一个碰撞，这个就是（8.9）的结论。现在判断 A_h 的运算时间，主要的 A_h 运算时间是 A_H 的运算时间。另外，h 的运算量等于加上运算数量为在 y_1 里的分组数加上在 y_2 里的分组数，有一些更小的运算是可以忽略的。

8.6 在隐密钥条件下的抗碰撞攻击

对隐密钥攻击，攻击者是不能在后向密钥阶段得到密钥 K，但是得到 H_K（ ）的随机预言机制。有三个可能的安全概念，在图 8.3 中得到一些印证。在后密钥攻击阶段，A 不能给出 K，但是同样可以得到 H_K（ ）的随机预言机制。CR0 的概念是在后密钥攻击阶段对已知密钥攻击的概念是一致的。一个 CR0 的攻击者输出这两个点，因此得到两个概念。先验定义了图 8.8 中的两个概念。

定义 8.9：假设 H ： K ×D → R，有一个 HASH 值，假设 A 是一个函数，可以使得：

$$Adv_H^{cr2\text{-}hk}(A) = Pr[Exp_H^{cr2\text{-}hk}(A)=1]$$

$$Adv_H^{cr1\text{-}hk}(A) = Pr[Exp_H^{cr1\text{-}hk}(A)=1]$$

8.7 难题

难题 8.10：HASH 函数是可以用分组密码算法构造的，通常这种情况是不太好的，假设 E：K × $\{0,1\}^n$ →$\{0,1\}^n$ 是一个分组密码算法考虑结构 H：K ×（$\{0,1\}^n$ ）$^+$→$\{0,1\}^n$ 使用 CBC 进行构造，假设 M_1，M_2，....，M_m 的 CBC 结果是 Y_m ，这里 $Y_0=0^n$ ，并且 $Y_i= E_K(H_{i\text{-}1}\oplus M_i)$ 对于 i≥1。选择 K 作为一个公共常数，可以看到，这个函数是不能抗碰撞攻击的。（无论分组密码算法设计得如何的好）

难题 8.11：假设 H：K ×$\{0,1\}^a$→$\{0,1\}^n$ 是一个ε-AU 单向函数簇，从 H 构造一个ε-AU 单向函数簇 H’：K ×$\{0,1\}^{2a}$→$\{0,1\}^{2n}$ 。

难题 8.12：假设 H：K ×$\{0,1\}^a$→$\{0,1\}^n$ 是一个ε-AU 单向函数簇，从 H 构造一个ε^2-AU 单向函数簇 H’：K^2×$\{0,1\}^a$→$\{0,1\}^{2n}$ 。

第九章 消息认证

一个消息认证体制可以让多方拥有者共享秘密密钥，保证消息完整性。这是私钥密码学的第二个目标。

9.1 简介

9.1.1 难题

假设收到一个从声称合法用户来源的消息，叫做 S。这里 S 也许就是许多不同整体其中之一，例如一个人，一个公司，或一个网址。也许已知这就是 S，有几个原因声称发送。例如，S 的身份可以通过通信确认。这里的确认是一个公众确认，已知属于 S：例如，如果 S 是一个人或公司，特别的仅是一个人或公司的名称，如果是一个网络地址就是网络地址本身。或者也许从前后关系，可以代替从通信中所获得的所有 S 的信息。

在所有这样的设置中，安全要求接收者可以确认数据确实来源于发送者。这就需要进行访问控制，这样服务和信息就可以提供给确定的用户。冒的风险是一个攻击者可以模仿 S，发送一个 S 的身份认证消息，使接收者相信通信来自 S。

一个在线存储代理 S 是回复一个引言要求发送这种存储的值。要求引用不同接受错误信息要求作出有损行动。这适用于从数据库得到的任何信息的价值在于作为担保的确认。或者认为 S 是需要发送仅仅两个数据，称为买或卖，或者解雇或留用。这也许把电文编成单比特，如果一个攻击者攻击这个比特，就采取了错误的行动。或者考虑电子银行。S 发送到银行一个消息，要求从的帐号转 200$到另一个帐户。A 扮演攻击者的角色，把转帐总数改为$2000。

事实上，通过一个网络的数据传输认证可以使数据的安全性更强于私密性，当数据用于网络和商用。

这个发送数据的能力在于声称从一个消息源并不需要攻击者部分的活动的攻击。也就是说攻击者必须有手段修改传输通信消息或插入一个。这些能力依靠与安装，也许很难介绍数据到一个专门的电话线，但不是在例如 INTERNET 的互联网上。这应该是可取的，假设对手确实具备这种能力。

证明难题是同加密难题非常不同的，并不担心数据的保密，假设数据是清楚的。令人担心的是攻击者修改它。

9.1.2 加密并不提供数据完整性

知道怎样为加密数据提供私密性。通常建议用加密提供如下的数据完整性，如下。固定一个加密体制 SE=（K,E,D），假设 S 方和 B 方对这个体制提供一个密钥 K。当 S 想发送一个消息 M 到 B，进行加密，传输一个密文 C，通过 C←E_k(M)，B 进行解密，恢复 D_k（C）。论点是用如下手段提供数据完整性。假设传输 S，对于上述的例子，一个消息 M 到 B 银行，要求从 S 银行传输$200 到 A 银行，A 想把$200 变为$2000。如果 M 明文发送，A 可以容易的修改。如果 M 进行加密，对密文 C 进行发送，A 怎样修改 C 发送到 B，恢复修改过的消息 M’。因为密钥 K 未知，因此不能加密修改消息 M’，消息的私密性更加困难。

这个争论是不合理的，为了看见这个缺点，首先看到反例，然后公开。考虑到，随机的 CTR 配置，使用一些分组 F，假设 RC6 是一个伪随机函数。证明在对称加密的章节中，对简单的说法，上述消息仅仅是 128 比特分组。对于参与的各方包括计数信息和一个美元数量的范围。更具体的，128 比特分组的最后 16 比特包括美元数量的分组作为 16 比特二进制数字。（因此数量最多是$65535）。因此，M 的最后 16 比特 0000000011001000，代表整数 200 的二进制表示。假设 A 已知，在这个电子帐单的美元数量是$200，这个信息是不保密的。现在回忆在 CTR 加密模式下的随机加密体制，用 S 的密文传送是构成 C= 〈r>y，并且设置一个 y'=y$\oplus 0^{112}$00000011100001000，设置成为 C' = 〈r>y'和前向 C'到 B。B 将解密这个，因此恢复消息 F_k（<r+1>$\oplus$ y'。用 M'表示值为：

M'= F_K（<r+1>）$\oplus$ y'

= F_k(<r+1>)$\oplus$ y$\oplus 0^{112}$00000011100001000

= M $\oplus 0^{112}$00000011100001000

= M_{prefix}0000011111000000

这里 M_{prefix} 是原始消息的前 112 比特。注意 M'的最后 16 比特是整数 2000 的二进制表示，这里 M 的前 112 比特与 M'的相同。这样结果结束了，银行 B 将错误导致执行交易。S 期望美元数量从 200 变到 2000。

对于这种记录有多种可能的反馈，有些是合理的，有些是不合理的。来仔细看看：从这点应该阻止的是加密并不提供数据完整性，这显然是与事无补的。事实上，对任何原因，任何数据加密不需要从提供接收者来阻止一个攻击者。首先，数据，或者一些数据，也许完全不是私密的，例如，A 已知上述关于 M 的信息：作为金钱的接受方，A 可以假设已知数量是$200，一个总的概率是预先知道的。然而甚至当数据预先已知，一个攻击者恢复的一些数据是不正确的，甚至不是预先知道的。例如使用随机化的 CTR 体制，一个攻击者可以有效的在消息 M 中跳动 1 比特。甚至未知原始比特值，对于相反的值，可以引起损坏。另一种可能是对于攻击者简单的传输 C 串。在许多加密体制里，包括 CTR 和 CBC 加密，C 将解密一些串得到 M。攻击者也许没有任何关于 M 的思想，将进行错误的观察，接收者收到 M，发送 S 到与实际相反的情况。

现在，在计数模式下有另外一种可能的选择，计数模式加密是不完善的，既然允许上述的攻击，因此这种模式不建议使用。使用 CBC 模式代替 CTR 模式，从得到密文比特不能还原明文比特。

这是一个不合理的 CTR 运算模式例子。但是这种模式不仅仅是用于声音，也用于文字。为什么这是不合理的呢？因为点子不仅仅是 CTR 模式的一种特殊攻击形式，但是认可对于目标的不同点。有简单的、没有理由的、期望使用加密提供完整性，加密并不提供完整性解决方案。解决这个问题的途径是首先盯住问题的细节特征，然后寻求解决。但是有许多存在的体制，并且加入著作中，这里加密和认证被关注，假定加密可以提供完整性。

CBC 加密也可以从完整性的角度观察，对于一个不好的加密体制也会在某些地方导致需求。对于一个加密体制的断层作用并没有提供例如反复认证，因为这样并没有作用。当然没有理由要求工具完成没有设计的任务。

有时暗示可以使用冗余加密来提供数据完整性。也就是说，发送者 S 的已知的底码，固定串，例如 128 比特 0，如果不是，接收者拒绝传送不可靠的消息。另外输出是其余的串作为实际数据，在通常意义下，这也许算做失败。例如，容易看到使用 CTR 模式加密，一个仅仅类似上述的加密。也可以攻击 CBC 模式加密。

好的加密设计是面向目标的。一个人必须明白形式化目标，仅仅当基于一个设计并且评价潜在的解答方案。因此，下一步将得到消息认证体制的定义和安全。

9.2 消息认证体制

在私钥体制前提下，简单使用提供数据的完整性是一个消息认证体制。这是一个用三个体制说明的算法：密钥产生算法 K，标签算法 T，初始化算法 V。发送者和接收者假设通过密钥产生算法拥有使用共用的密钥 k，这个算法攻击者未知。当发送者希望以认证的方式发送 M 到 B，为 M 计算一个标签 σ，在发送者和接收者之间的秘密密钥 k。在标签算法的特殊要求下，也就是说，S 传输 M，σ 到 B，（注意清晰的发送消息，同样通知不超过对标签 σ 的长度）。对于接收一个传送 M’，σ’，声称是 S 发送的，接收者 B 使用证实流程说明，依赖于消息，标签和共享密钥。换言之，计算 V_k（M’，σ’），值是 1 比特。可以读做数据是安全的，B 把数据当做从 S 发送的消息，也就是数据是不可靠的。

作为以前的讨论，有许多种办法接收者知道从 S 传输声明。例如，当已知与 S 的作用，S 的身份也许是伴随着发送或通信。因此，并不明确的忙于这个模型，宁可置于能力之外。一个可行的体制，一些安全特征。但这不是涉及到的，首先要固定制定一个特殊的体制。因此，需要达到哪几种目标的安全。假设总结上述的定义。

定义 8.1、 一个消息认证体制 MA=（K，T，V）有三个算法组成，如下：

• 密钥产生算法 K 是一个随机算法，返回一个密钥 k，记做：K←k

• 标签算法 T 是一个概率随机数算法，采用密钥 K 和一个消息 M，返回一个标签 σ，记做：$\sigma \leftarrow T_k(M)$

• 确认算法 V 是一个确定的算法，获得密钥 k，一个消息 M，一个 M 的候选标签 σ，返回 1 比特。记做：

联合的体制是一个明文消息空间，M 是可以被提取的，对于所有的 M∈明文空间，要求 $V_k(M,T_k(M))=1$。这个定义的最后一部分为可以恰当的产生标签，通过认证测试。这样简单的确保可信数据被接收者接受。标签数据也许会被随机化，意味着使用投币确定内在随机变化。因此，有许多的标签是仅仅与一个消息绑定，算法也许也是一种陈述，例如，使用一种发送者提供的计数。在那种情况下，正确的标签算法检查传输标签与正确的比较。也就是说，验证算法简单如下：

算法 V_k（M, σ)

σ’← T_k(M)

如果 σ=σ’那么返回 1 否则返回 0

因此当标签算法是确定的，确定算法不需要明确的规格，可以明确，这是上述的一种。

例8.2、假设 F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ 是一个分组密码，与基础的 F 簇是 CBC，MAC 确定的消息认证体制。消息的标签是密文的最后一个分组，使用零向量运行消息的 CBC 模式。对于更多的细节，体制是 $CBC\text{-}MAC^F$=（K，T，V）的算法定义如下，假设 K 是算法，选择 k 比特密钥 K，进行 k 的投币周期，返回成果。消息 M 的输入算法必须是 l 的倍数。

算法 T_k(M)
把 M 分割成 l 比特的分组，
$M=x_1 \ldots x_n$
$Y_0 \leftarrow 0^l$
For i=1,…,n do
$Y_i \leftarrow F_k(y_{i-1} \oplus x_i)$
Return y_n

算法 V_k（M，σ）
把 M 分割成 l 比特分组，
$M=x_1 \ldots x_n$
$Y_0 \leftarrow 0^l$
For i=1,…,n do
$Y_i \leftarrow F_k(y_{i-1} \oplus x_i)$
If y_n=σ 那么 return 1 else return 0

既然标签算法是确定的，确认算法简单的检查，是否 σ 是正确的标签，同上述的讨论相同。消息空间的选择对于 CBC，MAC 的安全是重要的，随后可以看到。如果消息的变化长度是允许的，那么体制是安全的。如果消息的长度是限制在一些单的预先说明和固定值，体制是安全的。

9.3 安全的体制

首先试着构建关于消息认证体制的什么特征，应该称为是安全的，然后获得一个正规的安全定义。

9.3.1 关于安全的几点

寻找需要使用消息认证达到的目标能够探测攻击者修改传输信号的尝试。有担心攻击者可以产生接收者接收的消息，当 S 不能发送时，作为合法发送 S。那就是说，A 可以产生 M，σ，使得 V_k（M，σ）=1，但是 M 并不从发送者 S 产生，这叫做伪造。也许，第一个问题在于一个人也许要求是否应该试着度量攻击者伪造的值。例如询问消息的内容是什么？例如消息希望有固定的格式，伪造仅仅是一个随机数。这个是否真的应该被认为是伪造？回答是肯定的。在之前看到了更普通的版本。这是没有文化的假设按照格式化或说明进行。好的协议设计意味着安全是有保障的，无论是什么应用。根据观点，攻击成功，如果产生出 M，σ，因此发送者不再鉴别 M，但是 V_k（M，σ）=1。

在一些安全问题的讨论中，攻击者的目标是从恢复共享秘密密钥 K。当然，如果可以做件事，可能是一个灾难，既然可以伪造。重要的是了解，不用恢复密钥而进行伪造。因此，一个安全体制有更强的安全保证。

在进行伪造时，必须考虑几种攻击，增强安全。最简单的安全是发送者还没发送任何的消息，攻击者必须简单的调试一个 M，σ 对，通过确定的测试，也就是因此 V_k（M，σ）=1。这叫做无消息攻击。

然而攻击者有获得传输内容的能力，也许试着进行基于信息的伪造。假设发送者发送一些由消息和标签组合传输内容 M，σ，接收者将首先收到这个信息。立刻，可以想到一个简单的攻击，攻击者仅仅复制这个攻击。也就是说，存储 M，σ，然后在随后重新传输这个消息，称为重放攻击。如果接收者首次接收，可以容易的重放这个攻击。这是一个典型的伪造？在现实生活中，大概可以这样考虑。第一个消息是传输$1000 到 B 帐户，B 突然看到使自己富足的办法，回复这个消息，并且保持帐户增加的平衡。

重要的是防止再次的重放攻击。但是将不再尝试这个。重放攻击不是一个有效的攻击。更有效的，攻击者必须是不是发送者发送的信息 M。随后将看到，可以对加了时间戳和消息认证机制的计数器进行抗重放攻击的安全加强。在这一点，在一个明显的问题前分开这部分消息要求在协议设计时使用模块化的设计。也就是，把问题分割为逻辑模块，然后一个一个的解决。从现在开始，不把重放攻击当作有效的攻击。

因此，如果攻击者想成功，必须在某种程度上使用有效的传输 M，σ，来调和一个对子 M’，σ’，因此，M≠M’，但是 V_k（M，σ）=1。如果不能做到这点，应该说这是成功的。因此，有一个非常自由的攻击者成功的概念。因此当说一个体制是安全的，是在很强的一个体制之下的概念。

允许攻击者看一个消息的例子，当然，可以看到比一个更多的，伪造还是困难的。希望攻击者的能力是看到更多的合理认证数据的例子。因此通常条件下，希望安全的概念是定量

的，同攻击者成功的概率是看到的一个合法的数字 q 函数对。

在许多设置里，攻击者影响合法消息标签的选择，在更坏的条件下，考虑攻击者自己选择这些消息，这叫做选择明文攻击。简单浏览一下选择明文攻击可以看做是不切实际的攻击。有两个论点反对这个观点。首先看到这个例子说明这样的攻击是非常现实的，第二，回忆常规法则。希望这个配置在任意的用法下是安全的，这要求安全在最坏条件下，这是在警告、允许攻击者具备更多的办法。然而最后，将可以设计体制，遇到这样严格的安全概念，只能得到过程的概念。

选择消息攻击的实例是一个设置 S 向 B 发送从 C 收到的数据，并且鉴别在 S 和 B 之间共享的一个密钥 K，在这个过程里，如果 C 想扮演一个攻击者的角色，C 可以选择希望的数据，然后看到 S 到 B 相应的标签，其余假设也是可能的。

简要的，希望一个安全的概念是得到如下的结论。允许一个攻击者跟上一个选择消息攻击的发送方，获得攻击者选择的消息正确的标签。那么，攻击者试图做一个伪造，并且声称是成功的，如果伪造是有效的（意味着被接受者接收）并且消息是不被发送者认证明的。

9.3.2 一个安全的概念

假设 MA =（K，T，V）是认证体制一个任意的消息。目标是对一个消息形式化，对这个体制的安装在发送者的选择明文攻击，那么，一个伪造试图直接发送到接收这。在形式化的角度下，开始筛选模型表面的密钥。事实上，作为一个整体明确的考虑发送者和接收者没必要。攻击者要求发送者做一个消息 M 的认证结果, 攻击者获得一个通过 $\sigma \leftarrow T_K$（M）产生的标签 σ，K 是发送者和接收者的共享密钥。因此，也许可以简化这个情况，考虑攻击者有随机预言机去逼近 T_K（·）。能够在 M 点询问 T_K（·）函数的消息空间并且得到结果。相应的，从图片和确定的过程中排除接收者。攻击者将最终输出一个对 M，σ。只要 V_k（M，σ）=1 这就是有效的伪造，并且 M 从不询问一个随机标签预言。

注意密钥 k 不能直接交给攻击者，即不是随机的选择，也不是标签算法的计数。攻击者只能看到产生的标签。然而，σ 是一个密钥的函数，是随机选择或是计数。因此，也许应该提供相应的信息，作为对该体制的扩展。如果标签允许攻击者推断密钥，这个体制的确是不安全的。攻击者的行为在于两个部分的观点。最初是一个学习的过程，给定一个 T_K（·）函数的逼近，这里 K 是一个优先的随机数选择。能够询问这个随机预言机制 q 次，在许多条件下是满足该结果的，只要这种询问在明文空间的条件范围内。一旦这个阶段结束，加入伪造段，因此输出一对（M，σ）。攻击者可以声称成功，如果 V_k（M，σ）=1 并且 M 从不是攻击者对随机预言标签的询问。联合任意攻击者 A 是一个后继概率（概率是在密钥选择的基础上的，任何 T 的概率选择，和 A 做的概率选择）。体制的不安全是概率攻击者的不安全体制，对一些固定的值，所有的攻击者限制的资源。选择攻击者运行时间资源，所做的询问次数，伪造的消息 M 包括所有询问的总数，消息 M 的比特总数。

正式的，定义攻击者运行的实验，A 在体制 MA=（K，T，V）的攻击如下：

测试 $Exp_{MA,A}^{uf\text{-}cma}$

假设 K←k

假设（M，σ）← $A^{Tk(\cdot)}$

如果 V_k（M，σ）=1 并且 M 不是随机语言机制的 A 询问

那么返回 1 否则返回 0

定义 8.3、 假设 MA=（K，T，V）是一个消息认证体制，假设 A 是可以接入随机预言机制的访问，假设 $Adv_{MA,A}^{uf\text{-}cma}$ 是经验值 $Exp_{MA,A}^{uf\text{-}cma}$ 返回 1 的概率。那么对于任意的 t,q,μ, 假设

$Adv_{MA,A}^{uf\text{-}cma}$（t,q,μ）= $\max_{a}$ { $Adv_{MA,A}^{uf\text{-}cma}$}。A 在运算时间的最大值，发展最多 q 预言询问。因此随机预言询问的长度加上消息 M 的长度，输出的伪造长度最多为 μ 比特。

事实上，合法用户回答标签消息的询问，应该用这个例子计算，也就是说，必须期望 q 比 t 小，这就是 q,μ 与 t 独立的资源的原因。

9.3.4 使用定义：一些例子

开始检查一些消息认证体制的例子，使用访问定义决定强弱。固定一个 PRF F：$\{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 。首先的体制是 MA_1=（K，T，V）按照如下方式工作：

Algorithm $T_k(M)$
把 M 分解为 l 比特分组，$M=x_1 \ldots x_n$
For i=1,…,n do $y_i \leftarrow F_k(x_i)$
$\sigma \leftarrow y_1 \oplus \ldots \oplus y_n$
返回 σ

Algorithm $V_k(M, \sigma)$
把 M 分解为 l 比特分组，$M=x_1 \ldots x_n$
For i=1,…,n do $y_i \leftarrow F_k(x_i)$
$\sigma' \leftarrow y_1 \oplus \ldots \oplus y_n$
If σ=σ'那么返回 1，否则返回 0

现在假设开始评估该消息认证体制的安全性。

假设攻击者想伪造指定消息 M 的标签，可以进行的优先级可以确定。攻击者不是秘密密钥 K 的拥有者，这样就不能计算 F_k，因此将有时间困难计算 σ。然而，记住定义的安全的概念是在攻击者只要可以用相应的消息得到标签就是成功的，没有必要给定一个。现在注意甚至没有选择消息攻击（事实上没有任何正确标签数据的一些例子）攻击者也可以做到这一点。可以选择一个消息 M 组成两个相同的分组，也就是说 M=x ‖ x，这里 x 是一些 l 比特串。假设 $\sigma \leftarrow 0^l$ ，并且输出 M，σ。注意 V_k（M，σ）=1，因为 $F_k(x) \oplus F_k(x)=0^l = \sigma$。因此攻击者是成功的。更细节的攻击者是：

攻击者 $A_1^{Tk(\cdot)}$
假设 x 是一些 l 比特串
假设 M ← x ‖ x
假设 $\sigma \leftarrow 0^l$
返回（M，σ）

那么 $Adv_{MA1,,A1}^{uf\text{-}cma} = 1$ ，此外，A_1 没有构造随机预言，使用 t=O(l)时间，在伪造中输出一个 l 比特的消息，因此可以看做：$Adv_{MA1}^{uf\text{-}cma}(t,0.l) = 1$，也就是说体制 MA1 是不安全的。

有一些其他的攻击，例如注意到如果 $\sigma=F_k(M_1) \oplus F_k(M_2)$是 M_1 和 M_2 的标签，那么 σ 也是 M_1 和 M_2 正确的标签。因此是可能的，给出一个消息的标签。构造一个旧消息置换的伪造一个新消息的标签。指定相应攻击者和计算优势留给读者。

假设试着加强体制避免这些攻击。代替回答 F_k 到一个数据分组，将首先用索引提炼数据。为了做这些，选择一些参数 m，1≤m≤l-1，并且做为 m 比特串写做索引。消息认证体制 MA1=（K，T，V），看做如下条件：

Algorithm $T_k(M)$
把 M 分解为 l-m 比特分组，$M=x_1 \ldots x_n$
For i=1,…,n do $y_i \leftarrow F_k(<i> \| x_i)$
$\sigma \leftarrow y_1 \oplus \ldots \oplus y_n$
返回 σ

Algorithm $V_k(M, \sigma)$
把 M 分解为 l-m 比特分组，$M=x_1 \ldots x_n$
For i=1,…,n do $y_i \leftarrow F_k(<i> \| x_i)$
$\sigma' \leftarrow y_1 \oplus \ldots \oplus y_n$
If σ=σ'那么返回 1，否则返回 0

作为代码指示，把 M 划分为更小的分组，没有 l 尺寸，但是有尺寸 l-m。那么提炼第 *i* 个消息分组是使用自己的 *i* 值。分组索引，用二进制记录。上述<i>定义为整数 *i* 的二进制 m 串表示，在 XOR 运算前进行 F_k 分组填充。

注意，当 $i<2^m$，分组的编码摘要 *i* 可以作为一个 m 比特的串。也就意味着不能认证一个消息 M 有大于 2^m 的分组。也就是说，消息空间限制在最多（l-m）（2^m-1），并且简单的，长度是 l-m 的倍数。实际应用中选择一个合理的值，是非常困难的一种限制，比如 m=32，典型的消息在消息空间是足够大的，l 是典型的至多是 64 比特，使用左边的 32 比特作为数据本身使用。

总之，真正关心的问题是安全。是否改进来适应 MA_1？开始关注攻击会发现 MA_1 没有进行工作。例如作为攻击者 A_1，需要一个较小的修正使新的体制更加安全，也就是说选择一个分组大小不是 l 就是 l-m。当作为一个攻击者攻击 MA_2，成功的可能性是多少？问题的总数是询问 V_k（M，σ）=1 的概率，当 V 是赔偿体制的修改算法，M，σ 是 A1 的输出。修改算法将计算 $\sigma'=F_k(<1>\|x) \oplus F_k(<2>\|x)$并且测试是否这是等于 0^l，A 输出的 σ 值。当满足下属条件时，这种概率才会发生：$F_k(<1>\|x) = F_k(<2>\|x)$，并且这确实是不相似的。例如，如果使用分组密码这种情况是一定不会发生的，因为 F_k 是一个置换。甚至当 F 不是分组密码时，当 F 是好的 PRF 时，这个事件有很小的概率。特别的， $Adv_{MA2,a1}^{uf\text{-}cma}$ 是最多 $Adv_F^{prf}(t,2)$，这里 t=O(l)(一个读者也许可以确定，当边界是真实的)。因此，攻击者有很低的成功概率。

相似的争论在于包括上述的第二轮攻击，也就是基于消息分组的置换是，同样有抗新体制的低的成功概率。为什么？在新的体制里：

$$T_k(M_1,M_2) = F_k(<1>\|M_1) \oplus F_k(<2>\|M_2)$$
$$T_k(M_2, M_1) = F_k(<1>\|M_2) \oplus F_k(<2>\|M_1)$$

这与上述讨论的相同问题的等式是不相同的。作为一个例子，一个读者的上界也许是这些值是相等的，对适当的参数值，在一定的程度上，F 的不安全值是相等的。

然而，MA_2 还是不安全的，攻击者要求一个有效的选择消息攻击更多的细节。攻击者就几个相关的点询问标签预言机制，合成几个新消息的相关回馈。称为 A_2：Aversary $A_2^{Tk(\cdot)}$

假设 x_1,x_1'是可区分的，l-m 比特串，并且假设 x_2,x_2'是可区分的 l-m 比特串

$\sigma_1 \leftarrow T_k(x_1x_2)$; $\sigma_2 \leftarrow T_k(x_1x_2')$; $\sigma_3 \leftarrow T_k(x_1'x_2)$;

$\sigma \leftarrow \sigma_1 \oplus \sigma_2 \oplus \sigma_3$

返回（x_1'， x_2'， σ）

主张 $Adv_{MA2,A2}^{uf\text{-}cma}$ =1。为什么？这需要两件事。首先，V_k（x_1'， x_2'， σ）=1，第二，x_1'，x_2'永远不是上述码型中的对 $T_k(\cdot)$的询问。后者是事实，因为坚持上述条件满足 $x_1\neq x_1'$，$x_2 \neq x_2'$。这样意味着 $x_1'x_2' \notin \{x_1x_2, x_1x_2', x_1'x_2\}$。因此现在假设首先的声明，使用标签算法可以看到：

$\sigma_1=F_k(<1>\|x_1) \oplus F_k(<2>\|x_2)$

$\sigma_2=F_k(<1>\|x_1) \oplus F_k(<2>\|x_2')$

$\sigma_3=F_k(<1>\|x_1') \oplus F_k(<2>\|x_2)$

现在看一看 A_2 怎样定义σ并做计算，用消去率可以得到：

$$\sigma = \sigma_1 \oplus \sigma_2 \oplus \sigma_3$$
$$= F_k(<1>\|x_1') \oplus F_k(<2>\|x_2')$$

这的确是 $x_1'x_2'$正确的标签，意味着σ'值可以通过 V_k（$x_1'x_2'$， σ）计算，正如所说，随后的算法返回 1。在摘要里已经得出：$Adv_{MA2,}^{uf\text{-}cma}$（t,3,4(l-m)）=1，这里 t=O(l)。因此，体制 MA_2 是不安全的。

随后将看到，实际产生一个安全体制的细微的修改。这时，需要加强上述攻击的一个特

征。也就是这些攻击不能分析 PRF F。消息认证体制的分析不关心 F 的结构，不论是 RC6，DES 或是其体制，发现的消息认证体制自身的弱点。特别的，攻击工作仅仅当 F_k 是随机函数或是一个完美密码体制条件下。这更再次说明了已经得到的关于一个工具和的作用的观点。需要对工具做进一步更好的使用，事实上体制的安全是这种攻击模式下的工具运行模式在假设工具安全的条件下是安全的。

9.4 XOR 体制

考虑一个消息认证体制的函数簇，称为一个源于[11]的 XOR MACs，并且看出这是安全的。

9.4.1 体制

对于上述讨论的第二个体制，这个体制进行了相应的论证。给定一个确定的分组大小 l，选择一个参数 1≤m≤l-2，把消息 M 分割为 l-m-1 个分组，定义第 *i* 个分组为 x_i，定义<*i*>是整数 *i* 的 m 比特二进制串。定义这个体制和分析的安全性，首先介绍一个附属函数定义为 XOR 标签？（.,.）。采用一个随机预言机制为 $f:\{0,1\}^l \rightarrow \{0,1\}^L$。这同样有两个输出，第一个是称为 s 的 l-1 比特的串，的作用随后将说明。第二是上述讨论的消息 M，这些输入使用的 *f* 过程如下指出，返回一个称为 τ 的值。

算法 Algorithm XOR-Tag^f(s,M)
把 M 划分为 l-m-1 个分组 $M=x_1 \ldots x_n$
$Y_0 \leftarrow f(0 \| s)$
For i=0,…,n do $y_i \leftarrow f(1 \| <i> \| x_i)$
$\tau \leftarrow y_0 \oplus y_1 \oplus \ldots \oplus y_n$
返回 τ

附属函数 *f* 适用于 n+1 个点，每一个这种点是 lbit 串。第一个点是 0‖s，也就是提炼包括一个 0 比特的（l-1）比特串，使得整个串为 l 比特，*f* 作用后得到 y_0 。其余的 n 点上，*f* 提炼为 1，随后如下的一个分组索引并且数据分组自身，对于一个 1+m+(l-m-1)=l 比特。

现在准备描述这个体制，固定一个函数簇 F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$，并且消息认证体制的密钥是簇 F 的简单的 k 比特密钥 K，说明了一个特殊的函数 F_K。多方可以使用 F_K 在上述 *f* 的角色，实际上这个体制有两个版本，一个是确定并且固定的，使用一个全局计数变量。另外一个是不固定和随机的，区别仅仅在于 s 的选择。开始使用计数版本，发送者保持一个整数计数器 ctr，初始化为 0。使用<ctr>定义一个 l-1 比特的整数，（计数器变化范围为 0 到 2^{l-1}-1，注意 *i* 是一个分组索引，<*i*>是一个二进制编码，但是作为一个 m 比特串，因此符号<·>是在串长度之外的比特，返回依赖于上下文的参数，但是有希望的是不引起过多的混乱）。基于使用 F 计数的 XOR MAC 体制，表示为 C-XOR^F = （K，T，V），工作如下：

Algorithm T_k(M)
$\tau \leftarrow$XOR-Tag^{Fk}(<ctr>,M)
$\sigma \leftarrow$(<ctr>, τ)
Ctr←ctr+1

Algorithm V_k(M, σ)
解析 σ 为（s， τ ）
τ’← XOR-Tag^{Fk} (s,M)
If τ = τ’时返回 1 否则返回 0

换言之，消息标签 $M=x_1\ldots x_n$ 是一个组成目前以二进制编码的计数值，一个子标签为 τ，

$$\tau = F_k(0 \| <ctr>) \oplus F_k(1 \| <1> \| x_1) \oplus \ldots \oplus F_k(1 \| <n> \| x_n)$$

为了校验收到的标签 σ =（<ctr>, τ ）确认算法重新计算子标签，称为 τ' 。对于给定计数值的一个函数，然后检查这个子标签在 σ 中配合一个假设。

这个体制的随机版本是使用 F 随机化 XOR MAC 体制，这表明 C-XOR^F=（K，T，V）。使用一个随机 l-1 比特值简单代替计数值，选择一个更新标签的算法。更细致的，算法如下工作：

Algorithm $T_k(M)$

R←$\{0,1\}^{l-1}$

τ ←XOR-Tag^{Fk}(<ctr>,M)

σ ←(r, τ)

返回 σ

Algorithm $V_k(M, \sigma)$

解析 σ 为（r, τ ）

τ'← XOR-Tag^{Fk} (s,M)

If $\tau = \tau'$时返回 1 否则返回 0

换言之，消息 $M=x_1\ldots x_n$ 的标签是一个随机值 r 组成的对，并且一个子标签 τ，这里，

$$\tau = F_k(0 \| r) \oplus F_k(1 \| <1> \| x_1) \oplus \ldots \oplus F_k(1 \| <n> \| x_n)$$

为了验证收到的标签 σ =（r, τ ），验证算法重新计算正确的子标签，称为 τ' ，作为一个给定值 *r* 的函数，那么检查这个标签与 σ 中相称的子标签。

9.4.2 安全考虑

在考虑安全因素前，重要的阐述关于概率的伪造的一个事件，一个伪造是一个对子：（s, τ ）这里 s 是一个（l-1）比特串，并且 τ 是一个 L 比特串。现在可以认为标签算法本身在一个特定的条件下产生第一个构成。对于具体的因素，用基于 XOR 体制的计数体制，因此对于合法标签的信息，一个值不断重复一个消息到另一个消息。对于具体的应用，采用基于 XOR 体制的计数模式。这里 s 是一个计数值，因此对于正常的标签消息，一个值永远不能重复使用。（假设没有多于 2^{l-1} 消息是认证，因此计数不能进行圈数包围）。这并不意味着攻击者被迫对 s 使用一个计数值在试图使用伪造的 M，（s, τ ）。攻击者是完全免费的使用 s 的任意的值，并且特别使用一个使用过的计数器。回忆攻击者的目标是得到认证算法收到一个对子 M，（s, τ ），目标仅仅限制在 M 不是一个标签预言机制的询问。观察认证算法的代码，在任意的方式下反馈 s 作为计数器的知识。试着检查任意计数相关的 s 的性质，事实上，认证算法根本不能维持一个计数器，不够稳定。因此，没有任何条件限制一个攻击者在进行伪造攻击的时候使用计数值。

一个类似的情况在于把握 XOR 体制的随机化版本。尽管合法用户随机选择 r，因此合法标签有首先构成标签的 r 的随机值，攻击者可以试图进行一个伪造 M，（r, τ ）这里 r 是完全不随机的。攻击者选择 r 并且得到任何想的结果。这是攻击者随意必须记住的，这个自由度在攻击者的一方，在分析体制时攻击的一方是必须记住的。

为了得到一些关于这个体制直观的消息，对返回攻击是有帮助的。在以前的体制中，用来打破体制 MA_2 的例子，并且可以看到这是失败的。将看到攻击者攻击 A_2 的说明，要求相关信息的标签并且要求返回三个相关消息的标签，在得到和取消之间进行公共开发。在新体制下的三个相同的值，返回标签预言机制看到子标签。

$$\text{XOR-Tag}^{Fk}(<0>,x_1x_2) = F_k(0 \| <0>) \oplus F_k(1 \| <1> \| x_1) \oplus F_k(1 \| <2> \| x_2)$$

$$\text{XOR-Tag}^{Fk}(<0>,x_1x_2') = F_k(0 \| <0>) \oplus F_k(1 \| <1> \| x_1) \oplus F_k(1 \| <2> \| x_2')$$

$$XOR\text{-}Tag^{Fk}(<0>,x_1'x_2) = F_k(0 \| <0>) \oplus F_k(1 \| <1> \| x_1') \oplus F_k(1 \| <2> \| x_2)$$

总共这三个值产生一个混乱，并没有看到其消息的子标签，因为这个值是相应的计数器是不用取消的，因此这个攻击没有效果。

是否存在另一些攻击？看起来很难达成一致，但是并不意味这么多。也许攻击者是聪明的。就这一点已有了进展，在函数簇 F 的是 PRF 的假设下称为可证安全性，这意味上述简单的攻击是不存在的。在这条线上的信心是比一个发现攻击更无力的，可以确定 F 簇是自身安全的。

9.5 例子

XOR 体制的安全结果

描述法则总结体制的安全，基于计数体制开始。称为体制中的整参数 m 和如下的分组索引参数。假设明文（l,m）表示所有串 M 的值。因此 M 的长度值是 n·(l-m-1)，对于整数 n，$1 \le n \le 2^n-1$。这就是 XOR 消息认证体制的消息空间。

下述定理有一个熟悉的格式，基于 XOR 消息认证体制上界是不安全的，如果在 PRF F 条件下是不安全的。换言之，如果上界是最大概率，攻击者可以攻击 XOR 体制，（也就是攻击伪造一个还没有认证的正确的标签），上界在这种体制的某种程度上是破解 PRF F 的最大能力。这是努力想要达到的一个另外例子，没有一个简单的攻击一个体制的保证，无论怎样聪明，只要已知潜在的工具是好的。特别的，可以确定这种攻击在上述的例子中是不能攻击这种体制的。

同样，边界可以量化，因此可以使用评价安全的质量，在函数 F 中，当使用一些 PRF 的说明。边界更合理：看到攻击消息认证体制的机会是与攻击 PRF 体制几乎是同样难度。

定理 8.4[11] 假设 $F、\{0，1\}^K \times \{0，1\}^l \to \{0，1\}^L$ 是一个 PRF，并且假设 $C\text{-}XOR^F$ = （K，T，V）是相应的基于上述 XOR 消息认证体制，使用分组索引参数 m≤l-2，明文消息空间(l,m)。然后对于任何的 t,q,μ，并且 $q<2^{l-1}$，有：$Adv_{c\text{-}xorF}^{uf\text{-}cma}(t,q,\mu) \le Adv_F^{prf}(t',q')+2^{-L}$，这里，t'=t+O(μ)并且 q'=q+1+μ/(l-m-1)，结果对于体制的随机版本是类似的，对于一个另外的条件相似的期望。这次，一个以 $q^2/2^l$ 有一个碰撞的概率上界，指出不能规定一个破解的概率，无论何种 PRF 的概率。将随后看到这是一个体制本身的特征，属于生日攻击中的一种。

定理 8.5[11]假设 $F、\{0，1\}^K \times \{0，1\}^l \to \{0，1\}^L$ 是一个 PRF，并且假设 $C\text{-}XOR^F$ = （K，T，V）是相应的基于上述 XOR 消息认证体制，使用分组索引参数 m≤l-2，明文消息空间(l,m)。然后对于任何的 t,q,μ，$Adv_{c\text{-}xorF}^{uf\text{-}cma}(t,q,\mu) \le Adv_F^{prf}(t',q') + q^2/2^l + 2^{-L}$。这里 t'=t+O(μ)并且 q'=q+1+μ/(l-m-1)

这次将不再证明这些结果，随后将可以做这些，当发展更多的技术。

9.6 随机函数构造好的 MACs

对于设计 MACs 的普通的方法利用这个事实，任何伪随机函数事实上是一个 MAC。这个归约是具体安全分析如下的形式[12]，可以指出归约总是紧的，是安全降级。这种关系意味着证明 CBC MAC 的安全作为 MAC 不足以展示 CBC 变化为伪随机的。为了简化 MAC 的定义域，对于一些整数，长度严格为 d 的串序列。

定理 8.6、假设 MAC：$KeysMAC \times \{0,1\}^d \to \{0,1\}^s$ 是一个函数簇，并且假设 q,t≥1 是整数，

那么：

$$Adv_{MAC}^{uf\text{-}cma}(t,q,dq) \le Adv_{MAC}^{prf}(t',q)+1/2^{s} \qquad (8.1)$$

这里 t'=t+O(s+d)藏在 O 符号的常数只能依靠计算模型的细节。这是一个小的常数，应该考虑 t'≈t。

证明：假设任意伪造攻击的消息认证码 MAC，假设经验伪造（MAC，A）的预言机制是至多调用 q 次，并且 A 的运算时间至多是 t，这些数量在定理 8.3 中可以作为讨论度量。为 MAC 设计一个可分辨的 B_A，与 $Rand^{d\to s}$ 因此，

$$Adv_{MAC}^{prf}(B_A) \ge Adv_{mac,a}^{uf\text{-}cma} - 1/2^{s} \qquad (8.2)$$

然而 B 将运行时间 t'，并且做至多 q 次询问，在定义 5.4 中使用时间测定的讨论。这意味着等式 8.1 成立。

$$Adv_{MAC}^{uf\text{-}cma}(t,q,dq) = \max_{A} \{Adv_{MAC,A}^{uf\text{-}cma}\}$$

$$\le \max_{A} \{Adv_{MAC}^{prf}(B_A)+2^{-s}\}$$

$$= \max_{A} \{Adv_{MAC}^{prf}(B_A)\}+ 2^{-s}$$

$$\le \max_{B} \{Adv_{MAC}^{prf}(B_A)\}+ 2^{-s}$$

$$= Adv_{MAC}^{prf}(t',q)+ 2^{-s}$$

上述第一个等式，是通过不安全函数的定义而定义的定义 8.3。如下的不等式使用等式 8.2。接下来，简单使用最大特征，用 5.4 定义的不安全函数定义的结论。因此保持设计 B_A 使得等式 8.2 是真的。记住 B_A 是给定函数 A 随机预言机制的询问：$f:\{0,1\}^{d}\to\{0,1\}^{s}$。将运行 A，提供一个环境 A 的随机预言机制是 B_A 回答的。当 A 最后输出伪造，B_A 检查是否正确，如果不这样计算，f 必须已经有一个 MAC 函数簇的例证，比一个随机函数更合适。

通过假设预言机制试验伪造（MAC，A）是至多调用 q 次，简单的，可以假设刚好是 q。这就意味着 A 做询问数字的随机预言机制是 q-1，现在编码完成 B_A 任务。

分辨机制 B_A^{f}

For i=1,…,q-1 do

当 A 询问预言机制，M_i,回答是 $f(M_i)$

End For

A 输出为（M，σ）

$\sigma' \leftarrow f(M)$

If $\sigma = \sigma'$并且 $M \notin \{M_1, \ldots, M_{q-1}\}$

然后返回 1 否则返回 0

这里 B_A 初始化 A 用一些随机事件的随机序列并且开始运算，如果是正确的，返回 1。

现在继续分析下去：可以声称：

$$P[B_A^{f}=1: f\leftarrow MAC] = Adv_{MAC,A}^{uf\text{-}cma} \qquad (8.3)$$

$$P[B_A^{f}=1: f\leftarrow Rand^{d\to s}] \le 1/2^{s} \qquad (8.4)$$

减去得到的等式（8.2），从编码看来，显然 B_A 开始 q 预言询问。考虑习惯的运算时间，提及所有的所有的经验，同样也是真的，B_A 的运算时间是 t+O(d+s)，因此保持上述两个等式的公正的。

在 f 的第一个例子中，MAC 的情况，因此模仿环境 B_A 提供了 A 的伪造正确的实验伪造（MAC，A）。既然 B_A 返回 1，当 A 进行连续的伪造，有等式 8.3。

在第二个条件下，在一个环境下进行运算是与这种做法不同的，也就是说，一个随机函

数用来计算 MACs，没有办法在这个环境做 A 的计算，但是无论怎样，知道概率是 $\sigma = f(M)$ 是 2^{-s}，因为 f 是一个随机函数，只要 A 不询问预言机制 M，等式 8.4 如下。

9.7 CBC MAC

最多消息认证码使用在 CBC MAC。假设 $f:\{0,1\}^l \rightarrow \{0,1\}^l$ 是一个函数，假设 $f^{(n)}:\{0,1\}^{nl} \rightarrow \{0,1\}^l$ 是一个函数，输入是 $x_1,\ldots,x_n$，输出是 y_n，这里，$y_i = f(y_{i-1} \oplus x_i)$并且 $y_0=0^l$。如果函数 F 是有限簇，输入长度 l 并且输出长度 l，那么假设 $F^{(n)}$ 定义函数簇，使用密钥 K 的索引是 $F_K^{(n)}$。新的函数簇是输入长度 nl，输出长度 l，并且叫做 F 的 CBC 模式。

当 F 是 DES，有 CBC 构造用于实践，称为 DES CBC MAC。一个构造是一个美国和国际标准，广泛的使用在金融和银行章节。安全性是值得研究的。

9.7.1 CBC MAC 的安全

在例 8.2 和第 5.11.1 节中，讨论 CBC 的构造，并且注意在定理 5.20 中，如果 F 是一个 PRF 簇，那么因此是 $F^{(n)}$。从定理 8.6，能够现在决定 CBC 构造，完成一个好的 MAC，只要下述函数是随机的。

定理 8.7[12] 假设 l,m≥1，并且 q,t≥1 是整数，因此，$qm \le 2^{(l+1)/2}$，假设 F：$KeysF\times\{0,1\}^l \rightarrow \{0,1\}^l$ 是一个函数簇。那么：$Adv_{f(m)}^{uf\text{-}cma}(t,q,mql) \le Adv_F^{prf}(t',q')+(3q^2m^2+2)/2^{l+1}$

$$\le Adv_F^{prp\text{-}cpa}(t',q')+(2q^2m^2+1)/2^l$$

这里 $q'=mq$ 并且 $t'= t + O(mql)$

特别的，如果 F=DES，有一个 DES CBC MAC 安全程度的评价，在某种程度上是 DES 作为 PRF 的加强型。不幸的，作为在 5.6.2 节的讨论，DES 作为 PRF 已经太强了。对于更强的分组密码可以做得更好。

9.7.2 对 CBC MAC 的生日攻击

攻击背后基本的思想是 Preneel, Oorschott[163]和 Krawczyk 提出的基于伪造的相互碰撞。攻击者[163]介绍函数是随机的条件下的分析假设，意味着对 CBC-MAC 变换簇的应用是 $Rand^{l\rightarrow l}$ 或 $Perm^l$。这里不必进行这种假设，这种攻击对任意置换簇的工作。攻击随机性从伪造的随机性开始。这使得攻击更普通。（把关注集中在置换的条件下，因为 CBC-MAC 的实践通常是基于分组函数）

命题 8.8、假设 l,m,q 是整数，并且满足 $1\le q\le 2^{(l+1)/2}$，并且 m≥2。假设 F：$KeysF\times\{0,1\}^l \rightarrow \{0,1\}^l$ 是置换簇。那么有一个伪造者 A 提出 q+1 个随机询问，运算时间为 O（lmq logq）并且得到 $Adv_{F(m),A}^{uf\text{-}cma} \ge 0.3\cdot q(q-1)/2^l$，作为 $q \ge 2$ 的推论，$Adv_{F(m)}^{uf\text{-}cma}(t,q,qml) \ge 0.3\cdot q(q-1)(q-2)/2^l$，对一个随机预言机询问消耗的时间评估称为一个时间单位。

与上述定理 8.7 比较，可以看到在消息分组数的平方上边界是紧的。

现在继续进行证明，从一对引理开始，考虑从生日攻击开始，引理考虑一个轻微的变化。并且攻击概率仍然与通常的生日攻击难题是相同的。

引理 8.9 、假设 l,q 是整数，满足 $1\le q \le 2^{(l+2)/2}$。固定 $b_1,\ldots,b_q \in \{0,1\}^l$。那么，$P[\exists i,j$ 使得 $i\ne j$ 并且 $b_i \oplus r_i = b_j \oplus r_j : r_1,\ldots,r_q \leftarrow \{0,1\}^l] \ge 0.3\cdot q(q-1)/2^l$

证明：这仅仅是扔 q 个球到 $N=2^l$ 个柜子里，并且一个碰撞概率的低边界，除去元素平

移一比特，箱子赋值是第 i 个球是 $r_i \oplus b_i$ 而不是 r_i ，这样可以使用假设，但是与 b_j 固定不同，r_i 是均匀分布的，因此是 $r_i \oplus b_i$。因此概率是附件 A.1 的标准生日攻击难题。

如下引理的第一部分证明陈述了一个 CBC-MAC 变换。真实的兴趣在于引理的第二部分，也就是说在这个条件下函数是置换，CBC-MAC 变换具有特征是：输出碰撞当且仅当产生输入碰撞。这是攻击的关键，随后将进行阐述。

引理 8.10、假设 l,m≥2 是整数，并且 $f:\{0,1\}^l \rightarrow \{0,1\}^l$ 是一个函数。假设 α_1，α_2，…，α_m，并且 β_1，β_2，…，β_m 在{0，1}^{ml}，是这样 $\alpha_k = \beta_k$ 对于 k=3，…，m，那么：

$$f(\alpha_1) \oplus \alpha_2 = f(\beta_1) \oplus \beta_2 \Rightarrow f^{(m)}(\alpha_1, \alpha_2, \ldots, \alpha_m) = f^{(m)}(\beta_1, \beta_2, \ldots, \beta_m)$$

如果 f 是一个置换，另外，逆是成立的：

$$f^{(m)}(\alpha_1, \alpha_2, \ldots, \alpha_m) = f^{(m)}(\beta_1, \beta_2, \ldots, \beta_m) \Rightarrow f(\alpha_1) \oplus \alpha_2 = f(\beta_1) \oplus \beta_2$$

证明：从 $f^{(m)}$ 的定义是第一部分，对于第二部分，定义 f^1 为置换 f 的逆。因此过程是 $y_m \leftarrow f^{(m)}(\alpha_1, \alpha_2, \ldots, \alpha_m)$，对于 k=m，最小数为 3，进行运算 $y_{k-1} \leftarrow f^1(y_k) \oplus \alpha_k$ ； 返回 $f^1(y_2)$，返回 $f(\alpha_1) \oplus \alpha_2$，这里过程为：

$y_m \leftarrow f^{(m)}(\beta_1, \beta_2, \ldots, \beta_m)$，对于 k=m，最小数为 3，进行运算 $y_{k-1} \leftarrow f^1(y_k) \oplus \beta_k$ ； 返回 $f^1(y_2)$，返回 $f(\beta_1) \oplus \beta_2$ ，但是通过假设这个过程有 y_m 相同的值，知道 $\alpha_k = \beta_k$ 对于 k=3，…，m，因此这个过程返回相同的值。命题 8.8 的证明：在伪造攻击前，讨论思想

伪造 A 有一个预言机 $g = f^{(m)}$ ，这里 f 是 F 的一个例子，伪造的策略是构造 q 序列询问，最后 m-2 分组。这些询问的第一个分组是清楚的但是固定的。第二分组，然而，在询问过程中是随机并且独立的。定义 α_n 的询问 n 是第一个分组，作为 r_n 的第二个分组，伪造者希望 i≠j 因此，$f(\alpha_i) \oplus r_i = f(\alpha_j) \oplus r_j$ ，这些情况发生的可能性是不充分的，伪造需要探测事件是怎样发生的。引理 8.10 确认为这个相互碰撞成功如果输出的 MAC 相同（这是事实，如果 f 是置换）。那么观察为如果两个碰撞询问的第二个分组是一些值 α 的相同值进行 XOR，那么询问结果仍然碰撞。伪造者仍然通过第二个分组进行修改，得到一个修改分组询问的 MAC 值，用第二个分组，做为第二个分组篡改的 MAC 值。

伪造者以下呈现细节。使用子程序找到给定的序列 σ_1，…，σ_q 的值，返回一个对（i,j），因此 $\sigma_i = \sigma_j$ ，如果这样的对子成立，否则返回（0，0）。

