密码学的新方向

Invited Paper

Whitfield Diffie and Martin E. Hellman

摘要：本文讨论了当代密码学的两个发展方向：加密和认证。随着远程通信的发展，特别是计算机网络的发展，密码学面临着两大难题：可靠的密钥传输通道问题，和如何提供与书面签名等效的认证体系。文章讨论了这些问题的解决方法。此外还讨论了怎样开始运用通信理论和计算理论来解决密码学长期存在的密码学问题。

1 介绍

今天我们正处于密码学革命的边缘。数字硬件的便宜化使得密码学能够从机器计算能力的设计限制中解脱出来，同时，它降低了高程度加密工具的价格，使得这些工具能够被类似远程自动取款机和计算机终端等商业应用所采用。反之，这些商业应用引发了对新类型的密码系统的需求，即要做到减少对密钥分发渠道的安全性需求和提取一个安全性等同于书面签名的认证体系。同时，信息理论和计算机科学的发展使得开发安全的密码系统成为可能，也使得密码学从真正意义上变为一门科学，而不再是一项古老的技术。

计算机远程通信网的发展为处于世界两端的人和计算机之间的通信降低了代价，使得远程通信手段取代了大部分的邮件和私人送信等古老手段。对于很多采用这种手段的计算机通信，要能够做到防范他人对通信内容的窃听和注入。不幸的是，现在这些安全保障技术的发展远远落后于通信领域的其它技术的发展。当代密码体制不能满足这些安全性要求，以至于采用了这些密码体制的系统，使用者在使用过程中会感到极为不便利，从而消弱了远程通信处理技术的优越性。

最广为人知的密码难题就是保密机制：为了阻止他人窃取在不安全传输通道上的通信数据，我们要采用加密技术来保护这些通信数据，因而，现在的问题在于通信双方怎样安全共享通信密钥。采用一些安全的通信渠道，例如私人送信或挂号信等渠道，可以事先共享通信密钥。但是，两个陌生人之间的商业会话是经常发生的事情。初始的商业会话被延迟到通信双方通过以上所列举的一些物理手段来共享通信密钥后再发生是不现实的。这种密钥分发问题所导致的代价和延迟是利用远程通信网络来进行商业通信的一个巨大障碍。

第三部分给出了如何利用公开（即不安全的）通信渠道传输密文而不损害安全性的两种方法。在公钥密码体制中，加密密钥E和解密密钥D是完全不相同的，而且利用E求解D是计算上不可行的（计算量为10^{100}）。公开加密密钥E不会损害解密密钥D的安全性。因而，网络上的每一个用户可以将他的加密密钥E放在一个公共加密密钥薄上。这样网络上用户就可以利用对方提供的加密密钥来给他发送信息，并且只有对方才能解密出这个信息。因此，公钥密码系统是多路存取密码机制。两个人，无论熟人还是陌生人，都可以进行一个安全的私人会话。彼此利用对方提供的公钥来加密所要发送的信息，用自己的私钥对收到的信息进行解密。

我们建议使用一些技术来改进和发展公钥密码机制，但是，它所存在的问题仍然是巨大的。

公钥分配系统不需要一个安全的密钥传输通道。在这个系统中，发送方和接受方重复的发送彼此的公钥，直到对方准确收到为止。第三方即使窃听到了这些信息，想破译双方的私钥在计算上也是不可行的。在第三部分给出了公钥分发系统存在的一些问题的解决方法，Merkle[1]还提供了不同的解决方案。

第二个阻碍远程事务处理系统取代当代现有商业通信手段的障碍是认证。现在商业合同

是通过签名来保证其有效性的。签名的合同是有法律效力的。然而，一般的签名是在书面合同的传输和存储中使用的。要用纯粹的数字取代纸张工具，必须做到每个使用者发送的信息，其他人都必须能准确验证它的真实性，并且不能被他人，包括接收者在内伪造。既然只有一个人能发送信息，其他人只能接收，这就好比是一个广播式密码系统。现有的数字签名技术不能满足这种需求。