伪造 A^g

假设 α_1，α_2，…，α_q 是不同的 l 比特串
For i=1,…,q do $r_i \leftarrow \{0,1\}^l$
For i=1,…,q do
  $X_{i,1} \leftarrow \alpha_i$ ； $X_{i,2} \leftarrow r_i$
  For k=3,…,m do $X_{i,k} \leftarrow 0^l$
  $X_i \leftarrow X_{i,1}, \ldots, X_{i,m}$
  $\sigma_i \leftarrow g(X_i)$
End For
$(i,j) \leftarrow$ Find $(\sigma_1, \ldots, \sigma_q)$
If(i,j)=(0,0)那么终止
Else
  假设 α 是任意 l-bit 串与 0^l 不同
  $X_{i,2}' \leftarrow X_{i,2} \oplus \alpha$，$X_{j,2}' \leftarrow X_{j,2} \oplus \alpha$
  $X_i' \leftarrow X_{i,1}, X_{i,2}', \ldots, X_{i,m}$ ； $X_j' \leftarrow X_{j,1}, X_{j,2}', \ldots, X_{j,m}$
  $\sigma_i' \leftarrow g(X_i')$
  Return (X_j', σ_i')
End If

为了评估成功的概率，假设 $g = f^{(m)}$ ，这里 f 是 F 的特例。假设（i,j）是通过寻找子集返回的值。假设（i,j）≠（0，0），那么可以知道：

$$f^{(m)}(x_{i,1},\ldots,x_{i,m}) = f^{(m)}(x_{j,1},\ldots,x_{j,m})$$

通过假设 f 是一个置换，并且通过设计 $x_{i,k} = x_{j,k}$ ，对于 k=3，…，m。推论 8.10 的第二部分意味着 $f(\alpha_i) \oplus r_i = f(\alpha_j) \oplus r_j$ ，两边同时加 α，得到 $f(\alpha_i) \oplus (r_i \oplus \alpha) = f(\alpha_j) \oplus r_j \oplus \alpha$），换言之，$f(\alpha_i) \oplus X_{i,2}' = f(\alpha_j) \oplus X_{j,2}'$。引理 8.10 的第一部分说明了 $f^{(m)}(X_i') = f^{(m)}(X_j')$，因此 σ_i' 是 X_j' 的正确的 MAC 值。此外，主张 X_j' 是新的，意味着对 g 预言机没有进行询问。既然 $\alpha_1, \alpha_2, \ldots, \alpha_q$ 是不同的，需要担心的仅仅是 $X_j' = X_j$ ，但这也是规则，因为 $\alpha=0^l$ 。

说明了只要找到子集返回（i,j)≠（0，0）就可以伪造成功，因此成功的概率是（i,j)≠（0，0）的概率，在 q 值为 $\sigma_1, \ldots, \sigma_q$ 范围无论如何有一个碰撞。引理 8.10，无论碰撞在哪些值，当且仅当在 q 值为 $f(\alpha_1) \oplus r_1$ ，…，$f(\alpha_q) \oplus r_q$ 有一个碰撞。在 r_1 ，…，r_q 的随机选择概率，对引理 8.9，后面的概率是低边界值的，通过在命题的需求数量。通过注记总结使用一个简单的 FindCol 的执行（使用一个平衡树配置）是声称的运算时间。

9.7.3 长度可变性

简单的，假设贯穿这一章的假设是需要认证的串具备长度是 lbits 的倍数。这个限制是容易使用简单和知名的明码方法进行分布的。例如，附加一个 1 并且附加最小限度的 0 来决定串是 lbits 的倍数。

CBC MAC 并不直接给出可变长度输入的消息认证码，事实上，当串的长度可变时，容易“破解”CBC MAC 构造。在这一章的最后，会要求进行这方面的工作。尝试这些，这是 MACs 的好的联系。

一个可变长度的消息认证攻击的企图是添加任意一串 $x=x_1\ldots x_n$，是数字 m，适当的对最后 l 比特进行编码，这样 CBC MAC 为 m+1 组。（当然这利用了一个限制 $m<2^l$，不是特别的涉及）定义：$f_a^*(x_1\ldots x_m)=f_a^{(m+1)}(x_1\ldots x_m m)$ 。

可以看到 f^* 是一个安全的 MAC，采用任意的 lbit 字 b,b’和 c，这里 b≠b’，容易检查到给定的一些值：

（1） $t_b = f^*(b)$

（2） $t_{b'} = f^*(b')$

（3） $t_{b1c} = f^*(b \| 1 \| c)$

攻击者是具备了 $f^*(b' \| 1 \| t_b \oplus t_{b'} \oplus c)$，一串的认证标签是没有必要询问以前，因为这正好是 t_{b1c} 。不管上述方法是否失败，有许多合适的方法产生 PRF，是好的输入变化。可以提到三条：在每一条中，假设 F 是 lbit 到 lbit 的有限函数簇。假设 $x=x_1\ldots x_m$ 是申请 f_a 的消息：

（1） 输入长度密钥分开，设置 $f_a^*(x)=f_{am}^{(m)}(x)$，这里 $a_m = f_a(m)$

（2） 长度预算，设置 $f_a^*(x)=f_{am}^{(m+1)}(m \| x)$

（3） 加密最后一个分组，设置 $f_{a1,\,a2}^*(x)= f_{a2}(f_{am}^{(m)}(x))$

前两种方法是从[12]中得到的，最后的方法在[112]的信息附件中得到。并且现在用 Petrank 和 Rackoff[156]进行分析，这是串中最吸引人的方法，既然在计算最后和帮助在线 MAC 的长度是不必要的。

9.8 基于 MAC 的普通的 HASH

对于快速消息认证有效范例，基于“几乎普遍的单向函数”，这些单向函数的设计得到

许多关注，得到一些快速的结果，因此基于 MAC 的普通的 HASH 是最快的 MACs，开始描述工具，然后看见怎样使用消息认证。

9.8.1 几乎普通的 HASH 函数

假设 H、Keys(H)×Dom(H)→$\{0,1\}^L$ 是函数簇，假设：Adv^{uh} H = $\max_{a_1,a_2}$ {P[$H_K(a_1)=H_K(a_2)$; K←Keys(H)]}，在所有不同的点 a_1 ,a_2 ∈Dom(H)进行最大化。

Adv^{uh} H 的较小值，作为几乎普通的函数，H 的性质是较好的，也就是说如果 Adv^{uh} $H=2^{-L}$ 是普通的 HASH 函数（随后可以看到不安全值的最低概率值）。一个强一些的特征是几乎 XOR 普遍的。

定义 8.12、 假设 H：Keys(H)×Dom(H)→$\{0,1\}^L$ 是一个函数值。

$$Adv^{uh} H = \max_{a_1,a_2} \{P[H_K(a_1) \oplus H_K(a_2)=b; K\leftarrow Keys(H)]\}$$

在所有不同的点 a_1 ,a_2 ∈Dom(H)和所有串 b∈$\{0,1\}^L$ 进行最大化。

Adv^{xuh} H 是小一些的值，作为 XOR 普通函数值 H 的特征越好，也就是说，H 是一个 XOR 通用的 HASH 函数，如果 Adv^{xuh} $H=2^{-L}$（这是不安全的最低的概率），几乎 XOR 普通的要求是比几乎普遍更强的要求。

命题 8.13、假设 H：Keys(H)×Dom(H)→$\{0,1\}^L$ 是一个函数簇。那么：

$$Adv^{uh} H \le Adv^{xuh} H$$

证明：在定义中产生 8.12 和 8.11 的数量是 $b=0^L$，最简单的例子是所有函数簇。

命题 8.14、所有 lbits 到 Lbits 的簇 $Rand^{l\to L}$ 是普通的或 XOR 普通的，意味着：$Adv^{uh}Rand^{l\to L} = Adv^{xuh}Rand^{l\to L}= 2^{-L}$

证明：就命题 8.13，需要知道 $Adv^{xuh}Rand^{l\to L} = 2^{-L}$，因为 h 是一个随机函数。例子的另一个源头是有限域上的多项式。

例 8.15、使用 GF（2^l）的识别，是 2^l 元素的有限域，锁定一个 GF（2）不可约、l 次的多项式，因此可以在域上数学运算，定义的 HASH 函数 H 拿走在$\{0，1\}^l$上的密钥对 α，β，因此，α≠0，领域是$\{0，1\}^l$并且值域是$\{0，1\}^L$，这里 L≤l，定义函数：

$H_{\alpha\beta}$（x）=$[\alpha x+\beta]_{1...L}$ ，也就是说，使用密钥 α，β，并且输入 x∈$\{0,1\}^l$，在有限域首先计算 αx+β，作为一个 lbit 串进行观察，输出前 L 比特。

命题 8.16、H 簇：根据上述定义，Keys(H)×$\{0,1\}^l$→$\{0,1\}^L$，这里 L≤l 并且 Keys(H)是 lbit 串的所有对（a,b），因此，a≠0，是一个 XOR 普通 HASH 值。

证明：需要知道 $AdvH = 2^{-L}$，根据 a_1 ,a_2 ∈$\{0,1\}^l$ ，因此，$a_1 \neq a_2$ ，并且固定 b∈$\{0,1\}^L$，对函数固定任意密钥，意味着任意 a≠0 和 β。注意 y =αx+β，如果 $x=\alpha^{-1}(y-\beta)$(这里的运算是有限域上的运算，并且假设 a≠0)，这意味着 GF（2^l）到 GF（2^l）的映射 x→αx+β 是一个置换。如下的命题也是如此。

从如下的证明可以看到对于一些术语是有用的，固定任意两个点 a_1 ,a_2 ，在这个簇的定义域 Dom(H)，对仅有的限制是不能相等。在$\{0，1\}^L$上固定一个点，H 固定，可以联合这三点的概率：

$$UHColPr_H(a_1,a_2，b)=P[H_K(a_1) \oplus H_K(a_2) = b: K\leftarrow Keys(H)]$$
$$= P [h(a_1) \oplus h(a_2) = b: h\leftarrow H]$$

上述的两个表示由定义是相等。

另外的更动态的解释几乎 XOR 普通的测量是可以使用的，可以假设点 a_1 ,a_2，b 是攻击者可以做到的，攻击者 C 知道 H 是目标簇，连续产生输出一些明确的值 a_1 ,a_2 ∈Dom(H)，

并且一个值 $b\in\{0,1\}^l$，现在一个密钥 K 可以随机选择，定义函数 H_K ： $Dom(H)\rightarrow\{0,1\}^L$，并且测试是否 $H_K(a_1) \oplus H_K(a_2) = b$。如果是这样，攻击者 C 成功。表示攻击者获胜的概率是 $Adv^{xuh}H$，C。这样可以声称概率是 $Adv^{xuh}H$。

原因是对于攻击者有一个简单的策略，也就是说，选择一个点(a_1 ,a_2，b)，最大上述的概率 $UHColPr_H(a_1 ,a_2$，b)，这应该是相对安全的，至少对于这个条件当攻击者是确定的。但是这种声明是真实的，甚至当攻击者是概率可能的。也就是说，输出三个点可以依靠不同的随机事件。（在这样的条件下，概率定义了 $Adv^{xuh}H,C$ 是接收 K 的选择，也就是 C 的随机事件）证明在下述命题 8.17 的声明中是正确的。因此有两个相同的途径考虑 $Adv^{xuh}H$，更多静态的和一些动态的。依靠这些设置，可以得到一些更有利的条件。

在开始之前，命题 8.17 是可以证明的，然而假设这个定义的一些特征。一个游戏密钥特征的如下特殊阶的步骤：首先攻击者选择 a_1 ，a_2，b，那么 K 是随机选择的，并且函数 H_K 是定义了的。攻击者不允许选择 a_1 ，a_2，b 作为 K 的函数，必须首先提交。然后有赢得这个游戏的概率。

也就是不是攻击者计算限制的不同于一些的概念。也就是说，只要选择 a_1 ，a_2，b 就可以进行运算。既然如此，安全条件是不真实的。因此，这是纯粹的信息论概念。这里是应答限制。

命题 8.17、假设 $HKeys(H)\times Dom(H)\rightarrow\{0,1\}^L$ 是一个函数簇，C 算法输出是 a_1，a_2，b，因此 a_1，a_2 是 Dom(H)中不同的点，并且 $b\in\{0,1\}^L$，那么：

$$Adv^{xuh}\ H,C\leq Adv^{xuh}H$$

证明：记住 C 是一个概率意味着长度为 r 的附加输入的随机比特的序列 ρ，使用进行计算。依靠 r 的值，三个输出 C 将变化。可以定义 三个 a_1（ρ），a_2（ρ），b（ρ）输出当随机事件是 ρ，对任意特殊的值 ρ 定理 8.11 是非常清楚的。

$$P[H_K（a_1（\rho）\oplus H_K（a_2（\rho））= b（\rho）：K\leftarrow Keys(H)）]= Adv^{xuh}\ H$$

使用这个可以得到：

$$\begin{aligned}Adv^{xuh}\ H,C &= P[H_K（a_1（\rho）\oplus H_K（a_2（\rho））= b（\rho）：\rho\leftarrow\{0,1\}^r; K\leftarrow Keys(H)）]\\ &= \sum_{\rho\in(0,1)^r}[H_K（a_1（\rho）\oplus H_K（a_2（\rho））= b（\rho）：K\leftarrow Keys(H)）]\cdot 2^{-r}\\ &\leq \sum_{\rho\in(0,1)^r} Adv^{xuh}\ H\cdot 2^{-r}\\ &= Adv^{xuh}\ H\end{aligned}$$

第一个等式是 Adv^{xuh} H,C 的定义。在第二行，使用 C 的随机事件是长度 r 的所有串的随机选择，在第三行，使用上述的观察。

Adv^{xuh} H 最低值是多少，可以证明最低概率值是 2^{-r}，那么值可以得到一个 XOR 普通簇，下述正好是这个声明。

命题 8.18 假设 $HKeys(H)\times Dom(H)\rightarrow\{0,1\}^L$ 是一个函数簇，那么：$Adv^{xuh}\ H\geq 2^{-r}$

证明：定义两个不同的值，a_1 ,$a_2\in\{0,1\}^l$ ，并且对于任何固定密钥 $K\in Keys(H)$，假设

$$C(K) = P[H_K（a_1）\oplus H_K（a_2）= b : b\leftarrow\{0,1\}^r]$$

那么 $c(K)= 2^{-r}$ 。为什么？对于固定的 K，a_1 ,a_2 ，所有 $H_K（a_1）\oplus H_K（a_2）$ 是固定的，称为 b’，上述是询问概率是什么？如果随机选取 b，那么 b=b’相同的概率是 2^{-r} 。

现在考虑对手 C，从{0，1}^{L} 随机选取 b，输出三个 a_1 ,a_2，b，（注意攻击者是概率的，因为是 b 的随机选择）那么：

$$\begin{aligned}Adv^{xuh}\ H,C &= P[H_K（a_1（\rho）\oplus H_K（a_2（\rho））= b（\rho）：\rho\leftarrow\{0,1\}^r; K\leftarrow Keys(H)）]\\ &= \sum_{K\in Keys(H)} c(K)\cdot P[K'=K : K'\leftarrow Keys(H)]\end{aligned}$$

$$= \sum_{K \in Keys(H)} 2^{-r} \cdot P[K'=K : K' \leftarrow Keys(H)]$$

$$= 2^{-r} \cdot 1$$

因此可以介绍一个攻击者 C，因此 Adv^{xuh} H，C = 2^{-r} ，从命题 8.17 随后可以得到：

$$Adv^{xuh} H \geq 2^{-r}$$

9.8.2 使用 UH 函数进行 MAC 运算

假设 H：Keys(H)×Plaintexts→$\{0,1\}^L$ 是一个 HASH 函数簇，并且假设 F：$\{0，1\}^k \times \{0，1\}^l \rightarrow \{0,1\}^L$ 是一个 PRF，联合基于 MAC 的普通函数，密钥是一个串对 K_1，K_2，第一个子密钥是 H 的，第二个子密钥是 F 的。（分别称为 Hashing 和 Masking 密钥）。消息首先对 x 使用 H_k 被 HASH，这个值再用 $F_{k2}(S)$产生一个作为标签的值 б，现在是这个体制完整的描述，$UHM^{H, F}$ = （K，T，V）

Algorithm $T_{k1,k2}(M)$	Algorithm $V_{k1,k2}(M,б)$
x←$H_{K1}(M)$	x←$H_{K1}(M)$
б←$F_{k2}(x)$	б'←$F_{k2}(x)$
Returnб	If б=б'则返回 1 否则返回 0

引理 8.19、假设 H：Keys(H)×Plaintexts → $\{0,1\}^L$ 是一个函数簇，并且 A 是一个攻击者攻击一个消息认证体制，$UHM^{H, RAND\ l \rightarrow L}$ 。那么对任意 q，μ，具备：

$$Adv_{UHMh,Randl \rightarrow L}^{uf\text{-}cma} \leq q(q-1) \cdot Adv^{uh}H/2$$

定理 8.20、假设 H：Keys(H)×Plaintexts → $\{0,1\}^L$ 是一个函数簇，假设 F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 是一个 PRF，那么对任意 t, q，μ，应该有：

$$Adv_{UHMH, F}^{f\text{-}cma}（t,q, \mu）\leq q(q-1) \cdot Adv^{uh}H/2 + Adv_F^{prf}(t',q+1)$$

这里 t'= t + O(μ)

9.8.3 使用 XUH 函数构造 MAC

假设 H：Keys(H)×Plaintexts → $\{0,1\}^L$ 是一个函数簇，并且假设 F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 是一个 PRF，是用基于 MACs 的 XOR 普通函数进行合作，有两个这样的 MACs，一人描述和确定，另外是不固定和随机的。密钥是一个对子串 K_1，K_2，那么 H 的第一个子密钥，第二个是 F 的子密钥。（分别称为 Hashing 和 Masking 密钥）。在两个条件中，基本的范例是相同的，消息首先对 x 使用 H_k 被 HASH，这个值再用 $F_{k2}(S)$进行 XOR 加密产生一个值 τ，现在是这个体制完整的描述，因此 s 也允许查证。两个版本的不同在于 s 是怎样选择的，在计数版本，是一个计数器，随机版本是一个使用标签算法的随机数选择更新。

现在使用完整的描述是基于体制的计数版本，$C\text{-}UHM^{H, F}$ = （K，T，V）

Algorithm $T_{k1,k2}(M)$	Algorithm $V_{k1,k2}(M,б)$
X←$H_{k1}(M)$	分解 б 为（s，τ）
τ ←$F_{k2}(ctr, \tau) \oplus x$	x'←$F_{k2}(s) \oplus \tau$
б←(ctr, τ)	x ←$H_{k1}(M)$
ctr←ctr+1	If x=x' 那么返回 1，否则返回 0

返回 б

随机版本 R-UHM$^{H,\ F}$ = （K，T，V）是与以下描述相似：

Algorithm $T_{k1,k2}(M)$	Algorithm $V_{k1,k2}(M, б)$
$X \leftarrow H_{k1}(M)$	分解 б 为（s, τ ）
$r \leftarrow \{0,1\}^l$	$x' \leftarrow F_{k2}(s) \oplus \tau$
$\tau \leftarrow F_{k2}(r) \oplus x$	$x \leftarrow H_{k1}(M)$
$б \leftarrow (r, \tau)$	If x=x’ 那么返回 1 否则返回 0
返回 б	

引理 8.21、假设 H：Keys(H)×Plaintexts → $\{0,1\}^L$ 是一个函数簇，并且 A 是一个攻击者攻击消息认证体制，C-UHM$^{H,\ Randl \to L}$ ，那么对于任意的 q，μ，这里 $q<2^l$，有：

$$Adv_{C\text{-}UHMH,Randl \to L,A}^{uf\text{-}cma} \le Adv^{xuh}\ H$$

引理 8.21 证明：攻击者 A 产生一条 $T_{k1,k2}(\cdot)$序列 M_1，…，M_q 预言机，根据上述配置有这样的描述：

$$M_1 \Rightarrow б_1 = （s_1，\tau_1）$$
$$M_2 \Rightarrow б_2 = （s_2，\tau_2）$$
$$\cdots$$
$$M_q \Rightarrow б_q = （s_q，\tau_q）$$

这里，s_i = <i-1>是简单的计数值，并且$\tau_I = f(s_i) \oplus h(M_i)$，这里 $h=H_k$ 是 hash 函数例子，并且 $f=Rand_{k2}^{l \to L}$，是使用第二个密钥的随机函数说明，如下的选择消息攻击，A 输出一个对 M，б，这里 б=（s, τ ）。可以假设 $M \notin \{M_1，…，M_q\}$。如果 $V_{k1k2}(M, б)=1$，知道 A 可以成功的考虑。希望限制这个事件的概率上界。

假设 NEW 是 $s \notin \{ s_1，…，s_q\}$事件，并且 OLD 是补充事件，对于一些值 $i \in \{1,…,q\}$也就是 $s = s_i$ 。假设 P[·]表示为事件 “·” 的概率，在实验时，ForgeExp(C-UHM$^{H,Randl \to L}$,A) ，考虑：

$$P_1 = P[V_{K1,\ K2}（M，б）=1|Old]$$
$$P_1 = P[V_{K1,\ K2}（M，б）=1|New]$$
$$Q = P[New]$$

将使用如下的两个声明：

声明 1： $p_1 \le Adv^{uh}\ H$

声明 2： $p_2 \le 2^{-L}$

随后将证明这些声明。首先检查希望得到的结果：

$$Adv^{uf\text{-}cma}{}_{C\text{-}UHM}{}^{h,Randl \to L},_A = P[V_{K1,K2}(M,\sigma)=1]$$
$$= p_1 q + p_2(1-q)$$
$$\le Adv^{uh}H \cdot q + 2^{-L} \cdot (1-q)$$
$$\le Adv^{uh}H \cdot q + Adv^{uh}H \cdot (1-q)$$
$$\le Adv^{uh}H$$

第一行是后继概率的简单定义，第二行是条件概率。第三行使用了命题，第四行使用了命题 8.18。这个证明将会留到以后，而现在开始第二个声明。

声明 2 的证明：既然攻击者的询问是在函数 *f* 在 s 点被评论是没有结果的，值 *f*(s)是从

A 的点的同一分布的，或者使用随机函数的动态观点，可以想象到 f 得到询问的回答。既然标签预言机制，可以假设随机事件可以进行伪造。明显有：$P_2 = P[f(s)\oplus h(M)=\tau]=2^{-L}$
注意这里没有使用关于 Hash 的任何信息，这个命题是真的当 f 是随机的。
声明 2 的证明：

Adversary C
Initialize counter ctr to 0
For i=1,…,q do
$A \rightarrow M_i$
$\tau_i \leftarrow \{0,1\}^L$; $s_i \leftarrow$ <ctr>; $\sigma_i \leftarrow (s_i , \tau_i)$
$A \leftarrow \sigma_i$; ctr ← ctr +1
$A \rightarrow M, \sigma$
Parse σ as (s, τ)
If $s \notin \{s_1 ,\ldots, s_q\}$ then FAIL
Else 假设 i 是使 $s=s_i$
假设 $b \leftarrow \tau_i \oplus \tau$，返回 M，$M_i$,b

可以声明 $Adv^{uh}H$，$C = p_1$

定理 8.22 、假设 H： $Keys(H) \times Plaintexts \rightarrow \{0,1\}^L$ 是一个函数簇，假设 F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^L$ 是一个 PRF，那么对于任何 p,q,μ，有：

$$Adv^{uf\text{-}cma}{}_{C\text{-}UHM}{}^{H,F}(t,q,\mu) \le Adv^{uh}H + Adv_F{}^{prf}(t',q+1)$$

这里 $t'= t + O(t)$

9.9 用密码 hash 函数的 Macing

最近引起了一个在 Macing 使用中的诸如 MD5，SHA 的 HASH 函数兴趣，容易明白是为什么，通俗的 HASH 函数例如 MD5，使用软件运算，SHA-1 比基于分组密码的 HASH 速度快，这些软件实现是容易的并且随便使用的，函数没有分 USA 或其国家的出口限制。更困难的问题是怎样把它做到最好。这些 HASH 函数不是原来使用的消息认证。（许多困难之一是 HASH 函数不是简单的键入的，不自然的调节秘密密钥的概念），因此特别的观点是在结束前必须使用。

一个变化的类型是已提出并且分析了。（见 Tsudik[195]，对于一个早期的这种构造的描述，并且与接入的互联协议是相互适应的。Preneel 和 Van Oorschot[163,162]是观察存在的构造和指出一些其他的特点和弱点，特别的，呈现了一种迭代函数的详细的生日攻击。特别的，呈现出了一种启发式的构造，MDx-MAC，基于这些发现。Kaliski 和 Robshaw [115]讨论并比较了不同的构造，在[194，11]中进行了讨论）。 最近，一个构造似乎被广泛接受了，这是 HMAC[18]的构造，特别，HMAC 最近被选为互联网协议的身份认证，对于这个目的的描述可以从 RFC 中查询。

9.9.1 HMAC 构造

假设 H 是 HASH 函数，对于简单的描述，可以假设 H 是 MD5 或 SHA-1，然而构造和

分析可以使用其他的函数，并且提出 l 比特输出，（l=128MD5 比特 l=160SHA-1），假设文本表示数据描述 MAC 数据是实用的，并且假设 K 是双方共享的消息认证密钥。（HASH 分组应该不会大于 64 字节串，如果更短，0 作为填充码）进一步如下定义两个固定并且不同的 64 字节串 IPAD 和 OPAD，（i 和 O 是输入和输出）：

IPAD=0X36 字节重复 64 次

OPAD=0X5C 字节重复 64 次

函数 HMAC 获得密钥 K 和文本，并且产生 $HMAC_K$ （Text）=H（K⊕IPAD，文本））

也就是

（1） 在 K 的结尾填充 0 产生 64 字节串

（2） XOR（面向比特异或）64byte 串，在（1）计算 IPAD

（3） 从（2）的结果产生的串填充数据流到 64 字节

（4） 在（3）中产生 H 数据流

（5） 使用 OPAD 计算 XOR（面向比特异或）64byte 串，在（1）中产生的数据流

（6） 从（5）中产生的结果添加 H 得到串

（7） 应用（6）H 到串数据产生输出结果

密钥的推荐长度是至少 l 比特，一个更长的密钥并没有加强函数的安全性，如果密钥随机性的安全函数被认为是弱的，尽管也许是明智的，密钥随机性是弱的。HMAC 随意使用输出的 80 比特。

作为一个结果，得到一个简单并有效的构造。认证一个流文本的开销是接近对序列流做 HASH 的开销是相同的，这使得使用 H 的图书馆代码更容易，并且使之容易产生一个特别的 HASH，例如 MD5，还有一些例如 SHA-1，需要使用这个原因。

9.9.2 HMAC 的安全

HMAC 的优势是下列 HASH 函数的强度是给定一些合理的假设安全是正规的。

HASH 函数的安全假设不是太强，既然对于足够的反对是目前产生的 MD5，SHA（特别的，现在已知 MD5 不能抗碰撞攻击，随后讨论 MD5 的情况），事实上，HASH 的安全特征越弱，合成的 MAC 构造就越强。

构造假设反映了更标准的 HASH 函数现有用处，要求的用途在于主要是一种特定的弱碰撞并且没有预期限制。这说明如果 HASH 函数具备这些特征，则 MAC 是安全的。如果 HASH 是失败的，则唯一 MAC 是失败的。[18]的分析是作为压缩函数（Compress Function, CF），应用 HASH 作为迭代函数，根据 Merkle 和 Damg_ard [71]。（在这个结构一个 l 比特初始变量 IV 是固定的，并且 H 在文字空间 x 的输出是把 x 分为每组 512 比特进行计算的，并且使用 CF 进行 HASH，用一个简单的方式说明，读者可以发现许多地方有类似的描述[125]）。简而言之[18]攻击者可以使用同样的努力伪造 HMAC 的（时间和搜集的信息），破解同样的体制使用如下任一种方式：

（1）攻击者发现 HASH 函数的碰撞当 IV 是随机和保密的

（2）攻击者可以计算一个压缩函数的输出当一个 IV 对于攻击者是随机的、保密的和未知的。（也就是说攻击者成功伪造压缩函数的秘密密钥并且当作一个 MAC 的固定长度消息）

这些攻击的灵活变化在于关于这些函数密码强度与最基本的一些假设不同。上述第一种攻击成功发现碰撞，密码学单向函数的设计的目的在于防止碰撞的发生。因此，可以经常假设碰撞难以发生。但是事实上，更多真实情况是：上述第一种攻击成功难度大于在 HASH 中找到碰撞。因为当 IV 是秘密的时候，HASH 的碰撞大大的难于 IV 已知时候。这是因为

前者要求交叉使用函数用户的空闲时间，（为了从函数产生输入/输出对）并且不允许并行传统的生日攻击。因此，甚至当 HASH 函数在传统的观念中是完全没有碰撞的，体制是安全的。在密钥产生和随机数产生的使用中，一些 HASH 被假设是随机的。（例如 SHA 的设计者建议 SHA 用于这种使用方法[85]）。函数的随机数也用于一种设计方法，获得抗碰撞设计。上述第二种设计的成功暗示这些 HASH 函数的随机特征是很贫乏的。

重要的是意识到这些结果是被简单的状态假设和一个简单的分析作为指导。事实上，这个构造甚至比分析指示更强。在某种程度上，甚至 HASH 函数没有满足状态假设，这个体制也许是安全的。例如，甚至抗弱碰撞的特征是技术含量非常高的，因为事实上，这样的构造中，攻击者必须发现密钥函数的碰撞，没有看到任何函数的输出，这是最困难的。

随后的评价是与最近在 MD5[74]的碰撞相关的。当这些攻击不能适应 MD5 的弱碰撞攻击，甚至使用 MD5 是不能导致 HMAC 的破解。

9.9.3 抗已知攻击

如同在[169，19]展示的，生日攻击，也就是基于密码学 HASH 函数的碰撞发现。可以应用攻击基于迭代函数加密 MAC 体制，（包括 CBC-MAC 体制）。这些攻击对大多数基于 HASH 的提出的 MACs 构造。特别的，可以构造众所周知的假冒 HMAC 构造的构造。考虑这些攻击是重要的，既然强烈建议提高单纯的穷尽搜索攻击。然而，给定典型的 HASH 长度 128，160 抗这些函数实际的免疫度是可忽略的。事实上，在大约 2^N 的空间上，这些攻击要求 MAC 值集合（这里 l 是 HASH 输出的长度）。对于 $l \geq 128$ 比特的值，攻击变得不容易了。与 HASH 减少密钥的生日攻击相反，新的攻击方法要求在一个巨大的数据空间对密钥拥有者产生 MAC 值，并且不能进行并行计算。例如，当使用 MD5，这种攻击可能要求 2^{64} 分组（或 2^{73} 比特）数据空间使用相同的密钥。在 1Gbit/sec 通信链接中，需要 250，000 年来处理这种攻击要求的所有攻击。这对于生日攻击有剧烈的冲突，在减少密钥 HASH 函数允许远远有效并且接近实际攻击[202]。

9.10 MACs 的最小假设

随着其他的私钥密码，安全消息认证的体制是与现存的单向函数是相等的。从定理 5.19、定理 8.6，单向函数产生消息认证体制如下。其他的方向是[114]。简而言之，

定理 8.23、消息空间$\{0，1\}^*$有一个安全消息认证体制，当且仅当存在一个单向函数。

9.11 问题和练习

难题 8.24、假设 F：$\{0，1\}^k \times \{0，1\}^l \rightarrow \{0，1\}^l$ 是一个 PRF，基于 F 的 CBC MAC 模式是消息认证体制 MA，标签和可确定的算法如下：

算法 T_K （$x_1,\ldots,x_n$）	算法 V_K （$x_1,\ldots,x_n$, б）
$Y_0 \leftarrow 0^l$	$Y_0 \leftarrow 0^l$
For i=1,…,n do $y_i \leftarrow F_K(y_{i-1} \oplus x_i)$	For i=1,…,n do $y_i \leftarrow F_K(y_{i-1} \oplus x_i)$
Return y_n	If y_n =б 那么返回 1 否则返回 0

假设消息空间是串 x 的集合，长度是 l 比特的倍数，（意味着在上述的变化中消息分组 n 的

数据）。可以看到在这个消息空间上的体制是不安全的。也就是出现一个攻击者 A 用时间 t 进行攻击，产生一个 q 询问预言机制，这些总共是 μ 比特，并且得到 $Adv_{MA,A}^{uf\text{-}cma}$ =1，这里 t,q, μ 是一些将要进行阐述的小的值。

第十章 数字签名

数字签名的概念在于可以证明最基础的部分和现代密码学有用的发明。一个签名体制提供了一个每个用户的途径进行消息签名，因此签名会随后对任何一个人进行证明。更特殊的是，只有产生用户签名（使用私钥）才能确认消息签名（使用公钥）。确认者认为可以使自己更方便，从消息被签名之后，消息目录被改变了。还有，签名者不能评判已签名的消息，因为只有签名者拥有私钥。

通过纸上谈兵，一个人可以在信封上进行字母签名和封铅，使用接收者的公钥进行签名。接收者可以分别使用自己的私钥和发送者的公钥执行打开信封和使用自己私钥进行逆操作。这些公钥技术的电子邮件在现在已广泛使用了。

如果公钥的目录在网络上进行访问控制，一个人需要从目录中可以获取公钥发送欺骗信息。一个文明的解决方式是使用一个证书：一个用户目录管理员或其可信方的公钥数字签名。如果用户 A 保持本地的一个目录管理员的公钥拷贝，可以使用所有有效的公钥签名进行通信，避免使用假密钥进行欺骗。此外，每个用户可以用公钥传送证书，使私钥签名的数据可以正常发送，从一个中心目录移动，允许一个验证消息签名对于没有信息甚至超过目录的公钥管理，一些协议条款有关这种组网方式，在这些注记中，这一节进行讨论。

10.1 数字签名的成分

一个在公钥框架下的数字签名体制，定义为一个三变元的函数（G，σ，V），因此：

•密钥产生算法是概率的，基于输入一个安全参数 1^K 多项式时间算法，产品对（P，S），这里 P 是公钥，S 是私钥。（使用符号（P，S）∈G（1^K）指对子（P，S）产生算法 G）

• 签名算法 σ 是一个概率多项式时间算法，给定安全参数算法 1^K ，在范围 G（1^K）内的一个秘密密钥 S，一个消息 m∈{0，1}^{K} ，并且产品作为输出串 s，称为 m 的签名。（使用概念 s∈σ(1^K,S,m)如果签名算法是概率的并且 s=σ(1^K,S,m)意味着 s 是消息 m 的签名。

• 验证算法 V 是一个概率多项式时间算法，给定一个公钥 P，一个消息 m 的数字签名 s，当上下文关系是清楚的。• 数字签名体制的最后特征是抗概率多项式伪造攻击安全的，随后延迟这个定义。

注意 V 是概率的，可以放松对 V 的要求对所有消息 m，接受有效的签名拒绝无效的签名，所有足够大的安全签名参数 K，并且所有密钥对（P，S）∈G（1^K）。概率是 V 和 S 的随机事件，签名消息是明文或加密消息，因为数字签名体制的消息空间是{0，1}* 的子集。

10.2 数字签名：陷门函数模型

DEFFIE 和 HELLMAN 提出了基于陷门函数模型的公钥密码体制（G，E，D），用户 A 签属任意消息 M，作为数字签名 D（M）=f^1(M)到 M，这里 f 是一个 A 的陷门公钥函数，A 可以已知相应的通信函数。任何人可以通过在公钥目录中查找 A 的公钥而检测通信的真实性，因此 E（D（M））=$f(f^1(M))$=M。注意，如果消息改变，这个签名失效，因此 A 是受保护的，在消息签署之后防止调整，检测签名的人是 A。

因此，在的原始提议中，DEFFIE 和 HELLMAN 与加密和数字签名两种功能有关，然而，作为研究者应该区分这两种功能。应该看到一些密码体制适合加密不适合签名，许多设

计仅仅针对高强度的签名体制设计的。

RSA 公钥体制落在 DEFFIE 和 HELLMAN 的范式中进行，允许一个以直接的方式进行数字签名。私钥指数 d 变成签名指数并且一个消息 Mi 的签名是 DEFFIE、HELLMAN 的范式 s 是 $M^d \bmod n$ 的值。任何人可以证实这个签名是有效的，使用相应的公钥认证指数 e，检测身份 $M=(M^d)^m \bmod n$ 。如果这个等式成立，那么通信签名指数 d 是 A 的签名 M^d 必须从 M 中产生，（实际上，可能是相反的，消息 M 可以从签名 M^d 中计算，使用可证实的等式和公共指数 e，然而，这样的消息似乎是不具备智能化的。实际上，这个难题是容易被避免的，用签名 $f(M)$代替 M, 这里 f 是一个标准的公共单向函数。

在数字签名体制的符号中，DEFFIE-HELLMAN 提议是如下的三变元算法（G，σ，V）：

- 密钥产生：G（1^K）从 F 选择一对（f_i, t_i），这里 $i \in I \cap \{0,1\}^K$
- 签名算法：σ（1^K，f_i, t_i, m）输出 $f_i^{-1}(m)$
- 验证算法：V（f_i, s, m）输出 1，如果 $f_i(s)=m$ 并且否则为 0

考虑这个和其提议的安全性，首先定义数字签名的安全性。

10.3 为安全签名体制定义和证明安全

一个数字签名体制的理论是从 Goldwasser、Micali 和 Yao[107]并且继续和[105，14，152，177，78]。

10.3.1 数字签名的攻击方法

区别三种基本的攻击，按照难度列出：

- 唯密钥攻击：在这种攻击中，攻击者知道签名者的公钥，因此，仅仅有能力检测给定的消息是否是伪造的。
- 已知签名攻击：攻击者知道签名者的公共密钥，并且看到了是合法用户签署的消息和签名对。事实上，这是一个攻击者可以做到的最小的努力。
- 选择消息攻击：攻击者允许询问选择攻击消息的合法签名者的签名，例如，一人可以判断是谁根据要求进行了签名。

对于攻击者可以选择的更好的子集，详细见[107]。

成功的进行签名伪造意味着什么？

对攻击成功也有几种不同的程度，为了增加攻击者成功的可能：

- 存在伪造：攻击者成功伪造一个消息的签名，的选择是不必要的。
- 选择伪造：攻击者成功对选择的消息进行伪造。
- 普遍伪造：攻击者虽然不能发现伪造者的密钥，能够伪造任意的签名。
- 全面破解：攻击者可以计算签名者的秘密密钥。

明显的，安全的不同级别也学要求不同的应用，有时，在选择伪造时，攻击者不能成功，可以看到，即使可以进行签名攻击。有时，可以看到一个在选择伪造时具备有能力进行中间人攻击，在其方面（例如攻击者是公证人或是一个返回调制机制），也许要求一个攻击者具备选择签名攻击的能力不能成功，甚至在不可忽略概率条件下存在的相关攻击。可以聚集的安全在于在这些注记中，一个多项式时间攻击者在选择消息攻击时，使用高概率甚至不能进行个种形式的伪造。

可以说数字签名是安全的，如果一个敌人可以使用真实的签名作为随机预言机制，对于任何消息进行伪造，在多项式时间里按照公钥尺寸，从真实的签名者那边得到签名。正式的，

假设 B 是一个黑盒子，把消息映射到一个无效签名，对于所有的消息，V（P，B（m）,m）=1。假设伪造函数 F 在输入公钥 P 的条件下可以访问 B，定义为 F^B（P）。伪造算法运行两步，首先完成选择消息攻击，然后输出一个新的伪造签名，可以定义成为任意消息-签名对，使得在签名之前，消息不能被签署。要求对所有伪造算法 F，所有的多项式 Q，对所有足够大的 k，Prob（v(P,s,m)）=1,(P,S)←G(1^K);(m,s) ←F^B（P）≤1/Q(k)。概率是在秘钥选择（P，S）∈G（1^k），伪造算法 F 是随机事件，事件时 B。DEFFIE，HELLMAN 的原始申请并不满足这些严格的安全要求。可能在公共信息部分存在伪造，随机选择一个 s，产生 m=*f*(s)，现在 s 是 m 的一个有效签名。

提出了许多数字签名体制，一个完成的单子在[107]中列出。

这里检查三个体制的安全。

10.3.2 RSA 数字签名体制

首先例子是基于 RSA 加密体制的。

公钥是一个数字对(n,e)，这里 n 是两个大素数的乘积，e 是与 Φ（n）互素的数，秘密密钥是 d，因此 ed≡1modΦ（n）。签名是计算 б（m）= m^d mod n ，验证是使用 e 阶签名，并且与原始消息进行比较。

命题 9.1、在选择消息攻击下，RSA 是可以伪造的（或者在已知消息攻击时存在伪造）

证明：如果可以产生两个消息的签名，两个消息乘积的签名是两个消息的签名的乘积。假设 m_1,m_2 是两个消息，对这些消息使用黑盒子产生签名：б（m_1）= m_1^d mod n，б（m_2）= m_2^d mod n，现在可以产生这两个消息的乘积的签名： б （m_1 m_2）=（ m_2m_1）^{d} = m_1^d m_2^d =б（m_1）б（m2）mod n

对一个消息 m 产生一个签名，开始选择一个随机数，r∈$2n^*$ ，现在定义 m_1,m_2 如下：（m_1）= mr mod n ，（m_2）= r^{-1} mod n , 使用上述的策略，可以对这些消息的乘积产生一个签名，原始消息 m，m_1 m_2 = （mr）r^{-1} = m。

10.3.3 El Gamal 体制

数字签名机制依赖于 DIEFFIE-HELLMAN 密钥交换难题的解决难度，也就是离散对数问题。DHKE 难题是基于输入是一个素数 p，一个生成元 g，并且 g^y,g^x∈Z_p^*，计算输出 g^{xy} mod p ，目前已知的最好的解决 DHKE 的方法是解决离散对数难题，是否计算一个离散对数是与 DIEFFIE-HELLMAN 难度相当的一个题目。如下的数字签名体制是概率的，的一个接近的变量叫做 DSS，被赋予了一个国家标准。

这个体制如下：

- 公钥：一个三变元（y,p,g），这里，y=g^x mod p , p 是素数并且 g 是 Z_p^*一个生成元。
- 秘密密钥：x,使得 y=g^x mod p
- 签名：消息 m 的签名是一个对（r,s），使得 0≠r,s≠p-1 并且 g^m = $y^r r^s$ mod p
- 校验：检查 g^m = $y^r r^s$ mod p 实际上是成立的

为了产生一个对（r,s），建立一个签名，签名者随机选择一个数字 k，使得 0≠k≠p-1 并且 GCD（k,p-1）=1，假设 r = g^k (mod p)。现在希望计算 s，使得：

g^m = $y^r r^s$ mod p = g^{xr+ks} mod p 。指数的阶的关系是 m=xr+ks(mod p-1)，因此，

s=（m-xr）$^{k-1}$ mod p-1，m 的签名的对是（r,s）。

明显的，如果一个攻击者可以解决一个离散对数难题，可以从公共信息的秘密密钥 x

完全计算出知道计算出整个密码体制。除此而外，如果一个攻击者为了一个消息找到 k，则可以解决离散对数难题，因此伪随机数产生的 k 有更好的性质。

命题 9.2、在存在已知消息攻击时，这个体制是存在伪造攻击的。

练习：

注意一个基于离散对数的密钥交换协议：这是非常有趣的，对于两个没有用离散对数产生陷门函数的人进行密钥交换是可能的，可以通过 A 和 B 进行协商得到一个素数 p 和一个生成元 g，A 选择一个秘密数字 x，并且发送 g^x(modp)给 B，B 选择一个秘密数字 y，发送 g^y(modp)给 A。现在每个用户可以计算 g^{xy}(modp)。假设这样可以共享密钥，这个体制的计算 xy 难度是与 DLP 的难度相当的。

10.3.4 Rabin 体制

Rabin[164]提出了一个为消息 M 签名的方案，本质上是模 n 的两个大素数 M 的平方根。既然获得平方根是与计算 n 的因数等概率的，一个攻击者不能伪造任何签名，除非能分解 n 。为了目的，假设考虑一个变量，当 n=pq，p=q=3 mod 4，因此签名被唯一确定了。

这个讨论假设攻击者只有访问签名者包含模 n 的公钥，一个敌人可以使用一个活动的方法破解这个体制：要求真正的签名者签署 $M=x^2 \bmod n$，这里 x 是随机选择的。如果签名者同意并且产生一个 M 的平方根 y，有一半的机会 gcd(n,x-y)将产生一个 n 的非平凡的因子，签署者因此与所拥有的秘密密钥矛盾。尽管 Rabin 体制提出了解决这一难题的一些实际技术，拥有建立伪造因素的归约构造条件。

现在来看一看这些细节：

这个数字签名体制是基于模 a 平方根的计算复杂数

- 公钥：n=pq
- 私钥：素数 p,q
- 签名：$s=\sqrt{m} \bmod n$（假设 WLOG 并且 m 是平方的）
- 校验：检查 $s^2 = m \bmod n$

命题 9.3、这个体制对于仅知密钥攻击是可伪造的。

证明：选择签名并且平方产生一个相应的消息

命题 9.4、这个系统在选择消息攻击时是完全可破的。

证明：如果找到一个消息的不同的平方根，就可以进行模数分解。选择一个值 s 并且假设 $m=s^2$，现在 s 是 m 的一个有效签名。把 m 放入黑盒子，有两个选择将产生同样的签名。如果这样，重复这个过程，如果不是，有两个 m 的平方根，可以恢复 n 的因子。当破解等价于因子分解时是安全的

给出 Rabin 体制在选择消息攻击的不安全，也许可以看到在因子分解时存在不安全的数字签名，也就是说，一个体制是：

- 破解等价于因子分解
- 抗选择消息攻击，签名体制是安全的

反证：在假设（1）时，（2）是不可能的。既然第一个陈述是破解体制是等价与因子分解的，对于一个合数 n，有如下的归约：

- 产生一个公钥 P
- 产生一个消息 m
- 产生一个有效签名 $s \in \sigma(P,m)$，使用破解算法（在多项式时间内重复这三项）
- 分解 n

总结在选择消息攻击时，体制必须是不安全的，既然可以在第 3 步用破解代替 CMA。QED 这个争论是错误的？首先有一个公共信息 P 的有效的定义，不需要包括数字 n，第二，CMA 将使用固定公共信息的方法产生签名，因此上述的归约在每次攻击者的呼叫中使用不同的公共信息是必要的。

10.4 概率签名

概率签名技术也使用在数字签名标准中。这个进展是 GOLDWASSER、MICALI、YAO[107]在基于因子分解难题的签名体制中是预先证明了的。在 RSA 的逆函数中，对攻击者的扩展伪造是使用了一个已知签名攻击。

Goldwasser、Micali 和 Rivest[107]已加强了这个结果，提供一个签名结果，在选择消息攻击中可以进行扩展伪造攻击。这个体制基于因子分解难题，更普通是存在陷门置换的缺陷，（也就是说，陷门的 f_0,f_1 对有一个共同的定义域，这个体制难于发现发现 x,y 使得：$f_0(x)=f_1(y)$）。

这个体制，作为原始的描述，尽管在理论上是吸引人的，但是非常没有效果。然而，可以调节允许更多的压缩签名，在签名者的公钥和私钥之间，对于内存没有用，甚至对于每个新的签名可以使用新的随机选择进行改变。特别的，Goldriche[95]提出建议使得基于因子分解的体制更实际的，当保持安全体制。

Bellare 和 Micali[14]已显示了一个数字签名，该体制的安全是基于现存的一些陷门置换的（一个减弱的条件是存在可忽略的要求）。那么 Naor 和 Yung[152]是已展示了怎样使用一个单向陷门函数设计一个数字签名体制，在选择签名攻击时抗扩展伪造是安全的。最后，Rompel[177]展示了怎样给定单向函数进行签名。这些工作建立在基于 Lamport 怎样在[130]中签署一比特的观点。这个观点如下：如果 f 是一个单向函数，并且 Alice 公开了两个数 $f(x_0)=y_0$，并且 $f(x_1)=y_1$，那么，她可以通过公布消息 x_0 签署 0，并且可以通过公布消息 x_1 签署 1。Merkle[146]介绍了一些基本观点的扩展，包括构建一个认证值的树，它的根储存在签名者的公钥中。

10.4.1 陷门置换的无缺陷

介绍了无缺陷陷门置换概念，并且构造一个签名体制，假设存在一个置换的无缺陷对。

定义 9.5[无缺陷]、假设 f_0，f_1 是共同的定义域 D 上的置换，如果 f_0（x）= f_1 (y)=z，那么，（x,y,z）是无缺陷的。

定义 9.6[一个无缺陷的置换函数簇]：一个簇 $F=\{f_{0,i}, \ f_{1,I} : D_i \rightarrow D_i\}_{i\in I}$，称做一个无缺陷的陷门置换簇，如果：

1. 存在一个算法 G，使得 G（1^K）输出两个对（f_0 , t_0）和（f_1 ,t_1）这里 t_i 是 f_i 的陷门信息。
2. 存在一个 PPT 算法，给定 f_i 并且 $x\in D_i$ ，计算 f_i （x）
3. 任意可逆算法 I，存在一些可忽略的函数 V_I ，使得对于足够大的 k，Prob（$f_0(x)=f_1(y)=z$:（f_0 , t_0），（f_1 ,t_1）←G(1^K）;(x,y,z) ←I(f_0 , f_1)）<V_I (K)

如下的观察可以看到，存在一个陷门置换并不直接暗示存在一个无缺陷的置换。例如，定义一个 RSA 的置换簇，$F_{0,n}(x) \equiv x^3 \bmod n$ ， $F_{1,n}(x) \equiv x^5 \bmod n$。(gcd(x,n)=1,并且 gcd (15,Φ(n))=1)，既然两个函数相关，容易通过随机选择 w 产生一个缺陷函数，并且定义 x=$f_{1,n}$（w），y= $f_{0,n}$ （w）并且：Z= $f_{1,n}$ （y）= $f_{0,n}$ （x）= $x^{15} \bmod n$，通常，如下的问题是：

公开问题 9.7、 一个陷门函数置换的存在是否暗示一个无缺陷的陷门置换的存在？

上述问题的逆显然是真实的，给定一个无缺陷的置换产生器，容易产生一个陷门置换。如果{ f_0 , f_1}是在公共定义域上的无缺陷对，也就是说，找到满足 Z= $f_{1,n}$ （y）= $f_{0,n}$ （x）对（x,y,z）是计算不可能的，那么（f_0 , f_0^{-1}）是陷门。（否则，给定逆算法 I，z = f_1(y);Z 在 D 上是分布非正规的，并且是不可忽略的概率，I 可以产生 x= f_0^{-1} (z)。因此（x,y,z）是一个缺陷，矛盾。

10.4.2 例子：如果是因子分解难题，无缺陷置换存在

假设 n=pq，这里 p,q 是素数（p,q∈H_k）并且 p≡3 mod 8 ,q≡7 mod 8, 偶素数对的 1/16 适合这个要求，假设 QR_n 赋给了模 n 立方剩余的集。

首先注意到：

1.（J_n(-1) = +1, 但是 -1 ∉ QR_n

2.（J_n(2) = -1, （并且 2 ∉ QR_n ）

3．X∈QR_n 有一个平方根 y ∈ QR_n (x 是一个 Blum 整数)，通常是四个平方根 y,-y,w,-w。根 w,-w 有 Jacobi 符号 -1，y 并且-y 有 Jacobi 符号+1。

现在定义一个函数对簇，并且证明，假设因子分解的相互性，也就是说在 QR_n 的无缺陷陷门函数簇。

定义，对于 X∈QR_n ： $f_{0,n}$ （x）= x^2 mod n $f_{1,n}$ （x） = $4x^2$ mod n

从上述注记中，函数 $f_{1,n}$ ， $f_{0,n}$ 是 QR_n 的置换。

命题：{ $f_{1,n}$ ， $f_{0,n}$}是无缺陷的。

证明：假设这个对是无缺陷的，假设 x,y∈QR_n 满足：$X^2 \equiv 4y^2$ mod n

这个暗示着（x-2y）(x+2y)≡0 mod n，然而检查两边的 Jacobi 符号有：

（J_n(x)）=+1 (J_n(2y))=(y/n)(2/n)= -1 (J_n(-2y))=(-1/n)= -1

也就是说，x 是一个平方剩余，但是±2y 不是。既然 x≠±2y mod n ,gcd(x±2y，n)将产生一个 n 的非平凡因子。

10.4.3 怎样签属一比特

首先描述签名体制中最基本的构造：签属一比特。

假设 D 是公共定义域的无缺陷对{f_0,f_1}，并且假设 x 是 D 中随机选择的。

Public	Secret
X∈D, f_0,f_1	f_0^{-1} ,f_1^{-1}

签属比特 b∈{0,1}，假设 s=б(b)= f_b^{-1} (x)

证实签名 s，检查 f_b（x）=x

命题 9.8、上述体制是抗选择消息攻击扩展安全的。

证明：假设，反证法，这个体制是不安全的。也就是说，存在一个伪造算法，F^{CMA}（P），可以伪造签名（给定公共信息），F 要求 b 的签名，并且任意多项式 Q 和无限 k 多）可以正确的签属 $\bar{b}$ ，以概率>1/Q（k）。为了得到反证结果，签属一个算法给定 F^{CMA}，可以得到缺陷：输入 f_0,f_1

输出 x,y,z，使得 Z= f_1,（y）= f_0（x）（以概率>1/Q（k））

（1） 随机选择 x∈D，投币列入公共文件：z=f_{coin}(x) ∈D，f_0, f_1 。（注意 f_0,f_1 是置换，因此，z 是 D 上的特例）

（2） 运行一个算法 F^{CMA}（P）：

1. F 要求签名 b = $\overline{coin}$，返回（1）
2. F 要求签名 b= coin, 回答 x = f_b^{-1} (f_{coin}(x))

（3） 根据假设，F 可以产生一个签名$\bar{b}$，y= $f_{\bar{b}}^{-1}$(f_{coin}(x)),例如，z=f_b(x)= (y)

10.4.4 怎样签属一个消息

与从前一样，D 在公共域上是一个无缺陷对（f_0,f_1），并且在 D 内可以随机选择 x。

Public	Secret
X∈D, f_0,f_1	f_0^{-1} ,f_1^{-1}

对于 x∈D，签署第一个消息 m^1 , S^1=б(m^1)=f_{m1}^{-1}(x)

通过验证：V（s^1,m^1）=$\begin{cases} 1 iff_{m1}(s^1)=x \\ 0否则 \end{cases}$

这里对于 m^1=$m_1^1m_2^1$…m_k^1 :

$F_{m1}^{-1}(x)=f_{mk1}^{-1}(\ldots(f_{m1}^{-1}(f_{m11}^{-1}(x))))$

$F_{m1}(x)=f_{mk1}(\ldots(f_{m1}(f_{m11}(x))))$

明显的，f_m 是 D 上的置换，并且是容易计算的。为了签署下一个消息 m^2，在上述的签名中使用一个新的置换：S^2 = б（m^2）=(f_{m2}^{-1}(s^1),m^1)

并且区分：V（s^2,m^2）= $\begin{cases} 1 iff_{m2}(s^2)=s^1 并且 f_{m1}(s^1)=x \\ 0否则 \end{cases}$

注记：

1、 使用这个体制，签名长度是数字消息的快速线性增加。
2、 在消息的尾部，很容易伪造签名。因此假设预处理声称没有消息剩余的编码体制（例如没有剩余消息作为其消息的内容）。

命题：在对已知消息攻击时，这个体制不是扩展安全的。

证明：假设存在 F（H，P）使得任意多项式 Q 和足够大的 k，给定公共信息 P，并且 H=（(m_1，б（m_1)），…，(m_l ，б（m_l)），对于使用运算 M（1^K）选择消息 m_1，m_2，…，m_l，可以找到一个消息$\bar{m}$≠m_i，（1≤i≤l），可以产生一个签名 б （$\bar{m}$）因此：

$$Prov\{V(\bar{m}),\bar{m}\}=1\}>1/Q(k)$$

这里概率是所有公共文件和 F 的随机事件。

现在签署一个算法 A，使用 F 得到一个缺陷：

输入 f_0,f_1

输出 a,b,c，因此，f_0(a)=f_1(b)=c(使用概率>1/Q(k))

(1) 选择 m_1m_2…m_i ∈ M(1^k)，x∈$_R$ D。假设 z=f_{ml}(…(f_{m1}(x)))，假设 P={f_0,f_1,x}是一个公

共文件。（注意 z 也是在 D 中均匀选择的）

(2) 产生 H=（（m_1，f_{m1}（z））, …,（m_l ，(f_{ml}(…f_{ml}（z））)），定义 $m=m_1 \cdot m_2 \cdot \ldots \cdot m_l$，所有消息的串产生。

(3) 选择伪造算法 F（H，P）产生（$\overline{m}$，б（$\overline{m}$））

(4) 与没有偏差的概率，б（$\overline{m}$）是一个有效签名，也就是说“来回散步”，从 б（$\overline{m}$）到 f_m。根据提供的历史数据，会得到 x，因此必须从 б（m_i）得到反馈。假设 l 是两条途径交叉的点，也就是说，m 是 $\overline{m}$ 的前 l-1 bit，并且定义 $w=f_{ml-1}^{-1}(\ldots(f_{m0}^{-1}(z)))$。

假设 w.l.o.g 是 m 的第 l 比特是 0，$\overline{m}$ 的第 i 比特是 1，假设 u,v 是相应的 $f_0^{-1}(w)$，$f_1^{-1}(w)$，输出是（u,v,w）。

明显的，（u,v,w）是一个缺陷，因此，提供一个公共数字 f_0,f_1 伪造算法 F 的输出以可忽略的概率，导致一个缺陷，矛盾。

然而，这个体制抗选择消息攻击是不安全的，至少，还不知道怎样证明这点，下一节回调整证明这点。

10.4.5 基于无缺陷的安全签名体制

假设 D_f 是无缺陷置换的公共域，考虑如下的体制，对于签名消息 $m_i \in \{0,1\}^k$ ，这里 $i \in \{1,\ldots,B(k)\}$并且 B（k）是 K 上的多项式。

选择两个没有缺陷的置换对（f_0,f_1）并且（g_0,g_1），这里 f_0^{-1},f_1^{-1}，g_0^{-1},g_1^{-1} 。选择 $X \in D_f$，假设公共密钥包括 D_f，X，f_0,f_1，g_0,g_1，并且假设秘密密钥包括 f_0^{-1},f_1^{-1}，g_0^{-1},g_1^{-1}，

PK	SK
$X \in D, f_0,f_1$	f_0^{-1},f_1^{-1}
g_0,g_1	g_0^{-1},g_1^{-1}

假设○是连接函数，并且设置 $H_1 = \Phi$。签署 m_i ，对于 $i \in \{1,\ldots,B(k)\}$:

1、 均匀选择 $R_i \in D_g$

2、 设置 $Z_1^i = f_{Hi \cdot Ri}^{-1}(X)$

3、 设置 $Z_2^i = g_{mi}^{-1}(R_i)$

4、 设置签名 б（m_i）=(Z_1^i,Z_2^i,H_i)

5、 设置 $H_{i+1} = H_i \cdot R_i$

为了验证一个消息签名对（m,s），这里 s=（z_1,z_2,H），

1、 假设 $R=g_m(z_2)$

2、 检查 $f_{H \cdot R}(z_1)=X$

如果是这样，那么签名是有效的并且验证函数 V（m,s）=1，否则，V（m,s）=0。这个体制采用实际优势，一个新的随机元素 Z_1^i 可以用每条消息代替 X 的使用。因此，伪造不能获得信息，对于消息的多项式得到签名。显然签名和验证过程是要求在多项式时间执行的，如下的定理证明了安全

定理 9.9、无缺陷置换签名体制是抗 CMA 扩展安全的，如果无缺陷置换存在。

证明：（反证法）假设不是定理描述。那么有一个伪造 F^{CMA}（f_0,f_1，g_0,g_1，X）组成如下两步：

1、 F 包括签名 б（m_i）对于 B（K）得到消息 m_i 作为选择

2、 F 输出（$\overline{m}, \overline{s}$）这里 $\overline{s}$ =（$\overline{z_1}, \overline{z_2}, \overline{H}$）因此 $\overline{m}$ 与第一步所有要求的 m_i ，并且 V（$\overline{m}, \overline{s}$）=1

可以看到如果 F 确实存在，那么有一个 PRM A，满足：

输入：均匀选择（h_0,h_1,D_h）无缺陷，并且 h_0^{-1},h_1^{-1} 未知

输出：一个以大于 1/Q（k），h － 缺陷概率，这里 Q（k）是一个 k 上的多项式

这与定义 h_0,h_1 是相互矛盾的。

PTM A 是基于事实 F 是连续的，并且是下列步骤中的一个：

形式 1：伪造：找到一个 g–缺陷

形式 2：伪造：找到一个 *f*–缺陷

形式 3：伪造：找到一个 f_0^{-1}(W)或者 f_1^{-1}(W)对于 $W=Z_1^{B(k)}$，是签名者在历史上最后签名一点。PTM A 由两个部分组成 PTM A_1，PTM A_2，一个运行后又运行另外一个。A_1 试着基于假设 F 产生 g 缺陷，找到一个 h 缺陷。A_2 试着基于假设 F 产生 *f* 缺陷，找到一个 h 缺陷。A_1 和 A_2 都会使用 h_0,h_1 到公钥中，为了使用 h_0,h_1 签署一个消息，这些 PTM 将计算 $v=h_i(R)$，对于一些 $R \in D_h$，并且使用 R 作为 $h_b^{-1}(v)$，因此 A_1 或 A_2 当 F 的询问时，不需要 h_b 的逆。注意既然 h_b 是一个置换，如果 R 是随机的，那么 v 是随机的。

A_1 的描述：

1、 选择（f_0,f_1，D_f）是无缺陷的，这样得到 f_0^{-1},f_1^{-1}，假设公钥包括了 X，D_f，f_0,f_1，$g_0=h_0$，并且 $g_1=h_1$，假设秘密密钥包括 f_0^{-1},f_1^{-1}，

PK	SK
D_f,X,f_0,f_1	f_0^{-1} ,f_1^{-1}
$g_0=h_0,g_1=h_1$	

2、 设置 $H_1=\Phi$ 并且运行 F（f_0,f_1，g_0,g_1，X）。当 F 要求消息 m_i 的签名，

（a） 随机选择 $z_2^i \in D_g$

（b） 设置 $R_i = g_{mi}(z_2^i)$

（c） 设置 $z_1^i = f_{Hi \cdot Ri}^{-1}(X)$

（d） 输出（z_1^i，z_2^i，H_i）

（e） 设置 $H_{i+1} = H_i \cdot R_i$

F 那么输出（$\overline{m}, \overline{s}$）这里 $\overline{s}$ =（$\overline{z_1}, \overline{z_2}, \overline{H}$）

3、 测试看到 V（$\overline{m}, \overline{s}$）=1。如果不是，$A_1$ 失败

4、 假设 $\overline{R}$ = g $\overline{m}$ （$\overline{Z_2}$）。如果对于任意的 i， $\overline{R} \neq R_j$，那么 A_1 失败，因此 F 并没有进行伪造。

5、 否则，假设 j 使得 $\overline{R}$ = R_j，现在有 h $\overline{m}$ （$\overline{Z_2}$）=h $\overline{m}_j$ (Z_2^j)=R_j。从这一点，容易获得一个 h-缺陷。

A_2 的描述：

1、 选择无缺陷（g_0,g_1,D_g），使用的可以得到 g_0^{-1},g_1^{-1}。假设 $f_0=h_0$ 并且 $f_1 = h_1$ ，选择 $R_1, R_2, \ldots, R_{B(K)} \in D_g$，$c \in \{0,1\}$并且均匀并且独立的选择 $z \in D_f$ ， 设置 $X = f_{R1 \cdot r2 \cdot \ldots \cdot RB(K) \cdot c}(z)$

2、 设置 $H_1=\Phi$ 并且运行 F（f_0,f_1，g_0,g_1，X）。当 F 要求消息 m_i 的签名，

（a） 设置 $Z_1^i = f_{Ri+1\cdot\ldots\cdot RB(x)}(X)$

（b） 设置 $Z_2^i = g_{mi}^{-1}(R_i)$

（c） 输出（Z_1^i ， Z_2^i ， H_i）

（d） 设置 $H_{i+1} = H_i \cdot R_i$

F 则输出（$\overline{m},\overline{s}$）这里 $\overline{s}$ =（$\overline{z_1},\overline{z_2},\overline{H}$ ）

3、 假设 $\overline{R}$ = g $\overline{m}$ （$\overline{Z}^2$）

4、 有三种情况可以考虑：

F 做第一类伪造，这就意味着 $\overline{H}\cdot\overline{R}$ =H_i 对于一些 i，在这种情况下，A_2 失败。

F 做第二类伪造：有 $\overline{H}$ • $\overline{R}$ 的前面部分比特，与 A_2 的最后部分比特 H_N。作为结果，

$\overline{H}$ •$\overline{R}$ =H·b·$\overline{s}$ 并且 H_N = H •$\overline{b}$ •S 对于一些 b∈{0,1}并且串 H，$\overline{S}$ ，S，从这一点，得到 $f_b(f_S(\overline{Z_1}))$= $f_b(f_S(Z_1^N))$以 h 缺陷提供了 A_2。

F 做第三类伪造：对于一些比特 b 和串 S，H·R = $H_N \cdot b \cdot S$。既然 A_2 选择了比特 d，为了跟着 H_N ，如果做另外一些要求，b 将与 d 以 1/2 的概率相同。在这个条件下，A_2 有 $h_0^{-1}(h_{HN}^{-1}(X))$和 $h_1^{-1}(h_{HN}^{-1}(X))$以 h 缺陷提供 A_2。

以 p_1 的概率 F（f_0,f_1，g_0,g_1，X）提供一个一类伪造，以 p_2 的概率 F（f_0,f_1，g_0,g_1，X）提供一个二类伪造，以 p_3 的概率 F（f_0,f_1，g_0,g_1，X）提供一个三类伪造。既然 f_0,f_1，g_0,g_1，h_0,h_1 通过无缺陷置换进行均匀选择，A_1 将以概率 p_1 成功，A_2 将以 p_2+（p_3/2）成功。因此 A_1 或 A_2 的成功至少将以 max（p_1 ， p_2+（p_3/2））≥1/3Q（k）。

注记：

1、 与之前的体制不同，这个签名不需要这个体制签署的所有前向消息。仅有签名 $R^i \in D_g$ 附加在后面。

2、 签名的长度不必要是消息签署是线性的，可能以线性的方式代替链接 R^i，去建立一个结构树，这里 R^1 验证 R^2，R^3，R^2 验证 R^4，R^5，这样，在 B（k）可以构造四个元素的二进制树，B（k）是上述签名的总量。那么，R^j 的标签是这个树的一片叶子上 $r^1\ldots r^{B(k)}$。在第 i 个消息的签名计算中，假设 $Z_2^i = g_{mi}^{-1}(r^i)$，并且在 R 的认证树中，假设 $Z_1^i = f_{ri}^{-1}(R)$这里 R 是 r^i 的父进程。第 i 个消息的签名需要包括，那么在所有 R 的进程是从叶子 r^i 到根的，只有这个消息树的离散值是被签署过的。

3、 计算一个 $f_m^{-1}(x)$的消耗是|m|（计算 f^1 的消耗）。接下来可以看到基于因子分解的无缺陷的执行，m 的因子可以储存。

例子：有效的途径计算 $f_m^{-1}(z)$

在例子 9.4.2 中可以看到，如果因子分解是难的，一个陷门置换的特殊簇是无缺陷的。假设 n=pq，这里 p 和 q 是素数，并且 p≡3 mod 8，q≡7 mod 8，对于 x∈QR_n ，

$$F_{0,n}(x) = x^2 \bmod n \qquad f_{1,n}(x) = 4x^2 \bmod n$$

是这个无缺陷陷门置换簇。

注意，记 $\sqrt{x}$ =y，$x=y^2$，并且 y∈QR_n 。

为了计算 $f_m^{-1}(z)$，首先计算（所有如下的计算是 mod n）

$F_{00}^{-1}(z)$ = $\sqrt{\sqrt{z}}$

$F_{01}^{-1}(z) = \sqrt{\sqrt{z}/4} = 1/\sqrt{4}\ \sqrt{\sqrt{z}}$

$F_{10}^{-1}(z) = \sqrt{\sqrt{z/4}} = 1/\sqrt{\sqrt{4}}\ \sqrt{\sqrt{z}}$

$F_{11}^{-1}(z) = \sqrt{\sqrt{z/4}/4}$

假设 m(i)是对应于存储的串 m，容易看到通常条件下得到：$f_m^{-1}(z) = (z/4^{i(m)})^{1/2|m|}$
现在需要计算 $2^{|m|}$ 次的模 n 根，可以有效的进行，提高一个模 Φ（n）的根。

10.4.6 基于陷门置换的一个安全签名体制

这一节包括陷门置换签名体制，开始看到签一比特的方法：

1、选择一个陷门置换 f，可以求逆。均匀独立选择 X_0 ， $X_1 \in D_f$ ，假设公钥包括 f，$f(X_0)$和 f（X_1）。假设秘密密钥包括 X_0 和 X_1 ，

PK	SK
$F, f(X_0), f(X_1)$	X_0, X_1

2 、b 的签名：б（b）= X_b

简单测试 $f(s)=f(X_b)$,证明（b,s）。

多个消息的签名体制，使用上述体制作为一个构建分组。多个消息签名的难度在于 f 不能重复使用，因此，对于一个签署了的消息，陷门置换签名体制产生并且签署一个新的陷门置换。新的陷门置换可以用来签署下一个消息。

描述陷门置换签名体制：

1、 选择一个已知逆的陷门置换 f_1，选择 α_0^j ，$\alpha_1^j \in \{0, 1\}^k$，$i \in \{1,\ldots,K\}$并且 β_0^j ，$\beta_1^j \in \{0, 1\}^k$，对于 $j \in \{1,\ldots,K(k)\}$，这里 K(k)是 K 上均匀独立分布的多项式。假设公包括 f_1，α 和 β。假设秘密密钥包括 f_1^{-1}，假设 $H_1=\Phi$。

PK	SK
$F_1, \alpha_b^j, \beta_b^j$	f_1^{-1}
For $b \in \{0, 1\}, i \in \{1,\ldots,K\},$	$j \in \{1,\ldots,K(k)\}$

签署消息 $m^i = m_1 m_2 \ldots m_k$

2、设置 $AUTH_{mi}^{aifi} = (f_i^{-1}(a_{m1}^{1}), f_i^{-1}(a_{m2}^{2}),\ldots, f_i^{-1}(a_{mk}^{k}))$。$AUTH_{mi}^{aifi}$ 是 m^i 使用 f_i 的签名和 a。

3、选择一个新的陷门函数 f_{i+1} 因此知道 f_{i+1}^{-1}。

4、设置 $AUTH_{fi+1}^{bifi} = (f_i^{-1}(b_{fi+1,1}^{1}), (f_i^{-1}(b_{fi+2,2}^{2})),\ldots, (f_i^{-1}(b_{fi+k,k}^{k}(k))))$这里 $f_{i+1} = f_{i+1,1} \cdot f_{i+1,2} \cdot \ldots f_{i+1,K(k)}$ 是 f_{i+1} 的二进制表示。

5、m^i 的签名是 б（m^i）= （$AUTH_{mi}^{aifi}$, $AUTH_{fi+1}^{bifi}$,H_i）$AUTH_{fi+1}^{bifi}$ 是 f_{i+1} 使用 f_i 和 β 的签名。

6、设置 $H_{i+1} = H_i \cdot (AUTH_{mi}^{aifi}, AUTH_{fi+1}^{bifi})$

注记：假设描述 f_{i+1} ， K（k）比特是满足的。

定理 9.10、陷门置换签名体制是抗 CMA 扩展安全的，如果陷门置换存在。

证明（反证法）：假设不是。有一个伪造 F，要求选择的消息，然后不要求伪造一个消

息以至少 1/Q（K）的概率，Q（k）是在 k 上的多项式。现在假设这样的一个 F 确实存在，可以发现一个 PTM A’满足：

输入：陷门函数 h，逆是未知的，并且 $W \in \{0，1\}^K$

输出：以至少 1/Q’（k），概率 $h^{-1}(W)$，这里 Q’（k）是 k 的一个多项式，概率是随机事件 A’上的 h,w。

A’的构造如下：

1、 A’将使用 h 作为一个陷门置换，回答 F 提出的一个签名要求。既然 A’不知道 h’，产生一个 α，β 的一个适当的子集：随机并且均匀的选择 γ_b^i ,$\delta_b^j \in \{0,1\}^K$，对所有 $b \in \{0,1\}$并且 $j \in \{1,\ldots,K\}$，假设 $\alpha_b^j = h(\gamma_b^i)$，$\beta_b^j = h(\delta_b^j)$在 b 和 j 的同样的范围。均匀选择 n∈{1，…，B（k）}。对于第一段，A’将作为一个单向置换的 f_n，选择 h。如果 A’将放弃交换 α 和 β。在这一点上，可以签 F 的协议，尽管不知道 f^{-1}，要求 m^n 。但是 A’改变 α 和 β 其中的一个。

2、 随机选择 α 和 β 其中的一个，并且设置一个 W 相等输入。假设公钥包括 f_1，（如果 n=1，是 h）, α 和 β。

3、 使用目前体制运行 F，注意以至少 1/B（k）的概率，F 将至少要求 n 个消息。同时注意，F 确实要求 m^n 的签名，A’可以以 1/2 的概率签署 m^n。这是因为以 1/2 的概率，A’不得不计算 α 或 β 的值，是设置 W。

4、 通过概率 1/Q（K），F 将成功的输出一个伪造（$\overline{m}, \overline{s}$），为了对 $\overline{s}$ 作一个好的伪造，必须不仅仅是针对验证，但是必须从 A’的历史分枝以至少 1/B（K）的概率伪造，从 n 个细致要求的历史分枝中选择伪造。因此，F 使用 h 作为陷门函数。

5、 如果基于这个原因，概率是 1/2（k+K（k）），对于集合 W，伪造 α 或 β 的逆。

6、 如果这样，A’输出 $h^{-1}(W)$

A’成功的概率在于以至少 $1/Q’（k）= 1/(4(k+K(k))B^2（k）)$ 是 k 上的一个多项式，这样得到反例。

10.5 具体安全和基于签名的 RSA 体制

在实际中，使用 RSA 签名最宽泛的使用是“HASH 然后解密”：首先把消息 HASH 到一个 RSA 的指定区域然后解密，（例如使用 RSA 求幂指数取幂），这个泛式的用处是显然的：仅仅使用一个 RSA 解密进行签名，并且仅仅使用一个 RSA 加密进行认证。此外只是简单的实现，因此，实际上，这是几种现存标准的基础。

在这一节，分析该范式。不巧的是，将看到标准体制的安全性不能在 RSA 的标准假设下进行验证判断。甚至在 HASH 函数是理想的假设条件下，有充分好的安全判断条件下是应该被推荐的。

可以看到这种体制是已经存在了的。不巧的是没有合适的 HASH 和解密体制可以有效和简单的适应这种方式。（比较见 9.5.12）。这样，还可以做什么呢？呈现一些体制在某种效果下适合“HASH 然后解密”，但是如果可以访问理想的 HASH 函数，应该是可证安全的。（在 7.4.6 章讨论的，意味着正式的，HASH 函数是模拟随机预言模型的，在执行过程中，HASH 函数是从密码学 HASH 函数中提取的。这代表了一个实际的折衷，可以进行有效安全的假设。见[15]中所有这个进展的讨论）。

现在提出和分析两个体制。首先是[15]的 FDH 体制。第二是[26]的 PSS 体制。此外，提出了一种体制，称为 PSS-R，具有消息恢复的特征，这是一个有效的途径缩短签名的长度。扩展上述的提议，开始在现在的实际使用里进行寻找。那么考虑[15、26]所有的 HASH 函数

定义域，最后得到 PSS，PSS-R，并且确实是安全的。

呈现 RSA 的这个体制，同样可以对 Rabin 体制进行使用。这一节的材料大部分从[26]中获得。为了确定这一节是自己包括的，重复这一章一些基本部分，这个观点是不同的。这个资料不是完全多余的。

10.5.1 数字签名体制

在公钥设置中，提供数字完整性的原始使用是数字签名体制。比如对一个不对称密钥结构体制的消息认证，密钥 SK 是一个产生标签（在这个设置中标签就是签名），与用来证实签名的密钥 PK 是不同的。此外 PK 是公钥，在这个条件下也知道这个。因此，仅仅拥有这个密钥的签名者可以签名。任何拥有这个通信公钥的人可以验证这个密钥。

定义 9.11、 一个数字签名体制 DS=（K，S，V）由三个算法组成，如下：

• 密钥产生算法 K 是一个随机算法，并且返回一个密钥对（pk,sk），分别代表公钥和私钥匹配，记做（pk,sk）←K

• 签名算法 S 是一个（可能随机）函数，使用密钥算法 sk ，一个消息 M 返回一个标签和签名 σ，记做：$\sigma \leftarrow S_{sk}(M)$

• 验证算法 V 是一个确定算法使用公钥 pk，一个消息 M，和一个 M 的候选签名 σ，返回一比特，记做：$d \leftarrow V_{pk}(M, \sigma)$。与每个公钥相关的 pk 是消息空间 PLAINTEXTS(PK)，对于每个 M 允许求取。要求 $V_{pk}(M,S_{mk}(M))=1$，对所有 $M \in$ Plaintexts(pk)。

假设 S 是一个实体，需要一个有实力的签名。第一步是密钥产生：S 运算 K 为自己产生一个密钥对（pk,sk），密钥产生算法是 S 在本地运行的。S 将使用 sk 进行签名，其他人使用 pk 验证签名。随后要求任何希望 S 签名的人必须拥有密钥 pk，这里 S 是产生的。此外，验证者必须保证验证者是可信的。意味着 S 的密钥不是其他人的。

有不同的体制保证预期验证的，是拥有签名者的验证公钥。这些实际上是密钥管理的名义下进行的。简言之，有一些选项。S 也许是管理验证者的公钥。更普通的，S 在 S 的名义下记录 pk，一些可信服务器被当作共用电话本，任何人希望获得 S 的公钥，要求发送以 S 命名的服务器并且返回公钥。步骤必须是安全的，这个通信是可鉴别的，意味着验证实际上是与合法服务器进行联系的。因此注册过程本身是可验证的。

事实上，密钥管理是自己拥有的话题，并且需要更深一步的观察。随后将会告知。对于这个时刻，重要的是抓住难题之间的区别。也就是，密钥管理过程不是数字签名本身的一部分。在数字签名的安全性构造和分析中，假设任何预期的验证是拥有一个签名者可信的公钥拷贝。这个假设可以如下使用：

一旦密钥结构是在合适的地方，S 可以产生一些文件 M 的数字签名，运行 $S_{sk}(M)$返回签名 σ，这个对（M，σ）那么验证文件的版本。在接受一个文件 M’和标签 σ’ 声称从 S，一个接收者 B 验证签名的真实性，通过使用特定的签名过程，依赖于消息、签名和公钥加密密钥。也就是计算 V_{PK}（M’，σ’），值是一点。如果这个值是 1，可以读到这个数据是可信的，所以 B 作为从 S 发送过来的信号接收。也就是放弃不可信的数据。

一个要求安全特征的变化过程的变化体制。但这些不是现在所关心的，首先希望固定一个体制的制定说明，因此知道想评估的目标的安全是多种的。

签名体制是随机的，意味着随机事件是相互的，使用这些投币事件决定输出。这样，也许有许多正确的标签，与单独的消息 M 相关。算法也许是正式的，例如，假设使用计数是由发送者提供的。在这个条件下，签名算法作为全局变量访问计数器，必须进行更新。

与加密体制不同，加密算法，是安全的，确定的，不固定的签名算法，必须对于体制既是随机又是正式的。

10.5.2 安全的一个概念

数字签名的目标是与消息验证提供同样安全的机制。不同的是更灵活的密钥结构。因此，可以在理解和安全消息认证基础上链接一个安全的概念。与数字签名不同之唯一处在于攻击者有访问公钥的权限。

攻击者 F 的目标是伪造：想要产生文件 M 和标签 σ，使得 V_{PK}（M，σ）=1，但是 M 并不是发送者 S 发送的。攻击者允许选择消息攻击，在试着产生伪造的过程中。并且体制是安全的，甚至当这样一个攻击者有产生伪造的低概率。

假设 DS=（K，S，V）是一个任意数字签名体制，目标是这个体制选择消息攻击者抗伪造不安全进行形式化。攻击者的行动是把这个过程分为两个阶段。首先是学习阶段，给定预机制访问 S_{SK}（·），这里（PK，SK）是根据 K 的随机一个预先选择。可以询问这个预言机制到 q 次。满足许多方式，只要所有询问是在消息空间 Plaintexts 的消息与这个密钥。一旦这个阶段结束，进入伪造阶段，在这个阶段，M∈Plaintexts（pk）输出（M，σ）。攻击者声称成功，如果 V_{PK}（M，σ）=1 并且 M 从不产生攻击者到预言机的询问。与任何攻击者相关因此是概率成功事件。（概率是基于密钥的选择，S 可能进行任何概率选择，如果是任意选择，那么 F 成功）。这个体制的不安全是最好条件概率攻击者的概率成功，对所有攻击者进行限制固定数量的资源。作为被选资源把攻击时间作为攻击者的一个参数，以及询问次数、所有询问的比特总数、加上伪造输出消息 M 的比特长度。

形式上，定义“攻击者的运算经验”F，在如下的数字签名体制 DS=（K，S，V）：注意公钥作为 F 的一个输入：

经验 $Exp_{DS,F}^{uf\text{-}cma}$

假设（pk,sk）←K

假设（M,σ）← $F^{Ssk(\cdot)}$(pk)

如果 V_{pk}(M, σ）=1 并且 M 不是对的预言机 F 的询问

那么返回 1，否则返回 0

定义 9.12、 假设 DS=（K，S，V）是一个数字签名体制，并且假设 F 是一个攻击者，这样可以访问一个预言机制，假设 $Adv_{DS,F}^{uf\text{-}cma}$ 服从概率分布，那么 $Exp_{DS,F}^{uf\text{-}cma}$ 返回 1。那么对于任何的 t,q,μ 假设：$Adv_{DS,F}^{uf\text{-}cma}$（t,q,μ）= $\max_{F}$ { $Adv_{DS,F}^{uf\text{-}cma}$}

这里是 F 的最大值，因此经验值的运行时间 $Adv_{DS,F}^{uf\text{-}cma}$ 最多是 t，F 的随机预言询问的数量最多是 q，随机预言机制长度的总数是消息 M 的所有询问长度输出伪造是至多 μ 比特。实际上，相应消息的回应是被合法发送者签署的，并且可能对这些例子是仅仅对一个所有者的计算比只具备自身计算能力是更贵的，也就是说，应该期望 q 比 t 小。这说明了为什么 q,μ 是与 t 独立的资源。

RSA 陷门置换是广泛的使用于数字签名机制，现在进行研究。

10.5.3 RSA 体制的密钥产生机

基于假设的单向 RSA 函数，考虑数字签名的各种方法。这些方法是与怎样数字签名和验证算法执行的差别，都使用相同的 RSA 密钥建立体制。也就是用户的公钥是模 N 的 RSA 并且一个加密的指数 e，这里 N=pq 是两个不同素数的乘积，每个长度是 k/2，并且 gcd(e,φ(N))=1。相应的秘密是解密指数 d，这里 $ed \equiv 1 (\mathrm{mod} \varphi(N))$，对于签名，可以方便的把 N 放到秘密密钥中，把看做一个对（N，d），甚至当 N 不是，当然，确实是秘密的。完整的

密钥产生算法如下：

算法 K

随机选择两个不同的素数，p,q,每个 k/2 长

$N \leftarrow pa$；$e \leftarrow Z_{\varphi(N)}^{*}$ ；$d \leftarrow e^{-1} \bmod \varphi(N)$

$Pk \leftarrow (N,e)$；$sk \leftarrow (N,d)$

返回 pk,sk

因为 φ(N)=（p-1）(q-1)，因此可以产生 p,q，上述密钥产生算法可以计算 φ(N)，因此可以计算剩余的步骤，依赖于这个值的知识，d 的计算是通过扩展的 GCD 算法进行使用的。

回忆映射 $RSA_{N,e}(x) = x^e \bmod N$ 是 Z_N^* 上的置换，逆是 $RSA^{-1}_{N,d}(y) = y^d \bmod N$，将进行一个好的信任假设：RSA 是单向函数。

以下将考虑各种签名体制，所有体制使用上述的密钥产生算法，并且为了安全签名，试着构建不同的单向 RSA 签名。

10.5.4 单向签名

RSA 的基本函数假设是单向的，意味着给定一个单向函数，N,e,y，难于计算 $RSA_{N,E}^{-1}(y)$，必须构成合适的思想。随机选择 y。

定义 10.17 假设 K_{RSA} 是一个 RSA 产生器，伴随着安全参数 k，假设 A 是一个算法。考虑如下的经验过程：

$$\begin{gathered}\text{Experiment } \mathrm{Exp}^{\text{ow-kea}}{}_{Krsa}(A) \\ ((N;e);(N;p;q;d)) \leftarrow K_{rsa} \\ x \leftarrow Z^{*}_{N};\ y \leftarrow x^{e} \bmod N \\ x' \leftarrow A(N;e;y) \\ \text{If } x' = x \text{ then return 1 else return 0}\end{gathered}$$

A 的 ow-kea 优势，定义为：

$$\mathrm{Adv}^{\text{ow-kea}}{}_{Krsa}(A) = \Pr[\mathrm{Exp}^{\text{ow-kea}}{}_{Krsa}(A)=1]$$

上述“kea”代表“已知指数攻击”。也许可以选择指数攻击，简短的“cea”，加密指数是用难题的例子。允许攻击者选择，唯一的条件是利用攻击者，不选择e=1。

定义10.18：假设 K_{mod} 是一个安全参数k的模型，假设A是一个算法。考虑如下的过程：

$$\begin{gathered}\text{Experiment } \mathrm{Exp}^{\text{ow-kea}}{}_{Krsa}(A) \\ ((N;e);(N;p;q;d)) \leftarrow K_{mod} \\ y \leftarrow Z^{*}_{N};\ (x,e) \leftarrow A(N;y) \\ \text{If } x^{e} \equiv y \pmod N \text{ and } e > 1 \\ \text{then return 1 else return 0}\end{gathered}$$

A 的 ow-cea 优势，定义为：

$$\mathrm{Adv}^{\text{ow-cea}}{}_{Krsa}(A) = \Pr[\mathrm{Exp}^{\text{ow-cea}}{}_{Krsa}(A)=1]$$

10.5.5 陷门签名

陷门签名代表最多直接的方向，努力构建为了签名体制的单向 RSA。相信签名者对密钥 N，d 的拥有仅仅在于可以计算 RSA 的逆 $RSA^{-1}_{N,d}$ ，对于任何其他人，仅仅知道 N，d 是计算不可行的。从而，签名者签署一个消息，难度在于运算复杂性。为了方便，假设要求消息是 Z_N^*的一个元。通过在需要签名的地方运行简单的签名计算 $RSA_{N,e}$ ，是可以进行签名的，并且看看是否可以返回消息。更细致的，体制 DS=（K，S，V）有上述的密钥产生算法，如下的记号和验证算法：

算法 $S_{N,d}$（M）　　算法 $V_{N,e}$（M，x）

$X \leftarrow M^d \bmod N$　　$M' \leftarrow x^e \bmod N$

返回 x　　如果 M=M'那么返回 1,否则返回 0

这个公钥的消息空间是 Plaintexts（N,e）=Z_N^* 意味着仅仅需要签名者可以签署的消息，是组 Z_N^* 的元。在这个体制，用 x 表示 M 的签名。这个体制是怎样安全的？如同上述所述，在这背后的直觉是签署运算是仅仅签名者可以执行的运算，既然在没有 d 的知识时，计算 $RSA_{N,e}^{-1}(M)$是具备难度的。然而，一个人应该记住，弱单向是在非常困难的情况下进行的，并且比签名的设置难度更大。弱单向函数说明如果随机的选择 M，并且伪造签名。记住攻击者的目标是产生一对（M，x），因此 $V_{N,e}$（M，x）=1，没有试着模仿签名算法，必须仅仅做一些满足验证的算法。特别的允许选择 M 或随机选择 M 比签署给定的输出，通过签名机制，相应在这些条件下得到有效的签名比 M 产生有效的输出 $RSA_{N,e}^{-1}(M)$有相同的预研机制。这些特征对攻击者容易得到伪造。最简单的策略是首先选择签名，并且作为函数定义消息。可以用如下的伪造进行说明：

伪造 $F^{Sn,e(\cdot,\cdot)}(N,e)$, $X \leftarrow Z_N^*$ ； $M \leftarrow x^e \bmod N$

返回（M，x）

这个伪造没有签名预言机制的询问，简单如下输出伪造：为了计算后继概率，注意到因为 $x^e \bmod N = M$,所以，$V_{N,e}(M,x)=1$。所以 $Adv_{DS,F}^{uf\text{-}cma} = 1$，这暗示着 $Adv_{DS}^{uf\text{-}cma}$ （t,0,k）=1，这里 $t=O(x^3)$是模 N 的时间指数，是 F 的计算时间指数，并且 μ=k 因为 M 的长度是不是 k。值 t,q, μ 非常小，可以说这个体制是非常不安全的。

消息 M 对于上述伪造者进行的伪造是随机的。这足够破解这个体制，作为每个安全的定义，因为构造了一个非常强的安全定义。事实上对于这个定义，给定一个消息 M 可能进行消息伪造，但是这次使用了签名机制。攻击依赖于 RSA 函数的倍数。

伪造 $F^{Sn,e(\cdot,\cdot)}(N,e)$

$M_1 \leftarrow Z_N^* - \{1, M\}$； $M_2 \leftarrow MM_1^{-1} \bmod N$

$X_1 \leftarrow S_{N,e}(M_1)$; $x_2 \leftarrow S_{N,e}(M_2)$

$X \leftarrow x_1x_2 \bmod N$

返回（M，x）

给定 M 伪造者想计算一个 M 的有效签名，产生 M_1 ，M_2 ，并且得到签名 x_1 ，x_2 。然后设置 $x= x_1x_2 \bmod N$，现在验证算法检查是否 $x^e \bmod N = M$。但是注意：

$$x^e \equiv (x_1x_2)^e \equiv x_1^e x_2^e \equiv M_1 M_2 \equiv M \pmod N$$

这里使用 RSA 函数的倍数，x_i 是 M_i 的签名 i=1,2，这意味着 x 是 M 的有效签名，既然 M_1 选择的是 1 或 M，同样 M_2 也是 F 的一个随机预言机制，因此 F 以概率 1 成立。

10.5.6 HASH 然后求逆范式

基于签名体制的 RSA 真实世界需要克服上述攻击，也可以加入其他的陷门设置。特别的，消息不是特别的分组的。可能是长文件，意味着任意长度的比特串，两方面的情况是典型的预处理，在给定消息 M，通过一个单向函数产生一个“定义域点”y，然后在逆函数 $RSA_{N,e}^{-1}$ 产生实际的签名。单向函数 HASH：$\{0,1\}^* \rightarrow Z_N^*$ 是公共的，意味这描述是已知的，并且任何人可以计算。（可能或不可能使用一个密钥，如果可以使用，这个密钥是一个公钥。）更细致的体制 DS=（K，S，V）有上述密钥产生算法，并且签名和验证算法如下：

Algorithm $S_{N,d}(M)$
　Y ← Hash(M)
　X ← y^d mod N
　返回 x

Algorithm $V_{N,e}(M,x)$
　y ← Hash (M)
　y' ← x^e mod N
　如果 y=y'那么返回 1 否则返回 0

研究为什么这可以帮助解决陷门函数签名的弱点，什么样的安全要需要单向函数实现。

返回上述研究的陷门签名体制的攻击问题，从提出的第一种伪造开始，仅仅是简单的设置 M 伪造一些随机数 $x \in Z_N^*$ 产生 x^e mod N。在 HASH 在求逆的条件下，这个策略成功的概率是什么？如果 x^e mod N = HASH（M）伪造者成功，（比之前 x^e mod N = M 更单纯），希望是用一个好的 HASH 函数，x^e mod N = HASH（M）是完全不同的。考虑上述提出的第二种攻击，依赖与 RSA 函数的倍数，如果这个攻击在单向再求逆的条件下工作的，下述条件是正确的：

$$Hash(M_1)Hash(M_2) \equiv Hash(M) \pmod N \quad (9.1)$$

然后通过好的 HASH 函数，希望这种攻击是不容易实现的。

这种攻击假设破坏了代数结构，使用上述攻击是可能的。怎样发现还不曾公开说明的。Hash 函数也许防止一些用于陷门体制的攻击，使用了一种新的基于 HASH 函数的碰撞攻击方法。如果一个攻击者可以发现两个不同的信息 M_1，M_2，产生相同的 HASH 值，也就是 $Hash(M_1) = Hash(M_2)$，那么可以容易的伪造签名，如下：

伪造 $F^{Sn,e(\cdot)}(N,e)$
　$X_1 \leftarrow S_{N,e}(M_1)$
　返回 （M_2 ，x_1）

这是因为有相同的 HASH 值，因为 x_1 是一个 M_1 有效的签名值，并且因为 M_1，M_2 有相同的 HASH 值，有：$X_1^e \equiv Hash(M_1) \equiv Hash(M_2) \pmod N$

这意味着验证过程是接受 x_1 作为 M_2 的签名，因此，一个单向函数必要的要求是抗碰撞攻击的，也就是对于两个不同的信息 M_1，M_2，产生相同的 HASH 值 $Hash(M_1) = Hash(M_2)$ 是计算不可行的。

以下将讨论更多的 HASH 再求逆的范式的具体例子。但是在之前，重要的是目前为止尝试访问。以上具备了单向函数的识别特征，这对于安全签名体制是重要的。抗碰撞是其中之一。其余的要求没有这样公式化的表明，但是使用（9.1）可以以高失败率进行代数结构攻击。经典的设计固定在单向函数上述的攻击上，目标是实现合适的单向函数。但是如果已明白通过努力在这门学科和注记上得到的进展和观点，应该有更深刻的看法。对注记关键点是需要不是真实的针对单向函数抗固定攻击的特征，而是单向函数针对足够深刻的特征，指定的特征还没有构思，这项工作还没有进行。当然，针对 HASH 函数的特定特征可以聚集可能具有的特征的一些观念。但是不仅如此，必须深入的考虑由此导出的特征是否足够，这

样就有了足够的关系。实践证明会产生一次又一次的错误。

怎样希望把这个做得更好？返回可证安全的基本原理。希望保障数字签名体制是安全的，在原始假设是安全的前提条件下。因此必须试着对 RSA 作为单向函数的签名体制的安全进行确认，一些在 HASH 条件下的安全。基于这种考虑，假设继续进行一些解决建议。

10.5.7 PKCS≠1 体制

RSA 公司基于密码学的 RSA 标准和软件是主要源头之一，RSA 实验室（现在是安全动力实验室公司的一部分）产生了一个标准叫做 PKCS（公钥密钥密码标准）。PKCS≠1 是基于 RSA 函数的签名（加密）体制。这个标准是广泛的使用的，并且根据这一点，将说明怎样看到所做的事情。

使用单向再求逆的范例，例如通过一个特殊的单向函数 PKCS-Hash 进行 Hash，在详细说明前，需要加入一些完成条款。到目前为止，考虑了作为返回一个群的元素的作为 Hash，作为集 Z_N^* 的一个点。这是必要的，因为 HASH 函数的输出必须是可以提供 $RSA_{N,e}^{-1}(M)$ 一些情况。然而，在一个执行过程中，可以仅仅处理比特串。因此实际上，单向函数必须返回一串比特，可以对 Z_N^* 的元素进行解释。这不是一个太大的考虑。模 N 有长度 k 比特（例如 k=1024 的值）并且 Z_N^* 是{1，…，N}的子集，包括{1，…，N}中与 N 互素的数。每个 Z_N^* 的元素因此可以写做一个 k 比特的串。因此，一个 HASH 函数返回一个 k 比特串 y，可以简单的把 Z_N^* 的元素表示为整数。好的，这里有一些警告。首先，整数必须与 N 互素，第二，必须至多是 N，如果 k 比特串的高阶比特串是 0 可以保证第二点。意味着一个整数 y 至多是 $2^{k-1}-1<N$。第一点不用担心，原因是在{1，…，N}中很少元素不与 N 互素。在{1，…，N}中这种元素部分是

$$（N-\varphi（N））/N = 1-\ （（p-1）(q-1)/pq）=(p+q-1)/pq< (2^{1+k/2})/2^{i-1} = 4\cdot 2^{-k/2}$$

这些事实是|p|=|q|=k/2，因此使用典型的长度例如 k=1024，不与 N 互素的部分值是可忽略的。不仅仅是希望有这些点可以进行攻击，甚至攻击者蓄意进行的概率是小的，因为这个点是 p,q 的倍数，使用 N 引入最大公因子 gcd 的概念，假设计算不可能。

这些技术被分配了，假设继续描述 PKCS≠1 单向函数。

回想已经讨论的抗碰撞攻击设计，假设固定一个函数 $h: \{0,1\}^* \rightarrow \{0,1\}^l$ ，这里 l≥128 并且抗碰撞在某种程度上是没人知道怎样找到一个不同的对 M，M’，使得 h(M)=h(M’)，目前的角色倾向于使用 SHA-1 代替，这样 l=160，在这之前是 MD5，这里 l=128,RSA PKCS≠1 标准定义，PKCS-Hash(M) = 0x0001FF FF…FF FF 00 ‖ h(M)。这里，‖定义的是连接，并且有足够的 FF 字节嵌入 PKCS-Hash(M)是等于 k 比特的，注意 HASH 首先输出的四比特是 0000，意味着作为一个整数的确至多是 N，根据上述，因此是一个群元。也需要注意 PKCS-Hash(M)是简单抗碰撞的，因为 h 是抗碰撞的，因此可以完成需要的条件。

回忆签名体制是准确的 HASH-求逆范式，更具体的，假设重写签署和验证算法：

算法 $S_{N,d}(M)$	算法 $V_{N,e}(M,x)$
$Y \leftarrow$ PKCS-Hash(M)	$y \leftarrow$ PKCS-Hash(M)
$X \leftarrow y^d \bmod N$	$y' \leftarrow x^e \bmod N$
Return x	If y=y’then return 1 else return 0

现在什么是关于签名体制的安全？首先关心的是看到过的几种陷门签名的代数攻击。作为在 9.5.5 节中的讨论，希望看到 9.1 的等式不成立。这也是不希望看到的；很难想象怎样

使 PKCS-Hash(M_1)，PKCS-Hash(M_2) mod N 有特殊的结构，要求与一些消息的 PKCS-Hash 相似。这不是一个证明攻击是不可能的，当然，至少是不明显的。

这是与经典的基于设计攻击的进展接触点，随后，上述体制是可接受的，因为已知攻击失败。但更深刻的看待这个问题是因为关心，需要得到的进展是看到怎样与签名体制相关的假设的期望安全或在原始条件下的明白安全，在这个原因下是 RSA 体制。

假设 RSA 是单向的，意味着对于随机选择的一个点 $y \in Z_N^*$ ，计算 $RSA_{N,e}^{-1}(M)$是计算不可能的，另一方面，在 $RSA_{N,e}^{-1}(M)$使用的一个点是应用与签名体制在范围，在 PKCS 的 HASH 函数的 $S=\{PKCS\text{-}Hash(M):M \in \{0,1\}^*\}$，S 的大小至多是 2^l ，因为输出 lbits，那么 PKCS-Hash 固定点的其比特。使用 SHA-1 意味着$|S| \le 2^{160}$ ，这看起来象一个大的集合，但是在 RSA 中的定义域 Z_N^* 是小的：$|S|/|Z_N^*| \le 2^{160}/2^{1023} = 1/2^{863}$ 。这是概率存在的，在 S 的 Z_N^* 中随机选择，对于所有实际的目的，是 0。因此 RSA 是非常好的单向函数并且容易在 S 上求逆，因为随机点在 S 范围中的概率很小。PKCS 体制的安全在 RSA 标准单向下的假设条件下可以保证是唯一的，注意这是真实的无论在 HASH h 条件下是否是好的，这构成了 PKCS-Hash 的基本。困难是 PKCS-Hash 设计自己，特别是底码。

PKCS 签名体制的安全要求假设在集 S 上 RSA 很难求逆，一个所有范围的函数，（并且这仅仅是种需要，对于签名体制的安全是足够条件的）。

假设试图澄清和强调次数的观点。现在并不声明知道怎样攻击 PKCS 体制。但是说一个已知攻击的缺乏应该不是满足这个体制的一个好的原因，可以识别设计缺陷，这样使用 RSA 体制的方式不是根据 RSA 作为单向函数的理解。这才是所关心的原因。

10.5.8 FDH 体制

从上述可以看出，如果 hash-求逆的范例产生一个签名体制，其安全是基于 RSA 函数的单向机制的，必须指出在 $RSA_{N,e}^{-1}$ 是随机应用在这个体制上的点 y。换言之，hash 函数的输出必须总是看起来是随机的。甚至这一点仅仅是一个必要的条件，不是（只要已知）一个足够的。现在问自己如下的问题，假设有一个“完美”的单向函数 HASH，在那种情况下，hash-求逆的体制是否安全？为了理解这个，必须首先决定什么是一个“完美”的单向函数。回答是非常中立的：一个是随机，也就是对任意询问，除了与过去的询问一致，返回一个随机回答（将回答更多的随机预言工作，但是假设继续）。因此问题是：在一个模型中，当 HASH 是完美的，现在证明签名体制是安全的，如果 RSA 是单向。

这确实是基本的问题，如果单向-求逆的范式是可行的，真正必须证明安全，在单向函数是完美的条件下。如果采用单向-求逆的范式是极端失策的，在这种模型下证明安全是不可能的。对于一个完美的单向函数是不可能的，怎样期望在任何真实条件下设置的，对于完美体制怎样变为可能的。

因而，现在关注“思考实验”，作为完美单向函数，包括数字签名体制的使用。这是一个彻底的实验，因为没有特定的单向函数是完美的。单向函数不再是固定的，仅仅是随机事件的一个盒子。然而，这个“思考实验”有一些重要的情况关于签名范式重要，在理解中，将随后看到更好更具体的关键步骤。

现在假设更多关于完美单向函数。假设 Hash 返回一个 Z_n^* 的随机数，每次进行调用，除了在同一个消息进行两次调用，两次返回同样的值。换言之，是一个在$\{0, 1\}^*$上的随机函数的实例，已经看到之前这样的目标，研究伪随机：记住定义伪随机函数，考虑包括随机函数的实验。因此这个概念不是新的。称 HASH 函数是一个随机预言机制，在这一章用 H 表示。对所有的方、签名者、验证者和攻击者是可访问的，但是作为一个预言机制。意味着仅仅通过一个说明界面，计算 H（M）的一方必须构造一个随机预言呼叫。意味着希望 H

（M）返回的一些表示和消息 M 和一个适当的值进行返回。说明可以输出一个对（hash,M），首先的构成是单纯的一种正式的符号使用指示这种随机预言符号。具备这样的输出，呼叫算法等待回应。一旦值 H（M）返回，就继续执行。

最好的方式考虑 H 是作为动态处理来维持一个输入、输出表，每次产生一个询问（hash,M），过程首先检查是否表对于部分 y 包括（M，y）对，并且如果是这样，返回 y。否则在 Z_n^* 中随机选择一个 y，把（M,y）置入表中，返回 y 作为询问回答。

考虑上述的在模式中的 HASH-求逆签名体制，Hash 是一个随机预演机制，这称做全定义域单向函数（Full Domain Hash, FDH），这个体制 DS=（K，S，V）使用 RSA 的 9.5.3 的密钥签名算法 K。写签名和验证算法如下：

Algorithm $S_{n,d}^{H}(M)$
Y ← H(M)
X ← y^d mod N
Return x

Algorithm $V_{n,d}^{H}(M)$
y ← H(M)
y' ← x^e mod N
If y=y'那么返回 1 否则返回 0

对于写算法的方式仅仅是改变，在 9.5.5 中的普通 HASH-求逆的体制可以记做：把 H 记为上标，仅仅通过说明预言机制界面进行随机接入。说明 y ← H(M)是通过使用询问（Hash, M）完成，并且使用 y 定义返回的回答，作为上述的讨论：

假设 RSA 是单向函数的假设条件下，是否上述签名体制是安全的？为了考虑这个问题，首先需要扩展包括新模型的定义。关键的不同在于攻击者的成功概率是 H 的随机选择加入之前考虑的随机选择。与以前的 F 伪造作为以前预言机制的访问控制，但是现在也可以访问 H。此外，S 可以访问 H。首先写实验意味着伪造 F 成功的估计并且讨论更多。

经验 $ForgeExp^{ro}(DS,F)$
假设（（N，e）,(N,d)）← K
随即找到 H：{0，1}* → Z_N^*
假设（M，x）← $F^{H,Sn,dH(\cdot)}(N,e)$
如果 $V_{N,e}^{H}(M,\sigma)=1$ 并且 M 不是 F 的询问对于签名预言机制
那么返回 1 否则返回 0