第四部分讨论了在提供一个真实的数字签名时存在的一些问题。考虑到某些原因，我们认为这是一个单向认证问题。对于这个问题的一些解决方案已给出，并且说明了怎样将一个公钥密码系统变为一个单向认证系统。

第五部分讨论了密码学不同问题的相关性，并且讨论了更为复杂的陷门问题。

当通信和计算引起新的密码学问题的同时，它们相关的信息理论和计算理论却开始为古典密码学存在的一些问题提供解决工具。

寻找不可破译的密码体系一直是密码学研究的最古老的主题之一，然而，到这个世纪初为止，所有被提出的加密系统都是最终可被破译的。直到1920年，“一次一密”加密体制才被创建，而且它被证明是不可破译的[2 , pp . 398-400]。这种加密机制和相关理论依据是建立在25年后的信息论基础上的[3]。“一次一密”加密机制加密需要很长时间，因而在实际应用时代价很大。

相比之下，大部分密码系统的安全性是建立在密码破译者在不知道密钥的情况下，对其破译的计算复杂性上基础的。这个问题属于算法复杂度研究领域，即计算复杂性领域和算法分析领域。运用这些理论的研究结果，在不久将来，将安全性的证明扩充到更有用系统中去将变为可能。第五部分将研究这些可能性。

2 常规密码体制

密码学是包含两种安全性问题的数学系统的研究：加密和认证。加密系统用来防止在公共通道传输的信息被非法窃取，即保证发送的信息只能被合法的接收者读取。认证系统用来防止他人注入非法信息到公开通道里，即保证接收者接受到的信息是发送者发送的。

如果一个通信通道的安全性不能满足用户的需求，那么它就被认为是公开的通信通道。因而，一个通信通道，例如电话线，可能是保密的也可能是公开的，这取决于它的使用者。根据使用方式通信通道可能面临着窃听或注入等威胁。在电话通信中，注入的威胁是巨大的，因为接收电话方不能确定到底是谁在给他打电话。而窃听需要使用窃听器，因而在技术上更为困难些，而且还要负法律责任。无线通信情况与之恰恰相反。窃听是被动的，不需冒违法的危险，而注入则需使用发射机并且容易被发现。

密码学问题包括加密和认证，认证又可分为消息认证和身份认证。消息认证处理的问题主要是以上讨论的问题，身份认证的任务是确保用户的合法身份。例如，某人使用了一个信用卡，那么他的身份必须得到证实，但是这个过程没有信息传输。如果不管在身份认证时明显缺少传输信息，消息认证和身份认证是相同的。其实在身份认证时，有隐藏信息“我是XX用户”。来自威胁环境和加密、认证这两个问题的其它方面不同，有时便于我们区分它们。

图1表明了在使用一般密码系统进行通信时信息的流动。它包含三部分：发送方，接收方和攻击者。发送方生成一个将在不安全通信通道上传输的明文 P 。为了防止攻击者窃取明文 P ，发送方利用一个可逆变换 S_K ,产生密文 $C = S_K(P)$ 。密钥 K 通过一个安全的通信通道发给且只发送给接收方。因为接收方知道密钥 K ，他能够利用 S_K 的逆变换 S_K^{-1} 来解密密文 C ，即 $S_K^{-1}(C) = S_K^{-1}(S_K(P)) = P$ 得到明文。当然，这种做法适用于传输密钥 K 的安全通道因载量或延时等问题不能传输明文 P 。举个例子，安全通信通道是每周一次的专人送信，不安全通道是电话线。

此密码系统有一个可逆变换 S_K ，

$$S_K : \{P\} \rightarrow \{C\} \qquad (1)$$

它将明文空间$\{P\}$变换为密文空间$\{C\}$。密钥K属于有限的密钥空间$\{K\}$，如果消息空间$\{P\}$和$\{C\}$相同，那我们标识它们为空间$\{M\}$。当我们讨论不同的加密变换S_K时，我们将不提及具体系统，而仅仅指密钥K的不同。