上标“ro”是实验名，在随机预言机制内指示。更具体的，已经写了实验说明，对于 RSA 的原因，基于作为 FDH-RSA 的体制，是容易产生的。开始选择 RSA 的公钥（N,e）和秘密密钥（N，d），作为每个 RSA 密钥签名的产生算法。选择下一个随机单向函数，选择是上述最好的动态考虑。立刻不用考虑 H 的选择，但是比这个过程完成表格的任务，因此随机选择仅仅是适时的 H 预言机制。伪造是给定访问 H 的预言机制，为了模拟选择消息攻击，同时给定一个签名预言机制 $S_{N,d}^{H}(\cdot)$的访问控制，通过这个机制，可以发送任何消息，收回在 FDH-RSA 条件下的 H（M）^{d} mod N 签名。为了返回一个签名，签名预言机制自身必须包括 H 预言机制。因此有两种方法可以包括 H：直接包括 F 或间接通过 $S_{N,d}^{H}(\cdot)$随后包括 F。在多次询问一些数之后伪造输出接受 M，x，但是 F 永不询问 M 的签名机制（F 当然允许进行一个随机询问 M，并且确实考虑伪造其方是难于实现的，但是不允许进行签名询问 M），假设 $Adv_{DS,F}^{uf\text{-}cma}$ 是概率的，实验 $ForgeExp^{ro}(DS,F)$返回 1，这个概念是与之前相同的，将知道从体制描述的随机预言机制。那么对于任意的 t,q_{sig},q_{hash},μ 假设：

$$Adv_{DS,F}^{uf\text{-}cma}(t,q_{sig},q_{hash},\mu) = \max_{F}\{Adv_{DS,F}^{uf\text{-}cma}\}$$

资源 t,q_{sig} ,q_{hash} , μ 可以使用整体的资源说明进行描述，比攻击者 F 的资源使用更好。首先定义执行时间，计算随机询问，产生密钥，甚至验证伪造。那么 t 假设是运算时间的上界加上 F 码的大小，F 签署询问的次数之多是 q_{sig} ，在计算 hash 询问时，再次看到整个的试验并且询问 H 的总数至多是 q_{hash} 。包括计数是直接的 F 的 Hash 询问验证，使签名预言机制的间接的 HASH 预言询问，甚至在最后一部分的验证算法是 HASH 询问。（随后意味着 q_{hash} 是对于一个验证至少要求的询问次数，对于 FDH-RSA 是一个，事实上，对于 FDH-RSA 有签名消息的长度是签名询问加最后输出消息 M 的长度。

在这个集合中，声称 FDH-RSA 体制是安全的。如下定理得上界是唯一不安全的，RSA 作为单向函数的单独不安全。

定理 9.13[24]、假设 DS 是 FDH-RSA 体制使用安全参数 k，那么对于任意的 t, q_{sig} , μ 和任意的 $q_{hash} \geq 1+ q_{sig}$ ，有：$Adv_{DS,F}^{uf\text{-}cma}(t,q_{sig},q_{hash},\mu) \leq q_{hash} \cdot Adv_{RSA}^{owf}(t')$。这里，$t'=t+q_{hash} \cdot O(k^3)$。

这个定理说明 FDH-RSA 仅有的伪造签名在随机点上试着求 RSA 函数的逆，在安全上有一些损失，也许是破解 RSA 签名体制的机会比在可对比的时间求 RSA 的逆更大。作为单向询问的数字的特征的因素，可以把 $Adv_{RSA}^{owf}(t')$是足够小的，在选择一个大的模数 k 的大小时，$q_{hash} \cdot Adv_{RSA}^{owf}(t')$也是足够小的。

必须记住：这是一个单向函数为随机函数的模型，甚至这点显示了一些信息，说明 hash-求逆的范例是合理的，至少是完美的单向函数。这是一个好的情况，开发具体的范例描述。

假设现在处理 9.13 的定理证明过程，范例是通常的一种情况。假设 F 是一个 FDH-RSA 的伪造攻击，假设伴随的源头攻击是 t,q_{sig} ,q_{hash} , μ，测量相对的经验值 $ForgeExp^{ro}(DS,F)$作为上述的讨论，对 RSA 函数设计一个逆 I：

$$Adv_{RSA,I}^{owf} \geq (Adv_{DS,F}^{uf\text{-}cma})/ q_{hash} \qquad (9.2)$$

此外，给定定理叙述，以 t'的时间进行运算时间，现在根据一些算数和最大采用的定理进行如下使用。

记住逆元 I 作为输入（N，e），描述 RSA 函数的一个例子 $RSA_{N,e}$，并且一个点 $y \in Z_N^*$，工作是试着输出 $RSA_{N,e}^{-1}(y) = y^d \bmod N$，在 RSA 函数下的 y 的逆，这里 d 是一个解密指数，相应与加密指数。当然，既不是 d 也不是 N 的因子是对 I 有用的。I 的成功在于在给定标准的 RSA 随机算法产生时给定（N,e）(N,d)的随机选择，也是从 Z_N^* 随机选择 y。为了完成任务，把 F 作为一个子集，在输入公钥 N，e 时，在某种程度上是使用 F 的功能伪造签名发现 $RSA_{N,e}^{-1}(y)$。在讨论怎样使用伪造决定点 y 的逆前，需要此外的观察怎样返回一个 F 作为子集。回想 F 访问两个预言机制，并且进行回应。在运算中的任何一个点上，可能输出（hash,M）。将一直等到返回值，转变为 H（M）。一旦这点被接受，可以继续进行。类似的可以输出（sign，M）并且等到一个转化值：$S_{N,d}^N(M)$。在得到这个值后，可以继续。重要的是说明 F，作为一个算法，单纯通过一个界面预言进行通信。并不控制这些预言的返回，也许考虑一个预言询问象一个系统呼叫。F 的考虑是在记忆中一些特殊的描述，写一个随机询问 M。这就是 F 读到的什么信息，并且继续。

当 I 执行 F，F 不知道这些。这将是一些点，作为一个随机预言机制，假设考虑预言机制，要求询问（hash,M），那么等待回答。如果 I 需要运算 F 作为实现，那么对 F 可以进行 I 运算，回答这个预言机制。F 将采取怎样采用和继续执行，如果不能提供回答，F 不能继续运算。仅仅在这里等待，看到同步的思想在几个证明中：I 产生了一个“虚拟真实”，在 F 可以相信自己是在普通环境下。

I 的策略是控制随机预言机制的优势，在特殊的方法下进行选择。不是仅仅在这种经验 $ForgeExp^{ro}(DS,F)$中进行选择，此外，x 将期望求 y 的逆，并不总是思考。I 是幸运的，并且足够幸运。

开始考虑一个简单的伪造 F。没有签名要求并且特别一个单向函数的请求（hash,M），那么输出一个对（M，x）作为伪造，在 hash 询问和伪造，消息 M 是相同的，（有 $q_{sig} = 0$，$q_{hash} = 2$，最后归因于 F 的 HASH 询问，最后一经验值进行询问），现在如果 F 是成功的，那么 x 是 M 的有效签名。意味着 $x^e \equiv H(M) \bmod N$，或者，对等的，$x \equiv H(M)^n \bmod N$。某种程度上，F 找到了 H（M）的逆，返回值是应答 M 的随机预言询问。现在记住 I 的目标是计算 $y^d \bmod N$。这里 y 有给定的输入。一个自然的思想是：如果 F 可以在 H（M）上求 $RSA_{N,e}$ 逆，那么设置 H（M）到 y，并且因此在 $RSA_{N,e}$ 的逆下得到 y 的逆。在这个方式下，可以设置 H（M）因为可以控制预言询问的回答。当 F 进行询问（hash,M），I 逆是作为回答简单返回 y。当 F 进行询问（hash,M），I 逆简单的返回 y 作为回答。如果 F 成立，输出一个有效伪造（M，x），有 $x=y^d \bmod N$，并且 I 可以输出 x，工作完成。

但是为什么 F 返回一个有效的伪造，当得到 y 作为应答对一个 hash 询问 M？也许将重新使用，逆 I 告知这点将不进行工作，但是这将不会存在。F 仅仅是一个算法并且对任意给定的环境进行工作。重要的是唯一做这些简单的应答分布。在经验值 $ForgeExp^{ro}(DS,F)$反馈（hash,M）是一个 Z_N^* 上的随机元素。但是 y 确实具备同样的分布，在把 RSA 作为单向函数进行破解时，怎样在经验值进行 I 的选择。因此 F 不能在虚拟真实的条件下进行更多不同的运算，比较在真实环境下。返回有效伪造的概率也还是 $Adv_{DS,F}^{uf\text{-}cma}$，因此对于简单的 F，找到 $y^d \bmod N$ 的逆的成功的概率确实是相同的，在伪造签名中 F 成功的概率。等式（9.2）少有声明，因此当然对此满意。

然而，更多的伪造者不是必须要求不进行签名，并且仅仅一个单向询问在伪造者中询问组成的每一个消息。I 必须可以应付任何伪造。

逆 I 将定义任何子路径集合，H-Sim（称为单向函数预言机制）并且 S-Sim(称为签名预言机制)扮演单向函数和伪造者的角色。名义上的，无论 F 构成一个询问（hash,M），逆 I 将返回 H-Sim（M）到 F 作为回答。（S-Sim 路径将包括 H-Sim）。作为执行，将构造几个不同的表（阵列）“定义”H。对于 $j=1,\ldots,q_{hash}$，在 H 上的第 j 串称为经验（每个方向直接起源于 F 的 HASH 询问，不直接起源于 F 的询问，并且产生最后的验证询问）将记做 Msg[i]是一个签名询问当负责返回 F 的询问作为签名是 X[j]，现在问题是 I 怎样定义所有这些值。假设第 j 个 HASH 询问在不直接进行的经验值，作为 F 签名询问的结果（sign,Msg[i]），在经验条件 $ForgeExp^{ro}(DS,F)$伪造将返回 $H(Msg[j])^d \bmod N$。如果 I 希望保持 F 运算，必须返回 F 作为签名是 X[j]。I 能做什么？可以尝试直接简化签名过程，设置 Y[j]作为随机数，（记住 Y[j]作为 H（Msg[j]）使用），并且返回 $(Y[j]^d) \bmod N$，但是随后不能计算既然不拥有秘密签名密钥指数 d。假冒者是 I 首先在 Z_N^* 上随机发现一个值，并且 $Y[j]=(X[j])^e \bmod N$，现在可以返回 X[j]作为签名询问的回答，这个回答在某种程度上是精确的，证明关系（F 也许检查）支持：有 $Y[j]\equiv(X[j])^e \bmod N$。

这剩下一个松散的结果，一方假设上述 I 在点构造的签名询问定义 Y[j]进行解释。但是也许 Msg[j]=Msg[l]对于一些 l<j，导致有一个单向询问在过去卷入相同的消息。那么单项值是 Y[j]已经定义，作为 Y[l]不能被改变，这可以被非常简单的表达：对任意单向询问 Msg[l]，单向激励机制可以跟随如下设置的策略回答 $Y[l]=(X[l])^e \bmod N$，同时进行单向询问，意味着对于概率准备超前的时间，Msg[l]是一个随后的签名询问，也许不是，但是没有任何损失。

总而言之，事实上一些事情是损失的，一个试图保持清醒的读者，也许注意到可以解决两个问题：怎样使用 F 找到 $y^d \bmod N$，这里 y 是输入到 I，怎样激励签名的回答和 F 的单向询问，但是者些过程是由冲突的。通过这个途径得到 $y^d \bmod N$，返回 y 作为询问（hash,M），这里 M 是伪造消息的签名。然而不知道在消息询问之前将有一个伪造。因此，知道回答一个单向询问 Msg[j]，返回 y，或者对一些 X[j]返回（$X[j])^e \bmod N$。如果首先做这些，将不能回答 Msg[j]签名询问。如果做第二部，Msg[j]等价于一个伪造的消息，这样不能找到 y 的

逆。回答是采取一个假设对付上述情况，有一些机会猜测是正确的，在这种情况，I 成功。

特别的，注意 Msg[q_{hash}]=M 是伪造定一消息，既然 Msg[q_{hash}]是最后认询问的消息。消息 M 也许在列表中不止产生一次，但是至少产生一次。现在 I 将随机选择一个在范围 1≤i≤q_{hash}，并且 y 作为单向询问（hash,Msg[j]），对所有其询问 j 将在 Z_N^* 中随机选择等于 Msg[j]作为询问并且设置 H（Msg[j]）=$(X[j])^e$ mod N。伪造消息 M 将以至少为 1/ q_{hash} 的概率等于 Msg[j]，并且这仅仅暗示等式（9.2）。如下将总结这些思想作为定理 9.13 的证明。对于上述描述建议总是选择 i= q_{hash} ，既然通过定义 Msg[q_{hash}]=M，为什么不做这个工作？因为 M 也许是等于 Msg[j]，对于一些 j< q_{hash} 并且如果有设置 i= q_{hash} ，那么同时希望返回 y 作为 M 的回答，发现有一些已经定义了的 H（M）作为其他的一些事情，并且改变思想应该是太晚了。

定理 9.13 的证明、假设 F 是攻击 FDH-RSA 体制的一些伪造，使用资源参数 t,q_{sig} ,q_{hash} , μ，测量相关的经验 $ForgeExp^{ro}$(DS,F)作为上述讨论，设计一个逆 I 对于 RSA 函数因此等式（9.2）是真实的并且 I 的运算时间是通过值 t’给定的在理论上的阐述，定理如下：

首先在两个子集的条件下描述 I，一个单向随机激励机制，H-Sim(·)和一个签名预言激励机制 S-Sim(·)，保持三个表，Msg,X 和 Y，每列的索引范围从 1 到 q_{hash} , 选择一个随机索引 i，所有这些是全局变量，也可以用于子程序。列输入的倾向意义如下，对于 j=1,…, q_{hash} 。

Msg[j] – 第 i 个在经验值中的单向询问

Y[j] - 关于上述的单向预言机制回答，意味着扮演 H（Msg[j]）角色的值

X[j] - 对于 j≠i，对于签名 Msg[j]的应答是意味着满足（X[j]）^{e} ≡ Y[j](mod N)。对于 j=i 是值 y

求逆的编码如下：

逆 I（N,e,y）

初始列 Msg[1…q_{hash}]，X[1…q_{hash}]，Y[1…q_{hash}]到空栈

J ← 0 ; i ← {1,…, q_{hash}}

在输入 N，e 上运算 F

如果 F 构造随机预言询问（hash,M）

那么 h ← H-Sim(M) ; 返回 h 到 F 作为回答

如果 F 构造随机预言询问（sign,M）

那么 x ← S-Sim(M) ; 返回 x 到 F 作为回答

知道 F 停止没有输出（M，x）

Y’ ← H-Sim(M)

返回 x

逆元应答随机预言询问通过使用合适的子集，一旦声称伪造，构造相应的单向询问和返回签名 x。

现在描述单向随机预言机制，对全局变量是参考，在 I 的主要编码过程中，作为争论，一个值 v 是一些简单的消息，它的单向值要求既然是直接由 F 产生的或者是签名机制，随后通过 F 包括。

将使用一个子集，找到一个给定的列 A，一个值 v 并且索引 m，返回 0，如果 v∉ {A[1],…,A[m]}，并且返回最小的索引 l 因此 v=A[l]。

子集 H-Sim(v)

L ← 找到（Msg,v,j）; j ← j+1 ;Msg[j] ← v

如果 l=0 那么

如果 j=i 那么 $Y[j] \leftarrow y$

那么 $X[j] \leftarrow Z_N^*$; $Y[j] \leftarrow (X[j])^e \bmod N$

结束如果

返回 Y[j]

否则

如果 j=i 那么推出

否则 $X[j] \leftarrow X[l]$; $Y[j] \leftarrow Y[l]$; 返回 Y[j]

结束如果

结束如果

在主要的方式下，单向询问可以回答如下的询问机制。

子集 S-Sim[M]

$h \leftarrow$ H-Sim(M)

如果 j=i，那么退出

否则 返回 X[j]

结束如果

逆 I 也许退出执行，停止每个子程序的说明。首先这种情况是单向预言机制不能返回 y 作为对第 i 个单向询问的应答，因为这个询问等于先前的一个询问的应答。第二个条件是 F 要求消息的签名，是第 i 个单向询问。并且 I 不能提供既然希望第 i 个消息是在伪造中的一个，并且返回 y 作为单向预言机制的应答。

现在希望 RSA 求逆成功的概率降低，也就是等式：

$$Adv_{RSA,I}^{owf} = P[x^e \equiv y(\bmod N) : ((N,e),(N,d)) \leftarrow K; y \leftarrow Z_N^* ; x \leftarrow I(N,e,y)]$$

有几个观察者在等式（9.2）卷入边界的验证。首先在任何时候是 F 的观点，在这种条件下，I 并不停止在相同的经验值 $ForgeExp^{ro}(DS,F)$。这意味着，通过 I 返回 F 的回答是精确分布的，可以在真实的经验值下。第二，F 得到没有信息，关于 I 随机选择的值 i。现在记住上一个单向激励机制，有 I 构造的询问是消息 M 的伪造。因此在执行最后，M 当然是在 Msg 的列中。假设 l=Find(Msg,M, q_{hash})是首先的索引，在这种条件下 M 发生，意味着 Msg[l]=M 但是没有先验消息是 M。i 的随机选择那么意味着有一个 $1/q_{hash}$ 机率产生，那么 i=l，反过来意味着 Y[i]=y，并且单向随机函数不会停止。如果 x 是 M 的正确的签名，将有 $x^e \equiv Y[i](\bmod N)$因为 Y[i]是 H（M）从 F 的观点。因此无论何时发生，I 是成功的。

10.5.9 PSS0：一个安全的提高

FDH-RSA 签名体制，虽然在随机预言机制条件，在 RSA 在单向函数条件下，执行签名过程的特殊安全分布。然而，定理 9.13 给定的安全数量是更好的。定理以概率分叉，一个人可以以概率伪造签名，是 q_{hash} 次概率是 RSA 函数在随机点可以求逆。两个行为可以测量，认为攻击者可以比较执行次数。既然 q_{hash} 可以非常大，比如 2^{60}，有一些可评估的安全泄露。现在有一个体制，其中的安全相关关系是很紧密的：伪造概率签名与对 RSA 是进行可比较 RSA 求逆不是仅仅一点差距。

体制称为 PSS0，对于“概率签名体制，VERSION.0”，强调密钥的一个方面，也就是说是随机的：签名算法选择一个新随机值，每次使用并且计算签名。体制 DS=（K，S，V）有通常在 9.5.3 节中的 RSA 密钥产生算法 K，像 FDH-RSA 使用一个公共的单向函数 $H: \{0,1\}^* \rightarrow Z_N^*$，模仿一个随机预言机制。另外有一个参数 S，签名算法进行随机值选择的长度。写出签名和证实算法如下：

算法 $S_{N,d}^{N}(M)$	算法 $V_{N,d}^{N}(M，\sigma)$
$R\leftarrow\{0,1\}^s$	分解 σ 为（r,x）这里|r|=s
$Y\leftarrow H(r \| M)$	$y \leftarrow H(r \| M)$
$X\leftarrow y^d \bmod N$	如果 $x^e \bmod N = y$
Return(r,x)	那么返回 1 否则返回 0

显然，“范围检查”是简单的，没有在验证码中明确写出。例如，在一个真实的实现中，随后可以检查 1≤x<N，并且 gcd(x,N)=1。

这个体制也许还是可以被观察，在 HASH-求逆的范式，除去 HASH 是随机的，使用签名算法进行一个值的选择。如果两次签署同一个消息，应该得到两个不同的签名。注意随机值必须包括在签名中，既然其他的方法不可能验证签名。因此不象上一级的体制，签名不是 Z_N^*中的数字。是这些成分中的一对是一个 s 比特签名并且其余的是 Z_N^*中的数字，签名的长度是 s+k 比特。对于确定的 HASH-求逆签名体制在长度上要长一些。通常 l 的长度设置为 160 比特，给定的 k 可能是 1024，长度的增加是可容忍的。

定理 9.14[26]假设 DS 是 PSS0 体制，安全参数是 k 和 s。那么对于任意 t, q_{hash} ,μ，并且任意 $q_{hash} > q_{sig} + 1$，有：$Adv_{DS}^{uf\text{-}cma}(t, q_{sig}, q_{hash}, \mu) \le Adv_{RSA}^{owf}(t') + (q_{hash} - 1)\cdot q_{sig} / 2^s$
这里 $t'=t+ q_{hash} \cdot O(K^3)$

也就是说，$q_{hash} = 2^{60}$ 并且 $q_{sig} = 2^{40}$ ，通过 l=160 上述的附加条件是大概 2^{-60} ，是非常小的。因此对于所有时间目的附加条款是可忽略的，并且 PSS0 的签名体制相对于 RSA 是紧的。

继续定理 9.14 的证明，在随机预言机制中给定一个攻击 DS 的伪造，使用资源 t, q_{sig} , q_{hash} , μ，对于 RSA 求一个逆 I，因此：

$$Adv_{RSA,I}^{owf} \ge Adv_{DS,F}^{uf\text{-}cma} - (q_{hash} - 1)\cdot q_{sig} / 2^s \qquad (9.3)$$

此外，I 将使用在定理描述中给定的值 t’运算时间的边界。现在定理通常如下，并且对于一些算数运算和采用最大值。

I 的设计是跟随相同的结构，使用 9.13 的证明。也就是 I，输入 N，e,y 将在输入 N，e 条件下执行 F，回答 F 的预言询问，因此 F 可以完成执行。从伪造，I 将在某种程度上找到 $y^d \bmod N$，I 将反馈 F 的单向随机询问，通过一个子集 H-Sim 称为单向预言模拟。设计的一大部分是这些子集的设计，为了得到一些直观的知识，希望返回 9.13 的证明。

在证明中可以看到，在等式（9.2）中，来源于 I 的假设，使用一个随机值 $i \in \{1,\ldots, q_{hash}\}$，$q_{hash}$ 的乘法因子，希望 $i=Find(Msg,M, q_{hash})$，这里 M 是一个伪造的消息。也就是说，必须假设时间，在消息伪造前必须先进行单向函数询问。

可以看到的最好情况是，这种猜测正确的概率是至少 $1/ q_{hash}$ 。然而如果现在想 I 的成功概率是在等式 9.3，猜测的时间是达不到的，在进行单向压缩后，伪造消息。当然不知道提前完成的时间。但是，I 不得不采取伪造的优势，但是返回 $y^d \bmod N$。

一个简单的办法是返回 y 作为所有单向函数的询问。那么一个询问消息产生期望值 $y^d \bmod N$，考虑 FDH 的策略，在那种条件下，存在两个问题。首先，这些回答也许不是随机独立的，作为回答的询问是单向询问。第二，如果询问中的一个消息是随后一个签名询问，I 也许没有办法回答签名询问。（记住，对于 j≠i 作为 $(X[j])^e \bmod N$，计算一个随机询问 Msg[j]，特别的为了能够随后返回 X[j]，如果 Msg[j]显示一个签名询问。但是有一个矛盾，I 可以运算这点，也可以返回 y，但是不能两者相同。如果必须选择，在 FDH 条件下，可以随机选择）。

第一个难题实际容易用一个小的代数欺骗，这种开发称为 RSA 的自还原。当希望 y 作为单向随机预言机制 Msg[j]的回答，在 Z_N^* 中随机选择 X[j]并且返回 $Y[j]=y\cdot(X[j])^e \bmod N$，

值 X[j]每次独立随机选择。现在的事实是 $RSA_{N, e}$ 是一个置换，意味着所有不同的 Y[j]值是独立随机分布。对于签名询问要求构造 r ‖ M 的回答，$Y[l]=y\cdot(X[j])^e \bmod N$。那么有（$x\cdot X[l]^{-1}$）≡y(mod N)，因此 y 的逆是 $x\cdot X[l]^{-1} \bmod N$。

然而，第二个难题在于，不能用 FDH 解决。也就是说在 HASH 之前，PSS0 在解决随机值 r 到消息。有效隔离两种 HASH 询问，F 对单向预言机制的直接询问和间接的单向随机询问是从签名预言机制中产生的。直接的单向预言询问对于一些比特串 r 有形态 r ‖ M 和一些消息 M，签名询问是消息 M。为了回答，首先随机选择一个值 r，当值 r ‖ M 有先验概率的特征是低概率的。因此，有时任意新的单向询问产生，可以假设将不会是一个间接询问，因此通过上述的陷门回答。以下是完整的证明。

定理 9.13 的证明：假设 F 是 PSS0 的伪造攻击，资源参数 t, q_{sig} , q_{hash} , μ 测量与经验值 $ForgeExp^{ro}$(DS,F)作为上述讨论。设计一个 RSA 函数逆 I 使得等式（9.3）是真的，并且在定理描述中，I 的运算时间是根据值 t’进行限制的，定理如下：

首先描述两个子程序，一个单向预言模拟 H-Sim 和一个签名预言模拟 S-Sim（·）。保持四个表 R，V，X，Y，使用索引的每个列范围从 1 到 q_{hash}，所有这些是全局变量，也可以用于子程序。排列的有意向的登录是 j=1，…， q_{hash}，

V[j] – 第 j 个基于经验的单向询问，有表格 R[j] ‖ Msg[j]

R[j] – 第 V[j]的第 l-bits

Y[j] – 值扮演一个 H（V[j]）的角色，使用单向模拟选择或使用签名模拟

X[j] – 如果 V[j]是一个直接的 F 单向函数预言机制满足 $Y[j]\cdot X[j]^e \equiv y(\text{mod } N)$。如果 V[j]是一个间接的单向预言询问机制满足 $X[j]^e \equiv Y[j]$（mod N）,意味着是一个 Msg[j]的签名。注意并不实际需要的储存列 Msg，仅仅指出上述解释。

一个子集找到一个给定的 A，一个值 v 和索引 m，返回 0 如果 v∉ {A[1]…A[m]}，否则返回最小的索引，因此 v=A[l]。

逆 I（N，e,y）

初始化列 R[1，…，q_{hash}]，V[1，…，q_{hash}]，Y[1，…，q_{hash}]，X[1，…，q_{hash}]清空

J←0

在输入 N，e 是运算 F

如果进行预言询问（hash,v）

那么返回 h←H-Sim(v);返回 h 到 F 作为回答

如果 F 进行预言询问（sign,M）

那么 σ←S-Sim(M)；返回 σ 到 F 作为回答

直到 F 输出（M,(r,x)），停止

y←H-Sim(r ‖ M)；l←Find(V, r ‖ M, q_{hash})

$w\leftarrow x\cdot X[l]^{-1} \bmod N$；返回 w

现在描述单向预言模拟，进行在 I 的主要代码的全局变量模拟。作为争辩，一个值 v 假设至少是 s 比特长度。意味着形如 r ‖ M 对一些 s 比特串 r（不必要考虑不是这种形式的单向询问，既然不是与签名体制相关的）

子集 H-Sim(v)

分解 v 作为 r ‖ M，这里|r|=s

L←Find(V,v,j) ;j←j+1 ;R[j]←r ;V[j]←v

如果 l=0 那么

$X[j]\leftarrow Z_N^*$； $Y[j]\leftarrow y\cdot(X[j])^e \bmod N$;返回 Y[j]

否则

X[j]←X[l] ;Y[j]←Y[l] ;返回 Y[j]

结束如果

每个串 v 的单向预言询问是通过这个子集输入表 V，因此 V[j]是在 F 中执行的第 j 个单向预言询问，如下的签名模拟是没有调用单向函数模拟。如果必要，可以自己填充必要的表格。

子集 S-Sim(M)

R←$\{0,1\}^s$

L←Find(R,r,j)

如果 l≠0，那么中止

否则

J←j+1 ;R[j]←r ;V[j]←r ‖ M ;X[j]←Z_N^* ;Y[j]←$(X[j])^e$ mod N

返回 X[j]

结束如果。

现在在逆 RSA 时，需要 I 的成功概率是小的边界，量化为：

$$Adv_{RSA,I}^{owf} = P\,[x^e \equiv y(\text{mod } N) : ((N,e),(N,d))\leftarrow K ; y\leftarrow Z_N^* ; x\leftarrow I(N,e,y)]$$

逆 I 中止执行是因为中止指令是在签名预言模拟机制。如果在签名预言模拟机制产生的随机值 r 在集合{R[1]，…，R[j]}中出现。这个集合的大小最多为 q_{hash} -1 在一个签名询问的时间，因此 r 落入集合的概率至多是（q_{hash} -1）/2^s，签名预言模拟最多调用 q_{sig} 次，因此在执行 I 同时中止的概率至多是（q_{hash} -1）$q_{sig}/2^s$ 。

在任意时间的 F 的概率是 I 不能中止是与 F 在经验值 $ForgeExp^{ro}$(DS,F)的观点是相同的。这意味着用 I 返回 F 的回答是精确分布的，可以作为真实的经验值。

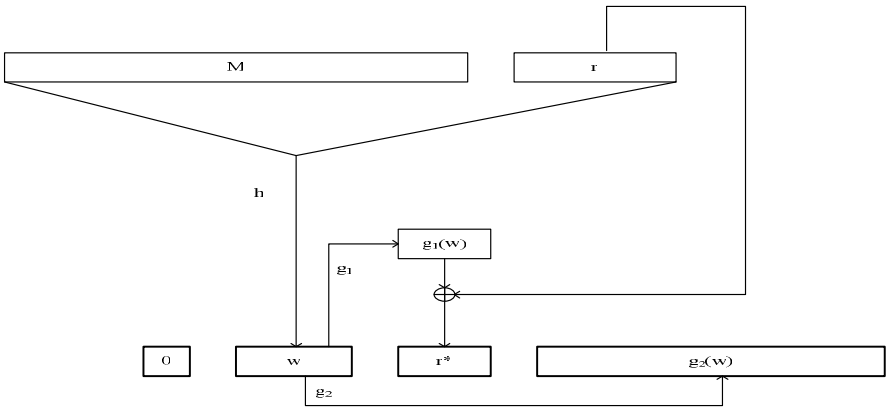

图9.1 PSS:构成为y=0| w| r*| g2(w)是在黑框中的,M的签名是y^d mod M

现在记住，最后的 I 提出的 HASH 模拟询问是 r ‖ M，这里 M 是消息的伪造，因此 r ‖ M 是 I 在阵列 V 最后的运行。因此，l=Find(V, r ‖ M, q_{hash})，可以知道通过签名模拟不能 r ‖ M 不能归入 V。这就意味着 Y[l]=y·$(X[l])^e$ mod N，因为通过这种方式，单向模拟询问选择回答。伪造正确性意味着 x^e≡Y[l] ≡y·X[l](mod N)。解决这个给定的（x·$X[l]^{-1}$）^{e} mod N=y，并且在返回（x·$X[l]^{-1}$） mod N 时是正确的。

10.5.10 概率签名体制-PSS

PSSO 在 FDH-RSA 上获得提高了的安全体制但是提高安全的代价在于增加签名大小的消耗。现在的体制减少了签名的大小，因此，既有提高了的安全又有 FDH-RSA 相同的大小。这是[24]的概率签名体制。

签名体制 PSS[k_0,k_1]=(K,SignPSS,VerifyPSS)是一个 k_0，k_1 的参数，是 1 到 k 的数字，满

足 $k_0+k_1 \le k-1$。为了更具体的，阅读这必须考虑 k=1024，$k_0=k_1=128$。这个体制有在 9.5.3 节产生通常的 RSA 密钥算法，签名和验证算法是两个单向函数的使用。首先 h 是压缩函数，映射为：$h:\{0,1\}^* \to \{0,1\}^{k1}$。并且第二个 g，称为产生器，映射为：$g:\{0,1\}^{k1} \to \{0，1\}^{k-k1-1}$。（分析假设这些是理想的，事实上，可以用密码学上简单的单向函数 MD5 进行实现。如同在附录 9.5.11 上的讨论。）假设 g_1 是函数，输入是 $w \in \{0，1\}^{k1}$，返回 g(w)剩余的 $k-k_0-k_1-1$ 比特。现在描述怎样签名和验证。参考图 9.1，写签名并验证算法如下：

算法 $SignPSS_{N,d}^{g,h}(M)$

$R \leftarrow \{0,1\}^{k0}; w \leftarrow h(M \| r)$

$R^* \leftarrow g_1(w) \oplus r$

$Y \leftarrow 0 \| w \| r^* \| g_2(w)$

$X \leftarrow y^d \bmod N$

返回 x

算法 $VerifyPSS_{N,e}^{g,h}(M,x)$

$y \leftarrow x^e \bmod N$

分解 y 作为 $b \| w \| r^* \| \gamma$ 这里

$|b|=1, |w|=k_1$，$|r^*|=k_0$

$r \leftarrow r^* \oplus g_1(w)$

如果（$h(M \| r)=w$ 并且 $g_2(w)=\gamma$ $b=0$）

那么返回 1 否则返回 0

显然“范围检测”是简单或者不用明确的在验证码中描写；例如，随后一个真实的实现应该检查 $1 \le x < N$，并且 gcd(x,N)=1。

步骤 $r \leftarrow \{0,1\}^{k0}$，指示签名者随机选择一个 k_0 比特的种子 r，然后连接这个种子到消息 M 后面。有效随机化消息，进行单向描述成为 k_1 比特的串 w。那么产生器 g 应用 w 产生一个 k_0 比特串 $r^* = g_1(w)$并且一个 $k-k_0-k_1-1$ 比特串 $g_2(w)$。首先使用 k_0 比特掩码种子 r，产生掩码种子 r^*，现在 $w \| r^*$ ，是使用 0 比特预先计算，并且添加 $g_2(w)$产生图象点 y，用于 RSA 函数定义签名条件下的解密。（0 比特保障 y 在 Z_n^*）

注意一个新的种子是每个消息的选择。特别的，一个给定的消息有许多种可能的签名，依靠签名者选择的 r 的值。给定（M，x），验证者首先计算 $y=x^e \bmod N$ 并且恢复 r^*，w，r。通常检查 y 可以正确的构造，如果所有检查成功，仅仅接受验证者。

注意已声明体制的效率，签名采用 h 的一个应用。两个安全机制中的关系是非常相同的，对于 PSS0，在定理 9.14 中可以看到，意味着本质是紧的，并且对于一个 FDH 体制是更紧的。这次可以不增加签名长度得到进展。

定理 9.15[26]：假设 DS 是使用参数 k_0，k_1 的 PSS 体制，那么对于任意的 t, q_{sig}，μ 并且任意 $q_{hash} \ge 1+ q_{sig}$，有：$Adv_{DS}^{uf\text{-}cma}(t, q_{sig}, q_{hash}, \mu) \le Adv_{RSA}^{owf}(t') + [3[q_{hash}-1]^2] \cdot (2^{-k0} + 2^{-k1})$ 这里 $t'=t+ q_{hash} \cdot k_0 \cdot O(k^3)$。证明是在[26]。扩展上述给定的 9.14 证明。

10.5.11 使用消息恢复 PSS-R 签名

消息恢复：在一个标准签名体制中，签名可以清晰地传输消息 M，粘上签名 x。在一个体制中，提供消息恢复，仅仅一个“加强签名”Γ 被传输。目标是保存签名消息的带宽，希望这个签名消息的长度比 |M|+|x| 小。（特别的，当 M 小时，希望 Γ 的长度是 k，签字长度。）从加强签名和同时检查认证中恢复消息 M。

从消息的可压缩部分完成签名，在这种渠道下，是验证者可恢复的。当 M 的长度 n 是小的，事实上，可以把所有消息压缩进签名，因此仅仅 k 比特的变量被传输。在如下的体制内，如果安全参数是 k=1024，可以压缩 767 比特到签名中。

定义：正式的，密钥产生和签名算法与之前相同，但是 V 被恢复者代替，采用 pk 和 x 返回 $Recover_{pk}(x) \in \{0,1\}^* \cup \{REJECT\}$。不同的点 REJECT 是用来指示接收者拒绝签名，一个 $M \in \{0,1\}^*$ 的返回值支出验证接收者作为可信方接受消息 M。安全公式是相同的期望值，意味着伪造者可以成功。应该提供一个 x，使得 $Revover_{pk}(x)= M \in \{0,1\}^*$，这里 M 不是一个

先验的签名询问。要求如果 x 通过 $x \leftarrow S_{sk}(M)$那么 $Revover_{pk}(x)= M$。PSS 的一个简单的变量完成消息恢复。现在描述体制和安全性。

配置：体制 RSS-R$[k_0,k_1]$=(K,SignPSSR,RecPSSR)是与以前相同的 k_0,k_1 参数。密钥产生算法是 k，与以前相同。与 PSS 相同，签名和验证算法依靠单向函数 $h: \{0,1\}^* \rightarrow \{0,1\}^{k1}$ 并且 $g: \{0,1\}^{k1} \rightarrow \{0,1\}^{k-k1-1}$。使用相同的 g_1,g_2 概念。为了简化说明，消息可以以长度 $n=k-k_0-k_1-1$ 的进行签名。（建议参数的选择是 k=1024，$k_0=k_1=128$ 并且 n=767）在这个条件下，产生仅有 k 比特的加强签名，从该签名验证者可以恢复 n 比特消息同时检查认证。签名产生和认证过程如下，如图 9.2 所示：

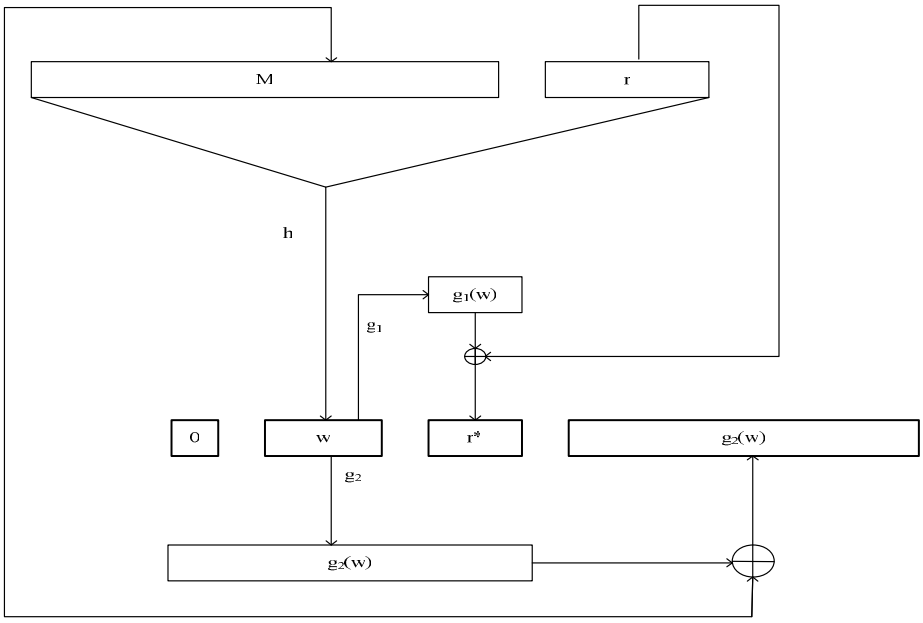

图 9.2 PSS-R:构成为y=0| w| r*| M*是在黑框里

算法 $SignPSSR_{N,d}^{g,h}(M)$

$R \leftarrow \{0,1\}^{k0}; w \leftarrow h(M \| r)$

$R^* \leftarrow g_1(w) \oplus r$

$M^* \leftarrow g_2(w) \oplus M$

$Y \leftarrow 0 \| w \| r^* \| M^*$

$X \leftarrow y^d \bmod N$

返回 x

算法 $RecPSSR_{N,e}^{g,h}(x)$

$y \leftarrow x^e \bmod N$

分解 y 作为 $b \| w \| r^* \| M^*$这里

$|b|=1, |w|=k_1, |r^*|=k_0$

$r \leftarrow r^* \oplus g_1(w)$

$M \leftarrow M^* \oplus g_2(w)$

如果（$h(M \| r)=w$ 并且 b=0）

那么返回 M，否则返回 REJECT

在 SignPSSR 中的签名是与 SignPSS 相应的，也就是 y 的最后部分不是 $g_2(w)$。作为代替，$g_2(w)$使用消息的掩码，并且掩码消息 M^*是映像点 y 的最后一部分。上述是简单的采用任意长度处理消息，一个充分说明体制能够使用大约 $\min\{k,k+k_0+k_1+16\}$比特。

安全：PSS-R 的安全是与 PSS 相同的

定理 9.16[26] 、假设 DS 是 PSS 使用安全参数 k_0，k_1 的恢复体制，那么对于任意 t, q_{sig}, μ 并且任意 $q_{hash} \geq q_{sig}+1$，$Adv_{DS}^{uf\text{-}cma}(t, q_{sig}, q_{hash}, \mu) \leq Adv_{RSA}^{owf}(t') + [3(q_{hash}-1)^2] \cdot (2^{-k0} + 2^{-k1})$

这里 $t'=t+ q_{hash} \cdot k_0 \cdot O(k^3)$

这个定理的证明与定理 9.15 非常相同。

10.5.12 怎样完成单向函数

在 PSS 体制中，需要一个具体的单向函数 h，对于一些给定的数 k_1 是输出长度。典型的将从一些密码学单向函数 H 构造 h，例如：H=MD5 或者 H=SHA-1，做这件事情的途径是在[15，33]中进行了讨论的。对于完整性，快速总结这些概率的部分。最简单的是把 h(x)定义适当长度的定义：H（const.<0>.x ‖ H(const.<1>.x) ‖ H(const.<2>.x ‖ …常数 const 对 h 是唯一的，为了构造另外的单向函数 g，简单选择一个不同的常数。

10.5.13 与其他体制的比较

已经讨论了 PKCS 标准[179，180]并且 ISO 标准[1]可以看到的标准是不充分的，基于 RSA 是单向陷门函数的假设。另外的标准，例如[9]，与[179]类似，是相同的陈述申请。在这一节剩余的章节中，讨论体制，不用使用 HASH 再加密范式。

签名体制的安全是基于包括[105，14，152，177，78]的 RSA 假设是可证明的。这些工作的主要附加条件是不使用理想单向函数（随机预言）模型，可证安全是标准的理解。在另一方面，是计算有效的。采用两个到六个 RSA 计算，尽管有一些高架的存储，签名体制比一个单独的 RSA 系数长。这个体制是最通用的选择，如果一方面可以允许一些其他的计算和存储。这个体制是最好的，没有假设一个理想单向函数，希望有一个好的判断安全。

返回签名体制，假设一个理想 hash，基于因式分解或其假设的难度，大数据量被提出。这些体制的大多数起源于认证体制，首先通过[83]进行构造。这些方法的部分是可证明的（在理想单向模式），一些不是，证明体制严格安全的部分条件被分析；通常不是。在没有的条件下，知道的是安全加强。有效变量。计算需求通常比单向然后解密的 RSA 体制低，尽管密钥长度典型较大。

最后，可以注意相关的新的工作， Pointcheval 和 Stern[165]认为随机预言机的签名是可证安全的，可以看做一个 Elgamal 体制[90]的可调整模型，与 Schnorr[184]体制，可以被证明是安全的。（并且[83]的体制是可以证明在没有询问条件下的抗攻击是安全），但是不能认为是特别安全的，一个有趣的问题是去考虑，或者提高，特别是的归约是特别安全的（发展，如果必要，体制调整）。

更进一步，一些基于签名的非常简单的 RSA 出现了，给出了一个基于加强和关于 RSA 减少标准假设，但不仅仅是依赖于随机预言[92，65]。

10.6 极限签名体制

使用极限签名体制、数字签名产生一组参与者比仅仅使用不当一个参与者更加合适。相对于正规签名体制，签名者是一个签名实体，用来保存密钥，在极限签名体制中，秘密密钥被一组的多个参与者共享。在给定的消息 m 中，为了产生有效签名，个体参与者在签名时，产生部分签名，那么整合为 m 上的所有签名。一个分布式签名体制达到极限 t<n，如果不是 t 个参与者的连接可以产生一个有效的签名，甚至在系统产生许多不同消息的签名。从极限签名产生的签名是通过单独的拥有完全私钥的签名者对签名过程是相同的。特别的，签名的有效性可以通过任何一个人验证，使用唯一公共验证密钥进行通信。换言之，签名的事实是在一个分布方式上产生相应的签名。

极限签名是在一些组织中有一群雇佣者同意在签署消息前给定消息进行推动，对于内部

和外部的攻击者与保护需求相同（例如政府认证、银行、签名认证），必须邀请攻击者尝试消耗这个能量。一个极限签名体制的目标是双重的：为了增加签名代理的有效性，并且同时增加保护对于攻击者在难于得到秘密密钥条件下进行伪造攻击，基于一个传统的密钥共享的极限逼近规律（见 11 章）是一个简单的解决办法。这里秘密密钥共享是组内共享，但是每次对于一个单独的签名者，可以产生签名。这种协议是与签名矛盾的，没有 t 个参与者可以产生一个新的有效签名。在极限体制中，在没有准确的秘密密钥公开和重构造的条件下可以产生多方签名。

极限签名体制是普通进展的一部分，发展成为极限密码学。在文献中，这个进展受到了合理的关注。介绍读者[70]作为这部分工作的综述，这个解决方案特别的例子是在[69，183]中描述的极限密码体制包括了 RSA，在[109]中包括了 Elgamal 体制。一个极限简明体制称做是充分的，如果不仅仅是 t 个或者更少参与者不能产生签名，但是也不能防止剩余的参与者通过自己持有的密钥进行签名。一个基于拒绝服务攻击的阻止概率有效的体制是对部分过失服务器的攻击，上述提到的解决是不充分的。在这一章，将集中充分的体制。将不给出技术细节。这一节的目的是介绍读者相关的概念和支出文献的来源。

10.6.1 极限密码体制的密钥产生

对于极限密码体制产生密钥的任务是比单独的签名体制者更复杂的。事实上，必须产生一个公钥 PK，与一个私钥 SK 是相应的，在服务器 P_1，…，P_n 进行某种程度上的共享。做这件事的一个方法是对于给定的签名体制，有一些信任的中间人可以产生一个密钥对（PK，SK），使用秘密签名协议在 P_i 中进行 PK 公钥和 SK 共享。然而注意这样一个签名体制与要求是矛盾的，没有个体可以进行签名，现在中间人知道私钥 SK 和可以签署自己的文件。这就是为什么大众可以试着在密钥产生阶段回避中间人的处理。对于对数签名体制的条件，这个要求是成功的。对于（[90，184，185]）中的 El Gamal, Schonorr 和 DSS 充分签名体制可以在[53，159，94]中发现。在 Feldman 和 Pedersen[82,160,161]中的结果下使用。

但是在一些条件下，中间解决是可以做到最好的。例如，在签名体制是 RSA 的条件下，在没有中间人使用的条件下就不知道怎样产生密钥。

10.6.2 签名协议

一旦密钥产生，在某些条件下在服务器中 P_1，…，P_n 进行共享，需要一个签名体制。思想是在输入消息 M 条件下，服务器产生通信的一些模式，将允许对 M 计算一个签名 σ，不用使用秘密密钥。这种协议不能泄露这种签名体制 σ 下的信息。也是为了获得充分的特征，这样的协议应该正确的计算签名，甚至 t 服务器 P_i 是损坏的和在协议中以任何方式进行运做。如果可能，计算方要求服务器 P_i 进行分布式签名，应该与有效签名可进行比较，如果 P_i 可以用自己进行签名。互作用应该减到最少。对于例如 El-gamal 体制充分的极限签名可以在[53，159]中发现。DSS 的特殊说明是非常难处理的，最好的解决在[94]中。

RSA 变化为甚至更少的复杂体制构造的原因，在某种程度上是无效解决（要求更多的计算和服务器之间更多的计算），可以在[86]中找到证明。一个非常有效和非交互的解决是在[93]中独立提出的。

第十一章 密钥分发

可以开阔的看一个加密体制和数字认证，对于者些任务使用多种方法设计体制。必须在这些体制下致力研究一个假设，这是可以用到其密钥的假设。对于密钥分布和密钥管理，这一章检查不同的方法，大量的努力在于理解者各领域最重要的实践困难，也就是说会话密钥分发。开始使用经典的 Diffie 和 Hellman 秘密密钥交换协议。

11.1 Diffie Hellman 秘密密钥交换

假设 Alice 和 Bob 没有密钥（分享或公用），并且想出现一个公共密钥，可以使用私钥密码。用 Diffie Hellman 秘密密钥交换协议[72]完成。

11.1.1 协议

固定一个素数 P，一个生成元 $g \in Z_p^*$ 。这些是公共的，不知道仅仅是所有方还有攻击者 E。

- A 随机选择 $x \in Z_{p-1}$ ，假设 $X=g^x \bmod p$ ，发送 X 到 B
- B 随机选择 $x \in Z_{p-1}$ ，假设 $Y=g^y \bmod p$ ，发送 Y 到 A

现在注意 ：$X^y = (g^x)^y = g^{xy} (g^y)^x = Y^x$

运算在 Z_p^* 中进行，假设叫这个共同的数量为 K，关键的问题是两者都可以进行计算。也就是 A 计算 Y^x ，是 K，并且 B 计算 X^y ，也就是 K，现在有一个共享密钥。

11.1.2 抗窃听安全：DH 难题

这是安全的吗？考虑一个攻击者，利用电线并且可以看到通过的电流。想要计算 K，她所看到的是 X 和 Y，但是知道既不是 x 也不是 y, 怎样得到 K？自然的攻击是既找到 x 也找到 y，从这一点可以简单的计算 K。然而，给定 X 注意计算 x 仅仅是 Z_n^* 上的离散对数算法难题。是广泛的相信是难处理的（适合素数 P 的选择）。类似从 Y 计算 y。相应的，相信这种攻击失败是正确的判断。

现在出现了版本的数据，首先是计算离散对数算法，现在不是仅仅从 X，Y 恢复 K，也许有其他的。为了检查其版本，假设阐明攻击者的计算难题尝试解决。如下：DH 难题：在 Z_{p-1} 中随机选择 x，y，给定 g^x 并且 g^y 计算 g^{xy} 。

因此这个问题是这个问题有怎样的难度？可以看到是否在 Z_p^* 上的离散对数难题是容易的因此是 DH 难题。例如，如果可以计算离散对数难题，可以解决 DH 难题。相反是真的吗？剩余的问题是公开问题。定时间是可行的，有一些聪明的进展是不用计算离散对数难题的。然而，还没有发现这样的进展。对解决 DH 难题最好的解决方法是计算 X 或 Y 的离散对数。这使得解译码者相信 DH 难题，尽管没有人知道离散难题相当的计算。还是一个计算难题，作为结论，DH 秘密密钥交换协议是安全的，在某种程度上，一个计算边界不能计算多方分享的密钥。

DH 假设：DH 难题是难以计算的

目前素数 p 推荐至少是 512 比特和也适合 1024 比特，作为已经看到的，为了假设离散

对数难题。已经看到，为了使模 p 的离散对数难题是难以处理的，p-1 至少有一个大的素数。在实践上，对一些大素数 q，采用 q=2p+1。在 DH 难题和离散算法难题的关系上是这样的研究目标，见例子 Maurer[141]。

11.1.3 DH 加密体制

DH 秘密密钥交换给出了一个非常方便的公共密钥体制。一个部分 A 随机选择点 $x \in Z_{p-1}$，并且假设 $X=g^y$ 是公钥。现在如果 B 方希望私下发送 A 一个消息，可以按照如下执行：首先，多方同意一个私钥加密体制（E，D）（cf,第六章）。对于具体的假设，是一个基于密码体制的 DES，因此需要使用 56 比特。现在 B 从 Z_{p-1} 随机选择 y，并且计算 DH 密钥 $K=X^y=g^{xy}$，从这一点，根据一些约定的协议，取出 56 比特作为秘密密钥加密体制，例如假设 K 的最初 56 比特。现在 a 下加密明文 M，使用私钥体制得到密文 $C=E_z(M)$，并且传输对（Y，C），这里 $Y=g^y$。

一个接收者（Y，C），使用秘密密钥 x，可以计算 DH 密钥对 $K=Y^x=g^{xy}$ ，因此恢复 a。现在可以根据私钥体制进行 C 解密，通过 $M=D_a(C)$，并且因此恢复明文。直观的，安全基于事实攻击者不能计算 K 并且因此是 a。然而，这是非常不正确的。这证实关于 DH 的安全的看法是有一些不正确的。

11.1.4 DH 密钥的比特安全

上述密钥的前 56 比特，$K=g^{xy}$ 是对私钥加密体制作为密钥。现在知道的（进行假设）是给定 g^s，g^y，攻击者不能恢复 K。对于私钥加密体制安全作为 K 是不够的，如果攻击者可以恢复密钥 K 的前 56 比特，但不是所有的密钥。那么肯定上述密码体制是不安全的，那么有密钥的前 56 比特不能找到 K，因此不与 DH 体制矛盾。

这是一个问题，在许多章的前后节，例如单向函数并且使用加密体制。这是部分信息的难题。如果 *f* 是单向的，意味给定 f(x),不能发现 x，不是不能找到一些比特。类似的，不能计算 K 不意味着不能计算 K 的一些比特。

事实上，可以计算 K 的最后几比特=g^{xy}，在给定 g^s，g^y 时是容易的。为了定义这些，不需要看到其部分信息的其可看穿的泄露。虽然如此，是不聪明是做为秘密密钥体制，在 DH 的私钥体制中使用这些比特的子集。假设这些比特是安全的是比 DH 假设更强的假设。什么是应该做的？实际上，也许单向 DH 密钥 K 得到对称密钥 a。例如，使用一个密码假设函数，例如 SHA-1 到 K 产生 160 比特也许有比 DH 体制有更好的随机特征，如果使用 DES，首先考虑 56 比特。

然而，上述也许在实际使用里好的启发式结构，如果没有单向函数的随机特征假设就不是有根据的。一个概率可以是有效的，通过定理 2.49 的一个对话 DH 体制。名义上，假设 r 是长度|p|的随机串，假设 b 是 K 和 r 的句点产生。预言 b 给定的 g^x，g^y 是不可行的，如果计算 $K=g^{xy}$ 在给定 g^x，g^y 是不可行的，这个进展的障碍在于一方给出非常少的比特。为了得到 56 比特，一方应该需要变换几个 DH 密钥，互相得到几比特。

在第二章看到，对每个固定的单向函数，可以出现核心预言，预言可以减少逆函数的难题。一个 DH 密钥交换协议类似的法则是好的，可以指示怎样使用对称密钥求函数的逆。最近这种结论别是 Boneh, Venkatesa[46]。

11.1.5 认证缺乏

首先看到，DH 秘密密钥交换也许看来解决对密码学的密钥获得难度。如果 A 希望和 B 获得共享密钥，能够仅仅做一个 DH 密钥交换得到一个，然后使用私钥密码学。不要做这种事情。困难是真实的，DH 密钥的安全是抗被动攻击，或者窃听。假设攻击者将恢复数据传输，但是不试着在网络上注入消息，黑客可以连接任何攻击。这样做的损失是什么？有攻击者进行的。可以呼叫 B，并且简单的 A 的角色。也就是，声称是 A，有某人 B 可以分享密钥，然后象 A 一样执行 DH 协议。也就是，随机选择 x，发送 $X=g^x$ 到 B。B 返回 $Y=g^y$ 并且现在 B 和攻击者共享密钥 $K=g^{xy}$ 。但是 B 想与 A 进行密钥共享。也许使用 K 加密机密数据，那么攻击者也许会恢复数据。

因此在一个主动攻击者的现实模型中，DH 密钥交换不直接使用。真实的困难是在身份识别模式下交换一个密钥，也就是说现在返回。

然而，注意在 DH 密钥交换时不是一个解，但是本身密钥分布难题是在一个活动攻击者的条件下的，这是一个有用的工具。将看到怎样与其工具使用，将发展到添加会话密钥分发协议，具有一些好的特征，例如前向安全。

11.2 会话密钥分发

假设现在是在活动攻击者的条件下，攻击者可以在线注入消息，通过合法方法改变消息发送，另外窃听通信。希望得到共享密钥。一点思想是清楚的，如果合法方信息攻击者都知道，没有可能交换攻击者知道的密钥。因为攻击者仅仅模仿一方到另一方，象是上述 DH 攻击。因此，为了获得基础，合法用户需要信息优势。有一些信息，在可信信道上预先分发这些信息，攻击者不知道，确保将来进行安全密钥交换。现在讨论几种方法认识信息优势，给定会话密钥分发难题。用更深的方法解释难题，部分见[16，17]。

11.2.1 可信模型和密钥分发难题

什么格局可以产生优势？有几种不同的可信模式并且相应的密钥分布难题。

三方模型

这个模型首先是 Needham, Schroeder[154] 提出的，比 Kebero 模型[199]更普及。在这个模型里，有一个可信方叫做认证服务器，指示 S。在这个系统下，每一方 A 有一个密钥 K_A，可以用服务器共享。这是两方的私钥，不知道任何一方。当两方 A、B 分别共享，使用 S 的密钥 K_A，K_B，希望从事一个通信过程，一个三方协议执行，包括 A、B 和 S。结果是发行一个共同密钥 K 给 A 和 B。可以使用这个密钥加密和鉴别互相通信的数据。

分布式密钥假设作为一个安全会话密钥，当多方完成通信过程时，将丢弃密钥 K，如果随后希望有其他的通信过程，三方协议是重新执行和得到一个新的、新鲜的会话密钥。什么安全特征是分布式会话密钥应该具备的？随后将更深一步的看到这个问题，这是一个重要的问题，既然，作为应该看到的，会话密钥分布协议是坚持几种新颖的攻击。

两方非对称模型

当公钥密码体制使用时，认证服务器活动的角色能够被消除。在这种可信模型下，假设是 A 有 B 的公钥 pk_B，B 有 A 的公钥 pk_A，这些密钥假设是可信的。也就是说 A 假设是 B 的公钥真正持有者，不是另外一些人的，并且类似 B 的情况。P203 现在假设 A 和 B 希望从事安全通信会话，希望考虑的问题是怎样通过双方协议，能够得到一个共享私钥和认证过程会话密钥。

关于如何精确的问题是难题，一个是私钥密码，至少在当前的技术下是比公钥密码更可行的。第二，然而，也许更重要是方便的得到会话密钥。允许一方共同使用唯一的密钥，这是如下原因的优势。

密钥实际上使用认证和加密数据得到更多的泄露，也许还不知道可以得到应用。或者预先可控。特别的，一个应用也许错误使用一个密钥，或者暴露。这也许是，也许不是当前密钥会话中的折衷，但是不能希望折衷长期的秘密密钥，因此其使用和会话。类似的，用户 A 的长效秘密密钥（也就是，秘密密钥 sk_A 相应的是公钥 pk_A），也许在存储中保护装置并且仅仅通过特殊界面，当会议密钥基于更公开的机器。

两方对称模型

大概最简单的模型是两方已分享一个长效密钥，每次在通信会话中，希望起源于一个会话密钥，运行一个协议。

再次，动机是方便的并且会话密钥是优势安全的，强调主要的一种。一个应用主机应该使用运行，所有对于一个原因到另一个原因希望的密钥。不想构成关于怎样使用密钥的一个假设。也许在这个方面对自己的目的是对全局变量的折衷，为了不影响全局安全，分配每个隔离会话密钥。

11.2.2 会话密钥分布历史

也许会话密钥分发式老问题，仅仅最近一个密码似乎可以处理，在可证安全或归约成为传统，这些文件注记是可描述的，合并在[16，17]。通过这种进展，现在有模型去讨论和证明正确的协议。并且几种协议是可证明安全的，在标准密码的假设条件下。

优先于这点的是困难的。会话秘钥分布是一个领域，大量的文章已发表，计划协议去解决问题。然而，许多随后可以破解，有可辨别的设计缺陷。

问题貌似简单的。容易产生协议，可以不断地加入细致安全问题。

在三方条件下，Needham, Schroeder[154]描述了候选协议的数量。在文章结束预言，对这种进展提出了警告，也就是说，这种协议的进展倾向于极端细致的错误，再普通探测下未必是可以得出结论的。技术的需求是验证这个协议的正确性…”，当在协议 1 中指出漏洞时，作者的证据声称是出乎意料的，许多相关的协议最后面临同样的命运。

作为这种攻击的长的历史的结论，有最后的普通意见，会话密钥分发，不是一个充分的目标，给定协议，对于设计者可以发现没有攻击。

这个工作的主要部分，由 Burrows,Abadi, Needham[49]，通过使用特殊的逻辑，目标是提高这个位置。目标是在分析的协议中，证明缺乏“论证难题”。在各种协议中帮助找到错误的技术，但是一个证明协议是“逻辑正确”的。暗示这是正确的（一旦抽象密码运行示例）。

真正的，容易出现具体的协议，逻辑正确但是否明显不安全。

检查在会话秘钥分发难题，可以找到方法，从基本的密码原则分离。例如，一旦找到一次又一次的数据加密和数据认证的例子。最流行的问题是缺乏什么是需要解决的问题关键的说明。没有攻击者的模型或安全的定义。

在这个领域的相关工作是 Bird[35]和 Diffie[73]，特别的，前者指出了一个新的攻击的种类，叫做“交叉攻击”，用于攻击存在的协议，并且假设一个协议[2PP]战胜认为的交互攻击。构造这些，Bellare，Rogaway 提供一个模型和安全的定义，可以证明基于标准密码算法的假设。也可以有效地转变协议。

现在其好的协议也可以合并，例如，Krawczyk[126]的 SKEME 协议是一个文雅的、多目的的两方会话秘钥协议，在全密钥分发协议指导下是需要 INTERNET 安全协议的。甚至最近，在会话密钥分发条件下，一个可证安全协议是在智能卡上通过 Shoup 和 Rubin[195]进行。

11.2.3 对于难题不正式的描述

正常考虑协议的一方，是单独贡献在协议上的。不做其他的其事情。会话密钥分发中，新颖的主要元素多种会话渠道同时保存。一方必须有多种实际例，这些例子是逻辑终点的重要例子，不包括自己。

假设$\{P_1....P_N\}$定义分布系统上的各方，如同上述的讨论，给定一对 P_i，P_j，可以保持多方会话（每个有各自的会话密钥），因此 P_i，P_j，不是真正意义上安全会话的逻辑节点，相应的，P_i 的范例 $\Pi^s_{i,j}$ ，P_j 的范例 $\Pi^t_{i,j}$ ，作为会话密钥分发难题的核心部分强调一个实例，这也是会话密钥分发与其他难题不同的一个关键。

会话密钥分发的目的是为 $\Pi^s_{i,j}$ 和 $\Pi^t_{i,j}$ 提供一个安全会话的密钥 $\delta^{s,t}{}_{i,j}$，$\Pi^s_{i,j}$ 和 $\Pi^t_{i,j}$ 必须在有 s,t 的先验条件下进行，而且无论在分发过程中还有其他的会话进程。

网络中的主动攻击在所有参与者中进行通信控制：控制方可以对没有要求的接收者进行密钥的随意分发，通过自己的选择进行充分的混合，并且对所有的参与者开始新的例证。进一步，可以增加将要讨论的各种攻击。

11.2.4 推导安全

最终，希望得到的是攻击者不能在合法的实例中进行安全密钥协商，必须考虑到两个方面：认证和秘钥保密。大至，最早的手段是：当一个 i 的范例接收 B，那么他们必须已与 J 进行会话。第二，在 $\Pi^s_{i,j}$ 和 $\Pi^t_{i,j}$ 进行密钥分发时，必须保证是安全的。

对会话密钥一个重要的要求是一个过程的会话密钥必须是与其他的会话密钥独立，这是因为不能确定会话密钥究竟是在哪个实例中进行使用。也许可以停止进行公开某一密钥，但是可以不对其他密钥有影响。这是在最坏条件下进行的构造，允许攻击者随意公开密钥。甚至在其他会话密钥公开的条件下，在没有公开的同伴中进行密钥共享可以保持秘密。

一个最重要的条件是密钥的共享意味着安全。用传统的观念是如果攻击者无法计算则密钥是安全的。但是对于不断进步的装置条件，这里没有确定的安全概念，必须阻止部分信息泄露。（这部分信息为什么对于会话密钥是重要的，显而易见，分析先前看到的一些例子可以阐述这点），定义要求进行概率加密的信息在会话密钥的条件下是不可预测的。

注意到不充分的会话密钥保护是有漏洞的，在除了[16，17]目前的密钥分发协议是找到了缺陷的，事实上，这种不安全是由于密钥欺诈所造成的。为了证实一方确实接受到了会话

密钥，加密一个消息，如果可以正确的进行加密并且可以被其他方正确的读出，说明接收了正确的密钥。但是这泄露了密钥的部分信息，这也许看来不重要，但是可以看到会话密钥是导致不安全的因素，这是一种密钥确认，事实上，还有其他的确认方式。

11.2.5 对密钥分布的完整性确认

考虑到的密钥分发目标是对于多方同时进行认证，进入一个安全领域进行会话密钥共享。有几种相应的办法定义认证的概念。

文献上考虑两种方法，首先在很强的条件下进行认证，考虑[16]中的两方实例子，在[17]中放宽了条件，考虑了三方实例。对于两方实例的条件讨论还在研究和不断进展中。

为什么更倾向装置条件的限制？在现存的文献里可以得到这样的进展，也就是将考虑两方装置模型更强的条件，弱的一方作为第三方的装置。在彻底使用弱概念的时候也许是正确的，在这些注记将来的版本里，也许会进行这方面的工作。目前的情况是对于弱的一方条件仍然是在两方中没有公开的前提下进行的。

11.3 认证密钥交换

先看到两方的认证：对称与非对称。看一看提供会话密钥交换的认证，意味着一方希望认证另外一方，同时共享秘密会话密钥，正式的协议是怎样构成一个安全认证会话密钥分发协议，在[16]中提及。这里仅仅描述部分协议。

首先，注意某些习惯，假设希望占用 l 比特，随机选择共享秘密密钥，l=56。（更通常的是能够从一些任意样本的分布，但是更简单的，假设坚持所有在共同的原因下，会话密钥是 a 定义的。

任意一方 A 发送一串到 B，可以明白她的身份 A 伴随这串比特。因此 B 知道目的串是从何而来，也就是没有密码学或者安全，仅仅是一个通信媒体的服务。注意这个身份是不安全的，攻击者可以改变这种情况。如果这些方希望是安全的，这就是使用密码进行这点，然后将看到怎样完成这点。

11.3.1 对称条件

假设 K 是多方共享的（长效）密钥，固定私钥加密体制（E，D）和一个私钥认证体制（T，V）。密钥 K 分成两部分，K^e 和 K^m，首先使用加密，第二是消息认证。协议，称为认证密钥交换协议 1，在图 10.1 中进行描述，一个更完整的描述如（图 10.1）：

有一个流程更完整的描述：

（1） A 选择一个随机串 R_A 并且设置到 B

（2） B 随机选择一个串 R_B，应该随机选择一个 l 比特的会话密钥 a。在 K^e 条件下加密，产生密文 $C=E_{Ke}$（a）。现在计算标签 $\mu= T_{Km}(B \| A \| R_A \| R_B \| C)$。发送 R_B ，C，μ 到 A。

（3） 一个 $V_{Km}(B \| A \| R_A \| R_B \| C, \mu)=1$。如果在这个条件下，计算标签 $T_{Km}(A \| R_B)$ 并且发送到 B。也通过 $a= D_{Ke}(C)$解密恢复会话密钥。

（4） B 证实最后的标签，并且接受（输出会话密钥 a），如果最后的标签是有效的。

评论 10.1、 注意两个加密和消息认证进行使用，上述提到的是，一个在会话密钥分发协议，试着使用加密提供证明。应该正确使用消息认证码。

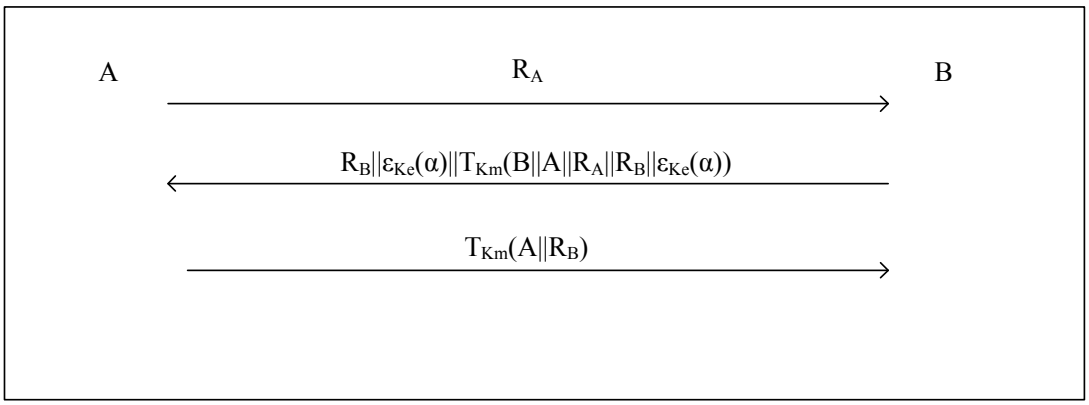

图10.1:协议AKEP1:对称密码设置中的会话密钥分发

评论 10.2、 在第六章中讨论的是，重要的是加密体制（E，D），特别的回忆意味着这是概率的。一个简单的明文有许多可能的明文，依赖于加密算法产生的概率选择。这些概率选择是 S 进行的，当随后加密会话密钥，执行两个独立的加密过程。这是在会话密钥安全中的关键因素。
这些评注也应用于以下的协议，适当的调整，当然，对于在设置中反应变化，将回答评注。

11.3.2 非对称条件

将使用公钥密码。特别的，同时使用公钥密码和数字签名。
固定一个公钥加密体制，假设分别定义了 E，D，为加密和解密体制。前者采用了公共加密密钥 pk^e 并且消息返回一个密文，随后采用了秘密解密密密钥 sk^e，并且密文返回明文。这个体制应该是安全的，在这个条件下，在第七章进行了讨论。

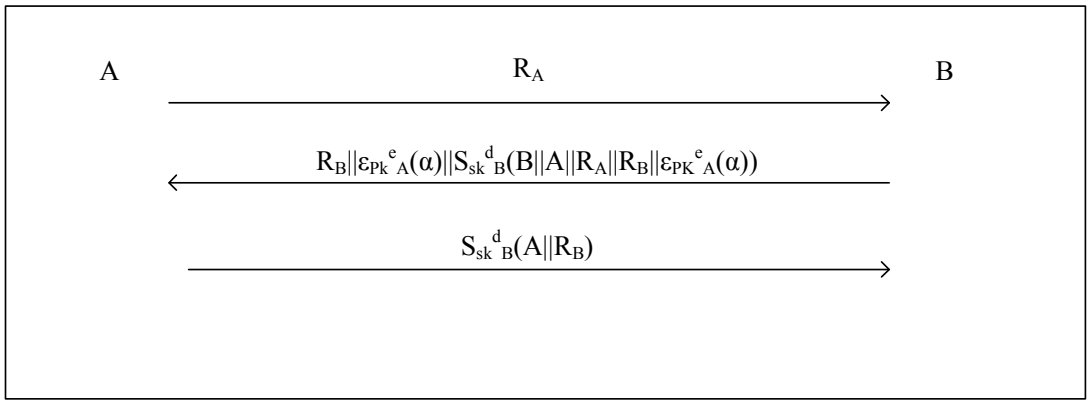

图10.2:在非对称体制中的对称密钥分发协议

固定一个数字签名体制，分别定义 S，V，作为签名和验证算法。前者采用了一个秘密签名密钥 sk^d 并且返回签名的消息。随后这个公钥验证密钥 pk^d，消息，候选签名返回一个指令，是否签名是有效的。这个体制应该是安全的，在这个条件下，在第九章已讨论。在系统中，

每个用户 I 有一个公钥 pk_I，事实上是一对公钥，pk_I=（pk_I^e ，pk_I^d ），一个对于加密体制，另一个是签名体制。对所有攻击者和用户，这些密钥是已知的。然而，用户保持私密和通信秘密密钥。也就是说，sk_I=（sk_I^e ，sk_I^d ），并且没人知道这些信息。回复这些密钥是 A 有 B 的公共密钥 pk_b，并且 B 有 A 的公钥 pk_A，多方协议得到一个连接、共享密钥 a 在图 10.2 进行描述。并且如下一个更好的解释。

有一个更完善的描述：

（1） A 随机选择一个串 R_A 并且发送到 B

（2） B 随机选择一个串 R_B ，同时随机选择 l 比特串会话密钥 a，用 A 的公钥 pk_A^e 加密产生密文 $C=E_{pkAe}(a)$，现在计算签名 $\mu=S_{SKBd}(A \| R_A \| R_B \| C)$，在秘密签名密钥 sk_B^d 下。发送 R_B，C，μ 到 A。

（3） A 验证 V_{PK}^B （$A \| R_A \| R_B \| C$）=1，如果在这个条件下，计算签名 S_{SKBd} （R_B）并且发送到 B。同时通过 $a=D_{skAe}(C)$解密 C 恢复会话密钥。

（4） B 验证最后的签名，并且如果如果签名是有效的，就接收。

11.4 三方会话密钥分发

固定一个私钥加密体制（E，D），在第六章里，在某种情况下是安全的。同时固定消息认证体制是（T，V），在第八章中某种程度上，就是安全的。密钥 K_1 在服务器 S 和 I 方共享是一对密钥（K_I^e ，K_I^m）。一旦对加密体制密钥并且对消息认证体制使用另外一个密钥，现在考虑 A、B 方，的密钥是 K_A，K_B，分别有这种形式。[17]的协议一个简洁的表述是在图 10.3 中给定的。并且更多是如下的解释：

现在有更完善的解释并且伴随计算：

Flow 1.	A→B:	R_A
Flow 2.	B→S:	$R_A \| R_B$
Flow 3A.	S→A:	$\varepsilon_K{}^e{}_A(\alpha) \| T_K{}^m{}_A(A \| B \| R_A \| \varepsilon_K{}^e{}_A(\alpha))$
Flow 3B.	S→B:	$\varepsilon_K{}^e{}_B(\alpha) \| T_K{}^m{}_B(A \| B \| R_B \| \varepsilon_K{}^e{}_B(\alpha))$

图10.3:三方会话密钥分发协议

（1） 在第一步，A 方选择挑战随机数 R_A 并且发送到 B

（2） 在第二步，B 方选择挑战随机数 R_B 并且发送 $R_A \| R_B$ 到 S

（3） 在第三步，S 随机选择将要分发的 l 比特会话密钥 a，在每方共享密钥的前提下，S 加密会话密钥。

（4） 在步骤 4，A 方收到消息 $a_A' \| \mu_A'$（resp. $a_B' \| \mu_B'$）并且接收，通过会话密钥 $D_{skAe}(a_A')$（resp.D_{Kbe}（a_B'）），当且仅当如果 V_{PA}^B （$A \| B \| R_A \| a_A'$，μ_A'）=1，（resp. V_{PA}^B（$A \| B \| R_A \| a_A'$，μ_A'）=1）

评论 10.3 这个协议有四个流程，典型的，三方密钥分发协议在文献中看到的有五个，事实上，四个就足够了。

11.5 前向保密

前向保密是一个特殊的安全特征，一个会话密钥有并且是非常值得的。考虑更具体的，图 10.2 的协议在非对称装置中用于对称密钥交换。假设 A 和 B 操作这个协议，并且交换一个会话密钥 a，并且用来交换密钥。假设攻击者记录这个交换的副本，这意味着攻击者拥有密文 $C=E_{pkAe}(a)$，加密会话密钥，并且任意密文在 a 下加密。多方已传输了，称为 C_1，C_2，…既然会话密钥分发协议是安全的，所拥有的信息是没用的。当然不能得到会话密钥 a。既然会话结束，现在假设，对于一些原因，A 的长效密钥暴露。意味着攻击者，但是，获得 $sk_A = (sk_A^e, sk_A^d)$。

当然，攻击者可以折衷所有 A 将要使用的密钥。但是在实际中，应该期望 A 可以不久认识到秘密信息丢失，并且宣告公钥 $pk_A = (pk_A^e, pk_A^d)$减轻损失。然而，还有另外一些论点。攻击者现在有 sk^e 并且能够解密密文 C 去得到 a。使用这个可以解密 C_1，C_2，…，因此在过去的会话中读机密信息。这称为前向秘密。

在这一章开始，前向秘密是通过 Deffie-Hellman 密钥交换可以完成的。假设给定一个协议，在非对称中，两方设置，类似的协议可以另外一种设置。这个协议是给定[69]的 STS 协议的一个扩展。在图 10.4 进行了描述和如下更完善的解释。

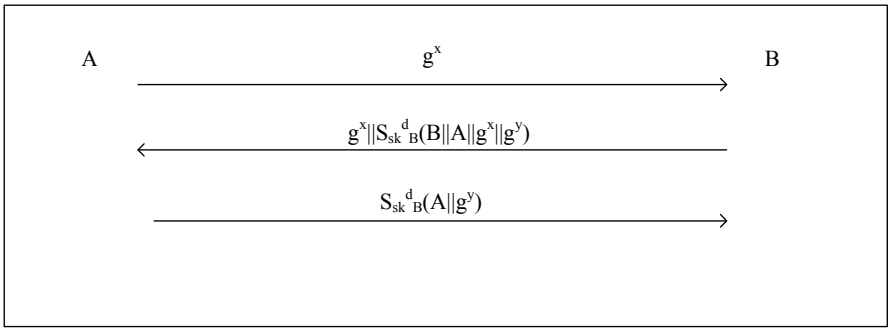

图10.4:前向安全的对称密钥交换协议

有如下更完善的描述：

（1） A 方随机选择串 x，计算 $X=g^x$，并且发送到 B

（2） B 方随机选择串 y，假设 $Y=g^y$。现在计算签名 $\mu=S_{skdb}(A \| X \| Y, \mu)$在秘密签名密钥下 sk_B^d。发送 Y, μ 到 A。

（3） 一个验证 $V_{PA}{}^B$ （$A \| X \| Y, \mu$）=1，在这个条件下计算签名=S_{skdb}（Y），并且发送到 B，解密输出 DH 密钥 $g^{xy} = Y^x$，作为会话密钥。

（4） B 证实最后的签名并且接收（输出会话密钥 $g^{xy} = X^x$）作为会话密钥。

DH 秘密密钥加密交换协议是有趣的，仅有的途径是得到前向安全。翻转出这些，Bellare 和 Rogaway 已注意到秘密密钥交换不仅仅是不够的，而且也需要前向安全特征。在 10.1.4 中注意到的，DH 密钥不是一个好的密钥，而且不能保证比特安全。因此，上述的会话密钥实际上是可以设置的，也就是说，H（g^{xy}）比 g^{xy} ，对于一个好的单向函数。

第十二章　协议

经典密码关心的是安全通信问题，在提供用户私密和认证，对于密钥管理的基础结构的需求自然导致密钥分发的话题。经过许多年，这些内容都包括在密码学中。

现代密码学一个主要贡献是高级协议的进展，这些协议促使使用者用电子解决许多现实中的难题，玩游戏，完成所有有趣和普通分布的任务。在这些中间，有零知识证明，安全分发计算和投票协议。这一章的目标是给定这个领域的简单介绍。

12.1 一些两方协议

在 C.6 节中参考一些数论附录。

12.1.1 健忘传输

这个协议是 M.Rabin[172]发明的。

一个健忘传输是一个普通协议，在这种方式下，Alice 传输一个秘密比特 m 给 Alice，以 1/2 概率传输到 Bob；Bob 知道当得到比特后，但是 Alice 不知道是否已经进行传送。这个合理的陌生而合理的协议（看到，例如[172，32]），事实上，Kilian 已看到[123]执行健忘传送的能力是足够强的，可以执行任意两方协议。

对于健忘传输协议如下的应用在理论上被提出了（因为 Rabin 和 Blum 的相关思想）。

（1） Alice 随机选择两个素数 p,q，并且进行模 N=pq 的乘积，在这些标准条件下，这个模的加密消息有这个特征，如果已知 p,q，就可以进行解密，否则不能。发送 N 和密文 C 到 Bob。

（2） Bob 选择随机的 $a \in Z_N^*$ ，并且发送 $w = a^2 \bmod N$ 到 Alice。

（3） Alice 计算 w 的四个平方根 x，-x，y，-y，随机选择和发送给 Bob。

（4） 如果 Bob 返回根，不是±a，可以分解 N 并且恢复 m，否则不能继续。

Alice 在 a 是随机的条件下不知道怎样发生。

这显然是公平的，对 A 没有在这个协议中进行欺骗，因为 A 不知道 z 的平方根，而 B 知道，因此 x 可以随机选择。首先看到，形如 B 不能得到另外一些随机平方根。然而，事实上这些形式化证明是未知的。还不太清楚是否 B 可以欺骗或不被欺骗。例如，如果 B 选择一个 z 的特殊值，代替随机选择 x，并且设置 $z=x^2 \pmod n$。那么这可以导致在因数 n 中的一个优势，这是可能的，例如已知（n-1）/2 mod n 的平方根，（或其随机值）可以允许 B 得到因数 n。因此满足条件 II，但是不能证明是否满足条件 I。

如果有一个方法，可以证明 A 是确实可以跟随协议和随机选择 x，对于 x 究竟是什么没有启迪作用。协议应该调整成为可能的工作，可以看到在下一节中零知识证明是怎样进行这样的证明。有另外一种 OT 的模式，称为二选一 OT。这里 Alice 有两个秘密 m_0 和 m_1，Bob 有一个选择比特 c。在协议的最后，Bob 得到 b_c，并且 Alice 也不知道 c，见[81]。

12.1.2 同时定约签名

Alice 和 Bob 希望签署协议，但是当且仅当其他人也相同。也就是，在一个人坐在签名

的位子上时候，没有其他人希望离开。因此，如果 Alice 首先签署，担心 B 不签字，并且代替 versa。（也许容易考虑有两个协议，首先允许某些给定 Alice，第二个给定 Bob。这是交易，显然的，每个希望另外的人可以得到签名）。这个难题在[81]中提到。

一个进展是 Alice 签署了她名字的第一个字母，并且发送协议给 Bob。同样进行并且发送返回。并且如此。假设的名字有同样的长度，那么面向一个解释有相同的进展。当然难题是最后必须可以停止。如果有一些可忽略的区别。例如，一次不是一个字母，但是一次是很少的毫米。没有任何一方在另外之前，如果指出同时停止。是同样的一点。

电子的交换串，是协议的数字签名。Alice 签署产生 σ_A 并且 Bob 签署产生 σ_B。现在同时交换这些串的比特，每次发送更多的比特。

在这点上有一个难题，如果一个人不发送签名，但是仅仅是一些垃圾串？另外一些甚至在最后是未知的，甚至，Goldreich 和 Lempel[77]表示怎样健忘传输可以得到这些。Alice 产生 L_A，用短语进行具体的签名“这是合同签名的左半部分”，类似的产生 R_A，因此这一段协议的签名是“合同签名的有半部分”。类似的 Bob 产生 L_B 和 R_B。

Alice 找到两个两个 DES 密钥，K_A^L 和 K_A^R，并且分别加密 L，R 产生 C_A^L 和 C_A^R。对 Bob 是类似的，用 B_s 代替 As。考虑签署合同，如果有两部分的个人签名。所有密文发送到其方。

Alice 发送二选一 OTs（K_A^L，K_A^R）到 Bob，然后随机选择一个随机选择比特，并且代替 VERSA。也就是说，Bob 得到 K_A^L 和 Alice 得到 K_B^R。Alice 和 Bob 同时发送两个 DES 密钥的第一比特，把持重复直到所有密钥比特发送成功。在这一段里，如果一方在比特通信中抓住一个错误发送到密钥，已有中断，否则继续。

12.1.3 比特承诺

Bob 希望 Alice 承诺一些值，也就是说是一个标价，因此以后不能改变这点，作为其事的一个函数。在另一方面，Alice 不希望 Bob 知道。这次，是委托的一个值，但是同时将随后开放。

Alice 安排了一个“电子安全”。有一个安全的密钥，在安全里设置了一个值并且安全发送到 Bob。随后不能发送到特别的目录。这是一个委托，随后，Alice 是发送密钥的转换。现在 Bob 可以打开了。必须是真实的在于 Alice 不能产生一个有两个密钥的安全，也不能打开，那么当进行调查时，看到了不同的值。

一个实现的方法是通过免碰撞的单向函数，对 x 进行提交，发送 yH(x)。对于这点 Bob 是不能计算出 x 的。既然 H 是单向函数。为了直接转换，Alice 发送 x 并且 Bob 检查 H（x）=y。但是 Alice 不能发现 $x' \neq x$ ，因此 H（x’）=y，因此不能欺骗。

这是因为有极小的比特安全，可以用核心比特进行固定。

另一个办法是使用二次剩余。首先，固定一个特别的数 $y \in Z_N^*$ 已知是一个非剩余。发送一个模 N 的随机串表示为 0，也就是说是 x^2，并且用 1 表示模 N 的非平方，形如 yx^2。QRA 通知 Bob 是不能区分哪里是哪里，在任何情况下展示 x。注意 QR 委托体制是安全的，甚至抗不充分计算能量边界的发送，但是不是接收者。

能反向做这些事情？是的，使用离散算法。假设 p 是一个已知的素数，$g \in Z_p^*$是一个已知 Z_p^*的发生器，并且 $s \in Z_p^*$是一个已知离散对数的元素，也就是 $\log_g(s)$是未知的。随机选择 x 为 0，发送 $y=g^x$ ；为 1，发送 sg^x。注意接收者，每个是随机元素的范围。但是发送者可以产生一个 y，两条路可以公开。应该有 s 的离散对数。

对于许多情况，委托体制是有用的。特别的，ZK 证明，但也是随机事件。

12.1.4 在一个好的条件下的随机事件

Blum[40]提议在电话上进行投币协议。Alice 和 Bob 等待一个公平合理的投币协议。希望做一个随机选择，但也不能去指示它。在开始，Alice 赢，在结尾 Bob 赢。如果 Bob 说“将进行投币并且发送给一个值”。这个协议不好，Bob 为了赢将进行投币，会影响这个值。

有一个思想。Alice 随机选择一个比特 a，发送给 Bob。并且 Bob 随机选择一个比特 b 发送给 Alice，投币的值是 a⊕b。难题是谁将首先走，如果 Alice 首先走，Bob 将选择 b 确认是否是想要的密钥。不公平。

因此 Alice 所做的是首先承认货币，发送 y=Committ(a) 到 Bob，现在 B 不能确信 a 的函数是 b 。显然，发送返回 b。Alice 也许希望 b 的函数做法 a 是诚实的，可以承认，投币是 a⊕b。

12.1.5 健忘电路赋值

Alice 和 Bob 希望知道哪一个更老。但是并不想揭露年龄，（意味着也不希望暴露年龄的区别，既然对于不同的年龄，每个也得到一些另外的年龄。）仅仅想挤压出单个比特，指出老的一个。

有时称为百万富翁难题，通过称为百万富翁。

普通的，难题是 Alice 是一个输入 x_A 并且 Bob 有一个输入 x_B 并且希望计算 $f(x_A, x_B)$这里 f 是一些已知函数，例如 $f(x_A, x_B)=1$，如果 $x_A \geq x_B$，否则是 0。希望进行健忘计算，因此游戏的结论是有值 $v= f(x_A, x_B)$但是并不知道其他的事。有这个任务的协议，并且是非常复杂的。希望读者为[10,47]

12.1.6 同步秘密交换协议

在[41，204，137，210]中已研究了。

协议给出了协议的例子，粗略的看到工作的过程，但实际上对类似的原因公开进行欺骗，因此实际上健忘传输协议进行欺骗。共同输入组成 1^k，$a \in E_{nA}(S_A)$，$\beta \in E_{nB}(S_B)$，n_A，和 n_B，这里 n_A 和 n_B 是两个大小相等的模 4 余 3 的素数组合，$E_{nA}(E_{nB})$是健忘传输协议中相同的加密方式，分别对于 n_A 和 n_B 。A 的私钥输入有素因子 $n_A = p_A q_A$ ，并且 B 包括同样的 n_B 。对于 A 和 B 希望做到的是对图形同时输出 S_A ，S_B 。假设相同的计算能量和算法知识，Blums 的建议协议如下：

步骤 1：A 选择 $\alpha_1, \alpha_2, \ldots, \alpha_k$ 在 Z_{nb}^* 随机选择，那么计算 $b_i = a_i^2$（mod n_b），对于 $1 \leq i \leq k$，在 Z_{nb}^*，B 随机选择 w_1，w_2，…，w_k，对于 $1 \leq i \leq k$，计算 x_i，$= w_{i*} \pmod{n_A}$。

步骤 2：A 对 B 发送所有 b_i ，并且 B 发送所有 x_i 到 A。

步骤 3：对每个 x_i ，A 计算 y_i 和 z_i ,因此 $y_i^2 = z_i^2 = x_i \pmod{n_A}$，但是 $y_i \neq \pm z_i \bmod n_B$ ，（注意：无论 y_i 或 z_i 等于$\pm w_i$ ）。对每个 b_i ，B 计算 c_i 和 d_i 使用类似的限制。（注意：无论 c_i 或 d_i 等于$\pm z_i$ ）。

步骤 4：当 $1 \leq j \leq k$，A 发送 By_i 的第 j 签名比特和 z_i ，$1 \leq i \leq k$。B 发送 A 第 j 个 c_i 的比特记号和 d_i ，$1 \leq i \leq k$。

步骤 5：完成上述循环后，A（和 B）指出 n_B 的因子是在第四步中获得的信息。（对每个 i，A 计算 gcd($c_i - d_i$ ，n_B)。对每个 i，B 计算 gcd（$y_i - z_i$ ，n_A））使用这个信息，通过解

密 α，β 计算出 S_B 和 S_A。

为什么选择 k 个数比选择一个数更合适？在 A 和 B 代表时，防止如下形式的欺骗。假设仅仅一个 x 会发送到 A，A 可以计算出 y 和 z，那么发送 y 的第 j 签名比特和在第四步发送一个垃圾串。希望 y=±w 并且 A 将注意到垃圾串将会被发送。如果 y=±w，那么直到最后一步，B 无法知道 A 的欺骗，这个时候，A 有所有需要找到 S_B 的消息，但是 B 没有获得任何新的信息找到 S_A。因此 A 可以 50%的欺骗成功。如果在另一方面，k 与 x 的不同是发送到 A，在这个条件下，A 有成功消去指数的机会。也就是 $Prob(y_i = \pm w_i \ \forall i)$，$\leq (1/2)^k$。

Shamir 和 Hastad 指出这个协议中成功欺骗的办法。如果随机选择 w_i ，A 随机选择 w_1，设置 $x_1 = w_1^2 \pmod{n_B}$，并且设置 $x_i = x_1/2^{i-1} \pmod{n_B}$，那么在第四步的一个迭代之后，A 有所有需要的信息，使用[102]的归约因子 n_B。因此，一个表面上好的协议失败，既然 B 没有办法检查是否 A 独立随机选择 x_i，对于每个协议的说明或者不是。注意：这个难题是类似的，因为健忘传输协议并且可以修正。如果 A 和 B 可以相互阻止如下的协议。

12.2 零知识协议

前一节列举的密码学应用协议的数字和提取的一些难题。在这一节，复习定理进展证明这些协议是安全的。并且对于协议设计是“结构可证安全的”。密钥的思想是把双方普通协议的难题归约为简单难题：A 怎样证明给 B 知道 x 在一种语言里，没有比 x∈L 更多的知识进行泄露了。如果这样做对任意 L∈NP，A 可以证明 B 按照步骤产生协议。继续定义“交互证明”（“一个协议证明”）并且“零知识”。

12.2.1 交互证明系统（IP）

在证明系统的证明概念前，定义交互 Turing 机的定义概念。

定义 11.1、一个交互图灵机（ITM）是一个图灵机和一个只读输入模式，一个只读随机模式，一个读/写工作模式，一个只读通信模式，一个只写通信模式，一个只写输出模式。随机模式包括一个比特有限序列，可以考虑没有偏差的随机输出。这个形式仅仅是从左向右扫描，可以说一个交互机器随机事件是意味着可以从随机模型读下一个比特。只写通信模式的内容可以考虑机器发送使用信息，当只读通信模式内容可以作为接收者的模型。

定义 11.2、 一个交互协议是一个 ITMs（A，B）安排好的对子，共享相同的输入模型。B 的只写通信模型是 A 的只读通信模型和副 VERSA。在 B 活动的条件下，机器变化是活动的。机器首先执行基于一些模式的内容进行一些交互计算，并且在只写模式下，第二次写一些串。A（B）的第 i 个消息是串 A（B）在第 *i* 步进行只写通信。在这一点上，机器停止并且其机器开始活动知道协议终止。机器可以终止协议，在活动期间不能发送任何消息。在输入一个接受或输入状态时，机器 B 接受（或拒绝）输入并且终止协议。在对子的第一个数中，A 是图灵机的极大计算边界。机器 B 的计算时间定义为活动期间 B 计算的时间的总数，在输入串长度为多项式计算时间限制。

定义 11.3、 假设 $L \in \{0,1\}^*$ ，也就是说 L 有一个交互证明体制，如果存在 ITM V s.t

1．∃ITM P s.t（P，V）是一个交互协议并且任意 x∈L s.t|x|足够大，prob(V 接受)>2/3（当概率使用 V 和 P 的随机投币事件）

2．∀ITM P s.t（P，V）是一个交互协议并且任意 x∉L s.t|x|足够大，prob(V 接受)>1/3（当概率使用 V 和 P 的随机投币事件）

注意：这是不够的，要求验证者不能自动预先校准。（如此温和的条件下，应该预示着

校验是信任模型)。NP 是交互证明的特殊条件，这里交互是细致的，验证者不参加投币。也就是说（P，V）是一个 L 的交互证明体制。定义 IP={L|L 有交互证明机制}。

12.2.2 例子

概念：通过讲义注记，任何交互式协议被证明，假设 B → A，定义一个机器 B 的活动阶段，在结束阶段，B 发送 A 一个消息，类似的，A → B，定义一个机器 A 的活动阶段。

例 1、（从数论的角度）

假设 Z_n^* = { x<n, ;(x,n)=1}

QR = {(x,n)|x<n,(x,n)并且$\exists$ y s.t $y^2 \equiv x \bmod n$}

QNR={(x,n)|x<n,(x,n)并且$\bar{\exists}$ y s.t $y^2 \equiv x \bmod n$}

对 QNR 进行一个交互式系统证明。

输入（x,n）到一个交互式协议（A，B）：

B → A，B 发送到 A 一个列表 $w_1 \ldots w_k$ ，这里 k=|n|,并且，

$$w_i = \begin{cases} z_i^2 \bmod n, if b_i = 1 \\ x \cdot z_i^2 \bmod n, if b_i = 0 \end{cases}$$

这里 B 选择 $z_i \in Z_n^*$ ，随机选择 $b_i \in \{0,1\}$。

A → B：A 发送 B 一个列表 $c_1, c_2, \ldots c_k$ s.t

$$c_i = \begin{cases} 1, if w_i 是模n的二次平反剩余 \\ 0, 其余情况 \end{cases}$$

B 接受，如果 $\forall_{1 \le i \le k}$ ， $c_i = b_i$

明显的，B 解释 $b_i = c_i$,（x,n）∈QRN，当 $b_i \neq c_i$,使得拒绝。声明（A，B）是一个 QRN 的交互式证明体制。如果(x,n)∈QRN，这里 w_i 是一个 mod n 的二次平方剩余。因此，所有强大的 A 可以容易计算，是否 w_i 是一个 mod n 的平方剩余或不是。

正确计算 c_i ，并且使 B 以概率 1 接受。如果（x,n）∉ QNR，并且（x,n）∈QR，那么 w_i 是一个 mod n 的随机平方剩余，不管是否 b_i =0 或 1。因此，概率是 A（无论能力有多强），可以发送 c_i s.t $c_i = b_i$ ，对每个 i 是有限的。概率是以至多（1/2）^{K} 的接受。

例 2（图论）

说明一个互相证明的定义，介绍一个非同构图论的相互证明。输入是图 G_1 和 G_2 的一个对，一个要求证明存在不是 1-1 不变边界的映射，第一个图的顶点到第二个图的顶点。（一个映射 π 从 G_1 的顶点到 G_2 的顶点，是无变量边界的，如果节点 v 和节点 u，是在 G_1 临近的。如果节点 π（v）和 π（u）是在 G_2 中临近的）。这是有趣的注意没有短的 NP 证明是已知这个难题的。也就是 Graph 非同构映射是在 NP 条件下未知的。

在输入（G_1，G_2）上的交互式证明是如下：

B → A，B 随机选择两个输入中的一个输入图 $G_{\alpha i}$ ，这里 $\alpha_i \in \{1, 2\}$，B 随机产生 $G_{\alpha i}$ 的同态，发送到 A。（这个过程重复 k 次，1≤i≤k，独立随机选择）

A → B：A 发送 B$\beta_i \in \{1, 2\}$，对所有 1≤i≤k

B 接受，如果 $\beta_i = \alpha_i$ 做为证据，图表是不同构的，这里 $\beta_i \neq \alpha_i$ 导致被拒绝。

如果两个图是不同构的，校正对于正确的回答是不难的。（例如 α 和 β 等于 a）。并且验证会被接受。如果两个图是同构的，简单的区分第一个表的随机拷贝到第二个表的随机拷贝。

并且概率是校准回答正确，询问概率至多是 1/2。校准正确的概率是所有 k 询问是（1/2）^{k}。

12.2.3 零知识

既然已扩展了概念，是一个有效的证明系统，提出的问题是多少知识需要传输，为了确定多项式时间的确定，一个陈述的确定。通过“知识”意味着什么？例如，考虑 SAT，满足微积分提议的判断 NP-完全语言。最明显的证明数据是在逻辑公式 F 的校对，给定验证满足分配 I，在多项式时间可以进行验证。如果发现分配 I，可以采用多项式时间的验证（如果 P≠NP）。可以说验证获得另外的知识，对仅仅的信息 F∈SAT。

Goldwasser,Micali 和 Rackoff[103]，使这个观念更细致。对于语言 L 零知识，如果∀x∈L，无论验证者可以在多方交互验证后可以计算验证。能够在多项式时间计算，在输入 x，对于一个图灵机的概率多项式时间，叫做交互证明体制。给定技术的零知识证明体制的定义，并且在 11.2.4 节中变化，并且简要提出这个领域中有趣的结论。

12.2.4 定义

假设（A，B）是一个交互协议，假设通过一个随机变量表示验证观点，在输入 x 通过协议。也就是，对于 A 到 B 的随机事件的投币协议，在验证和校验时，观察信息交换次序。串 h 表示一个秘密的输入，验证者也许只有输入。输入长度是公共输入的多项式时间长度限制。（观察在 A 和 B 的随机事件是分布式）。

也就是说（A，B）是 L 的完美零知识证明，有一个概率，多项式时间概率图零机 M s.t ∀x∈L，对于所有 α>0，对所有串 h，因此|h|<|x|$^\alpha$，随机变量 M（x,h）并且观察同一分布。（M（x,h）是在 M 上的投币事件上分布的，并且输入是 x 和 h）。

也就是（A，B）是满足 L 的零知识变量，有一个概率多项式时间图灵机 M s.t ∀x∈L，对所有 α>0，对所有串 h，因此|h|<|x|$^\alpha$，随机变量 M（x,h），并且观点是对等分布的。（M（x,h）并且观察同一分布，当输入是 x 和 h 时）。

也就是说（A，B）对于 L，满足零知识，如果存在一个概率多项式时间图灵机 M s.t 。∀x∈L，对所有 α>0，对所有串 h，因此|h|<|x|$^\alpha$

$$\sum_{\alpha} | \ prob(M(x,h)=\alpha) - prob(view=\alpha) | < 1/|x|^c$$

对所有的常数 c>0，并且有足够大的|x|。

直观的，满足零知识协议的考虑是一个主考者的有限能量，给定了仅仅是多项式大样本的{M（x,h）|M 的随机事件}并且{观点|A 的和 B 的随机事件}，不能进行两个区分。最后，可以说这个协议（A，B）是计算零知识，如果概率多项式时间边界检查给定了样本的多项式数字。上述设置不能区别，正式的，

也就是说（A，B）是对 L 的计算零知识，如果∃概率多项式时间图灵机 M s.t∀任意多长度 C={$C_{|x|}$}，∀a,d >0，对于足够大的|x| s.t 是 x∈L，并且对所有的串 h，因此，|h|<|x|$^\alpha$，

Prob($C_{|x|}$ (α)=1|α 在 M（x,h）)-prob($C_{|x|}$(α)=1|α 随机选择 view(x)|<1/|x|^d

也就是说 L 有（计算/统计/完美）零知识证明体制，如果：

1．对 L，∃交互式的证明体制（A，B）

2．对 L，∀ITM 的 B’的交互协议证明（A，B’）是（计算/统计/完美）零知识证明。明显的，最后的证明是三个中最普通的一个。因此假设 KC[0]={L|L 有可计算的零知识证明}。

12.2.5 如果有单向函数，NP 在 KC[0]中

到目前为止，关于零知识证明的最重要的结果是由 Goldreich,Micali 并且 Wigderson[99]。可以看到如下的结果：

定理[99]、如果有（非正规）的多项式时间不可区别的加密体制，那么每个 NP 语言有一个计算零知识交互系统。非正规的条件下，对技术原因是必要的（例如加密体制应该是抗非正规攻击者是安全的，见 3.7 节）。在这个加密体制条件下存在单向函数最后假设（非正规攻击者是需要重视的），对于假设 Levin-Luby 和 Naor 是有假设结果的。

对于 NP 完全问题，证明的轮廓展示了一个零知识证明结果，三色图。这里给出协议轮廓。假设验证者希望验证者相信部分确定输入是三色图，没有对校验者知道的验证着色。验证可以这样进行，在$|E|^2$条件下的次序，每一步验证如下：

• 验证人随意转换三色图，（e.g.转换所有红色节点到蓝色节点，蓝色节点变为黄色节点，所有黄色节点变换成红色节点）

• 验证人加密每个节点的颜色，对每个节点使用一个不同的概率加密体制，并且展示校验人所有这些加密，与每个顶点标明相应的密文的走向。

• 验证人随机选择一个图的边界。

• 验证人恢复两个节点颜色的加密是恢复相应解密密钥的事件。

• 验证人确定解密是适合的，两个边的终点是用两个不同但是合法的颜色进行着色的。

（任意秘密的概率加密体制是多项式时间可分辨的），如果图确实是三色的，（校验人是知道颜色的），那么验证人将不能验证边界不正确的标签。然而，如果图不是三色的，那么有一个至少$|E|^{-1}$在每一步验证者将会被导致盲验证。在$|E|^2$步校验人可以错误引导验证者，没有导致小指数。

注意通信联系的历史——在图是三色图的条件下——组成消息的串联发送在每一步。可能证明（在假设安全加密条件是允许的条件下），在这个历史下，通过可能的交互测试条件是不可分辨的，概率分布定义在多项式时间产生自己的历史。这个事实意味着验证者获得 0（附加），从协议的知识，此图是三色图更是事实。证明图是三色的概念是有这样的零知识交互证明系统可以使用在 NP 条件下，证明每种语言，有这样的零知识证明系统。

12.2.6 用户认证应用

零知识证明提供一个革命性的全新途径去认识口令[104，83]。观点是对每个用户储存定理的描述，在公共易读的目录中，证明只有自己知道。

对于注册，用户忙碌于定理的正确性零知识证明。如果证明是令人信服的，访问许可被授权。这些保证甚至是一个攻击者窃听了零知识证明而不能获取足够的方法进行未授权访问。这是一个新奇的特征，不能用传统的密码体制得到进展。Fiat 和 Shamir[83]已进行变化，提供零知识协议[104]的先验提议，是相当有效和非常有用的，对用户的验证和口令。

12.3 多方协议

在一个典型的多方协议难题里，参与者的数量希望调整活动数量完成一些目标，甚至一些（足够小）的子集合是已被攻击者腐蚀了。协议应该保证好的一方能够达到目的，甚至是被发送方腐蚀误导信息，而另外有敌意的一方行为不端，在试图阻止好的一方成功。

12.3.1 秘密共享

秘密共享体制是 Blakley 和 Shamir[37,187]独立发明的。在多方设置中，秘密共享是基础协议和工具。

基础的观点是分布信息的私密性保护。也就是说到一个重要的体制中是有密钥的，恐怕应该放松，希望给一些思想。但不是一个人可以相信密钥，不仅仅因为个人可以成为可信的，是因为位置是保持密钥可以折衷。因此在这样的一簇人中，密钥是可以共享的。假设秘密密钥 s。A 与五个人进行共享，把它分为 $s=s_1 \oplus s_2 \oplus s_3 \oplus s_4 \oplus s_5$，并且给定 s_i 到 i。没有一个人可以计算出 s，而且，四个人也不能计算出 s。必须五人才能计算。如果聚齐密钥，可以恢复 s。（一旦这样进行，也许会放弃，也可能是一次密钥。因为没有任何一个人知道）。

称 s_i 是一个共享。谁产生了共享？s 的最初占有者，有时是多方中的一个。有时不是，称这方是经营者。

注意 s_i 必须给定私钥到第 i 个参与者。如果参与者看到这点，那么当然，这些 就不工作了。希望一些事情更灵活，也就是有 n 个人，希望任意的 t+1 可以恢复秘钥，但是 t 不能找到任何信息。例如，说 n=5，并且 t=2，任意三个朋友可以打开的系统，但是任意两个不可以。这是比较好的，既然上述的一个是共享的，这个体制不能开放。

Shamir 的观点是使用多项式[181]。假设 F 是一个有限域，类似 Z_p^*。一个阶 t 多项式形如：$f(x)=a_0+a_1x+a_2x+\ldots+a_tx^t$，对系数 $a_0,\ldots,a_t \in F$。有 t+1 期限，但不是 t。一旦比次数有更多的级数，多项式有如下好的计算：

— 篡改：给定多项式上的 t+1 点，也就是（x_1,y_1）,…,(x_{t+1},y_{t+1})，这里 $x_1 \ldots x_{t=1}$ 是有区别的，并且 $y_i = f(x_i)$，是可能的去找到 $\alpha_0,\alpha_1,\ldots,\alpha_t$，做这个的算法称为篡改。在许多书中可以发现。