密码系统$\{S_K\}$的设计目的是为了加密和解密运算经济化，但是要确保密码分析运算是复杂的，以至于破译所要付出的代价是昂贵的。有两种方法可以实现这种要求。如果一个系统的安全性是建立在密码分析的计算复杂性上的，且能被穷尽计算破译，那么它是计算安全的；如果一个系统能够抵御任何形式的密码分析攻击，包括蛮力攻击，那么它是无条件安全的。无条件安全性系统在[3]和[4]中讨论，它属于信息理论的一部分，被称作香农理论，即通过穷尽计算获得最优解。

无条件安全性密码体制源于对密文多样意义的处理。例如，简单替代密文XMD的明文可能是：now , and , the 等等。相反，计算安全性密码体制本身就包含足够的信息来最终确定明文和密钥，它的安全性取决于计算它们的代价。

唯一一个广泛使用的无条件安全密码系统是“一次一密”系统。它使用与消息一样长的随机密钥。虽然这种系统是不可破译的，但是大量密钥的需求使得它不实用。本文着重讨论计算安全性密码系统，因为它们更为实用。我们讨论的开发可证明安全性密码系统，不包括这些不实用的系统，例如“一次一密”密码系统。我们趋向于开发使用密钥只有几百比特的系统，且要求这种系统无论在小规模的数字硬件上还是在只有几百行代码的软件上都能运行。

如果计算一份作业的内存使用量、运行时间虽然是有限的，然而是无法想象的巨大，那么我们就认为这份作业是计算不可行的。

像纠错码可分为卷积码和分组码一样，密码系统可分为两大类：流密码系统和分组密码系统。流密码系统将明文划分为一小块一小块（比特或者字符）。它经常产生一个伪随机比特流，将其模2加到明文的比特中去。而分组密码是变换大块内容的组合形式。在这种组合中，输入块的每一个小的变化都将导致输出结果的巨大变化。本文主要讨论分组密码系统，因为这种系统的差错传播机制对许多认证软件有巨大价值。

在认证系统中，系统要保证接收者收到的信息是真实的。它不仅仅要能防止攻击者向通信通道中注入貌似真实的新信息，也要能防止攻击者通过组合或者重复他过去拷贝留下的老合法消息等手段制造的新真实信息。一个仅仅用于信息保密性的密码系统，总的来说，不能防止后者形式的破坏。

为保证消息的真实性，我们需要在传播的消息中添加一些额外的内容。例如，在消息中添加时间和日期信息，然后对这新消息加密。这样就确保了只有密钥持有者才能得到正确的日期和时间。必须注意的是，要采用能够实现密文每一微小的变化都能导致解密后的明文大变化的加密算法。如果攻击者故意将干扰信息注入到通信通道中，从而将消息”erase file 7”变为消息”erase file 8”，那么这种刻意的差错传播机制确保了验证信息已被破坏。这个消息也因非真实性而被拒绝。

评估一个密码系统是否合适，第一步要将其面临的威胁分类。无论是用于加密还是用于认证的密码系统都可能面临以下的威胁。

惟密文攻击，这种形式的密码分析者只拥有密文。

已知明文攻击，这种形式的密码分析者拥有一定量的明文—密文对。

选择明文攻击，这种形式的密码分析者可以选定任意的明文信息，并且知道其对应的密文。

以上这些攻击形式，假定了密码分析者已经知道了密码系统$\{S_K\}$的加密算法，因为这个加密算法可以通过研究这个系统得到。许多密码系统的使用者试图对他们的加密工具对外保密，但是商业应用软件要求加密系统的算法对外公开，而且要求这种算法已成为一个商业

标准算法。

惟密文攻击在实际中是经常发生的。密码分析者能利用的信息只有某种语言的统计特性（比如，在英文中，字母“e”的出现概率是13%）和一些密文单词的可能明文（例如，信件的可能开头“Dear Sir”）。惟密文攻击的威胁性最小，如果一个系统抵御不了它的攻击，那么这个系统是彻底不安全的。

如果系统对已知明文攻击是安全的，那么它的使用者可以公开他们以前发送的保密消息，也无需在消息分类之前再对它们进行解释了。保密以前发送的消息对系统使用者来说是一个无理的负担，特别是通信稿稍候会被公开的商业情况下。虽然已知明文攻击并不是时常方生的，但如果一个系统无法承受它的攻击，那么这个系统就是不安全的。