— 保密：给定在多项式上的任意 t 点，也就是说（x_1,y_1）,…,(x_t,y_t)，这里 $x_1 \ldots x_t$ 是不同的，并且 $y_i = f(x_i)$，一人不能计算出关于 α_0 的一些事情。更细致的，对任意的值 v ，多项式数是满足这些 t 限制，不用依靠 v(事实上是中的一个)。这些给一个秘密共享，伴随每个参与者 *i*，一个点 $x_i \in F$，这些点是不同的。(因此|F|≥n)，为了共享密钥 s，中间人随机选择 $\alpha_0,\alpha_1,\ldots,\alpha_t$，设置 $\alpha_0 = s$ 并且形式如多项式 $f(x)=a_0+a_1x+a_2x+\ldots+a_tx^t$。现在计算 $s_i = f(x_i)$，并且发送这个私钥给 *i* 方。现在如果 t+1 参与方聚齐，能够计算出 f 并且因此是 s ； 至多 t 个参与者的设置不能计算出 s 的任何信息。

12.3.2 可验证秘密共享

Shamir 的体制遇到两个困难。如果秘密管理人是不诚实的，能够给定修补，放在一起不能唯一定义一个秘密。第二，如果一些参与者不诚实，在重改阶段，可以使用与接收到不同的接缝提供其他的参与者，然后又导致一个不正确的秘密被改造。Chor, Gildwasser, Micali, Awerbuch[59]观察了上述难题并且表示出怎样获得基于因子分解难题的秘密共享，与上述难题不同。称新的协议可验证秘密共享，因此每一方可以验证，收到的秘密的补充是需要进行调整的。协议容忍 O（log n）篡谋，Benaloh[29]和其[99，82]表示怎样获得可验证的秘密共享，既然每一方可验证所收到的秘密补充，如果存在单向函数容忍少数篡谋。在[28]中，最近已显示怎样得到完成验证秘密共享。既然现在每一方使用正确的纠错码，抗第三方串谋，不进行秘密学假设。这提供了在[173]中的少数串谋。

12.3.3 匿名处理

Chaum 提倡匿名传输处理，作为从所有传输的一个数据库链的源头，并且提供使用数字伪造。使用假名，个体可以对安全进行电子交易，传输不能对个体随后的跟踪。然而，既然个体是匿名的，其余方可以希望假设个体是可认证的，进行传输或者可以付款[54，57]。

12.3.4 多方 Ping-Pong 协议

证明的一个方法是一个密码协议是安全的，可以看到原始操作是每一方执行，不能进行任意的秘密信息恢复。

考虑一个起因于 Dolev 和 Yao[77]，包括公共密钥的使用。Alice 发送一个消息 M 到 Bob，使用公钥加密，因此密文 C 是 E_B（M）。这里 E_B 是 Bob 的公共加密密钥。然后 Bob 反馈消息给 Alice ，使用 Alice 的公钥加密，因此密文返回是 $C'=E_A(M)$。这完成了协议的描述。这是安全的？既然消息 M 从两个途径加密，明显的对于一个被动窃听者获得消息 M 是不可行的。然而，一个积极的窃听者 X 可以攻击协议。有 HOW：窃听者 X 窃听以前的会话，记录密文 $C=E_B$（M）。随后，X 开始与 Bob 的一个会话，已记录的加密消息 $C=E_B$（M）。现在 Bob 忠实地返回 X 密文 E_X（M），给定 X 一个希望的消息。

道德上，一个攻击者可以“剪贴和复制”不同协议补丁去攻击系统，这里，每个补丁是一个基本的传输，通过协议的一个合法方执行，或者在攻击者执行自己的步骤。

在一个攻击的模式下，有时可以证明一个协议是无懈可击的。Dolev 和 Yao[77]是这个证明的风格的先驱，另外的工作是 Dlev,Even,Karp[77],Yao[209],Even 和 Goldreich[76]进行的。在其他条件下，一个协议的调整是排除和消去危险；见[174]作为抗攻击者在公钥交换协议中进行抗插入攻击危险。

12.3.5 基于多数方诚实的多方协议

Goldreich, Micali,和 Wigderson[99]已展示怎样编辑协议，对诚实方进行设计，还可以正确的工作，甚至以高于一半的概率试图欺骗。当协议对诚实方可以包括秘密揭露，编辑协议的最后是没有任何一方是知道原来已知，加上任何信息作为官方输出协议的透露。基于陷门函数存在条件下的编辑器是正确并且私密的。

Ben 或 Wigderson[28],Chaum,Crepeau,和 Damgard[55]更进一步。假设秘密通信是使用者在原始通信条件下，没有假设限制，表现了一个编辑器，给定一个多项式时间函数 *f* 的描述（一个多项式时间算法或电路），产生一个协议总是计算正确的函数，并且保证一个没有函数的附加信息值，泄露给不诚实的参与者。编辑者是保证 1/3 的参与者是在条件下，用一个坏的条件下极大计算时间攻击者在诚实的前提下参与。

这些“主人法则”希望和强的工具在将来设计安全协议。

12.4 电子选举

电子选举能够考虑典型的安全多方计算例子。这种困难普通的实例有 m 个人，其中每一个使用自己的输入 x_i，希望计算 n 列函数 *f* 的计算结果，没有启迪作用。

在电子选举的条件下，多方是选举者。输入的比特值是计算函数，使用简单的总数进行计算总的结果。

普通的，希望选举协议有这样的特征：

1、 仅有授权选取者可以选举
2、 没有人可以选举超过一次
3、 保持选举的秘密
4、 没有人可以复制其他人的选举
5、 可以正确计算得分
6、 任何人可以检查 5
7、 协议应该是容错的，意味着在甚至是在有破坏方时也可正常工作
8、 应该是不可能的，强制选举者恢复怎样选举

通常的，在选举协议时，不希望在计算过程中，所有选举者 V_i 进行参与。因此假设有 n 个政府中心 $C_1 \cdots C_n$ ，任务是搜集投票和计算点数。

12.4.1 Merritt 选举协议

考虑由 Michael Merritt[147]提出的体制，每个中心 C_i 公布一个公共密钥 E_i，并且保持通信秘密密钥的秘密。为了投出选票 v_j，每个选票 V_j 随机选择一个数 s_j 并且计算

$$E_1(E_2(\cdots E_n(v_j, s_j))) = y_{n+1,j} \qquad (11.1)$$

(对第二个索引 n+1 的在一分种内的需要，现在仅仅是不相关的)

现在有值 y 的消息，为了从 C_n 到 C_1，每个中心 C_i 进行如下的工作。对于每个 $y_{j+1,j}$ ，C_j 选择一个随机值 $r_{i,j}$，并且广播 $y_{i,j'} = E_i(y_{i+1,j}, j)$。新的索引 j'通过整数[1…n]的一个随机置换 π_i。也就是说 $j' = \pi_i(j)$，C_j 保持置换秘密。在最后，有 $Y_{1,j} = E_1(E_2(\ldots E_n(y_{n+1,j}, r_{n,j})\ldots r_{2,j})r_{1,j})$

在这一点上，验证圈开始。由两个解密圈组成：$C_1 \rightarrow C_2 \ldots \rightarrow C_n$。

解密值是有效的，总数由选票 v_j 进行计算。（1）和（2）明显是满足的。（3）是满足的，甚至是在选举透露的条件下，需要保持隐藏的是在选举和投票选举的联系下进行的。事实上，为了构造这样的链接，需要知道所有的置换 π_i，（4）不满足选举者 V_1 可以容易的拷贝 V_2，对于铸造相同的加密串。（5）和（6）满足使用随机串，在第一个解密圈里，每个中心检查随机串的值，在解密值，确信所有这些密文可被计算。在第二个解密圈结束时，每个投票者检查自己的串 s_j，确信选择是被计算了的（选择随机串的一个足够大的空间，可以消除复制的风险）。注意为了验证选举的正确性，需要与所有的选举者合作。（一个否定的特征特别在一个大的协议里）。（7）要求一个较长的讨论。如果关心选举的秘密会丢失，因为多方不守信用，那么协议是理想的。事实上，甚至如果 n-1 个中心合作，将不能知道哪个团体进行了怎样的选举。事实上，需要知道所有的置换 π_i 。然而甚至如果一个政府代理失败，那么对于一个彻底的例子，所有的系统分割。所有选举需要重复。（8）不满足，事实上，v_j 和 s_j 选举者是能够被强行显示的。试着对选举者展开发现和既然公开的值，不能与密文 $y_{n+1,j}$。

12.4.2 一个容错选举协议

在这一节描述协议有如下的特征：

- 满足（4），意味着将是不可能的，复制其他人的选举。（没有协议）
- 不能要求每个选举者公开验证总数，要求合作（对（6）的解决多于上述要求）
- 介绍容错：固定一个开端 t 并且假设是否这里比 t 坏的中心少的协议将是完全恰当的，

每个选举的秘密是被保护的。（对于（7）有比上述更好的解释）
协议还是容易受强迫的影响，最后将讨论这一点。
在这个进展之后的观点是归于 Josh Benaloh[31]的。在如下章节描述的协议在 Cramer, ranklin, choemakers 和 Yung[64]的文献中是最有效的。

同型委托：

假设 B 是一个许诺体制（一个基本的单向函数）。
也就是说一个承诺体制 B 是一个同形（+，×），如果：B（X+Y）=B（X）×B（Y）
这种承诺一个可能的例子是如下的（Pederson[155]发明的）
基于同型委托的离散对数：假设 p 是一个形如 p=kq+1 的素数，假设 g，h 是阶为 q 的两个子群，假设没人知道离散对数，在 h 的基 g。在[1，…，q]中委托一个数据 m。

$$B_a(m) = g^a h^m \quad (11.2)$$

对于一个随机选择模 q 的 a，公开一个承诺 a 和 m 必须显露。
注意有一个 a（+，×）-同形委托体制。

$$B_{a1}(m_1)B_{a2}(m_2) = g^{a1}h^{m1}g^{a2}h^{m2} = g^{a1+a2}h^{m1+m2} = B_{a1+a2}(m_1 + m_2)$$

现在假设 E 是一个（+，×）-同形承诺体制。

12.4.3 协议

因为介绍的理由，将看到两个版本的协议。首先假设仅仅有一个中心。然后表示怎样产生许多中心的观点。

投票构造——1 个中心

假设既然仅有一个中心 C，假设 E 是加密体制。

假设选举既不是-1 也不是 1，每个投票者 V_j 通过计算进行加密投票，并且记录 $B_{aj}(v_j)$ ，对一个随机选择 a_j 。V_j 也发送值 a_j 和 v_j 到 C 加密。

选举者必须证明选举正确，（例如，加密 a-1 或 a1）。通过构造有效的零知识证明。

对于基于上述体制的同形承诺体制，有一个非常有效的协议，假设简单的降下索引 j。
对于 v=1：

1、 选举者 V 随机选择模 q 的 a, r_1, d_1, w_2。张贴 $B_a(v) = g^a h$ 并且也粘贴 $a_1 = g^{r1}(B_a(v)h)^{-d1}$，$a_2 = g^{w2}$ 。
2、 中心 C 发送一个模 q 的挑战随机数 c。
3、 投票者 V 应答如下：V 计算 $d_2 = c-d_1$ 并且 $r_2 = w_2+ad_2$ 并且粘贴 d_1，d_2，r_1，r_2。
4、 中心检查

- $d_1+d_2 = c$
- $g^{r1} = a_1(B_a(v)h)^{d1}$
- $g^{r2} = a_2(B_a(v)h)^{d2}$

对于 v= —1
1、 选举者 V 随机选择模 q 的 a，r_2，d_2，w_1。张贴 $Ba(v) = g^a /h$ 并且也张贴 $a_1 = g^{w1}$ ，$a_2 = G^{r2}(B_a(v)/h)^{-d2}$ 。
2、 中心 C 发送一个模 q 的挑战随机数 c。
3、 选举者 V 应当如下：V 计算 $d_1 = c – d_2$，并且 $r_1 = w_1 + ad_1$。并且张贴 d_1，d_2，r_1，r_2 。
4、 中心 C 检查：

·$d_1+d_2 = c$
·$g^{r1} = a_1(B_a(v)h)^{d1}$

$\cdot g^{r2} = a_2(B_a(v)h)^{d2}$
从现在开始，指出上述协议作为 Proof($B_a(v)$)。

记录计算—1 中心

在前一阶段的结束时，对于每个投票者 V_j 剩下了 $B_{aj}(v_j)$。中心剩余了总共 $T=\sum_j v_j$ 。并且值 $A=\sum_j a_j$ 。每个人检查计数器的正确性，计算：$B_A(T) = \prod_j B_{aj}(v_j)$ 。因为 B 的同型特征，对于正确的计数是真的。协议的一个中心版本也有缺点，中心知道每个人的选票。

N 中心选举构造

假设 n 中心 $C_1 \ldots C_n$ ，并且假设 E_i 是 C_i 的加密函数。
在这个条件下，投票者 V_j 按照如下的方式加密选举者 v_j ，首先通过记录委托选举者：

$$B_j = B_{aj}(v_j)$$

对于一个随机模 q 选择 a_j ，同时也证明这是一个通过执行 Proof($B_{aj}(v_j)$)。在使用 Shamir(t,n)开始秘密共享体制时，中心可以共享值 a_j 和 v_j 。也就是说，随机选择阶数为 t 的多项式 $H_j(X)$和 $A_j(X)$，因此，$H_j(0)= v_j$ ，并且 A_j （0）= a_j 。假设：

$$R_j(X)=v_j + r_{1,j}X + \ldots + r_{t,j} X^t$$
$$S_j(X)=a_j + s_{1,j}X + \ldots + s_{t,j} X^t$$

所有系数是模 q 的。
现在选举者发送值 $u_{i,j} = R_j(i)$，并且 $w_{i,j} = S_j(i)$到中心 C_i （用 E_i 加密）
最后，通过记录委托多项式 H_j 的系数：$B_{l,j} = B_{sl,j}(r_{l,j})$
中心执行如下的检查：

$$G^{wi,j} h^{ui,j} = B_j \prod_{l=1}^{t} (B_{l,j})^{il} \qquad (11.3)$$

确认收到的加密共享是正确的。
N 中心的总数计算

每个中心 C_i 张贴部分总数：$T_i = \sum_j u_{i,j}$
这是每个参与者选举共享的随机串 a_j 的共享总数。
没有人可以检测中心是有启迪作用的，使用承诺体制 B 的同型特征。事实上，必须保证：

$$g^{Ai}h^{Ti} = \prod_{j=1}^{m} (B_j \prod_{l=1}^{t} (B_{l,j})^{jl} \qquad (11.4)$$

注意：正确的 T_i 是计数（t,n）Shamir 共享体制是正确的。因此，足够的获取 t+1 个篡

改总数。

注意：等式（11.3）和（11.4）是有根据的，在假设没有知道基于 h 的 g 离散对数。实际上，在两个办法中，知道可以打开承诺 B 的一些值，因此展现满足等式的不正确的值。

分析：开始一个接一个的通过特点，（1）和（2）明显是满足的。假设至多 t 中心可以合作学习选择假设。如果 t+1 中心合作，那么选举的私密性就失去了。（4）对于如下的理由是真的：假设 V_1 是 V_2 的复制作用，当进行证明选举的正确性时，是证明的点（例如 Proof(B) 的任务，将比 V_2 收到不同的挑战数 c。不能回答是否可以从选举中排除。（5）在离散对数假设条件下是真的（见上述注记）。（6）是真的，任何人可以检查 ZK 证明并且等式（11.3）和（11.4）。（7）是真实的，仅仅需要 t+1 个好的中心恢复计数。容易看到因为需要 t+1 个好的中心，至多 t 个中心是坏的，破坏中心的最大数对于可容忍的协议是 n/2 – 1。（8）不满足，这是因为一些人能够强制通过 a 和 v 进行启迪，当粘贴委托 $B_a(v)$。

12.4.4 非强制条件

选举者的强制条件，可能是至多复杂的一个。这意味着什么是严格的？在多少条路里能够进行强制，试着强制一个投票者给定一个投票。假设试着简化困难，考虑两个强制的可能。一个在选举前联系投票者，一个在选举结束后可以联系投票者。

在强制前有很大的能量，可以区别选举者试着投票，也就是在协议过程中怎样随机。这个基本上是固定协议过程中的投票和行为。如果投票者不遵守，对于强制测试这类事件是容易的。有一些解决办法是使用一些物理假设。例如，一个人可以允许选举者交换一个有限的数，使用投票中心[30，181]的安全信道。有希望从通知防止投票者不是如下的用法。或者一个人可以假设投票者对进行随机加密信息使用一些篡改证明设计。这可以防止强制使用者进行随机投币，作为使用者没有将要产生的篡改证明控制。

投票后的强制执行少量的能量。能够仅仅进行投票，要求看投票 v 和随机 ρ 在协议中对投票者的使用。也许对选举者是一个办法构造不同的 v’和 ρ’与协议的完成相配。在协议里这是不可能的（除非投票者可以解决离散对数问题），最近，仍然有这样目的的协议被 Canetti 和 Gennaro[51]提出。使用了一个叫做可否认加密的新技术，（由 Canetti，Dwork,Naor 和 Ostrovsky[50]发明），这是公钥概率加密 E 的一种新的模式，有如下的性质：

假设 m 是消息，r 是发送者的随机投币事件。发送者计算密文 $C=E_r(m)$。在一些人接近后，询问 m 的值，发送者将可以产生 m’和 r’，使得 $E_{r'}(m') = c$。

12.5 数字现金

在 Internet 上使用金钱传输是第一位的意思，进行发送信用卡信息和在之前通过买方建立一个帐户。

基于 Internet 购物的主要的反对派是因为不是匿名的。事实上，对检测是容易受影响的，既然消费者的身份是每次在购买时确立的。在实际生活中，对使用现金有二选一，然而在没有建立同一性买一些东西。数字现今的期限是描述密码技术和协议体制的，目标是产生基于 Internet 购物的现金概念。

首先，在公钥密码学上描述一个普通的数字现金进展，这个进展是最初由 David Chaum[54]建议的。基于这种进展的体制是完成匿名的特征。

12.5.1 数字现金要求的特征

从数字现金体制中的特征至少是：

- 伪造困难
- 副本应该是既可防止又可探测的
- 保持消费者的匿名
- 在大数据库中保持极小在线操作

12.5.2 第一次尝试协议

一个数字现金体制通常由三个协议组成，撤消协议允许使用者从银行获得数字货币。一个支付协议是使用者通过数字交换购买货物。最后，一个存款协议是买主在帐户上返回银行协议。在如下的协议里，假设银行有秘密密钥 SK_B 签署消息，并且相应的强公共密钥 PK_B 对每个人是已知的。使用符号$\{M\}_{SK}$ 定义消息 M 在 SK 下的签名。假设看到可能的数字现金协议。

撤回协议：

1、 用户告知银行希望撤回$100。

2、 银行返回一个$100 的帐单，如同：

$\{I\ am\ a\ \$100\ bill\ ,\ \#\ 4527)\}_{SKB}$，从用户帐户抽走$100。

3、用户检查签名并且根据接收到的帐单是正确的。

付款协议：

1、 用户付给买主帐单

2、 买主检查签名，如果是有效的，接收帐单

存款协议：

1、 买主把帐单提供给银行

2、 买主检查签名是否有效，相信买主帐户

在签名体制的安全上，给定一些适当的假设，清楚的是可以伪造数字货币。然而非常容易复制两边几次消耗一些数字货币。同时也是夹板的，匿名体制没有被保护，链接用户的名字和使用连续的数字，出现在帐单上，并且知道用户花销了货币。

12.5.3 盲签名

假设首先解决匿名难题，这个进展包括 —em 盲签名。用户使用一个帐单里的容器提交给银行，这个办法，银行不能决定帐单的来源，当一个商人作为存款提交。

一个有用的推理：用户使用一个碳纸代替一个检查，然后在一个信封里封装两个内容。用户把信封交给银行，银行在信封外签名然后返回信封到银行。（还没打开之前，实际上银行不能在数字版本下打开信封）。用户然后移动签名检查并可以改动。银行从来没有看到签名是什么样，也不能用用户合作，当被返回时被保存。当然能够通过检查验证签名，同时保证检查有效。

当然，这里有一个问题：银行可以被签名假冒欺骗。例如，一个用户可以告知银行，可以构造 1$的撤消程序，而提交一个$100 的帐单进行签名。银行将不知道签署一个$ 100 的帐

单，并且允许用户欺骗银行$99。随后将处理这个难题，现在显示怎样构造盲签名。

12.5.4 RSA 盲签名

回忆 RSA 盲签名：如果 M 是将要签署的消息，那么签名是 $s = M^{e-1} \bmod n$，这里 m 和 n 是公共已知值。秘密信息是银行执行 $e^{-1} \bmod (\Phi(n))$，用 d 定义。签名是可通过计算 $s^e \bmod n$ 验证，并且验证等于 M mod n。在盲签名的条件下，用户希望银行用 s 提供信息，没有返回 M 给银行。有一个可能的匿名撤消协议。假设 M 是一个$100 的帐单。

撤消协议：

1、 用户选择一些随机数，r mod n

2、 用户计算 $M' = M \cdot r^e \bmod n$

3、 用户给定银行 M

4、 银行返回一个签名 M’，也就是说，$s' = (M')^d \bmod n$，注意：

$$S' = (M')^d = M^d \cdot (r^e)^d = M^d \cdot r$$

5、 银行提交用户一个$100

6、 既然用户知道 r，可以用 r 分割 s，得到：$S = M^d$

支付和存取协议与上述保持相同，这解决了用户匿名的问题，当货币返回银行，这里没有用户和银行之间的链接。

有两个难题。

1、 银行能够被签名欺骗，也就是说，（象$100 帐单是一个$1 的帐单）

2、 货币可以被复制并且两倍开销

12.5.5 固定美元数量

上述难题一个可能的解决是仅仅有一个命名（例如，每个公钥）。也就是说银行有几个公钥 $PK1_B$，…并且签名使用 PKi_B 是有效的，仅仅在 i 美元的帐单上。

另一种可能是使用“剪贴和复制”过程：

1、 用户产生 100 个$20 帐单

2、 进行匿名

3、 把交到银行

4、 银行捡出一个签名（随机），要求用户对所有剩余的不再匿名。（通过恢复 r’s）。在签名返回前，银行保证所有被匿名的帐单是正确的。

在这个条件下，用户仅仅有 1/100 欺骗的机会。当然，一个人可以建立协议产生一个更小的欺骗机会。（例如，要求用户提供更多的盲签名机会）因此，现在有一个协议满足匿名要求，提供足够小的机会欺骗，还需要处理双倍开销的机会。

12.5.6 在线数字签名

在在线数字现金体制的版本上，一人要求所有这些帐单在数据库里被检查。在付款协议过程中，买主应该传送帐单到银行，并询问是否收到帐单。如果这是帐单第一次被使用，买主接受，尽管这是简单的解决，会招致一个作为付款协议的通信，看来更象银行卡的传输，当买主等到完成交易认证。数据库的大小可以尝试解决现有的银行问题。

注意预防双倍消耗，将介绍一个不需要在线认证的双倍探测消耗。

12.5.7 离线数字现金

离线数字现金背后的思想如下：在支付协议中，用户在帐单上被迫写一个“随机验证串”，或 RIS。

RIS 必须有如下的特征：

- 对货币的每个付款必须是不同的
- 仅仅用户可以产生一个有效的 RIS
- 在两个相同的货币上，两个不同的 RIS 允许银行返回用户名

如果银行收到不同值的两个同样的帐户，那么用户被欺骗，上述观点首先在[52]中出现。有一个可能的解决，假设 H 是一个单向函数。

撤消协议：

1、 用户使用$20 的 100 个帐单，看起来是：

M_i = (I’m $20 bill, # 4527i, $y_{i,1}$， $y'_{I,1}$， $y_{i,2}$， $y'_{I,2}$， …， $y_{i,K}$， $y'_{I,K}$)

这里，$y_{i,j} = H(x_{i,j})$,$y'_{I,j} = H(x'_{I,j})$，这里 $x_{i,j}$ 和 $x'_{I,j}$ 是随机选择的，条件是：

$$x_{i,j} \oplus x'_{I,j} = \text{Username} \quad \forall i,j$$

2、用户匿名所有 M_i 成为随机信息 M'_i ，（使用上述盲协议的轮廓）。并且发送到银行。

3、 银行要求使用者在 100 个匿名帐户中选择使用 99 个用户。

4、 当用户打开帐号是，恰当的选择 $x_{i,j}$ 和 $x'_{I,j}$。

5、银行检查不仅仅是帐号是$20，并且 $y_{i,j} = H(x_{i,j})$ ， $y'_{i,j} = H(x'_{i,j})$并且 $x_{i,j} \oplus x'_{I,j}$ = Username $\forall$ i,j，对于没有盲签名的帐户。

6、银行仅仅在盲消息（M'_{17}）时返回签名。

7、用户限制在 M_{17} 上的 s_{17}。

现在假设简单停止索引 i=17，付款协议调整强制用户产生一个在货币上的 RIS。RIS 是对每个 j=1,…,K，是 x_j 或 x'_j 中的一个。其中一个是从买主中依靠一个随机挑战。

支付协议：

1、 用户给定 M，s 到买主

2、 买主检查帐单上的银行签名是否有效，应当一个长度为 K 的随机比特串，$b_1,\ldots,b_k$

3、 如果 $b_j = 0$，用户使用 x_j。否则返回 x'_j

4、 买主检查 $y_{i,j} = H(x_{i,j})$ ，$y'_{i,j} = H(x'_{i,j})$，无论哪一个是这个条件，如果保留上述等式，接受帐单。

注意上述特征和 RIS 是否满足。实际概率是相同的 RIS 中不同的支付，产生 2^{-k} ，既然买主选择随机挑战，仅仅用户可以产生一个有效的 RIS，既然函数是单向的。最后在没有用户名的两个不同的 RIS 中，似乎两个 RIS 是不同的，必须有一个索引 j，对每一个同时有 x_j 或 x'_j。

存款协议：

1、 买主带来货币 M,s,RIS 返回银行

2、 银行证实签名，检查是否是货币 M,s,已经返回银行

3、 如果货币已经在数据库中了，银行比较两个货币的 RIS，如果 RIS 是不同的，那么用户双倍花销货币，否则是买主试着存放货币两次。

Bibliography

[1] ISO/IEC 9796. Information technology security techniques { digital signature scheme giving message recovery, 1991. International Organization for Standards.

[2] L. M. Adleman. On breaking generalized knapsack public key cryptosystems. In *Proc. 15th ACM Symp. on Theory of Computing*, pages 402{412, Boston, 1983. ACM.

[3] L. M. Adleman. Factoring numbers using singular integers. Technical Report TR 90-20, U.S.C. Computer Science Department, September 1990.

[4] L. M. Adleman and M. A. Huang. Recognizing primes in random polynomial time. In *Proc. 19th ACM Symp. on Theory of Computing*, pages 462{469, New York City, 1987. ACM.

[5] L. M. Adleman, C. Pomerance, and R. S. Rumely. On distinguishing prime numbers from composite numbers. *Ann. Math.*, 117:173{206, 1983.

[6] W. B. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA/Rabin functions: certain parts are as hard as the whole. *SIAM J. Computing*, 17(2):194{209, April 1988.

[7] D. Angluin. Lecture notes on the complexity of some problems in number theory. Technical Report TR-243, Yale University Computer Science Department, August 1982.

[8] Eric Bach. How to generate factored random numbers. *SIAM J. Computing*, 17(2):179{193, April 1988.

[9] D. Balenson. *RFC 1423: Privacy Enhancement for Internet Electronic Mail: Part III { Algorithms, Modes, and Identi¯ers*. Internet Activities Board, February 1993.

[10] D. Beaver. E±cient multiparty protocols using circuit randomization. In J. Feigenbaum, editor, *Proc. CRYPTO 91*, pages 420{432. Springer, 1992. Lecture Notes in Computer Science No. 576.

[11] M. Bellare, R. Gu¶erin, and P. Rogaway. XOR MACs: New methods for message authentication using ¯nite pseudorandom functions. In Don Coppersmith, editor, *Proc. CRYPTO 95*, pages 15{28. Springer, 1995. Lecture Notes in Computer Science No. 963.

[12] M. Bellare, J. Kilian, and P. Rogaway. The security of cipher block chaining. In Yvo G. Desmedt, editor, *Proceedings of Crypto 94*, volume 839 of *Lecture Notes in Computer Science*, pages 341{358. Springer-Verlag, 1994. Full version to appear in *J. Computer and System Sciences*, available via http://www-cse.ucsd.edu/users/mihir.

[13] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362{399, December 2000. 238 Cryptography: Lecture Notes 239

[14] M. Bellare and S. Micali. How to sign given any trapdoor permutation. *Journal of the ACM*, 39(1):214{233,January 1992.

[15] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing e±cient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62{73, Fairfax, 1993. ACM.

[16] M. Bellare and P. Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Proc. CRYPTO 93*, pages 232{249. Springer, 1994. Lecture Notes in Computer Science No. 773.

[17] M. Bellare and P. Rogaway. Provably secure session key distribution{ the three party case. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 57{66, Las Vegas, 1995. ACM.

[18] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Proceedings of Crypto 96*, volume 1109 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[19] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade con-struction and its concrete security. In *Proc. 37th IEEE Symp. on Foundations of Comp. Science*. IEEE,

1996.

[20] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security treatment of sym-metric encryption: Analysis of the DES modes of operation. In *Proc. 38th IEEE Symp. on Foundations of Comp. Science*. IEEE, 1997.

[21] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Proceedings of Crypto 98*, volume 1462 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[22] Mihir Bellare and Tadayoshi Kohno. Hash function balance and its impact on birthday attacks. In *Proceedings of EUROCRYP'04*, volume 3027 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[23] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography. Notes, 1996-2004.

[24] Mihir Bellare and Phillip Rogaway. Distributing keys with perfect forward secrecy. Manuscript, 1994.

[25] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *Proceedings of EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[26] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In *Proceedings of EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[27] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In *Proceedings of CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[28] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for fault-tolerant distributed computing. In *Proc. 20th ACM Symp. on Theory of Computing*, pages 1{10, Chicago, 1988. ACM.

[29] J. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret sharing. In A. M. Odlyzko, editor, *Proc. CRYPTO 86*. Springer, 1987. Lecture Notes in Computer Science No. 263.

[30] J. Benaloh and D. Tuinstra. Receipt-free secret ballot elections. In *26th ACM Symposium on Theory of Computing*, pages 544{553, 1994.

[31] Josh Benaloh. Veri¯able secret ballot elections Technical Report TR{561, Yale Department of Computer Science, September 1987.

[32] R. Berger, R. Peralta, and T. Tedrick. A provably secure oblivious transfer protocol. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Proc. EUROCRYPT 84*, pages 379{386. Springer-Verlag, 1985. Lecture Notes in Computer Science No. 209.240 Goldwasser and Bellare

[33] E. Biham and A. Shamir. Di®erential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*,4(1):3{72, 1991.

[34] Eli Biham and Adi Shamir. Di®erential cryptanalysis of the full 16-round DES. In Ernest F. Brickell,editor, *Proc. CRYPTO 92*, pages 487{496. Springer-Verlag, 1992. Lecture Notes in Computer Science No.740.

[35] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. Systematic design of two-party authentication protocols. In J. Feigenbaum, editor, *Proc. CRYPTO 91*, pages 44{61. Springer, 1992. Lecture Notes in Computer Science No. 576.

[36] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: fast and secure message authentication. In *Proceedings of CRYPTO'99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[37] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 National Computer Conference*, pages 313{317. AFIPS, 1979.

[38] D. Bleichenbacher. A chosen ciphertext attack against protocols based on the RSA encryption standard pkcs #1. In *Proceedings of Crypto 98*, volume 1462 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[39] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM J.Computing*, 15(2):364{383, May 1986.

[40] M. Blum. Coin °ipping by telephone. In *Proc. IEEE Spring COMPCOM*, pages 133{137. IEEE, 1982.

[41] M. Blum. How to exchange (secret) keys. *Trans. Computer Systems*, 1:175{193, May 1983. (Previously published in ACM STOC '83 proceedings, pages 440{447.).

[42] M. Blum. Independent unbiased coin °ips from a correlated biased source: A ¯nite state Markov chain. In *Proc. 25th IEEE Symp. on Foundations of Comp. Science*, pages 425{433, Singer Island, 1984. IEEE.

[43] M. Blum and S. Goldwasser. An e±cient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and D. C. Chaum, editors, *Proc. CRYPTO 84*, pages 289{302. Springer, 1985. Lecture Notes in Computer Science No. 196.

[44] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Computing*, 13(4):850{863, November 1984.

[45] M. Blum, A. De Santis, S. Micali, and G. Persiano. Noninteractive zero-knowledge. *SIAM J. Computing*, 20(6):1084{1118, December 1991.

[46] D. Boneh and R. Venkatesan. Hardness of computing the most signi¯cant bits of secret keys in Di±e-Hellman and related schemes. In *Proceedings of CRYPTO'96*, Lecture Notes in Computer Science. Springer-Verlag, 1996.

[47] Gilles Brassard and Claude Cr¶epeau. Zero-knowledge simulation of boolean circuits. In A.M. Odlyzko, editor, *Proc. CRYPTO 86*, pages 223{233. Springer-Verlag, 1987. Lecture Notes in Computer Science No.263.

[48] E. F. Brickell. Solving low density knapsacks. In D. Chaum, editor, *Proc. CRYPTO 83*, pages 25{37, New York, 1984. Plenum Press.

[49] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18{36, February 1990.

[50] Ran Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable encryption. In *Proc. CRYPTO 97*, Lecture Notes in Computer Science. Springer-Verlag, 1997.

[51] Ran Canetti and R. Gennaro. Incoercible multiparty computation. In *Proc. 37th IEEE Symp. on Foun-dations of Comp. Science*, 1996. Cryptography: Lecture Notes 241

[52] E.R. Can¯eld, P. ErdÄos, and C. Pomerance. On a problem of Oppenheim concerning `Factorisatio Nu-merorum'. *J. Number Theory*, 17:1{28, 1983.

[53] M. Cerecedo, T. Matsumoto, and H. Imai. E±cient and secure multiparty generation of digital signatures based on discrete logarithm. *IEICE Trans. on Fund. Electr. Comm. and Comp. Sci.*, E76{A(4):532{545,1993.

[54] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84{88, February 1981.

[55] D. Chaum, C. Cr¶epeau, and I. Damgºard. Multiparty unconditionally secure protocols. In Carl Pomerance, editor, *Proc. CRYPTO 87*, pages 462{462. Springer-Verlag, 1988. Lecture Notes in Computer Science No.293.

[56] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *Proc. CRYPTO 88*, pages 319{327. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 403.

[57] D. L. Chaum. Veri¯cation by anonymous monitors. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81*, pages 138{139. U.C. Santa Barbara Dept. of Elec. and Computer Eng., 1982. Tech Report 82-04.

[58] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communica-tion complexity. *SIAM J. Computing*, 17(2):230{261, April 1988.

[59] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Veri¯able secret sharing and achieving simultaneity in the presence of faults. In *Proc. 26th IEEE Symp. on Foundations of Comp. Science*, pages 383{395, Portland, 1985. IEEE.

[60] B. Chor and R. L. Rivest. A knapsack type public-key cryptosystem based on arithmetic in ¯nite ¯elds. *IEEE Trans. Inform. Theory*, 34(5):901{909, September 1988.

[61] D. Coppersmith. Evaluating logarithms in *GF*(2*n*). In *Proc. 16th ACM Symp. on Theory of Computing*, pages 201{207, Washington, D.C., 1984. ACM.

[62] D. Coppersmith, M. K. Franklin, J. Patarin, and M. K. Reiter. Low-exponent RSA with related messages. In Ueli Maurer, editor, *Advances in Cryptology - EuroCrypt '96*, pages 1{9, Berlin, 1996. Springer-Verlag. Lecture Notes in Computer Science Volume 1070.

[63] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[64] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung. Multi-authority secret-ballot elections with linear work. In *EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 72{83.Springer-Verlag, 1996.

[65] R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. Theory of Cryptog-raphy Library Record 99-01, 1999.

[66] J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer, Berlin, 2001.

[67] I. Damg°ard. A design principle for hash functions. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 416{427. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[68] D. Denning and G. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):533{536, 1981.

[69] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In J. Feigenbaum, editor, *Proc. CRYPTO 91*, pages 457{469. Springer, 1992. Lecture Notes in Computer Science No. 576.

[70] Yvo G. Desmedt. Threshold cryptography. *European Transactions on Telecommunications*, 5(4):449{457, July 1994.242 Goldwasser and Bellare

[71] W. Di±e and M. E. Hellman. Multiuser cryptographic techniques. In *Proc. AFIPS 1976 National Com-puter Conference*, pages 109{112, Montvale, N.J., 1976. AFIPS.

[72] W. Di±e and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644{654, November 1976.

[73] Whit¯eld Di±e, Paul C. Van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes, and Cryptography*, 2(2):107{125, June 1992.

[74] H. Dobbertin. Cryptanalysis of MD5. Rump session of Eurocrypt 96, 1996.

[75] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 542{552. ACM, 1991.

[76] D. Dolev, S. Even, and R. M. Karp. On the security of ping-pong protocols. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Proc. CRYPTO 82*, pages 177{186, New York, 1983. Plenum Press.

[77] D. Dolev and A. C. Yao. On the security of public key protocols. In *Proc. 22nd IEEE Symp. on Foundations of Comp. Science*, pages 350{357, Nashville, 1981. IEEE.

[78] C. Dwork and M. Naor. An e±cient existentially unforgeable signature scheme and its applications. In Yvo G. Desmedt, editor, *Proc. CRYPTO 94*, pages 234{246. Springer, 1994. Lecture Notes in Computer Science No. 839.

[79] P. Elias. The e±cient construction of an unbiased random sequence. *Ann. Math. Statist.*, 43(3):865{870, 1972.

[80] S. Even and O. Goldreich. On the security of multi-party ping-pong protocols. In *Proc. 24th IEEE Symp. on*

Foundations of Comp. Science, pages 34{39, Tucson, 1983. IEEE.

[81] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28:637{647, 1985.

[82] P. Feldman. A practical scheme for non-interactive veri¯able secret sharing. In *Proc. 28th IEEE Symp. on Foundations of Comp. Science*, pages 427{438, Los Angeles, 1987. IEEE.

[83] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identi¯cation and signature problems. In A. M. Odlyzko, editor, *Proc. CRYPTO 86*, pages 186{194. Springer, 1987. Lecture Notes in Computer Science No. 263.

[84] R. Fischlin and C. Schnorr. Stronger security proofs for RSA and Rabin bits. In *EUROCRYPT'97*, volume 1223 of *Lecture Notes in Computer Science*, pages 267{279. Springer-Verlag, 1997.

[85] National Institute for Standards and Technology. A proposed federal information processing standard for digital signature standard (DSS). Technical Report FIPS PUB XX, National Institute for Standards and Technology, August 1991. DRAFT.

[86] Y. Frankel, P. Gemmell, and M. Yung. Witness-based cryptographic program checking and robust function sharing. In *28th ACM Symposium on Theory of Computing*, 1996.

[87] A. M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias, and A. Shamir. Reconstructing truncated integer variables satisfying linear congruences. *SIAM J. Computing*, 17(2):262{280, April 1988.

[88] Electronic Frontier Foundation. E® des cracker project. ttp://www.eff.org/Privacy/Crypto/Crypto_ misc/DESCracker/.

[89] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. In *CRYPTO'01*, volume 2139 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[90] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31:469{472, 1985. Cryptography: Lecture Notes 243

[91] M. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[92] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT'99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[93] R. Gennaro, S. Jarecki, Hugo Krawczyk, and T. Rabin. Robust and e±cient sharing of rsa functions. In *CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[94] R. Gennaro, S. Jarecki, Hugo Krawczyk, and T. Rabin. Robust threshold dss signatures. In *EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 354{371. Springer-Verlag, 1996.

[95] O. Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. Technical Report MIT/LCS/TM-315, MIT Laboratory for Computer Science, September 1986.

[96] O. Goldreich. A uniform complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21{53, 1993.

[97] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792{807, October 1984.

[98] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *21st ACM Symposium on Theory of Computing*, 1989.

[99] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Proc. 27th IEEE Symp. on Foundations of Comp. Science*, pages 174{187, Toronto, 1986. IEEE.

[100] S. Goldwasser and J. Kilian. Almost all primes can be quickly certi¯ed. In *Proc. 18th ACM Symp. On Theory of Computing*, pages 316{329, Berkeley, 1986. ACM.

[101] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proc. 14th ACM Symp. on Theory of Computing*, pages 365{377, San Francisco, 1982. ACM.

[102] S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, 28(2):270{299, April 1984.

[103] S. Goldwasser, S. Micali, and C. Racko®. The knowledge complexity of interactive proof-systems. In *Proc. 17th ACM Symp. on Theory of Computing*, pages 291{304, Providence, 1985. ACM.

[104] S. Goldwasser, S. Micali, and C. Racko®. The knowledge complexity of interactive proof-systems. *SIAM. J. Computing*, 18(1):186{208, February 1989.

[105] S. Goldwasser, S. Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281{308, April 1988.

[106] S. Goldwasser, S. Micali, and P. Tong. Why and how to establish a private code on a public network. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 134{144, Chicago, 1982. IEEE.

[107] S. Goldwasser, S. Micali, and A. Yao. Strong signature schemes. In *Proc. 15th ACM Symp. on Theory of Computing*, pages 431{439, Boston, 1983. ACM.

[108] S. W. Golomb. *Shift Register Sequences*. Aegean Park Press, Laguna Hills, 1982. Revised edition.

[109] L. Harn. Group{oriented (*t; n*) threshold digital signature scheme and digital multisignature. *IEE Proc. Comput. Digit. Tech.*, 141(5):307{313, 1994.

[110] J. Hastad. Solving simultaneous modular equations of low degree. *SIAM J. Computing*, 17(2):336{341, April 1988.

[111] J. Hºastad. Pseudo-random generators under uniform assumptions. In *22nd ACM Symposium on Theoryof Computing*, 1990. 244 Goldwasser and Bellare

[112] J. Hastad, A.W. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides *o*(*n*) bits. *Journal of Computer and Systems Sciences*, 47:376{404, 1993.

[113] Johan Hºastad, Russell Impagliazzo, Leonid Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364{1396, 1999.

[114] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proc.30th IEEE Symp. on Foundations of Comp. Science*, 1989.

[115] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way func-tions. In *Proc. 21st ACM Symp. on Theory of Computing*, pages 12{24, Seattle, 1989. ACM.

[116] ISO. Data cryptographic techniques { data integrity mechanism using a cryptographic check function employing a block cipher algorithm. ISO/IEC 9797, 1989.

[117] D. Johnson, A. Lee, W. Martin, S. Matyas, and J. Wilkins. Hybrid key distribution scheme giving key record recovery. *IBM Technical Disclosure Bulletin*, 37(2A):5{16, February 1994. See also U.S. Patent 5,142,578.

[118] D. Johnson and M. Matyas. Asymmetric encryption: Evolution and enhancements. *RSA Labs Cryptobytes*, 2(1), Spring 1996.

[119] B. Kaliski and M. Robshaw. Message authentication with MD5. *CryptoBytes*, 1(1):5{8, Spring 1995.

[120] B. S. Kaliski, Jr. A pseudo-random bit generator based on elliptic logarithms. In A.M. Odlyzko, editor, *Proc. CRYPTO 86*, pages 84{103. Springer-Verlag, 1987. Lecture Notes in Computer Science No. 263.

[121] B. S. Kaliski, Jr. *Elliptic Curves and Cryptography: A Pseudorandom Bit Generator and Other Tools*. PhD thesis, MIT EECS Dept., January 1988. Published as MIT LCS Technical Report MIT/LCS/TR-411 (Jan. 1988).

[122] R. Kannan, A. Lenstra, and L. Lov¶asz. Polynomial factorization and non-randomness of bits of algebraic and some transcendental numbers. In *Proc. 16th ACM Symp. on Theory of Computing*, pages 191{200,Washington, D.C., 1984. ACM.

[123] J. Kilian. Founding cryptography on oblivious transfer. In *Proc. 20th ACM Symp. on Theory of Computing*,

pages 20{31, Chicago, 1988. ACM.

[124] L. Knudsen and J. E. Mathiassen. A chosen-plaintext linear attack on des. In B. Schneier, editor, *Fast Software Encryption*. Springer, 2000. Lecture Notes in Computer Science No. 1978.

[125] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 1969. Second edition, 1981.

[126] Hugo Krawczyk. Skeme: A versatile secure key exchange mechanism for internet. In *Proceedings of the Symposium on Network and Distributed System Security*, 1996.

[127] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. Hmac: Keyed-hashing for message authentication, February 1997. Internet RFC 2104.

[128] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. In *Proc. 24th IEEE Symp. on Foundations of Comp. Science*, pages 1{10, Tucson, 1983. IEEE.

[129] X. Lai and J. Massey. A proposal for a new block encryption standard. In I.B. Damg°ard, editor, *Proc. EUROCRYPT 90*, pages 389{404. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 473.

[130] L. Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, October 1979.

[131] A. K. Lenstra and H. W. Lenstra, Jr. Algorithms in number theory. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science (Volume A: Algorithms and Complexity)*, chapter 12, pages 673{715. Elsevier and MIT Press, 1990. Cryptography: Lecture Notes 245

[132] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lov¶asz. Factoring polynomials with rational coe±cients. *Mathematische Ann.*, 261:513{534, 1982.

[133] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard. The number ¯eld sieve. In *Proc.22nd ACM Symp. on Theory of Computing*, pages 564{572, Baltimore, Maryland, 1990. ACM.

[134] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649{673, 1987.

[135] R. Lipton. How to cheat at mental poker. In *Proc. AMS Short Course on Cryptography*, 1981.

[136] D. L. Long and A. Wigderson. The discrete logarithm problem hides $O(\log n)$ bits. *SIAM J. Computing*, 17(2):363{372, April 1988.

[137] M. Luby, S. Micali, and C. Racko®. How to simultaneously exchange a secret bit by °ipping a symmetrically biased coin. In *Proc. 24th IEEE Symp. on Foundations of Comp. Science*, pages 11{22, Tucson, 1983. IEEE.

[138] M. Luby and C. Racko®. How to construct pseudorandom permutations and pseudorandom functions. *SIAM J. Computing*, 17(2):373{386, April 1988.

[139] Maurice P. Luby and C. Racko®. A study of password security. In Carl Pomerance, editor, *Proc. CRYPTO 87*, pages 392{397. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 293.

[140] M. Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology-EuroCrypt '93*, pages 386{397, Berlin, 1993. Springer-Verlag. Lecture Notes in Computer Science Volume 765.

[141] Ueli M. Maurer. Towards the equivalence of breaking the Di±e-Hellman protocol and computing discrete algorithms. In Yvo G. Desmedt, editor, *Proc. CRYPTO 94*, pages 271{281. Springer, 1994. Lecture Notes in Computer Science No. 839.

[142] R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114{116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

[143] R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, IT-24:525{530, September 1978.

[144] Ralph C. Merkle. A certi¯ed digital signature. In G. Brassard, editor, *Proc. CRYPTO 89*, pages

218{238.Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[145] Ralph C. Merkle. One way hash functions and DES. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 428{446. Springer, 1990. Lecture Notes in Computer Science No. 435.

[146] Ralph Charles Merkle. Secrecy, authentication, and public key systems. Technical report, Stanford University, Jun 1979.

[147] M. Merritt. *Cryptographic Protocols*. PhD thesis, Georgia Institute of Technology, February 1983.

[148] S. Micali, C. Racko®, and R. H. Sloan. The notion of security for probabilistic cryptosystems. *SIAM J.Computing*, 17(2):412{426, April 1988.

[149] Gary L. Miller. Riemann's hypothesis and tests for primality. *JCSS*, 13(3):300{317, 1976.

[150] M. Naor and O. Reingold. Synthesizers and their application to the parallel construction of pseudorandom functions. In *Proc. 36th IEEE Symp. on Foundations of Comp. Science*. IEEE, 1995.

[151] M. Naor and O. Reingold. Number-theoretic constructions of e±cient pseudo-random functions. In *Proc.38th IEEE Symp. on Foundations of Comp. Science*. IEEE, 1997.

[152] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. 21st ACM Symp. on Theory of Computing*, pages 33{43, Seattle, 1989. ACM. 246 Goldwasser and Bellare

[153] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *Proc. of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pages 427{437, Baltimore, Maryland, 1990. ACM.

[154] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993{999, December 1978.

[155] R. M. Needham and M. D. Schroeder. Authentication revisited. *Operating Systems Review*, 21(1):7,January 1987.

[156] I. Niven and H. S. Zuckerman. *An Introduction to the Theory of Numbers*. Wiley, 1972.

[157] A. M. Odlyzko. Cryptanalytic attacks on the multiplicative knapsack scheme and on Shamir's fast signa-ture scheme. *IEEE Trans. Inform. Theory*, IT-30:594{601, July 1984.

[158] A. M. Odlyzko. Discrete logarithms in ¯nite ¯elds and their cryptographic signi¯cance. In T. Beth, N. Cot,and I. Ingemarsson, editors, *Proc. EUROCRYPT 84*, pages 224{314, Paris, 1985. Springer. Lecture Notes in Computer Science No. 209.

[159] C. Park and K. Kurosawa. New elgamal type threshold signature scheme. *IEICE Trans. on Fund. Electr. Comm. and Comp. Sci.*, E79{A(1):86{93, 1996.

[160] T. Pedersen. Distributed provers with applications to undeniable signatures. In *EuroCrypt'91*, 1991.

[161] T.P. Pedersen. Non-interactive and information-theoretic secure veri¯able secret sharing. In J. Feigen-baum, editor, *Proc. CRYPTO 91*, pages 129{140. Springer, 1992. Lecture Notes in Computer Science No.576.

[162] E. Petrank and C. Racko®. Cbc mac for real-time data sources. Manuscript, 1997.

[163] J. Plumstead. Inferring a sequence generated by a linear congruence. In *Proc. 23rd IEEE Symp. On Foundations of Comp. Science*, pages 153{159, Chicago, 1982. IEEE.

[164] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over *GF*(*p*) and its cryptographic signi¯cance. *IEEE Trans. Inform. Theory*, IT-24:106{110, January 1978.

[165] D. Pointcheval and J. Stern. Security proofs for signatures. In *Proceedings of EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387{398. Springer-Verlag, 1996.

[166] J. M. Pollard. Theorems on factorization and primality testing. *Proc. Cambridge Philosophical Society* , 76:521{528, 1974.

[167] V. Pratt. Every prime has a succinct certi¯cate. *SIAM J. Comput.*, 4:214{220, 1975.

[168] B. Preneel and P.C. van Oorschot. On the security of two MAC algorithms. In *Proceedings of*

EURO-CRYPT'96, volume 1070 of *Lecture Notes in Computer Science*, pages 19{32. Springer-Verlag, 1996.

[169] Bart Preneel and Paul C. van Oorschot. MDx-MAC and building fast MACs from hash functions. In Don Coppersmith, editor, *Proc. CRYPTO 94*, pages 1{14. Springer, 1995. Lecture Notes in Computer Science No. 963.

[170] M. Rabin. Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, January 1979.

[171] M. Rabin. Probabilistic algorithms for testing primality. *J. Number Theory*, 12:128{138, 1980.

[172] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[173] T. Rabin and M. Ben-Or. Veri¯able secret sharing and multiparty protocols with honest majority. In *21st ACM Symposium on Theory of Computing*, pages 73{85, 1989. Cryptography: Lecture Notes 247

[174] R. L. Rivest and A. Shamir. How to expose an eavesdropper. *Communications of the ACM*, 27:393{395,April 1984.

[175] Ronald L. Rivest. The MD5 message-digest algorithm. Internet Request for Comments, April 1992. RFC 1321.

[176] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120{126, 1978.

[177] John Rompel. One-way functions are necessary and su±cient for secure signatures. In *Proc. 22nd ACM Symp. on Theory of Computing*, pages 387{394, Baltimore, Maryland, 1990. ACM.

[178] J. Rosser and L. Schoen¯eld. Approximate formulas for some functions of prime numbers. *Illinois J.Math.*, 6:64{94, 1962.

[179] RSA Data Security, Inc. *PKCS #1: RSA Encryption Standard*, June 1991. Version 1.4.

[180] RSA Data Security, Inc. *PKCS #7: Cryptographic Message Syntax Standard*, June 1991. Version 1.4.

[181] K. Sako and J. Kilian. Receipt-free mix-type voting schemes. a practical implementation of a voting booth. In *EUROCRYPT'95*, volume 921 of *Lecture Notes in Computer Science*, pages 393{403. Springer-Verlag, 1995.