选择明文攻击又叫做IFF攻击，这种叫法源于第二次世界大战时“识别系统”的发展。一个安装了IFF系统的雷达能够自动识别自己的朋友和敌人。这个雷达发送一个随时间变化的频谱给待识别的飞机。如果这架飞机是本方飞机且在本方上空领域飞行，那么它能在收到信息后，用正确的密钥对其加密，并将密文发送回来。反之，如果我方飞机在敌人上空，敌人密码分析专家也能向其发送信息，并且希望通过收到的密文来解密我方的密钥，所以这是一个选择明文攻击类型。

有一些针对识别系统的攻击，普通系统是抵御不了的，这就需要使用到我们将在文中介绍的新想法和技术。泄露识别系统的验证信息的威胁来源于多用户网络环境。多用户网络中接收方的密码表和其它验证信息安全性比单用户系统的更脆弱。后面我们将会知道，一些针对这些问题的保护技术也可用于处理发送方和接收方之间的冲突问题。发送方可能因某些因素拒绝承认他发送了的信息，同理，接收方也可能会拒绝承认他已收到了的信息；第三方冒充发送方发送信息，这些都会引起发送方和接收方之间的冲突。举个例子，一个不诚实的股票经纪人可能为了自己的利益，伪造客户的命令，未经授权的进行商业买卖；一个客户由于他的错误命令使其遭受了巨大的损失，他事后就拒绝承认这条命令是他发出的。由于这些问题的存在，数字签名技术就必不可缺了。稍后我们将介绍一些技术用来帮助接收方验证信息的真实性，同时阻止接收方伪造那些显然真实的信息，即解决接收方验证信息泄露问题和发送方和接收方之间的冲突问题。

3 公钥密码体制

正如图1所示的，密码学源于安全问题。一旦存在一个能够安全传输密钥的通信通道，那么就得考虑其它高带宽的通信通道的安全性，和怎样快速实现信息加密。

开发一个大型安全的远程通信系统，必须克服以上种种难题。N 个用户之间相互通信，就的需要 $(n^2-n)/2$ 个密钥对。两个陌生人之间的通信，他们通信密钥如果通过一个安全的物理手段传输是不现实的。把 $(n^2-n)/2$ 个密钥对预先发送给用户也是不现实的。在另外一篇论文[5]，作者采用了一个保守的方法来实现通信，不需要开发新的密码体制。但是这种方法降低了安全性、使用便利性，且只适用于遵守初级通信协议的星型网络。

我们建议开发如图2所示的密码系统。通信双方利用的是一个公开的通信通道，使用的是公开的加密算法，却能建立一个安全的连接。公钥密码体制包括公钥密码算法和公钥分配算法。公钥密码算法有利于实现认证机制，而公钥分配算法更接近于实现。

公钥密码算法是指定义在有限信息空间{M}上的，基于算法{ E_k }和{ D_k }的可逆变换

$$E_k:\{M\}\to\{M\} \qquad (2)$$

$$D_k:\{M\}\to\{M\} \qquad (3)$$

算法必须满足下列条件：

⑴对任给 $K \in \{K\}$， E_k 是 D_k 的互逆变换

⑵对任意的 $K \in \{K\}$ 和 $M \in \{M\}$，用 E_k 和 D_k 进行加密和解密是容易计算的

⑶对几乎所有的 $K \in \{K\}$，从 E_k 推出 D_k 在计算上是不可行的

⑷对任意的 $K \in \{K\}$，从 K 计算 E_k 和 D_k 是可行的

性质⑶保证了我们可以公开 E_k 而不损害 D_k 的安全性，这样才保证了公钥密码算法的安全性。因此，公钥密码算法可以分为两部分：加密算法和解密算法。利用加密算法推导解密算法是计算不可行的，反之亦然。

性质⑷确保了在对加密和解密函数没有限制的条件下，计算这两个函数的可行性。实际运用中，公钥加密工具必须包含一个随机数生成器用来生成 K，利用 K 计算 E_k 和 D_k。

公公钥分配算法是相当简单的。系统的每一个用户在自己的机器终端产生加密密钥 E 和 D。解密密钥 D 必须保密，而且不能在通信通道中传播。加密密钥 E 可以放置在公开的公钥密码薄上，并附上自己的姓名和地址。任何人都可以利用这个公开的加密密钥 E 给用户发送消息。公钥密码系统可以被认为是一个多路存取密码机制。

要防止公开的加密密钥被他人修改。这个保护很容易实现。公开的加密密钥文件使用只读保护机制是不必要的。既然密钥的修改是偶然性的，那么从经济上考虑，详细地写保护机制更适用。

举个例子，虽然这个例子并不实用，但对解释公钥密码算法是有用的。以加密二进制 n 维向量 m 为例，加密算法是将明文乘以一个 $n \times n$ 可逆矩阵 E，即密文是 Em，解密则乘其逆矩阵 $D = E^{-1}$，因此， $m = Dc$。加密和解密算法所需运算时间为 n^2。通过 E 求 D 涉及到求矩阵的逆矩阵，这是困难的。理论上讲，寻找一对可逆矩阵比求解给定矩阵的逆矩阵要容易。可逆矩阵 E 可通过对单位矩阵 I 做一系列的行和列的初等变换得到，而其逆矩阵是经过逆序的行和列的逆变换得到 $D = E^{-1}$。任意给定了的二进制位串，决定了一个初等变换。

不幸的是，矩阵求逆的时间代价只有 n^3，密码分析者利用 E 计算 D 用时与正常解密用时之比至多是 n，而理论要求至少在 10^6 以上。这个例子不能说明从 E 计算 D 比从 I 计算 D 要花费更多的时间。此外，因为二进制运算没有四舍五入的差错，所以在矩阵求逆过程中数值稳定性是不重要的。虽然这个例子并不实用，但有助于我们理解公钥密码算法。

另外一个能够快速计算可逆运算 E 和 D，并且从 E 推导 D 是困难的方法是利用机器语言的难读性。如果一个人的计算机正在运行一个用机器语言编写的算法 E，那么其他人即使得到了这个算法 E，也很难确定 E 的功能。如果这个人还刻意在算法 E 中添加一些无用的变量和代码，那么其他人就更难获得 E 的逆变换了。当然， E 要足够复杂到不能被一些公开的输入—输出信息对破解。

因而我们本质上需要的是一个单向编译器：可以将容易理解的高级语言程序翻译成等价的难以理解的机器语言程序。编译器是单向的，即它的逆翻译过程是计算上不可行的。我们基本上无需考虑这种编译器的大小和运行时间的长短，因而只要机器语言的结构能被优化到辅助混淆，那么设计出这种编译器是可能的。

Merkle[1]已经独立研究了如何在不安全通信通道上安全分发密钥。他的方法不同于我们上面讨论的公钥密码算法，可以被称作是公钥分配算法。算法的目的是让两个用户 A 和 B 如何在不安全的通信通道上安全传输密钥。利用这个密钥，用常规的密码体制对消息进行加密和解密。Merkle 的算法的时间代价是：合法用户加密、解密时间数量级为 n，而密码分析者的时间数量级为 n^2。不幸的是，因为 Merkle 的算法要求用户在使用一个密钥前，必须发送 n 个潜在密钥，这样一来，密钥传输的时间代价和加密、解密的时间代价在一个数量级上。Merkle 也注意到了密钥传输的高时间代价阻止了算法在实际中的广泛应用。如果这种算法被应用在一个百万数量级上，那么它的时间代价比大约为 10000:1，对于实际应用来说，太小了。如果高带宽的数据链路通道经济代价不是很大，且能让算法的时间比超过 1000000:1，那么它还是有实用价值的。

我们建议使用一种新的密钥分配算法，这种算法有一些优点：第一，只要传输一个密钥。第二，密码分析者的时间开销是系统使用者的指数级倍。第三，算法的公开参数可以放在包含用户信息的公开文件上，这样用户 A 和用户 B 可以利用它们彼此鉴别对方。将这个公开文件放在一个只读存储器中，这样用户就容易向其他用户表明自己的合法身份。