[182] M. Santha and U. V. Vazirani. Generating quasi-random sequences from slightly-random sources. In *Proc.25th IEEE Symp. on Foundations of Comp. Science*, pages 434{440, Singer Island, 1984. IEEE.

[183] Alredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung. How to share a function securely. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 522{533, Montreal, Canada, 1994. ACM.

[184] C. P. Schnorr. E±cient signature generation by smart cards. *Journal of Cryptology*, 4:161{174, 1991.

[185] R. J. Schoof. Elliptic curves over ¯nite ¯eldsand the computation of square roots mod *p*. *Math. Comp.*, 44:483{494, 1985.

[186] R. Schroeppel and A. Shamir. A *TS*2 = *O*(2*n*) time/space tradeo® for certain NP-complete problems. In *Proc. 20th IEEE Symp. on Foundations of Comp. Science*, pages 328{336, San Juan, Puerto Rico, 1979. IEEE.

[187] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612{613, November 1979.

[188] A. Shamir. On the cryptocomplexity of knapsack schemes. In *Proc. 11th ACM Symp. on Theory of Computing*, pages 118{129, Atlanta, 1979. ACM.

[189] A. Shamir. On the generation of cryptographically strong pseudo-random sequences. In *Proc. ICALP*, pages 544{550. Springer, 1981.

[190] A. Shamir. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 145{152, Chicago, 1982. IEEE.

[191] A. Shamir, R. L. Rivest, and L. M. Adleman. Mental poker. In D. Klarner, editor, *The Mathematical Gardner*, pages 37{43. Wadsworth, Belmont, California, 1981.

[192] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:623{656, 1948.

[193] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:657{715, 1949.

[194] V. Shoup. Oaep reconsidered. In *CRYPTO'01*, volume 2139 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[195] V. Shoup and A. Rubin. Session key distribution for smart cards. In U. Maurer, editor, *Proc. CRYPTO 96*. Springer-Verlag, 1996. Lecture Notes in Computer Science No. 1070. 248 Goldwasser and Bellare

[196] R.D. Silverman. The multiple polynomial quadratic sieve. *Mathematics of Computation*, 48:329{339,1987.

[197] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM J. Computing*, 6:84{85, 1977.

[198] William Stallings. *Network and Internetwork Security Principles and Practice*. Prentice Hall, 1995.

[199] J.G. Steiner, B.C. Neuman, and J.I. Schiller. Kerberos: an authentication service for open network systems. In *Usenix Conference Proceedings*, pages 191{202, Dallas, Texas, February 1988.

[200] Joseph D. Touch. Performance analysis of MD5. *Proceedings SIGCOMM*, 25(4):77{86, October 1995. Also at ftp://ftp.isi.edu/pub/hpcc-papers/touch/sigcomm95.ps.Z.

[201] Gene Tsudik. Message authentication with one-way hash functions. *ACM SIGCOMM, Computer Com-munication Review*, 22(5):29{38, October 1992.

[202] P. van Oorschot and M. Wiener. Parallel collision search with applications to hash functions and discrete logarithms. In *Proceedings of the 2nd ACM Conf. Computer and Communications Security*, November 1994.

[203] U. V. Vazirani. Towards a strong communication complexity theory, or generating quasi-random sequences from two communicating slightly-random sources. In *Proc. 17th ACM Symp. on Theory of Computing*, pages 366{378, Providence, 1985. ACM.

[204] U. V. Vazirani and V. V. Vazirani. Trapdoor pseudo-random number generators, with applications to protocol design. In *Proc. 24th IEEE Symp. on Foundations of Comp. Science*, pages 23{30, Tucson, 1983. IEEE.

[205] Umesh V. Vazirani and Vijay V. Vazirani. RSA bits are $732 + \sqrt{}$ secure. In D. Chaum, editor, *Proc. CRYPTO 83*, pages 369{375, New York, 1984. Plenum Press.

[206] J. von Neumann. Various techniques for use in connection with random digits. In *von Neumann's Collected Works*, volume 5, pages 768{770. Pergamon, 1963.

[207] M. Wiener. E±cient des key search. Practical Cryptography for Data Internetworks, 1996. http: //www3.sympatico.ca/wienerfamily/Michael/MichaelPapers/dessearch.pdf.

[208] A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 80{91, Chicago, 1982. IEEE.

[209] A.C. Yao. Protocols for secure computations. In *Proc. 23rd IEEE Symp. on Foundations of Comp. Science*, pages 160{164, Chicago, 1982. IEEE.

[210] A.C. Yao. How to generate and exchange secrets. In *Proc. 27th IEEE Symp. on Foundations of Comp. Science*, pages 162{167, Toronto, 1986. IEEE.

附录 A：一些概率事实

A.1 生日难题

在第6、8、5章的统计中，要求细致的生日概率边界，如下[12]：
模型是有q个球，把记为1，2，…，q。有N个柜子，N≥q，把这些球随机的扔入柜子，从第1个球开始，随机扔。随机意味着每个球是独立的，产生一个碰撞是两个球被扔入了同一个柜子，感兴趣的是C（N，q），是一个碰撞的概率。

生日现象是当N=365时，询问碰撞概率，在q个人的群中，有两个人有相同的生日，假设生日在每年365天是随机、独立分布的。可以变换为当q采样为$\sqrt{365}$=19.1，是产生碰撞的概率已经非常高了，例如，以q=20的概率产生碰撞的概率是0.328。

生日现象是真的，碰撞概率是C（N，q）产生概率的比例是q^2/N，这是记忆的事实，如下给出更精确的描写，提供这个概率的上界或下界。

命题A.1、假设C（N，q）定义至少产生一个碰撞的概率，当扔q≥1个球到N≥q个桶里，那么：C（N，q）≤q(q-1)/2N，也就是

$$C（N，q）\geq 1-e^{-q(q-1)/2N}，并且对于1\leq q\leq\sqrt{2N}，C（N，q）\geq 0.3\cdot q(q-1)/N$$

在这个证明中，发现如下不平等的使用导致错误。

命题A.2、对于任意实数x∈[0,1]，

$$（1-1/e）\cdot x\leq 1-e^{-x}\leq x$$

命题A.1的证明：假设C_i是第i个球与前面的球产生碰撞的事件，那么$P[C_i]$至多是（i-1）/N，因此当第i个球投入时，有至多i-1个不同的投递可能，第i个球是以相等的机会投入到盒子里，现在：

$$\begin{aligned}C（N，q）&= P[C_1 \vee C_2 \vee \ldots \vee C_q]\\ &\leq P[C_1]+ P[C_2]+\ldots+ P[C_q]\\ &\leq 0/N+1/N+\ldots+(q-1)/N\\ &=q(q-1)/2N\end{aligned}$$

这证明了上界，对于下界，假设D_i是一个事件：在扔进第i个球时没有碰撞产生。如果在扔进第i个球时没有碰撞产生，则一定有第i+1个投递地址产生，因此没有碰撞产生的概率是在第i+1个球时是严格的（N-i）/N，也就是说：$P[D_{i+1}|D_i]=$ （N-i）/N=1-i/N
$P[D_1]=1$，在游戏最后无碰撞的概率是可以通过计算的：

$$\begin{aligned}C（N，q）&=P[D_q]= P[D_q|D_{q-1}]\cdot P[D_{q-1}]\\ &=\ldots\\ &=\prod_{i=1}^{q-1} P[D_i|D_{i-1}]=\prod_{i=1}^{q-1}(1-i/N)\end{aligned}$$

也就是说注意：i/N≤1，因此，使用不等式$1-x\leq e^{-x}$是上述表示的限制，这意味着上述表达式不超过

$$\prod_{i=1}^{q-1} e^{-i/N} = e^{-1/N-2/N-\ldots-(q-1)/N} = e^{-q(q-1)/2N}$$

把这些放在一起可以得到：C（N，q）≥1- $e^{-q(q-1)/2N}$

这在命题A.1中是第二个不等式，为了得到最后一个，需要进行更多的评估。因为$q<\sqrt{2N}$，

知道$q(q-1)/2N\leq 1$，可以使用不等式$1-e^{-x}\geq(1-e^{-1})x$去得到
C（N，q）≥（1-1/e）·q(q-1)/2N，这里常数的计算完成证明。

附录 B：一些复杂性理论背景

到目前为止，甚至不知道怎样证明在要求解决NP-难题的时间段里去证明一个低的界限。因此，一个密码学理论进展标识存在一个攻击者的计算边界，必须求助于假设难题的存在。事实上，在密码学里一个目前的研究话题（在这点上，最近几年获得一些进展），找到存在这种安全体制的最小假设。

假设应该能够快速产生难题的例子，另外的人一定比产生这个难题的人更难于解这个问题。例如，对于消息的发送者应该产生一个密文，对于任何攻击者是难于破解的。（自然的，在这个例子里，对于为了破解密文进行蓄意的消息接收是容易的）为了正式的描述假设（存在单向函数和陷门函数）。首先需要回忆一些复杂度理论的定义。

B.1 复杂度分类和标准定义

B.1.1 复杂度分类 P

一个语言L是P类中的，当且仅当存在一个图灵机M（x），和一个多项式函数Q（y），因此输入串x，

1、 x∈L，如果M承认x（定义为M（x））

2、 在至少Q（|x|）步后M停止。

语言P的分类是经典的，考虑这些语言是容易计算的。使用这个条件在图灵机的多项式时间计算中是有效算法。

B.1.2 复杂度分类 NP

一个语言L是在NP类中的，当且仅当存在一个图灵机M（x, y），多项式p和l，使得在输入串x。

1. $x \in L \Rightarrow \exists y, |y| \le l(|x|)$,使得M（x,y）接收，并且知多p(|x|)步骤后 M 停止
2. $x \notin L \Rightarrow \forall y, |y| \le l(|x|)$, M（x,y）拒绝

注意这是相等的，定义L∈NP，如果有一个非确定的多项式时间的图灵机M，当且仅当x∈L，序列 y 相应上述的非确定多项式图灵机。

B.1.3 复杂簇 BPP

一个语言L是在BPP中的，当且仅当存在图灵机M（x,y），并且多项式p和l，输入串x，

1. $x \in L \Rightarrow Pr_{|y|<l(|x|)}$ [M(x,y)接收] ≥ 2/3
2. $x \notin L \Rightarrow Pr_{|y|<l(|x|)}$ [M(x,y)接收] ≤ 1/3
3. M(x,y)在至多p（|x|）步后结束

作为一个练习，可以试着展示如果常数 2/3 和 1/3 被 1/2+1/p(|x|)和1/2-1/p(|x|)，这里p是一个任意固定的多项式当BPP有相同的剩余。

提示：简单运行机器M（x,y），在许多y上，接收当且仅当运行多数接收。多数量那么依赖于多项式时间p。

知道P⊆NP并且P⊆BPP。如果不知道这些是否被限制，虽然经常推测事例。在BPP中的一个已知语言的例子，但是不知道在P里是所有素整数。不知道是否BPP是NP的子集。

B.2 概率算法

BPP簇是可选的，如果定义概率图零机（概率算法），一个概率多项式时间图灵机。一个概率多项式图灵机M是传统简单步骤的投币，并且在传统的多项式时间至多|x|步。可以定义BPP解释一个语言L在BPP里，如果存在概率多项式时间图灵机M（x），因此，当x∈L，（投币事件）的概率是M（x）的接收是大于2/3，并且当x∉ L，（投币事件）的概率M（x）拒绝的概率是大于2/3。在以前定义的串 y 中，定义相应的随机序列串，在输入为x时机器为M。

到目前为止，考虑概率多项式时间。图灵机是有效算法，（扩展期限在确定的图灵机多项式时间内完成）。同时也称在BPP簇是容易计算的。注意在一个非确定的与概率图灵机的不同点。一个非确定的图灵机在实际使用里通过投币产生概率随机序列（当然假设存在自然随机投币源）。当谈到概率图灵机时，一些概率是有用的。

B.2.1 对于概率图灵机的概念

假设 M 定义一个概率图灵机（PTM）。M（x）定义输入是x的M一个输出概率空间，声明z∈M(x)，指出z是M的输出，当输入是x时。Pr[M(x)=z]是输出为M，z的概率，输入x是M的输出。（这里概率是在完成M的相互投币概念是概率的）。M（x,y）定义M的输出，输入为x，这里投币为y。

B.2.2 概率算法的不同类型

Monte Carlo 算法和Las Vegas算法是两个不同类型的概率算法。在这两个类型的不同是Monte Carlo 算法总是在多项式时间结束的，但是以高概率输出是正确的，反之，Las Vegas算法在一个期望的多项式时间步骤中结束并且输出总是正确的。正式的，定义这些算法如下：
定义B.1、 一个Monte Carlo 算法是概率算法M，对于所有x选择一个概率P，M在P（|x|）中结束。更远的，Pr[M(x)是正确的]>2/3
（这里是M的投币概率）
一个Las Vegas算法是概率算法M，存在一个多项式p，使得对于所有x，

$$E（返回时间x）=\sum_{t=1}^{\infty} t\cdot Pr[M(x),特别获取t步]<P(|x|)$$

更远的，M（x）的输出总是正确的。

所有Las Vegas算法可以代替Monte Carlo 算法，但是不知道是否所有Monte Carlo 算法可以代替Las Vegas算法。一些Monte Carlo 算法的例子是首先要测试的，例如Solovay Strassen[197]或Miller-Rabin[171]测试，相当于多变元多项式和Las Vegas算法的一些例子，计算模素数p的平方根，计算一个模合数n的平方根（如果n的因子分解已知），并且基于椭园曲线的素性测试。（[4]或[100]）

B.2.3 非统一多项式时间

一个重要的概念是多项式时间算法，可以代表不同的大小的输入和也许是甚至多项式的输入大小。（一个图灵机的定义是不变的）

定义B.2、一个非统一的算法：A是一个{M_i}的无限序列，（每个输入大小i），因此输入是x，运算$M_{|x|}$ (x)，也就是说A（x）接收当且仅当$M_{|x|}$ (x)接收。也就是说A是一个多项式时间非统一算法，如果有多项式P和Q，使得$M_{|x|}$ (x)在P（|x|）中结束。并且M_i 描述的大小是（根据所有算法编码相同的一些标准），在Q（|i|）中是有限制的。

定义B.3、可以说一个语言L是在P/poly里的。明显的，P⊂P/poly是清楚的，Adleman是BPP⊂P/poly的。将使用条件“有效非统一算法”指一个非统一多项式时间并且限制“有效非统一可计算”指出P/poly簇。

B.3 攻击者

用两个方式，攻击者是可以仿真计算阶。首先统一的敌人是（将通常落下一致的，当谈及到这点时）。一个相同的攻击者是任何多项式时间多项式时间的概率算法。因此，攻击者可以使用不同的算法对于不同大小的输入。明显的，非统一的攻击者是更强的，比同一的一个。因此证明一些事情是安全的，甚至在非统一的攻击者是一个更好的结果，在统一的条件下，比仅仅证明是安全的。

B.3.1 假定存在

存在一个统一的攻击者时，最弱的假设是必须从密码学的角度定制的，P≠NP，也就是，∃L∈NP，因此，L∉P，不幸的是，没有足够的假设使得攻击者可以使用多项式时间完成攻击。因此，进一步假设BPP≠NP，是否足够？好的，事实上可以对一个攻击者在大多数时间对体制进行破解。但是对于攻击者不能立刻破解这种体制。假设BPP≠NP，仅仅意味着在L∈NP时，存在一种语言，每个相同的攻击者以高概率的机会得出错误的决定，当确定是否x∈L，使得关于无限多的输入x。这是错误的结论，尽管在数字上是无限的，但是很少发生的（例如对每个输入只进行一此）

因此需要一个更强的假设，保证满足下列条件。存在一个语言L∈NP使得对每个足够大的输入尺寸n，每个普通的攻击者以高概率在对多长度为n的输入x上做出错误的决定，是否x∈L。这些错误的决定尽管在数量上是无限的。然而，希望尽可能对入不同长度n的输入普通攻击者无法做出正确判断。假设将保证可以存在单向函数。假设可以保证对于不相同的攻击者存在不相同的单向函数，对于单向函数的定义、特征和可能存在的例子详见第2章。

B.4 从概率论得到的一些不等式

命题8.4 [Makov 不等式]如果Z是随机变量，仅仅对于非负值，那么对任意值a>0，Pr[Z≥a]≤E[Z]/a。

命题8.5［大数弱定理］假设Z_1，…，Z_n 是独立随机分布的0、1值（Bernoulli 随机值）平均为μ。那么Pr[$|\sum_{i=1}^{n} Z_i$ /n-μ|<ε]>1-δ，假若n>1/4$\varepsilon^2\delta$。

附录 C： 一些数论背景

许多密码原语的重要结构是基于数论难题的，看起来是计算不可能的。这些难题最知名是合数分解问题，为了对这些难题进行工作，需要发展一些数论理论和数论定理算法。因此，在这个学科提供一个小型过程。这里的资料是随后诶使用的，在随后讨论单向函数和陷门函数。

在数论领域文献中有许多信息源，例如假设Angluin's注记[7]和Cormen, Leiserson and Rivest [63]的第33章。

C.1 群，基础

一个群G是包含运算*的集合，定义一些运算*，a*b是*作用在a，b上的结果，一个群有如下的特征：

（1） 如果a，b∈G，则a*b∈G

（2） 满足结合率：(a*b)*c = a*(b*c)

（3） 有一个单位元I，使得I*a = a* I = a，对于任意a∈G

（4） 每个a∈G有一个逆元，定义a^{-1}，使得a* a^{-1} = a^{-1} *a = I

将遇到很多这种结构，首先回忆Z,N和R，现在定义

— 整数加 ： I=0；a^{-1} = -a

— 实数乘： I=1；a^{-1} = 1/a

— 自然数加不是一个群

— 整数乘不是一个群

符号：a^m 是自身的m次乘积，也就是，符号是指数集的。那么a^{-m} 是指（a^{-1}）^{m} ，注意按照规定工作。这些群都是无限的，通常的兴趣在于发现有限群。

Def：称|G|是G的阶：

公理C.1、假设m=|G|，那么a^m = 1对于每个a∈G

这个公理将随后使用。

A≡b （mod n） 意味着如果用n除a，那么剩余是b。（在C中，可以表示为 a%n = b）

一个重要的集是模n整数集，也就是Z_n = { 0,1,…,n-1}，在模n加下成群。另一个重要的群是Z_n^* = {m:1≤m≤n，并且gcd(m,n)=1}，小于n且与n互素的集，可以看到这是一个模n乘法的一个群。假设Φ（n）= | Z_n^* |，这是一个欧拉函数。

一个子集S⊆G称为一个子群，如果在自己的权限范围内对于如下的定义成群：特别的，如果x,y∈S，因此xy，x^{-1}和1∈S。

命题C.2、假设S是G的子群，|S|可以整除|G|

C.2 算数：+，*，GCD

算法的复杂度，运算一个数a，测量 a 的大小，|a|=log a ，做基本的运算需要多少时间，在数字中的k比特数字：

— 加是线性时间，例如，两个k比特长度的数字相加，需要时间为O(k)

— A和b的数乘是O(|a|*|b|)的运算，名义上是O(K^2)算法
— 用b除a得到的余数时间是O((1+|q|)|b|)，这里q是得到的商，因此这是一个二次时间算法

欧几里得算法可以在多项式时间内用于计算GCDs，工作的方式是重复使用同一个gcd(a,b)=gcd(b,a mod b)，例如，见[7]的第10页。

什么是运算时间？每个划分阶段是采用二次时间和有k阶段，可以说这是一个O（k^3）时间。但是见难题 33-2，[63]的850页。这样可以如下产生：

定理C.3、欧几里得算法时间可以仅仅用O（|a|·|b|）比特运算直接计算gcd(a,b)，也就是说，对k比特数字，得到O（k^2）算法。

命题C.4、gcd(a,b)=1 当且仅当存在整数u,v，使得：1=au+bv
扩展欧几里得算法由a,b给定，并且返回不仅是 d= gcd(a,b)，但是整数u,v，使得：d = au+bv，这与欧几里得算法非常类似。在每一步试着跟踪额外的信息。

C.3 模运算组

C.3.1 简单运算

现在进行模运算，关注的一点是：
— 加法如下：给定a,b,n，当a,b∈Z_n，计算 a + b mod n，这也是线性时间。例如两个b比特数可以在O（k）时间内进行，为什么？不能在N上进行运算。如果这样做，仅仅减去n。这也是线性时间。
— 使用a mod n，意味着用 n 除 a 并且得到剩余，因此，采用二次时间
— A和b的模n数乘：首先相乘，采用O（|a|·|b|）比特运算，然后用n去除得到剩余，随后也可以看到剩余。因此所有的事是二次方。

Z_n对模n加成群，这意味着可以模 n 加两个数，返回一个集合的元素，因此减少是可能的。在加的情况下，可以向考虑的方向进行。

现在考虑Z_n^*，这里对数乘有兴趣。希望看到这是一个群，在某种情况下，可以进行数乘或是除法，已经知道是怎样数乘的。

定理C.5、有一个O（k^2）算法，给定a,n，a∈Z_n^*，输出b∈Z_n^*，满足ab≡1 (mod n)，这里k=|n|。见7的12页，算法使用扩展欧几里得算法，知道 1= gcd(a,n)。因此可以发现整数u,v，使得：1 = au+bv，意味着 gcd(u,n)= 1。

在定理中可以发现b是唯一的，因此，b在定理中定义为 a^{-1}。

C.3.2 主群 Z_n 和 Z_n^* 。

定理C.6、对任意正整数n，Z_n^* 在模n数乘运算下构成群。
这意味着a,b ∈Z_n^*，意味着ab mod n 是在Z_n^* 里。有些事情在不是特别困难的情况下可以确定，同样意味着能够数乘和除法，有一个单位元和一个削去率。
符号：典型的停止mod n 的写法，每个是真实的，这显然是明白的。
考虑Z_n^* 的方式是实数，可以象以前一样操作，随后是C.1的推论。

定理C.7、对于任意a ∈Z_n^*，这是原因 $a^{\Phi(n)} = 1$

推论C.8、(费马小定理) 如果p是素数，那么 $a^{p-1} \equiv 1 \pmod p$，对于任意 a ∈{0,1,…,p-1}

为什么？因为 Φ(n)= p-1

C.3.3 求幂

这是公钥密码的基本运算，运算仅给定a,n,m，这里$a \in Z_n$，m是一个整数，计算$a^m \bmod n$。例 C.9、计算 $2^{21} \bmod 22$，单纯的办法：使用21次幂乘。这种运算的难题是什么？是一个指数时间运算，因为需要多项式时间poly(K)，这里K=|n|。使用反复平方进行运算：

$$2^1 \equiv 2$$
$$2^2 \equiv 4$$
$$2^4 \equiv 16$$
$$2^8 \equiv 14$$
$$2^{16} \equiv 20$$

现在计算$2^{21} \equiv 2^{16+4+1} \equiv 20*16*2 \equiv 10 \bmod 21$
这是一个求幂的反复平方算法，消耗立方指数时间。

定理C.10、有一个算法，给定a,n,m，$a,m \in Z_n$ ，在时间O（k^3）内输出$a^m \bmod n$，这里K=|n|。更细致的是，算法至多使用2k模kbits数的数乘。

算法可以看做是k进制展开的表出方式，在上述的例子中，21=10101，需要做的只是得到各次幂的数值，并且进行乘积运算。

4	3	2	1	0
α^{16}	α^{8}	α^{4}	α^{2}	α^{1}
1	0	1	0	1

求幂（a,m,n）
假设$b_{k-1}\ldots b_1 b_0$ 是m的二进制比特表示
假设b_0 =a
假设y=1
For i=0,…,k-1，do
　　If b_i=1 let $y=y*x_i \bmod n$
　　Let $x_{i+1} = x_i^2 \bmod n$
输出 y

C.4 中国剩余

假设$m=m_1 m_2$ ，假设$y \in Z_m$,考虑数组：
$a_1=y \bmod m_1 \in Z_{m1}$

$$a_2=y \bmod m_2 \in Z_{m2}$$

中国剩余定理考虑是是重组a_1 a_2得到y。也就是说有唯一的办法在这个条件下的运算。

例C.11 m=6=3*2

$$0 \rightarrow (0,0)$$
$$1 \rightarrow (1,1)$$

2→(2,0)
3→(0,1)
4→(1,0)
5→(2,1)

例C.11 m=4=2*2

0→(0,0)
1→(1,1)
2→(0,0)
3→(1,1)

这里的区别是在第一个例子里，结合结果是唯一的，在第二个中不是唯一的。当m_1和m_2只是相应的素数，有这个定理简单的外形。

定理C.13[中国剩余定理]、假设$m_1m_2\ldots m_k$是相应素整数对，$\gcd(m_i,m_j)=1$，对于$1\le i<j\le K$。假设$a_i\in Z_{mi}$,使得$y\equiv a_i \bmod m_i$，对于i=1,…,k，更进一步，有一个$O(k^2)$算法，给定a_1,a_2,m_1,m_2,计算y，这里$k=\max(|m_1|,|m_2|)$

证明：对于每个i，假设$n_i=(m/m_i)\in Z_m$，通过假设，$\gcd(m_i,m_j)=1$，因此，$\exists b_i\in Z_{mi}$，因此，$n_ib_i\equiv 1 \bmod m_i$，假设$c_i= n_ib_i$，那么$c_i = \begin{cases} 1 \bmod m_i \\ 0 \bmod m_j \text{对于} i \neq j \end{cases}$

令$y=\sum_{i=1}^{k} c_i a_i \bmod m$，那么，$y\equiv a_i \bmod m_i$ 对于每个i。

更进一步，如果$y'\equiv \bmod m$，唯一性得到确认。

推论C.14、在上述证明中的整数c_i 是指中国剩余定理的系数。注意证明产生了一个多项式时间算法找到y，因为元素$b_i\in Z_{mi}$，可以决定使用欧几里德算法和其他的运算，包括除法、乘法和加法。

一个更普通的方法是中国剩余定理如下的结论。

定理C.15、设$a_i\in Z_{mi}$,对于$1\le i\le k$，在一个必要和足够的条件下，一致的系统$x\equiv a_i \bmod (m_i)$对于$1\le i\le k$是有解的，$\gcd(m_i,m_j)|(a_i-a_j)$,对于$1\le i<j\le k$，如果一个解是存在的，那么它是模$\text{lcm}(m_1m_2\ldots m_k)$唯一的。

二次剩余$a\equiv x^2 \bmod n$，当$a\in Z_n$。

首先观察对于p一个素数加，并且$a\in Z_{p^*}$，因此，$a\equiv x^2 \bmod p$，因为x和-x有两个模p不同的解，如果$y^2\equiv a\equiv x^2 \bmod p$，那么，$p|[(x-y)(x+y)]\Rightarrow p|(x-y)$或$p|(x+y)$，因此，$y\equiv \pm x \bmod p$，（注意，$x\overline{\equiv} y \bmod p$，否则，$2x\equiv 0 \bmod p \Rightarrow p|x$，因为p是奇数。因此，对于$a\in Z_p^*$，$a\equiv x^2 \bmod p$是0解或2解。

下一个考虑是$a\equiv x^2 \bmod p_1p_2$，这里p_1，p_2 是不同的奇素数，当且仅当$a\equiv x^2 \bmod p_1$和$a\equiv x^2 \bmod p_2$有解时，这个二次同余方程有一个解。对于每个对（x_1,x_2）使得$a\equiv x_1^2 \bmod p_1$和$a\equiv x_2^2 \bmod p_2$，可以用中国剩余定理产生一个解y，对于$a\equiv x^2 \bmod p_1p_2$，有$y\equiv x_1 \bmod p_1$，$y\equiv x_2 \bmod p_2$，因此，$a\equiv x^2 \bmod p_1p_2$，有0解或有4个解。

更普通的，如果$p_1p_2\ldots p_k$，是不同的奇素数，等式$a\equiv x^2 \bmod p_1^{a1}\ldots p_k^{ak}$有0或$2^k$个解，另外，这些解可以用中国剩余定理使用，$a\equiv x^2 \bmod p_1^{a1}$。更进一步，对一个素数，一个素数p是一个解y，$a\equiv x^2 \bmod p^k$，可以首先发现一个解$x_0$，使用推论2.39的一个算法作为平方根的一个近似值进行运算，近似值以迭代次数$x_j\equiv (x_{j-1}+a/x_{j-1})/2$，对于$j\ge 1$。

命题C.16、对于每个整数j≥0，$a\equiv x^2 \bmod p_1^{a1}$。证明：对于j=0，命题显然正确。假设对于j>0，$a\equiv x^2 \bmod p_1^{a1}$。

那么：$x_j - a\, x_j^{-1} \equiv 0 \bmod p^{2j} \Rightarrow (x_j - a\, x_j^{-1})^2 \equiv 0 \bmod p^{2j+1}$

扩展和加4a对于两边给定 $x_j^2 + 2a + a^2 x_j^{-2} \equiv 4a \bmod p^{2j+1}$ 并且因此，$((x_{j-1} + a/x_{j-1})/2)^2 \equiv a \bmod p^{2j+1}$ 或$x_{j+1}^2 \equiv a \bmod p^{2j+1}$

因此，归纳有效。

从命题可以得到在「logk」次迭代后，可以得到$a\equiv x^2 \bmod p^k$ 的解。

C.5 素元和 Z_p^*

C.5.1 定义

假设G是一个群，令a∈G，观察a的阶，也就是：$a^0, a^1, \ldots,$.假设：

$$<a>=\{a^i : i\geq 0\}$$

假设m=|G|，是G的阶。已知a^0是单位元，称为1，并且$a^m=1$，序列在m步骤后重复，$a^{m+1}=a$，但是也可以在之前重复，来看一个例子。

例C.17: $Z_9^* = \{1, 2, 4, 5, 7, 8\}$，Φ（9）=6。那么：

<1>={1}

<2>={1,2,4,8,7,5}

<4>={1,4,7}

<5>={1,5,7,8,4,2,}

这里给出了所有元，有些成为了小圈。

公理C.18、<a>是G的一个子群，称为由a产生的子群。

假设t=|<a>|，那么可以知道t整除m，可以知道事实上，$<a>=\{a^0, a^1, \ldots, a^{t-1}\}$,这些都是不同的元，其余的元落在其子集。

定义C.19、一个元a的阶是最小正整数t，使得$a^t=1$，也就是说，order（a）=|<a>|。

在指数域可以进行模t运算，也就是说，$a^i=a^{i \bmod t}$ ，这是因为$a^t=a^0=1$。

a^i 的逆是什么？直观上看应该是a^{-i}，是这样的吗？好，考虑这是$(a^{-1})^i$是正确的，另一方面，这个子群的数量是多少？对一些j而言，一定是 a^j，那么j=t-i。特别的，如果给出指数表出，逆元是很容易实现的。

类似的，对于实数乘法可以转化为指数进行加法运算，例如：$a^{i+j} = a^i a^j$

定义C.20、一个元g∈G是一个素元，或一个生成元，如果g的阶产生G，也就是，<g>=G，是所有的群G，是循环群，如果有一个素元。

注记：对于任何g∈G是一个唯一的元，i∈{0,1,…,m-1}，也就是说，$g^i=y$，这里m=|G|。

符号：唯一i可以定义为 $\log_g(y)$，并且称为基于g的离散对数x。

考虑如下的难题，给定g,y，指出$\log_g(y)$，怎样处理这些？一个办法是所有i=0,…,m-1，对于每个i, 计算$g^i=y$，但是这个过程是个指数时间过程。对于许多群，计算处理离散对数。也就是，没有已知多项式时间算法，特别的，对于Z_p^* 为真,p是素数。

C.5.2 群 Z_p^*

命题C.21 、[第9节，7]群Z_p^* 是循环的。

推论C.22、群Z_p^*的阶是多少？是Φ(p)，满足与P互素小于p的数的个数。既然p是素数，则Φ(p)=p-1，注意阶不是互素。特别的，是偶数（对于p≥3）

一个单向函数，假设p是素数，假设$g\in Z_p^*$是一个生成元，那么函数$f_{pg}:Z_p\to Z_p^*$,定义：$x\to g^x$被猜测是单向的，只要是p上的技术条件。也就是说，没有足够的办法求逆，对于足够大的参数值。见第2章。

同型特征，函数f_{pg}是$g^{a+b}=g^a g^b$。

现在怎样使用这个函数？首先建立一个函数，这要求两个条件，首先可以找到素数，其次可以找到生成元。

C.5.3 查询生成元

从第二步开始，必须深入研究Z_p^*并且发现一个生成元，怎样办？甚至在有一个候选元的情况下，怎样去测试其素性。条件是<g>=G，可以使用|G|步去检测。

事实上，找一个生成元p在通常意义下是一个难题。实际上，g是一个生成元，在给定p,g条件下是一个难题。但是可以发掘的条件是型如p=2q+1的素数形式，注意群Z_p^*的阶是p-1=2q。

命题C.23、p=2q+1是一个素数，当q是一个素数。那么$g\in Z_p^*$是Z_p^*的一个生成元，如果$g^q\neq 1$，并且$g^2\neq 1$。

换言之，在给定q时，很容易测试p是否是一个素元。现在给定p=2q+1，怎样确定一个生成元。

命题C.24、如果g是一个生成元，并且i不能被2或q整除，那么g^i是一个生成元。

证明：$g^{iq}=g^{q+(i-1)q}=g^q(g^{2q})^{(i-1)/2}=g^q\cdot 1=g^q$不等于1，是因为g不是一个生成元。类似的，假设i=r+jq并且有$g^{2i}=g^{2r}\cdot g^{2jq}=g^{2r}$，但是2r<2q，因此r<q，所以$g^{2r}\neq 1$。

因此，在Z_p^*中有多少生成元？形如g^i的并且i不能被2或q整除，并且i=1,…,2q，也就是说所有Z_{2p}^*，所以Φ（2q）=q-1。

发现生成元的具体步骤是：随机从Z_p^*中选择一个元g，使得$g^q\neq 1$，$g^2\neq 1$。如果不满足条件，重现选择，直到找到这种元。失败的概率是多少？一次失败的概率是（q+1）/2q，通过1次测试，包括$((q+1)/2q)^1$,因为q值很大，因此可以简要描述为2^{-1}。

C.6 二次剩余

一个元素$x\in Z_p^*$是一个平方根，或者是二次剩余，如果存在一个平方根，也就是有一个$y\in Z_N^*$，使得$y^2\equiv x \bmod N$，如果不是，则是一个平方或是一个非平方剩余。注意一个数也许有许多平方根。

容易计算一个模素数的平方根，在已知一个合数的素因数分解时，通过中国剩余定理也是容易计算的。（在两个条件下，可以计算出所有的根），但是在未知分解的条件下是难于计算的。事实上，计算平方根的难度与因数分解难度是等价的。所以决定一个合数的平方剩余是困难的。

命题C.25、如果N是两个素数的乘积，每个平方$w\in Z_N^*$有四个平方根, x,-x,y,-y。对于一些$x,y\in Z_N^*$。如果有两个平方根x,y，因此，$x\neq\pm y$，那么，可以容易的分解N。

第一个事实是基于数论的，第二个可以这样观察。也就是说x>y是平方根，考虑$x^2-y^2=(x+y)(x-y)\equiv 0 \bmod N$，假设a= x+y，并且b= x-ymodN,因此ab被N整除。因此p整除ab。既然p是素数，也就是说，p既整除a，p也整除b。因此，1≤a, b≤N，意味着gcd(a,N)=p，

gcd(b,N)=p。可以计算出gcd，并且检查是否可以得到N的一个除数。

C.7 贾可比符号

首先定义勒让德符号，表示$a\in Z_p^*$ 的平方符号，这里 p 是一个素数。特别的，对于素数p并且$a\in Z_p$ 是

$$J_p(a)=\begin{cases} 1 \text{如果} a \text{是一个} Z_p^* \text{的一个平方根} \\ 0 \text{如果} a=0 \\ -1 \text{如果} a \text{不是一个} Z_p^* \text{的一个平方根} \end{cases}$$

对于一个合数，这个定义是可扩展的，如下给定Jacobi符号，假设$n=\prod_{i=1}^{k} p_i^{ai}$ 是n的素因数，

对于 $a\in Z_n$，定义：$J_n(a)=\prod_{i=1}^{k} J_{pj}(a)^{ai}$

然而，不能产生贾可比符号，不能产生勒让德符号，当n是合数的时候，指$a\in Z_n^*$ ，例如$J_9(2)=J_3(2)J_3(2)=1$，尽管等式$x^2\equiv 2 \bmod 9$没有解。

贾可比符号的满足唯一的勒让德符号条件。从[150]可以看到这些结论：

1、 如果$a\equiv b \bmod n$ 那么 $J_n(a)=J_n(b)$
2、 $J_n(1)=1$
3、 $J_n(-1)=(-1)^{(n-1)/2}$
4、 $J_n(ab)=J_n(a)J_n(b)$
5、 $J_n(2)=(-1)^{(n2-1)/8}$
6、 如果m和n是相应的素整数，那么$J_n(m)=(-1)^{(n-1)/2(m-1)/2}J_m(n)$

使用这些恒等式，贾可比符号$J_m(a)$，这里$a\in Z_p$，可以在不知道n的因子分解的条件下，多项式时间内完成计算。回忆在多项式时间计算勒让德符号使用的欧拉定理，也就是说，$a\in Z_p^*$，p是素数，有$J_p(a)\equiv a^{(n-1)/2} \bmod n$，对于$a\in Z_n^*$。事实上，对$Z_n^*$最多有一半的元素满足条件。从这个结论可以看出，可以使用蒙特卡罗素性测试。

C.8 RSA

这里有一个合数N，是两个长度大致相同的素数的乘积，N=pq。假设k=|N|，大约是1024位。通常条件下相信这样一个数是难于分解的。

回想Φ（N）=| Z_n^*|是欧拉PHI函数。注意Φ（N）=（p-1）(q-1)，（是与N相关的素数，一个数必须同时可以被p或q整除，除去这个产生的倍数，注意这里p≠q）

现在假设gcd(e, Φ（N）)=1，也就是说，$e\in Z_{\Phi(N)}^*$。RSA函数定义为：

$$F: Z_n^* \rightarrow Z_n^*$$
$$X \rightarrow x^e \bmod N$$

这样可以知道$Z_{\Phi(N)}^*$ 是一个群。因此e有一个逆$d\in Z_{\Phi(N)}^*$ 。既然d是e的一个逆，那么满足：

$$ed \equiv 1 \bmod (\Phi(N))$$

现在假设$x\in Z_{\Phi(N)}^*$是任意的，观察如下的计算：

$$(x^e)^d \bmod N = x^{ed \bmod \Phi(N)} \bmod N = x \bmod N = x$$

换言之，$y \rightarrow y^d$ 是函数f的一个逆，也就是，$f^{-1}(y)=y^d \bmod N$。

可以发现d？容易：在平方时间内可以计算逆，如同已说的，使用扩展GCD算法。但是注意一个关键点，在模Φ（N）的条件下工作。因此，在已知Φ（N）时反复计算p,q。

看起来是这个条件，仅仅给定N和e，难以发现d。当然同意难于发现p,q。但是更多的，难以找到d。产生推测RSA是一个单向陷门置换，也就是给定N，e，定义f，$x \rightarrow f(x)$是容易的。$y \rightarrow f^{-1}(y)$是困难的，但是f^{-1} 在给定p,q（或 d）。注意这个陷门是没有一个离散算法的特征。F的计算称为加密，计算f^{-1} 称为解密计算。

对于加密和解密是求幂，一个优先在k次运算是立方的。然而，通常选择的e是小的，因此加密速度是快的。对于装置，RSA比DES慢1000倍。对于软件，对于小的加密指数大概慢100倍。

正式的，RSA定义了一个陷门置换簇。这个簇是用一个安全参数k进行检索的，是模数的大小。RSA发生器是一个算法G，选择两个不同的1^k ，随机（k/2）比特素数p,q，乘法产生N=pq，计算e,d。输出N，e，描述f和N，d，如同描述f^{-1} 。见第2章。
RSA提供了公开密码学的基础。

C.9 素性测试

对于许多加密目的，需要找到素数。没有多项式时间算法，测试给定n的原始测试。所使用的是概率多项式时间（PPT）算法。可以看到确定一个整树是否是素数是NP难题，那么将讨论 Solovay-Strassen和 Miller-Rabin 的概率素性测试，可以找到有效的合数证明。

最后，将给定一个 Solovay-Strassen 和 Miller-Rabin 的素性测试，使用椭园曲线找到有效的素性测试。

C.9.1 素性∈NP 难题

首先考虑测试整数的素性是NP难度问题，两者都是无效的，因为要求因子是一个子集。然而，每个算法可以用于解决整数的素性测试问题。事实上，第二个算法进一步说明，证明决定素性为 UP ∩ coUP 。这里UP定义了语言簇L，使用非确定的图灵机进行多项式时间计算，对每个x∈L 有一个唯一的接收途径。

定义C.26、假设素数={p:p是一个素整数}

C.9.2 Pratt's 的素性测试

Pratt's 的素性测试是基于如下的结论。

命题C.27、对一个整数n>1，如下的表述是等价的。

1、$|Z_n^*|=n-1$

2、整数n是素数

3、有一个元素$g \in Z_n^*$，使得$g^{n-1} \equiv 1 \bmod n$，对于n-1的每个素除数q，$g^{(n-1)/q} \not\equiv 1 \bmod n$

Pratt's 的算法对于如下的一个素数输入p和输出一个证明p的确是一个素数。

1、 寻找一个元素$g \in Z_n^*$，阶是p-1

2、 决定一个阶为p-1的素数分解 $\prod_{i=1}^{k} q_i^{ai}$

3、 有g是Z_p^* 的生成元，证明p是素数。说明检查$g^{p-1} \equiv 1 \bmod p$，对于每个素数q_i 检测

$$G^{(p-1)/qi} \equiv 1 \bmod p$$

4、 递归显示对于$1 \le i \le k$，q_i是素数

注意，如果p是素数，那么Z_p^* 是φ（p-1）= Ω(p/loglogp)生成元（见[172]），因此，为了找到一个从Z_p^*中随机选择的生成元g，希望不得不选择g 的O（loglogp）元。如果找到一个Z_p^* 的生成元，并且如果可以分解p-1，递归证明p-1的素因数分解是真正的素数，这样得到了一个关于p的素性证明。不幸的，对于产生 P 的未知怎样分解因数p-1。Pratt's 的素性测试算法得到证明。然而，PRIMES ∈NP，因为在第1步找到生成元，第2步要求推测因式分解。而且，在多项式时间的分解可以证明是正确的，每个q_i的素性可以用算法证明是递归的。同时注意，Pratt在[161]中使用一个简单的推倒证明，这种素数的数量是O（logp），使用至多p的模数乘法。

C.9.3 概率素性测试

C.9.4 Solovay-Strassen 素性测试

可以进行一个蒙特卡罗测试，这个算法随后将进行描述，是由Solovay-Strassen发明的（见[197]）。Solovay-Strassen素性测试由如下的算法进行测试，在输入奇正数为n时，一个整数为k，标识相应的可靠性：

1、 测试对于整数b,e>1，如果$n=b^e$;如果是这样，输出复合数并且停止

2、 随机选择$a_1,a_2,\ldots,a_k \in \{1, 2, \ldots, n-1\}$

3、 如果gcd（a_i,n）≠1，对于任意的$1 \le i \le k$，输出复合数并且停止

4、 计算$a_i = a_i^{(n-1)/2} \bmod n$，并且$\beta_i = J_n(a_i)$

既然计算包括了Solovay-Strassen素性测试在多项式时间是可计算的，（这个描述对于第1步为真），显然算法一logn和k是在多项式时间运行。如下的结论保证如果n是合数，在算法的第5步有$Pr[\alpha_i=\beta_i \bmod n$ 对于$1 \le i \le k] \le (1/2)^k$。

推论C.28、假设n是一个奇正整数，不是一个完全平方，假设：

$G=\{a \in Z_n^*$，因此，$J_n(a) \equiv a_i^{(n-1)/2} \bmod n\}$。那么$|G| \le |Z_n^*|/2$。

证明：既然Z_n^* 是G的一个子群，可以看出$G \ne Z_n^*$

既然n是合数，并且不是一个完全平方。有一个非平凡的因子，$n=rp^a$，这里p是一个素数，a是一个奇数，gcd(r,p)=1。

假设$a^{(n-1)/2} \equiv J_n(a) \bmod n$，对于所有的$a \in Z_n^*$，那么，

$$A^{(n-1)/2} \equiv \pm 1 \bmod n \text{ 对于所有 } a \in Z_n^* \quad (C.1)$$

首先看到，事实上，$a^{(n-1)/2} \equiv 1 \bmod n$，对于所有这样的a。如果不是，有一个$a \in Z_n^*$，使得$a^{(n-1)/2} \equiv -1 \bmod n$。通过中国剩余定理，有一个唯一的元$b \in Z_n$，因此，$b \equiv 1 \bmod r$并且$b \equiv a \bmod p^a$ 。那么$b \in Z_n^*$，并且$b^{(n-1)/2} \equiv 1 \bmod r$并且$b^{(n-1)/2} \equiv -1 \bmod p^a$ 同等式C.1相矛盾。因此，$J_n(a)=1$对所有$a \in Z_n^*$ 。

然而，通过中国剩余定理，有一个唯一的元$a \in Z_{rp}$，因此，$a \equiv 1 \bmod r$，并且$a \equiv z \bmod p$。这里z是（p-1）/2模p的二次剩余，那么$a \in Z_n^*$ 并且因此，$J_n(a)=J_r(1)J_p(z)=-1$，因此a是一个奇数。这是一个矛盾。

注意，如果达到Solovay-Strassen的第5步，那么n不是一个完全平方数。每个 $a_i \in Z_n^*$，因为算法在第1步进行了完全阶检查，在第3步计算$\gcd(a_i,n)$。因此C.28的推论满足任意$1\le i\le k$，$\Pr[a_i=\beta_i \bmod n]\le 1/2$。

评注：在命题C.28中的论断在事实上是成立的，甚至n是一个完全平方根。更普通层面的证明是与C.28中的论断非常相似的。详细细节见[7]。

最后，从命题C.28可以看到Solovay-Strassen算法以高概率运算正确，特别的，

Pr[输出可能为Solovay-Strassen的素数|n是合数]≤（1/2）^{k}

Pr[输出可能为Solovay-Strassen的素数|n是素数]=1

C.9.5 Miller-Rabin 素性测试

费马小定理是描述对于一个素数p，并且$a\in Z_n^*$，$a^{(p-1)} \equiv 1 \bmod p$。这对于用一个概率的办法测试n是否是一个素数，例如计算$a^{(n-1)} \equiv 1 \bmod n$。不幸的，有一些合整数n（称为基为2的偶素数），$a^{(n-1)} \equiv 1 \bmod n$（称为Carmichael数）。例如$2^{340} \equiv 1 \bmod 341$，并且$341=11\cdot 31$。事实上，在上述整数中使用$a\in Z_n^*$ 代替2，对于n的一些值是没有帮助的，因为有一些合整数n，$a^{(n-1)} \equiv 1 \bmod n$，对于所有$a\in Z_n^*$，561，1105，1729是三个最小的Carmichael数。

Miller-Rabin 素数测试是通过简单建议的难题，如上所说的$a^{(n-1)} \bmod n$对于$a\in Z_n^*$有重复的平方根。在计算每个模指数时，检查是否a的一些根是一个1模n的非平方根，（也就是说，1的一个根是不适合的，对于±1 mod n）。如果是这样，算法可以确定n是合数。这个测试的质量依赖于命题C.30，在[171]中Rabin 已提出。对于一个简单的证明，仅仅产生|{b:W_n(b)保存}≥（n-1）/2}，（然而对于Miller-Rabin算法是有目的的）见33章，[63]的P842-843。

定义C.29、假设n是一个奇正整数，定义如下的条件w_n(b) 是在一个整数b上的。

1、$1\le b<n$ 并且

2、（I）$b^{n-1} \not\equiv 1 \bmod n$ 或

（II）存在一个整数i，使得$2^i|(n-1)$，并且$b^{(n-1)/2i} \not\equiv \pm 1 \bmod n$，但是，

$(b^{(n-1)/2i})^2 \equiv 1 \bmod n$

一个整数b ，对于所有W_n(b)称为一个合数n的复杂度的证词。

评注：Rabin最初定义了一个条件W_n(b)保留是否$1\le b<n$，并且$b^{n-1} \equiv 1 \bmod n$，或者对于一些整数i，使得$2^i|(n-1)$，$1<\gcd(b^{(n-1)/2i}-1, n)<n$。在[171]中，Rabin证明了$W_n$(b)的两个定义是等价的。这个条件事实上是由Miller在[149]中首先想到的，用来给定一个非概率测试，对于扩展Riemann的素性假设证明是正确的。Rabin的结论不要求任何没有证明的假设。

命题C.30、如果n是一个奇正合数，那么|{b：W_n(b)保存}|>3(n-1)/4。

Miller-Rabin算法如下运算，输入一个奇正整数n和一个整数k，指出相关的希望：

1、 随机选择$b_1,b_2,\ldots,b_k\in\{1,2,\ldots,n-1\}$

2、 假设$n-1=2^l m$，这里m是奇数

3、 对于$1\le i\le k$计算$b_i^m \bmod n$，通过反复平方

4、 计算$b_i^{2jm} \bmod n$ 对于j=1,2,…,l。如果对于一些j，$b_i^{2^{(j-1)}m} \not\equiv \pm 1 \bmod n$，但是$b_i^{2jm} \equiv 1 \bmod n$，那么$W_n$(b)保存

5、 如果$b_i^{n-1} \not\equiv 1 \bmod n$，那么$W_n$(b)保存

6、 对于$1\le i\le k$，$W_n(b_i)$保存并且输出合数。如果对于所有$1\le i\le k$，$W_n(b_i)$不保存，输出可能的素数

命题C.30给出Miller-Rabin算法以高概率正确运行，特别的

Pr[输出可能为Solovay-Strassen的素数|n是合数]≤（1/4）^{k}

Pr[输出可能为Solovay-Strassen的素数|n是素数]=1

更进一步，Miller-Rabin在多项式时间以logn和k是所有可以在多项式时间内进行的计算。

C.9.6 素性的多项式时间证明

两个算法中的一个在前一节从挑战中进行了讨论，无论算法怎样进行显示，输入n是素数，以高概率证明是素数，但是不确定是否提供挑战。（换言之，算法是蒙特卡罗素性测试）。然而当每个算法的输出和输入n 是合数时，需要确定n是否真是合数。因此，Solovay-Strassen和Miller-Rabin算法可以观察合成证明。在这一节，将讨论一个素性测试，产生一个期望中的多项式时间短（在确定的多项式时间进行确定）一个素数确实是素数。因此，对于一个通常的完整输入可以运行一个素性测试，与一个合成测试是两个中的一个将最终停止产生一个证明输入是合成的，提供给了一个拉斯维加斯决定一个整数是素数还是合数。

C.9.7 对于一些素数工作的算法

假设发现了一个p-1的除数q，使得q>$\sqrt{p}$，因此下述的算法可以检测素性：

1、 确定一个p-1的除数q，使得q>$\sqrt{p}$

2、 随机选择a∈Z_p^*-{1}
3、 如果1<gcd(a-1,p)<p，那么输出p是合数
4、 检测$a^q \equiv 1 \bmod p$
5、 递归证明p 是素数

这个算法的正确性是从下两个结论中得到

命题C.31、如果q>$\sqrt{p}$ 是一个素数，对于一些a∈Z_p^* gcd(a-1,p)=1并且$a^q \equiv 1 \bmod p$，那么p是一个素数。

证明：假设p不是素数，那么有一个素数d≤$\sqrt{p}$，因此， d|p并且因此，通过假设，a |≡ 1 mod d，$a^q \equiv 1 \bmod d$，因此，在Z_d^*中，ord(a)|q，但是q 是素数并且a的阶不是1。因此，q=ord(a) ≤| Z_d^*|=d-1<$\sqrt{p}$，这与假设q>$\sqrt{p}$ 矛盾。

注意，如果p是一个素数，那么在第4部，条件，$a^q \equiv 1 \bmod p$是以概率（p-1）/(q-1)>1/$\sqrt{p}$ (既然q>$\sqrt{p}$)。然而为了算法成功，必须存在一个p-1的素因数q，q>$\sqrt{p}$，并且必须在每个递归关系下满足条件。也就是说，必须有一个素数序列，$q=q_0,q_1,q_2, \cdots, q_k$, 是足够小的素数使得：$p_i|(q_i-1)$并且$q_i > \sqrt{p_i - 1}$ 对于i=1,…,k，并且这是很不可靠的。

这个障碍可以通过模p上的椭园曲线代替Z_p^* 完成。这允许产生任何模素数，变换阶的椭园群，对于如下节，使用在通常意义下的传统的lenstra的椭园因子算法。

C.9.8 Goldwasser-Kilian 素性测试

Goldwasser-Kilian 素性测试基于椭园曲线特征。算法的观点与C.9.7节中的素性测试是类似的，希望使用椭园曲线$E_{a,b}$（Z_p）代替Z_p^*，通过变化a,b可以找到椭园曲线，展示相应的特征。Goldwasser-Kilian算法进行如下运算，输入一个长度为l的素整数，p≠2,3，输出证明p是素数。

1、 随机选择一个$a,b \in Z_p$，拒绝选择$gdc(4a^3+27b^2;p)\neq 1$

2、 计算$|E_{ab}(Z_p)|$使用Schoof在[179]中介绍的多项式时间算法

3、 使用一个概率伪素性测试（例如Solovay-Strassen 和 Miller-Rabin）决定是否$|E_{ab}(Z_p)|$是形如cq，这里$1<c\leq O(\log^2 p)$并且q是一个概率素数。如果$|E_{ab}(Z_p)|$不是这种形式，重复第1步。

4、 选择在$E_{ab}(Z_p)$上的一个点，M=（x,y），随机选择$x \in Z_p$，采用y进行一个$x^3 + ax + b$的平方根，如果存在。如果$x^3 + ax + b$是一个模p 的二次平方剩余，重复选择过程。

5、 计算q·M

（I） 如果q·M=o，输出（a,b,q,M）。那么如果$q>2^{l(1/\log\log l)}$，[5]）反复证明q是素数，（特别的，从第1步使用 p 代替 q 重复）。否则使用Adleman, Pomerance, and Rumely在[5]中使用的确定性测试，显示q是素数并且停止。

(II) 如果q·M≠o，那么重第4步开始重复。

评注：在第5i步提到的测试是最好的确定算法，确定一个输入是素数还是合数。在输入为n时以（log n）$^{o(\log\log\log n)}$ 步终止。

C.9.9 Goldwasser-Kilian 算法的正确性

注意，首先看到第9.3节发现，在第3步的发现错误的概率可以减少幂指数。Goldwasser-Kilian算法的正确性是从定理C.32得到的，这个结论是与C.31类似的。

定理C.32 假设n>1是一个整数，gcd（n，6） = 1。假设$E_{ab}(Z_p)$是一个模n的椭园曲线，并且假设M≠o是在$E_{ab}(Z_n)$上的点。如果有一个素整数q，使得$q>(n^{1/4}+1)^2$，并且q·M=o，那么n是素数。

证明：假设n是合数，那么有一个n的素除数，使得$p<\sqrt{n}$

假设$ord_E(M)$定义椭园曲线E上的点M的阶，如果q·M=o，那么$q\cdot M_p = o_p$，意味着M=O，是一个矛盾。

定理C.33、使用Goldwasser-Kilian算法的四倍输出序列，可以确认在时间O（$\log^4 p$）确实是一个素数。

证明：假设$p_0=p$，算法四倍输出的序列是形如（a1，b1，p1，M1）；（a2，b2，p2，M2）；这里$gcd(4a_i^3+27b_i^2,p^{i-1})\neq 1$，$M_i\neq 0$，是在$E_{aibi}(Z_{pi-1})$, $p_i>p_{i-1}^{1/2}+1+2p_{i-1}^{1/4}$，并且$p_i\cdot M_i=0$，对于1≤i≤k。这些事实是在O（$\log^3 p$）时间内对任何 i 的值，使用定理C.32，根据$p_i$素数如下。根据定理C.32，对于1≤i≤k，跟随如下素数，p_i 素数$\Rightarrow p_{i-1}$素数，更进一步，在算法的第3步，c≥2 ，因此，$p_i \leq (p_{i-1}+2\sqrt{p_{i-1}})/2$。因此，k的大小是O（logp）给定O（$\log^4 p$）步，最后，$p_k$可以确信在O（logp）时间的素数是可以被证实是小尺寸的。

C.9.10 Goldwasser-Kilian 的预期运算时间

算法是归因于Shoof计算在O（$\log^9 p$）时间$E_{ab}(Z_p)$。那么检查，$|E_{ab}(Z_p)|= cq$，这里，$1<c\leq O$（$\log^2 p$）并且q是素数，在O（$\log^6 p$）步，如果使用Solovay-Strassen或Miller-Rabin在迭代足够的次数后，使用一个小的错误指数算法可以在O（$\log^4 p$）时间运算所有算法。

接下来挑选点M（x,y）要求选择一个希望值，对于 x 最多找到一个模p平方剩余的$x^3 + ax + b$的解。注意模素数p的平方根可以在0（$\log^4 p$）时间内找到解。既然$|E_{ab}(Z_p)|= cq$，这里q是一个素数，$E_{ab}(Z_p)$是同态于$Z_{c1q} + Z_{c2}$，这里$c=c_1c_2$并且，$c_2|c_1$，因此，$E_{ab}(Z_p)$以概率（q-1）/cq ≈1/c，至少有q−1个阶为q的点，在第4步中选择的M的阶是q。因此，在找到阶为q的M点的期望值是$c=O(\log^2 p)$，更进一步，计算qM，要求进行O（logp）次加，利用双倍乘法，并且在$O(\log^3 p)$可以完成。因此，处理第4和第5步需要$O(\log^5 p)$期望时间。

作为先前的评价，递归深度是O（logp），因此，仅仅剩余的考虑是确定怎样使用一个椭园曲线$E_{ab}(Z_p)$，在$|E_{ab}(Z_p)|= cq$中进行选择。这里c= O（$\log^2 p$）并且q是素数，对于Lenstra的考虑$|E_{ab}(Z_p)|$的分布，在期望cp的形式（$p+1-\sqrt{p}$，$p+1+\sqrt{p}$）。注意

$|S| \geq \pi$（$p+1+\sqrt{p}$）/2-π（$p+1-\sqrt{p}$）/2，因为S包括这些形式（$p+1-\sqrt{p}$，$p+1+\sqrt{p}$）是一个素数的两倍。因此，如果假设逼近素数的分布，保持小的距离，那么离散对数必须考虑的是O（$\log^2 p$），然而，仅仅有明显的假设推测小间隔简单的假设。

推论C.34、有一个正常数s使得对于所有的$x\in R_{\geq 2}$，在x和$x+\sqrt{2x}$是Ω（$\sqrt{x}/\log^8 x$）在这种假设条件下，Goldwasser-Kilian算法证明p的素性需要O（$(\log p)^{11+s}$）时间。

C.9.11 几乎所有素数的期望运算时间

尽管在C.9.10节分析中可以看出，推测C.34的没有证明的结论就可以看出，一个归于Heath-Brown关心的理论证明在小的间隔下关心素数的稠密度。长度为l的素数的碎片，Goldwasser-Kilian在多项式时间l运算至少1-O（$2^{-l(l/\log\log l)}$），Heath-Brown有如下的定理：

定理C.35 假设$\#_p[a,b]$表示素数x的数量满足a≤x≤b，

假设 $i(a,b)=\begin{cases}1 if \#p[a,b]\leq (b-a)/2\lfloor \log a\rfloor \\ 0 otherwise\end{cases}$ 存在一个正常数a，因此，

$$\sum_{x\leq a\leq 2x} i(a,a+\sqrt{a}) \leq x^{5/6}\log^a x$$

使用这个定理，Goldwasser and Kilian可以证明在[96]，的算法在期望时间O（l^{12}）在至少1-O（$2^{-l(l/\log\log l)}$）长度为l的这些素数上。在[4]中，Adleman and Huang已给出对Goldwasser-Kilian算法更详细的证明，在多项式时间内消去指数。更进一步，提出了超椭园曲线的素性证明方案，因此，获得一个拉斯维加斯算法的结论最终会获得。

C.10 因子分解算法

在这个注记里，讨论一些椭园曲线的普通特征，最近Lenstra的椭园曲线因子分解算法，使用在Z_n上的椭园曲线整数因子分解。

Pollard的p-1方法

开始介绍一个椭园曲线的因子算法的原语模型，在椭园曲线因字分解中使用类似思想。这个算法，已知为p-1的Pollard方法[166]。假设n是合数，希望可以进行分解，Pollard方法使用思想是可以找到整数e和a，使$a^e \equiv 1 \bmod p$，并且$a^e \not\equiv \bmod p$ ，对于一些素因数p，并且是n的素因数，那么，既然$p|(a^e-1)$并且$q\|(a^e-1)$，$\gcd(a^e-1,n)$是一个n的非平凡因子，可以被p整除，但是不能被q整除。

在输入为n时，算法的过程如下：

1、 选择一个整数e，在限制条件B内所有整数的积。例如，e也许是所有a≤B的整数的积。为了简化这些，假设$e=\prod_{i=1}^{\pi(B)} p_i^{a_i}$ 当$p1,p2,\ldots,p_{\pi(B)}$是小于等于B的素数是最小选择，因此，$\sqrt{\sqrt{\ }}\ p_i^{a_i} \geq \sqrt{n} > \min_{p|n}\{p-1\}$

2、 随机选择一个在2和n-2之间的整数

3、 通过反复平方计算$a^e \bmod n$

4、 使用欧几里德算法，计算$d=\gcd(a^e-1, n)$。如果1<d<n，输出为非平凡因子d，否则，重第二步重复一个a的选择。

当这个算法工作时，假设整数e 是对每个整数B可分解的，p是n的素因子，因此p-1是≤B的素幂乘积。那么，e=m(p-1)对于一些整数m，因此$a^e=(a^{(p-1)})^m \equiv 1^m=1 \bmod p$。因此，$p|\gcd(a^e-1, n)$并且唯一的办法是在第4步获得一个n的非平凡因子可能失败，如果$a^e \equiv 1 \bmod n$。换言之，仅仅对于n的每一个素因数q，a mod q的阶整除e，并且这是不可靠的。

不幸的，对于普通的n是不确定的，有一个n的素因数p，p-1不会被大于B的素阶整除。如果p-1有大的素阶整数，因此将是无效的，因为算法实际上是在O（B）时间内确定的。例如，如果n是两个不同素数p,q的乘积，这里$|p| \approx |q|$是素数，并且p-1和q-1是O（$\sqrt{n}$）平滑的，那么方法相当于要求一个大小为O（$\sqrt{n}$）的边界。

反复的说，难题给出了输入 $n=\prod p_i^{a_i}$，这里，p_i 是n的不同的素因子。被可能的限制，没有一个整数p_i-1是足够平滑的。然而可以改进限制为在椭园曲线上定义的点是可能的，对每个素数p，得到一个大的群集，阶在间隔（$p+1-\sqrt{p}$，$p+1+\sqrt{p}$）是均匀变化的。通过改变群，希望可以发现一个平滑阶。可以看到怎样得到n的因数分解得到这样一个集。

C.11 椭园曲线

定义C.36、一个在域F上的椭园曲线是点（x,y）的集，这里x,y∈F，满足Weierstrass等式，$Y^2=x^3+ax+b$，这里a,b∈F并且，$4a^3+27b^2 \neq 0$，使用一个特殊的输出O称为无限的分数。

使用符号$E_{a,b}(F)$定义集合的这个点。

注记：条件$4a^3+27b^2\neq 0$确保曲线是非退化的，也就是说，当域F是R，椭园曲线上每点的切线是唯一的。

假设P，Q是椭园曲线上$E_{a,b}(F)$的两点，可以定义P的负数，并且根据下述规则定义P+Q的和：

1、 如果P是在无限点O上的点，在无限远处O上定义-P

2、 O+P=P+O=P

3、 假设P，$Q\neq O$，并且假设P=（x_1,y_1）Q=(x_2,y_2)

(I) 如果P=-Q，（也就是x1=x2，并且y1=-y2），那么定义：P+Q=O

(II) 否则，假设

$$\alpha=(y_1-y_2)/(x_1-x_2) \quad \text{如果} P\neq Q \qquad (C.2)$$

$$\alpha=(3x_1^2+a)/\ (y_1+y_2) \quad \text{如果} P=Q \qquad (C.3)$$

也就是说，如果域F=R，α是有P，Q定义的行范围。如果P≠Q，在P点的正切线范围，如果P=Q。那么P+Q=R，这里R=（x_3,y_3）满足：$x_3=\alpha^2-(x_1+x_2)$并且$y_3=\alpha(x_1-x_3)-y_1$

可以看到在椭园曲线上的加法定义满足结合率和定义，并且利用一个加法拉贝儿群在集$E_{a,b}(F)$集成加法恒等群。关注$F=Z_p$ ，这里$p\neq 2,3$是个素数。在这个条件下，在$E_{a,b}(Z_p)$可以在多项式时间内计算，作为等式（C.2）和(C.3)仅仅包括加法、减法和模p除法。注意去计算$z^{-1} \bmod p$ 这里，$z\in Z_p^*$，可以使用扩展欧几里德算法计算一个整数t，使得$tz\equiv 1 \bmod p$，然后可得：$z^{-1}=t \bmod p$。

为了说明加法和求逆，考虑椭园曲线$y^2=x^3-x$

图表关于x轴是均衡的，因此点P是在椭园曲线上的当且仅当-P是在椭园曲线上的。同样，如果线l穿过椭园曲线的两点P、$Q\neq O$，在曲线E（R）不是垂直的，那么有一个更多的点，这条线与椭园曲线相交。为了看到这点，假设P=（x_1,y_1），Q=（x_2,y_2）。并且假设$y=\alpha x+\beta$，是通过P、Q的线性等式。这里，$\alpha=(y_1-y_2)/(x_1-x_2)$，如果P≠Q，或者$\alpha=(3x_1^2+\alpha)/(y_1+y_2)$，如果P=Q，并且$\beta=y_1-ax_1$。注意在这个条件下，P=Q，根据曲线E（R）上的加法规则，在P点做l的切线。一个在椭园曲线l上的点（$x, \alpha x+\beta$），当且仅当$(\alpha x+\beta)^2=x^3+\alpha x+\beta$。因此，对立方曲线$x^3+(\alpha x+\beta)^2x+\alpha x+\beta$有一个插入点，因为（$x_1, \alpha x_1+\beta$）和（$x_2, \alpha x_2+\beta$）分别是椭园曲线的根所以$x_1, x_2$ 是等式的根。因此，等式有第三个根x_3 ，这里$x_1+x_2+x_3=\alpha^2$，这就导出提到的x_3的表出，根据在椭园曲线上E（R）的加法规则。因此，几何学上，在E（R）上的加法运算是从相应的点P、Q引出的线，假设线的第三个概念是−R=（x,y），并且采用R=（x,-y）是总数P+Q。

C.11.1 在 Z_n 上的椭园曲线

Lenstra's 的椭园曲线因子算法在$E_{a,b}(Z_n)$在环Z_n上做验证，这里n是一个奇数、合整数。在非奇异性曲线条件$4a^3+27b^2\neq 0$由（$4a^3+27b^2$，n）=1。否定和加法条件在制定椭园曲线域上是给定。然而，两个点的加法是一个分割（指C.2和C.3在这一节给定的椭园曲线），不是总定义在环Z_n上，对于附加的定义，这个等式的分母必须是素数。因此，$E_{a,b}(Z_n)$不必要成一个群。然而，定义一个办法计算点$P\in E_{a,b}(Z_n)$计算乘法e.P：

1、 假设$a_0+a_1 2+\ldots+a_{m-1}2_{m-1}$是e的二进制扩展，假设j=0,S=0

2、 如果$a_j=1$，那么$S\leftarrow S+2^j p$，如果这个总数没有定义（名义上，在等式（C.2）和（C.3）的划分是失败的）

3、 $j\leftarrow j+1$。如果 j=m，那么输出S作为定义e·P和停止

4、 计算2^jP ：$= 2^{j-1}P+2^{j-1}P$ 。如果总数没有定义，那么输出e ·P 不能计算并且停止。否则，从第二步开始重复。

5、 椭园曲线算法将使用这个方法，指双倍重复。注意，如果重复两个方法是没有用的，不能计算给定的倍数e·P，没有定义输出，那么遇到在$E_{a,b}(Z_n)$上的点Q1= (x_1,y_1)，Q=(x_2,y_2)，使得Q_1+Q_2是没有模n定义的。（在等式（C.2）和（C.3）的划分是失败的）。因此，gcd（$x_1 - x_2$; n）或gcd（y_1+y_2, n）是n的非平凡因子。

接下来，描述一些现象，关于在Z_n 上的椭园曲线定义和在Z_p上的椭园曲线定义，当p是n的一个因子。假设a_p=a mod p，b_p=b mod p

公理C.37、$E_{apbp}(Zp)$是一个附加交换群

更进一步，给定P=（x,y）∈$E_{ab}(Z_n)$，定义P_p=（x mod p，y mod p）。P_p是椭园曲线$E_{apbp}(Z_p)$上的一个点。

公理C.38、假设P和Q是$E_{apbp}(Z_n)$上的两点，并且假设p是n的一个因数。如果P+Q是模n上的，那么P_p+Q_p是定义在$E_{apbp}(Z_q)$上的。并且P_p+Q_p = （P+Q）$_p$ 。然而，如果P≠-q，那么P+Q和是模n定义的当且仅当有n的素因子q，使得点P_q和Q_q 的总数的点在$E_{apbp}(Z_p)$是是无限的O。（在$E_{apbp}(Z_p)$是P_p = - Q_p）

C.11.2 使用椭园曲线的因子分解

Lenstra's曲线分解算法的主要观点是在$E_{apbp}(Zp)$上发现点P和Q，因此P+Q在$E_{apbp}(Zp)$上是不定义的，将假设n 没有因子2或3，因为在算法开始前可以被区分。

在输入为n时，算法运行如下：

1、 产生一个椭园曲线$E_{a,b}(Z_n)$，随机在$E_{a,b}(Z_n)$上选择一个点P（x,y），在Z_n上选择x,y,a，并且选择安置b = $y^2 - x^3$- ax mod n

2、 计算gcd（$4a^3+27b^2$，n），如果1< gcd（$4a^3+27b^2$，n）<n，那么当发现n的一个因子，停止。如果gcd（$4a^3+27b^2$，n）=1，那么$4a^3+27b^2$ |≡ 0 mod p，对于每个n的素因数，并且因此，$Z_{a,b}$是一个Z_p上的椭园曲线，可以继续进行。但是如果gcd（$4a^3+27b^2$，n）=n，那么重新选择椭园曲线$E_{a,b}$

3、 假设e=$\prod_{i=1}^{\pi(B)} p_i^{a_i}$ ，这里$p_1,p_2,\ldots,p_{\pi(B)}$是≤B的素数，并且$a_i$进行极大选择，因此，

p_i^{ai} ≤C。B和C是限制在随后的决定中去优化运算时间和保证算法必须可以继续

4、 在$E_{a,b}$（Z_n）使用双倍重复进行计算e.P,在增加两个中间点P_1 （x_1 ,y_1 ）和P_2 （x_2 ,y_2 ）之前，检查gcd(x_1-x_2,n)或gcd(y_1+y_2,n)是一个n的非平方因子。如果是这样，输出因子并且停止，否则重复第1步

一个椭园曲线$E_{a,b}$（Z_n）会导致一个非平方的因数p，如果对于n的素因子p和q，e.P_p = O，但是，P_p 没有可以分解e的因子，在$E_{apbp}(Zp)$上，注意类推Lenstra's椭园曲线并且Pollard's的p-1算法。在Pollard's的p-1算法可以找到n上的素因子p,q，那么e是a∈Z_p^* 是一个复数阶，但不是一个a∈Z_p^* 的倍数。类似的，在Lenstra's上的算法寻找n的素因子p,q，因此，e是一个倍数P_p∈$E_{a,b}(Z_n)$，但是不是一个阶P_p∈$E_{a,b}(Z_n)$。然而，有一个不同的密钥，在两个算法中的多样化的不同。在Pollard的算法，群Z_p^* 是p在n上的因子是固定，因此，如果这些群有阶整除e，那么这个方法失败。在Lenstra的椭园曲线算法群是$E_{apbp}(Zp)$，可以被变量a,b划分。因此，对于每个n的因子p，| $E_{apbp}(Zp)$| || e，那么，还是继续用简单的工作代替其他的椭园曲线。也就是说，对a,b选择新的值。

C.11.3 Lenstra 算法的正确性

假设有n的素因数p,q，使得e是一个$|E_{apbp}(Zp)|$的乘积，但是在$E_{apbp}(Zq)$上Pq没有阶因子。那么$e\cdot Pp = O$在$E_{apbp}(Zp)$里，但是 $e\cdot Pq \neq O$ $E_{apbp}(Zq)$，因此有两个点 $P_1=(x_1,y_1)$,$Q_1=(x_2,y_2)$的一个中间加计算$e\cdot P$：

$$X_1 \equiv x_2 \bmod p \text{ 但是 } X_1 \not\equiv x_2 \bmod q\text{，如果 } P_1 \neq P_2$$

或者，$y_1 \equiv -y_2 \bmod p$ 但是，$y_1 \not\equiv -y_2 \bmod q$ 如果 $P_1 = P_2$

因此，$\gcd(x_1-x_2,n)$或$\gcd(y_1+y_2,n)$是一个n的非平凡因子，点P_1和P_2是相等的，如果$(P_1)_p+(P_2)_p = O$ 在$E_{apbp}(Z_p)$上，但是$(P_1)_q+(P_2)_q \neq O$在$E_{apbp}(Z_q)$上。

C.11.4 运算时间分析

在椭园曲线上计算加法的时间可以花费$M(n) = O(\log^2 n)$如果使用欧几里德算法。因此，计算$e\cdot P$花费双倍的时间。处理一个给定的椭园曲线时间是O（（loge）(M(n))。回忆，

$e=\prod_{i=1}^{\pi(B)} p_i^{ai}$ ，这里$p_i^{ai} \le C$，那么$e \le C^{\pi(B)}$，因此，$\log e \le \pi(B)\log C$。现在假设p是n的最小素因数，考虑选择 $B= L^{\beta}(p) = \exp[\beta(\log p)^{1/2}(\log\log p)^{1/2}]$这里β是最优化的。那么：$\log B =\beta(\log p)^{1/2}(\log\log p)^{1/2} = e^{[\log\beta+(\log\log p)/2 + (\log\log\log p)/2]}$ 因此，

$$\pi(B) \approx B/\log B \approx O(e^{[\beta(\log p)^{1/2}(\log\log p)^{1/2} - (\log\log p)/2]}) \approx O(L^{\beta}(p))$$

因此，对每个算法的迭代时间要求是$O(L^{\beta}(p)M(n)(\log C))$。在选择C时，注意e是$|E_{apbp}(Zp)|$里n的素因数的指数。称$|E_{apbp}(Zp)|$是B平滑的，p的值未知，但是如果p是n的最小的素因数，那么$p<\sqrt{n}$，通过Hasse's的不等式是已知的（例如[196]的131页）。因此，

$P+1-2\sqrt{p} < |E_{apbp}(Zp)| < P+1+2\sqrt{p}$ 并且因此，$|E_{apbp}(Zp)| < \sqrt{n} +1 +2\sqrt[4]{n}$，因此，获得

$C = \sqrt{n} +1 +2\sqrt[4]{n}$

剩下唯一考虑的是确定椭园曲线的测定，必须检查获得 n 的因数分解。部分分析在Lenstra是依靠如下结论，在[134]的结果中命题。

命题C.29、假设$S=\{s\in Z:|s-(p+1)| < \sqrt{p}$,并且s是$L(p)^{\beta}$平滑的}，假设n是一个合整数，有至少两个大于3的不同素数。那么

$$\Pr[\text{Lenstra's算法因子n}] \ge \Omega(|S|-2/2\sqrt{p})(1/\log p)$$

(这里概率是在Z_n上的x,y和a)

换言之，命题断言，概率是一个随机三维向量（x,y,a），导出一个n的因子分解本质上是随机的，在域$(p+1-\sqrt{p}; p+1+\sqrt{p})$的随机整数是$L(p)^{\beta}$平滑的。随后的概率是$|S|/(2\lfloor\sqrt{p}\rfloor+1)$。

回想处理整数的平滑整数，比以前的限制小，可以看到一个Canfield，Erdos的定理并且暗示：Pr[m≤x是L（x）$^{\alpha}$平滑的］= $L^{1/2\alpha}$ （x）

然而，才看到有未证明推测同样的结果，相同的结果是无效的，如果m是在小区间（p + 1 -$\sqrt{p}$; p + 1 + $\sqrt{p}$ ）的随机整数。说明：

Pr[m∈≤x，（p + 1 -$\sqrt{p}$; p + 1 + $\sqrt{p}$ ）是L（x）$^{\alpha}$平滑的］= $L^{-1/2\beta}$ （x）

因此，在命题C.39成功的概率的低边界是清楚的，有：

Pr[Lenstra's算法的因子n]≥Ω($L^{-1/2\beta}$ (p))1/logp

因此，在计算一个希望尝试($L^{1/2\beta}$ (p))logp的椭园曲线L^{α}（p）平滑阶，因此，总的运算时间要求希望值是O（($L^{1/2\beta}$ (p) （logp) L^{α}（p）M（n）logC）= O($L^{\beta+1/2\beta}$(p)（$\log^4$n）)，当β =1/$\sqrt{2}$ 。

评注：在Lenstra's算法的第3步，一个较小的实践难题在于选择$B=L^{\beta}$（p），因为在算法开始前，p是n的最小素数，在算法开始前是未知的。这个难题可以解决，采用$B=L^{\alpha}$ （v），对于v，对于一个逐渐增长的值执行一个算法，当v最后超过$\sqrt{n}$ 因数分解在于连续失败和宣称失败，因为n的最小素因子不超过$\sqrt{n}$ 。

附录 D：关于 PGP

PGP是免费软件包，在电子邮件系统中执行密码任务。在这个短附录中，将回顾一些特征，对于这个函数读者的研究在[198]章的第9章中进行完全的描述。

D.1 认证

PGP使用一个单向签名执行消息认证的范式，给定了一个消息M，过程如下：
消息是有时间戳的，数据和时间附在其后
使用MD5算法进行HASH[175]
128比特的摘要结果是用RSA[176]私钥进行发送的
签名附在消息后

D.2 私密性

PGP使用一个特异的系统确保私密，也就是说在一次一密条件下，使用一个快速对称加密体制进行每个消息的加密。使用接收者的公钥进行密钥加密，并且与加密消息一起进行发送。

详细的，假设A希望发送一个加密消息到B。
一个使用ZIP加密包的压缩消息，假设M是消息压缩的结果
产生一个128比特的随机密钥
消息M在密钥k下进行加密，使用对称加密体制IDEA（见[129]或[198]的第7章），假设C是相应的密文，K是使用RSA的B的加密体制的，假设c是相应的密文数据对（c,C）发送到B，认证和签名都是需要的，消息首先进行签名，然后压缩和加密。

D.3 密钥大小

PGP需要使用RSA的三种密钥长度
临时的384比特
商用512比特
军用1024比特

D.4 E-mail 兼容性

既然E-mail允许ASCAII字符传输，PGP需要从ASCAII字符恢复消息的部分加密。

为了使用64位基的变化把二进制比特流转换为ASCAII字符集，这个变换以33%进行消息扩展，但是因为原始的ZIP压缩，密文结果是还是比原始数据少1/3。

万一密文结果还是比仅仅限制在一些E-mail系统中的长度大，PGP分成碎片并且单独的发送消息。

D.5 一次 IDEA 密钥产生

注意PGP没有会话密钥，事实上，每个消息在密钥k下产生一个消息的ad hoc，种子是起源于用户的键盘，也就是说，从实际的键盘进行输入，时间插入其中。

D.6 公钥管理

假设考虑PK是用户B的一个公钥，当替代为C，了解相应的秘密密钥SK：
这个产生两个主要的难题：

1、 C可以读加密消息，A考虑可以使用B
2、 C可以有认为从B来源的消息A

建立信任的难题在于公钥和拥有者之间的信任问题，不仅仅是PGP。
有不同的渠道解决这个问题。

B的物理交换，个人给定一个A，储存在软盘中。
确认A可以在电话里呼叫B，并且确认密钥。
授权认证，有一个信任中心AUTH，签署用户的公钥，在密钥有用户ID之间建立沟通，对于一个范围的可信任密钥，相信从使用的用户中进行确认，在[198]中可以发现细节。

附录 E：问题

E.1 秘密密钥加密

E.1.1 DES

假设$\overline{m}$是串m的面向比特的补，假设$DES_k(m)$表示在DES下使用K定义的m的加密，不难看出如果 $c=DES_k(m)$ 那么 $\overline{c}=DES_{\overline{k}}(m)$。知道在DES的一个强力破解问题，要求搜索一个$2^{56}$空间，这意味着有许多DES加密的执行问题，是为了发现密钥，在最坏的情况下：

1、 在已知明文攻击条件下，（例如，给定一个单独的对（m,c），这里$c=DES_k(m)$）使用上述等式，变换DES的数字，执行一个强力破解去恢复K？
2、 在选择明文攻击的条件下（当允许选择许多m，对于获得的对（m,c），使用$c=DES_k(m)$）？

E.1.2 在 DES 密文条件下的错误检测

假设n明文分组是$x_1,\ldots,x_n$加密产生密文$y_1,\ldots,y_n$，假设一个密文分组，也就是y_i，不正确传输（例如一些1变换为0，并且进行恶性循环），有多少明文分组会被不正确的解密，如果用ECB模式进行解密。如果用CBC模式呢？

E.1.3 对 CBC 模式的强力破解

对DES在ECB模式下的一个已知明文强力密钥搜索攻击是很直接的，给定64bit明文和64比特密文，尝试所有的2^{56} 种密钥，直到从一个明文产生一个密文。对于CBC加密模式是更复杂的，包括使用一个64比特的IV，这看来介绍一个不确定的另外64比特。

1、 建议在CBC模式下，已知明文攻击的策略，是与ECB模式攻击相同的运算数量。
2、 现在考虑仅知密文攻击，对于ECB模式，这个策略是试着在给定明文条件下去解密，对所有的可能2^{56}密钥，并且测试每个结果，如果试着在语句上证明是正确的，对于CBC模式的工作策略是什么？如果是这样，解释。对CBC模式描述一个攻击策略，并且估计其努力水平。

E.1.4 E-mail

电子邮件系统在不同的用途对于不同的收件人是有把握的，在一些系统里，起源邮件处理者是所有必要的备份，这些单独的发出。所有变化的进展是首先对每个目的确定路由，那么一个单独的信息是从路径的一部分发出的。当路径分叉，进行拷贝（这个系统作为邮件包已知）

1、 把安全考虑放到一边，讨论两个方法的相对优势和劣势

2、 讨论安全要求和两个方法的含义

E.2 口令字

这是UNIX口令字的框架，固定一些函数，$h:\{0,1\}^k \rightarrow \{0,1\}^L$，用户选择一个k比特口令字，系统储存值y=h(K)在口令文件，当用户进行注册，必须提供K，系统然后计算h(K)，并且断言如果这个值等于y，通过认证。

假设攻击者可以访问口令字文件，因此，发送到y。直觉是从y恢复K是计算灵活的，因此，h 必须选择这是为真的。

H的说明选择是UNIX构造的，是$h(K)=DES_k(0)$，这里'0'代表64比特零串，因此，k=56，并且L=64。

在这个难题下，将分析普通配置和基于事例的特殊的DES。目的是怎样给出一个如同这样的体制。为了使用这个模型，必须进行继续进展，特别把DES当作伪随机函数簇。

为了配置模型，应该分析一般的配置，并且基于特殊的DES，目的是看怎样给定一个这样的体制。为了使用这个模型，在经典中使用模型。特别考虑DES，作为一个伪随机函数簇。

为了安排配置，假设F：$\{0，1\}^k \times \{0,1\}^l \rightarrow \{01\}^L$ 是一个伪随机函数簇，有一些给定的不安全的元素 $Adv_F^{prf}(.,.)$,并且L>k。假设T_F定义时间计算F。（也就是时间，给定K，x，计算$F_k(x).$）如下定义一个单向函数，现在将要指出。

（a） 定义 h：$\{0，1\}^k \rightarrow \{0,1\}^L$ $h(K) = F_K(0)$，这里0代表全0的1比特串，证明h是一个单向函数：

$$Adv_F^{prf}(D) \geq Adv_{h,I}^{owf}/2$$

使用起源结果。

（b） 能考虑可能的威胁或者弱点，也许在真实的条件下是使用的。但是不包括模型，能考虑怎样对抗？能考虑这是一个实际上好的口令体制。

现在提供一个安全的定义，对上述使用的单向函数。

假设h、$\{0，1\}^k \rightarrow \{0,1\}^L$是一个函数，是一个单向函数，如果直观的表述，是困难的，计算一个点x' 使得h(x')=y，当选择y，通过随机从$\{0，1\}_k$中选择x，并且设置y=h(x)，为了更正式，一个h的逆是一个算法I，给定一个点$y \in \{0,1\}^L$并且设置并且假设x'，假设：

$$Adv_{h,I}^{owf} = P[h(x')=y : x \leftarrow \{0,1\}^k，\ y \leftarrow h(x)，\ x' \leftarrow I(y)]$$

是求逆成功的概率，采用一个x的随机选择，任何投币是随机事件，假设：

$$Adv_h^{owf}(t') = \max_I \{Adv_{h,I}^{owf}\}$$

这里求逆I的最大消耗时间是t'。

E.3 数论

E.3.1 数论定理

证明如下的事实：

1、 如果 k 是 n 的不同素因数，那么等式 $x^2 = 1 \bmod n$ 有2^k，在Z_n^* 上有不同的解。隐藏，使用中国剩余定理。

2、 如果p是素数，并且$x \in Z_p^*$，那么，$(x/p) = x^{(p-1)/2}$

3、 G是一个Z_p^*的生成元，如果$g^{p-1} = 1 \bmod p$，对所有q素数是p-1的整数。

E.3.2 难题之间的关系

假设n是两个素数的乘积，描述如下剩余的两个难题（如果希望可以求RSA的逆的因数），不须要正式证明任何事情。仅仅表述如下的结果：

计算Φ（n）

因子n

对一些 $a \in Z_n^*$

计算模n平方根

计算模n的第k个根，这里gcd(k,Φ(n))=1

E.3.3 概率素性测试

假设SQRT(p,a)定义一个期望的多项式时间算法，输入p,a，输出x，因此，$x^2=a \bmod p$，如果a是一个模p的平方剩余根。考虑如下的概率素性测试，采用一个输入的奇整数p>1，并且输出是“合数”或“素数”。

1、 测试存在b,c>1，使得$p=b^c$，如果是这样，输出合数

2、 随机选择$i \in Z_p^*$，并且设置$y=i^2$

3、 计算x=SQRT(p,y)

4、 如果$x=i \bmod p$或者 $x=-i \bmod p$ 输出素数，否则输出合数

（A） 上述素性测试总是在希望的多项式时间内进行？证明的回答

（B） 如果p是素数，上述算法出现一个错误的概率

（C） 如果p是合数，上述算法出现一个错误的概率

E.4 公钥加密

E.4.1 简单的 RSA 问题

假设有一个使用RSA算法的分组加密，并且没有私钥，假设n=pq，e是公钥，假设有人告知一个明文分组有一个n的公共因子，这在其方面帮助了。

E.4.2 另一些简单的 RSA 问题

在RSA公钥加密体制里，每个用户有一个公钥n,e和一个私钥d。假设Bob，泄露了私钥，另外产生了一些新的模式，决定产生一个新的对子e’,d’，这是一个好思想？

E.4.3 导致 RSA 协议失败

回忆一个RSA体制公钥是一个对（n,e），这里n是两个素数的乘积。

$$RSA_{(n,e)}(m) = m^e \bmod n$$

假设在网络上的三个用户Alice，Bob 和Carl 使用RSA分别加密体制$(n_A,3),(n_b,3)$和$(n_c,3)$，假设David希望发送相应相同的消息到中的三个。这样David计算

$$y_A = m^3 \bmod n_A\ ,\ y_B = m^3 \bmod n_B,\ y_C = m^3 \bmod n_C$$

并且发送密文到相应的用户。

看看窃听者能够计算消息m，甚至在不知道Alice，Bob 和Carl任何密钥的基础上。

E4.4 RSA 的猜想

最好的因子分解算法知道时间（素域筛法）：$e^{o(\log 1/3 \log\log 2/3 n)}$

也就是说运算时间并不依赖于最小因数，一定在整个合数的大小范围内。

上述观察看起来建议为了保护RSA的安全，也许不必要增加两个素数的大小，但是仅仅是中的一个。Shamir建议如下的RSA的版本，称为非平衡RSA（已知RSA的猜想）。选择模n的RSA为5000比特长度，一个500比特素数p和一个4500比特素数，既然通常RSA通常使用仅仅交换密钥，可以假设消息加密到比p更小。

（A） 怎样选择公共指数，3是否是好的选择？

既然选择公共指数选择e，一个计算$d = e^{-1} \bmod \Phi(n)$ 并且保持秘密。难题是这样大的一个模数，是加密一个密文 $c = m^e \bmod n$ 可以消耗很长的时间（既然必需计算$c^d \bmod n$），但是既然知道m<p，可以仅仅使用中国剩余定理，并且计算$m1=c^d \bmod p=m$，Shamir声明在没有能效损失的条件下，抗进一步的因式分解攻击。

（B） 同选择一个单独的消息攻击，（i,e，获得选择的一个消息解密）通过n 的因子分解，可以完全破解RSA体制。

E4.4.5 Diffie-Hellman 的难题

回忆Diffie-Hellman密钥交换协议，p是一个素数，g是Z_p^*的一个生成元。Alice的密钥是一个随机密钥a<p，并且公钥是$g^a \bmod p$，公钥是g^{ab}。

在这个难题里，将证明如果Diffie-Hellman密钥交换协议是安全的，对于值（a,b）是一个小的分数，但是对于几乎所有的值（a,b）是安全的。

假设有一个PPT算法A，$Prob[A(g^a,g^b)=g^{ab}]>1/2+\varepsilon$（这里概率产生（a,b）是相互选择，并且是一个相互概率事件A）的任务是证明，对于任何$\delta<1$，存在一个PPT算法B使得对于所有（a,b），$Prob[B(g^a,g^b)=g^{ab}]>1-\varepsilon$。（这里概率是随机事件B）

E4.4.6 比特承诺

考虑如下的真实情况，Alice和Bob正在玩“猜测正在考虑哪个比特？”Alice考虑一个比特b=0,1并且Bob试着猜测正确或错误。

然而Bob失去所有的时间，因此怀疑Alice欺骗。她听到Bob的猜测，断言可以考虑相反的比特。因此Bob要求Alice写下一比特，封装后递交并且把信封放置在桌上。在这一点上，Alice是诚信的。然而Bob没有关于这一点的信息。

目标是在没有信封的条件下，达到关于安全的承诺。考虑如下的方法，Alice和Bob一起选择一个素数和一个Z_p^*的一个生成元p，当Alice希望进行一个比特承诺b，随机选择一个x

$\in Z_p^*$，因此，lsb（x）= b并且公布$y=g^x \bmod p$，有更好的比特承诺？有一个好的建议？

E.4.7 完善前向保密

假设两方Alice 和 Bob，希望私下沟通。在传统Diffie-Hellman密钥交换协议模型下持有两方的公钥。

一个窃听者Eve储存之间所有的加密信息，某天希望攻击Alice 和 Bob的计算机，并且找到私钥，进行公钥通信。

看到怎样使用一个公钥密码学，可以达到一个完善的前向安全。Eve将不能得到任何信息的知识，Alice和Bob是在私钥公开前的交换。

E.4.8 已知明文和非延展性

可以说一个已知明文的加密体制，如果没有知道相应的明文产生一个有效的明文是不可能的。

通常已知明文加密体制是通过一个明文冗余实施加法，一个密文的解密结果既在一个有效消息和一个非法的指标（如果不是正确的冗余）。正确的解密确信接受者是在发送者已知明文进行解密。

已知明文的概念是与有延展性相关的，说一个加密体制E 是不可延展的，如果给定义个密文 c = E(m)，是不可能产生一个相关信息m'的正确的密文C'的。

比较两个定义，告诉如果一个则意味着另一个。

E.4.9 概率加密

假设有一个消息m，希望用一个概率的方法加密。对于每个如下的方法，考虑一个好的或坏的方法：

1、固定一个大素数p，假设g是一个生成元。对于在m中的每个比特b_i，随机选择$x_i \in Z_{p-1}$，使得$lsb(x_i)= b_i$，（lsb(x)=x的低位比特），密文是串联 $y_i=g^{xi} \pmod p$，如果使用x 的其比特会使什么结果，$msb(x_i)= b_i$。

2、选择一个RSA公钥n，e，使得 |n|> 2m，任何m比特的底码变换成n的倍数。

3、选择一个RSA公钥n，e，假设|m|是比loglog n 小的数，把m比特底码随机化为相同的长度n，假设m'是明文底码，加密 $c = m^e \bmod n$。

4、选择两个大素数p,q=3 mod 4，假设n=pq，对于每个在m中的比特b_i，随机选择$x_i \in Z_n^*$并且假设$y_i=x_i^2 \bmod n$，如果bi=0或$y_i=-x_i^2 \bmod n$，如果bi=1，密文是y_i的串联。

E.5 秘密加密体制

E.5.1 同步加密和认证

假设（E，D）是对称加密体制（cf第6章和MAC消息认证码（cf第8章）），假设Alice和Bob共享两个密钥K_1，K_2，用于私密和认证，想要交换消息m以私密和认证的方法。假设

发送如下的消息：

1、M，$MAC_{k2}(E_{k1}(M))$

2、$E_{k1}(M，MAC_{k2}(M))$

3、$E_{k1}(M)，MAC_{k2}(M)$

4、$E_{k1}(M)，E_{k1}(MAC_{k2}(M))$

5、$E_{k1}(M)，MAC_{k2}(E_{k1}（M）)$

6、$E_{k1}(M，A)$这里A是Alice的加密认证，Bob解密密文，检查明文的一半是A

对于每次条件，如果安全或不安全，简单的调整的回答。

E.6 单向函数

E.6.1 生日攻击

假设H是单向函数，输出m比特值，假设H是随机预言机制。对于每个串s，H（s）在0和2^m-1是统一的和独立分布的。

考虑对每一个碰撞的强力搜索，假设各种合作$s_1,s_2,\ldots$,直到发现一个碰撞。（也就是说，保持单向直到发现不同的串产生一个相同的HASH值）

证明HASH的期望值是接近$2^{m/2}$。

E.6.2 从 DES 构造单向函数

在这个难题下，将考虑从形如DES的分组密码进行构造HASH的两个提案。

假设E表示对称密码体制的分组密码算法，假设$E_k(M)$表示一个消息M分组，假设$M=M_0 \cdot M_1 \cdot \ldots \cdot M_n$，定义一个n+1个消息分组。

首先被提议的单向函数h_1用如下的方式进行：假设$H_0=M_0$并且定义：

$$H_i=E_{mi}(H_{i-1}) \oplus H_{i-1} \quad 对于i=1,\ldots,n$$

单向函数值如下：

$$h_1(M)= H_n$$

第二个提议的单向函数h_2用如下的方式进行：

$$H_i=E_{Hi}(H_{i-1}) \oplus M_i \quad 对于i=1,\ldots,n$$

单向函数的值如下：

$$h_2(M)= H_n$$

对于两个提议，怎样从DES构造的HASH中找到碰撞。

E.6.3 从 RSA 构造单向函数

考虑如下的单向函数H，固定一个RSA体制密钥n,e，定义$RSA_{n,e}(m)=m^e \bmod n$。假设消息可以被HASH为$m=m_1 \ldots m_k$，定义$h_1=m_1$，并且对于i>1：

$H_i= RSA_{n,e}(h_{i-1}) \oplus m_i$ ，那么H（m）$=h_n$，怎样发现一个碰撞？

E.7 伪随机

E.7.1 扩展 PRGs

假设给定一个PRG G，使用一个k比特的初始密钥加入2比特随机序列，构造一个k比特的初始密钥加入3比特随机序列。假设G_1（s）定义G（s）的k比特串，并且假设G_2（s）是最后k比特（因此，G（s）= G_1（s）·G_2（s）,这里a·b定义为序列a,b的串联）
考虑两种构造：

1、 $G'(s)=G_1(s)\cdot G(G_1(s))$

2、 $G''(s)=G_1(s)\cdot G(G_2(s))$

对于每个构造，是否构造证明的回答，也就是说，如果回答不提供一个简单的统计测试，也就是区分输出，G'从一个随机的3k序列提出。如果回答肯定，则证明。

E.7.2 从 PRG 到 PRF

回想PRFs到PRGs 的构造，假设G是一个双倍长度的PRG，对于长度为k的初试向量到长度为2k的序列。

假设G_0（k）为G（x）的前k比特，并且G_1（k）为G（x）的后k比特，换言之，G_0（x）·G_1（x）= G（x）并且| G_0（x）|=| G_1（x）|

对于任意比特串z，递归定义$G_{0\cdot z}(x)\cdot G_{1\cdot z}(x)=G(G_z(x))$当| $G_{0\cdot z}(x)$|=| $G_{1\cdot z}(x)$|
构造的PRF簇是定义为F={f_i}。f_i （x）=G_x(i)。假设代替定义f_i （x）=G_x(x)，这是PRF簇？

E.8 数字签名

E.8.1 伪造表

对于两个RSA和ElGamal说明体制是
1、 普通伪造
2、 可选择伪造
3、 存在伪造
如果在下一种攻击。

E.8.2 ElGamal

假设Bob使用Elgmal签名体制，Bob使用两个签名（r,s_1）（r,s_2）签两个消息m_1,m_2，（两个签名中使用相同的值r）假设gcd（s_1-s_2,p-1）=1

1、 在给出信息的条件下，怎样有效计算k
2、 给出签名体制怎样被破解

E.8.3 建议签名体制

考虑如下的离散对数签名体制，假设p是一个大素数，q是一个生成元，私钥x<p，公钥$y=g^x \bmod p$。

签名一个消息M，计算单向函数h=H(M)，如果gcd(h,p-1)是与1不同的，那么把h附加到M后，并且再进行HASH。重复直到gcd(h,p-1)=1，求Z的解：Zh=X mod (p-1)

消息的签名是$x=g^Z \bmod p$，为了证实这个签名，使用者检查$s^h=y \bmod p$。

1、 可以看出有效签名总是会被接受

2、 这个体制是否安全

E.8.4 Ong-Schnorr-Shamir

Ong,Schnorr和Shamir建议如下的签名体制：

假设n是一个大整数，（不必知道n的因子），那么选择$k \in Z_n^*$ 假设：

$$H=-k^2 \bmod n = -(k^{-1})^2 \bmod n$$

公钥是（n,h），私钥是k。

为了签署一个消息M，产生一个随机数r, 使得r和n是相关的素数，那么计算：

S_1=（M/r+r）/2 mod n

S_2=k/2(M/r-r)

(S_1,S_2)是签名。

为了证明这个签名，检查：

$M=s_1^2+hs_2^2 \bmod n$

1、 证明重构私钥，公钥相当于n的因子

2、 这个体制是否足够安全

E.9 协议

E.9.1 无条件的安全秘密共享

考虑一个通常意义下的秘密共享，A的经销商D希望在s 和 n中共享一个密钥托管，并且中的t个没有关于s的任何信息，但是t+1可以重构秘密。假设s_i是T_i的托管，假设v表示s可能的值，并且假设w表示不同可能的值，因此s是变量。（假设s是对于每个托管是相同的）假设w≥v，对于秘密共享体制，（那么跟随如下的比特值，代表一个比秘密共享是更小的数）暗示：使用这个事实，t个玩家没有关于秘密的信息，无论接受到的t值，s的值是可能的。

E.9.2 欺骗者的秘密共享

不诚实的托管可以防止秘密的重构，重构一个 $s_i \neq s_i$，使用密码工具在这个分类中怎样阻止拒绝服务攻击。

E.9.3 对离散对数的零知识证明

假设p是一个素数，q是一个模p的生成元，给定$y=g^x$，Alice声称已知y的离散指数x，希望相信Bob，但是不想使知道x，怎样可以知道这些？（对这个难题给定零知识证明）

E.9.4 健忘传输协议

一个健忘传输协议是一个在Alice和Bob中间的通信协议，Alice 在输入值为s时进行。在协议的最后Bob知道s ，在协议的最后，Bob知道s，并且没有相关的消息，Alice没有已发生的事件的信息。

一个1-2的健忘传输协议是一个在Alice和Bob中间的通信协议，Alice在两个值s_0,s_1 之间运行两个值，Bob在输入是比特b时进行运算，在协议的最后，Bob已知s_b 但是没有关于 s_{1-b} ，Alice 关于b没有先验信息。

给定一个作为黑盒子的健忘传输协议，可以设计一个1-2健忘传输协议。

E.9.5 电子货币

真实生活中的货币有两个特征：

1、 是匿名的，意味着当使用货币购买商品时，身份是隐匿的，与的信用卡使用可以得出的身份和消费习惯

2、 是可传输的，卖主得到的现金可以继续购买其商品，没有这种可能如果只付钱而不准许别人消费

电子现金提议，看到在一群中是所有非健忘传输，也就是说，用户从银行获得一个货币进行消费，但是卖主必须为了信誉返回银行货币。真的是作为一个非健忘传输的检查，在这个难题下，进行修改这个协议为了完成可传递。

看到的这个提案可以提炼如下：有三个代理，银行，用户和卖主。

银行有一个密钥对（S，P），一个用S进行签名是一个货币是一个固定的数量（$1），可能进行盲签名，意味着用户进行一个消息m上的签名S（m），但是银行不能得到关于m的消息。

撤回协议

1、 用户选择一个消息m

2、 银行盲签消息m和从用户帐户撤回$1

3、 用户恢复S（m），货币是对子（m,S(m)）

支付协议

1、 用户给定（m,S(m)）到卖主

2、 用户证实银行签名并且发送一个挑战随机数给用户

3、 用户给定一个回答r

4、 卖主证实回答正确

挑战应答协议需要侦测两倍货币的开销m,S(m)，c,r, 如果使用不同的伪随机事件传送给两个不同的接收者，既然她从不回答两个不同的挑战在同一个伪随机事件中，她的证实将永远不被透露，当难题将被解决时，希望传输真正的货币。

怎样达到上述目标，需要证实上述传输协议和支付协议。

暗示：如果卖主希望在支付协议中进行传输真实的货币，希望在真实货币和伪随机货币之间建立一个链接，用于随后的传输。注意卖主随机选择c，也许c 可以在一些不同的方式下被选择。

E.9.6 撤消协议的原子签名

回想协议允许用户从银行撤回一个$1，假设（n,3）是银行对公钥 RSA。

1、 用户准备100个消息，$m_1,m_2,\ldots,m_{100}$，并且计算$w_i=r_i^3\ m_i$ ，用户发送$w_1,w_2,\ldots,w_{100}$到银行
2、 银行随机选择盲签名中的99个，要求用户开通，银行选择$i_1,\ldots i_{99}$发送到用户。
3、 用户检测盲签名进行正确构造，最后在没有打开的盲体制中进行签名，W.l.o.g，假设这是第一个。因此，银行通过发送用户$w_1^{1/3}=r_1m_1^{1/3}$签名。
4、 用户区分r_1的签名，使用一个有效硬币得到签名m_1。

注意用户有1/100的概率进行欺骗。

假设协议不是原子的，也就是说银行和用户通信在每部的最后进行。协议可以在如下的每一步最后进行防止每一方的欺骗和防止滥用。

E.9.7 使用 Elgamal/DSS 进行盲签

在这一节里，看到一个使用RSA进行签名的消息，将进行如下考虑，假设p是一个大素数，q是p-1的一个因子，g是在Z_p^*中的一个阶为q的盲签名。X是银行的密钥，并且$y=g^x$是相应的公钥。假设H是一个自由碰撞。

当银行希望签署一个消息m，计算

$$a=g^k \bmod p$$

对于一个随机的密钥k,

$$C=H(m,a)$$

并且最后：

$$b=kc+xa \bmod q$$

消息m的签名是sig(m)=(a,b)，给定三元离子（m,a,b）证实是计算c=H(m,a)并且检测

$$g^b=a^cy^a$$

因此撤回如下的协议：

1、 用户告诉银行希望得到$1货币
2、 银行回复100个值$a_i=g^{ki}$ 对于随机k_i
3、 用户返回$c_i=H(m_i,a_i)$这里m_i是所有的$1
4、 银行需要用户开通这些99个帐户
5、 用户使用m_i的99个
6、 对于没有开通的指数I, 银行回复$b_i=k_ic_i+xa_i \bmod (p-1)$

然而，这不是匿名的，既然当货币回流时可以认出用户，为了真正进行匿名的协议，用户在第三步改变挑战值c_i，这个调整允许计算不同的签名m_i在不同的环境，当货币回流时自己不知道银行情况。在协议中，银行将象通常情况下一样检查，调整正确的运算，要求用户打开99个盲签名。