这种新算法的有效性依赖于计算有限域中离散对数的困难性。令 q 是一个素数，有限域为 $GF(q)$。计算

$$Y = \alpha^X \bmod q，\quad 其中 1 \le X \le q-1 \tag{4}$$

其中 α 是 $GF(q)$ 上的一个固定本原元。然后计算 X：

$$X = \log_\alpha Y \bmod q \quad ，其中 1 \le Y \le q-1 \tag{5}$$

不难得出由 X 计算 Y 是较容易的，约需要 $2 \times \log_2 q$ 次乘法运算[6 , pp . 398-422]；例如，X=18，则

$$Y = a^{18}(((a^2)^2)^2)^2 \times a^2 \tag{6}$$

然而用 Y 计算 X 是困难的。对于某些精心选定了的 q，采用最快的算法也需要 $q^{1/2}$ 次运算[7 , pp . 9 , 575-576] , [8]。

此密码算法的安全性是建立在求对数模 q 的运算困难性上的。如果能找到一种求对数模 q 计算量是随着 $\log_2 q$ 增长的算法，那么这种加密算法就是无效的。

每一个用户，从 $[1,2,\cdots,q-1]$ 中随机的选一个 X_i，计算出 Y_i

$$Y_i = Y_i = a^{x_i} \bmod q \tag{7}$$

并将 Y_i 公布，X_i 保密。那么当用户 i 和 j 通信时，使用

$$K_{ij} = \alpha^{X_i X_j} \bmod q \tag{8}$$

作为他们的公共密钥。此密钥用户 i 通过 j 公布的 Y_j 得到，即

$$K_{ij} = Y_j^{X_i} \bmod q \tag{9}$$

$$= (a^{X_j})^{X_i} \bmod q \tag{10}$$

$$= a^{X_j X_i} = a^{X_j X_i} \bmod q \tag{11}$$

用户 j 的计算 K_{ij} 同理。

$$K_{ij} = Y_i^{X_j} \bmod q \tag{12}$$

对于第三方要获得此密钥就必须计算

$$K_{ij} = Y_j^{(\log_\alpha Y_i)} \bmod q \tag{13}$$

如果 $\log_a^{Y_j}$ 容易计算，那么 K_{ij} 就容易计算，那么这个系统就会轻易被破解掉。我们没有证据说明即使 $\log_a^{Y_j}$ 被轻易计算出，这个系统也是安全的。也没有证据说明能够在不知道 X_i 和 X_j 的情况下，能够计算出 K_{ij}。

如果素数 q 小于 2^b，那么其它参数都可以用 b 个二进制位来表示。假设 $\log s$ 运算需要 $q^{1/2} = (2b)^{1/2}$ 运算，则求幂运算至多需要 $2b$ 次模 q 运算。密码分析者的运算量相对于系统使用者是指数倍增长的。如果 $b = 200$，那么从 X_i 计算 Y_i 或者从 Y_i 和 X_j 计算 K_{ij} 需要 400 次运算，然而计算 $\log s$ 模 q 需要 2^{100} 次运算，即大约为 10^{30}。

4 单向认证

相对于如何进行密钥安全分发难题，认证问题是妨碍我们采用远程通信手段进行商业交易的更大障碍。认证是涉及到使用合同和帐单的密码系统的核心，没有它，商业交易就无法

完成。现有的认证体制不能满足安全性要求。它们只能保证发送方不被第三方冒名顶替，但不能解决发送方和接收方之间的冲突。

为了能用数字式合同取代书面合同，我们需要设计一个能取代书面签名的数字签名系统。这个数字签名必须能标示使用方的身份，且不能被他人伪造。我们称这种技术为单向认证机制。

讨论一下多用户系统中的登录问题。当用户在这个系统中建立一个帐户时，他的登录密码被记录到系统密码文件中去。每次用户登录时，系统都要求用户输入正确的密码。我们要保护用户的密码不能被他人知道，否则他人就会伪造登录。因而保护系统密码文件的安全是十分重要的，否则通过它，任何人都可以冒名他人登录系统并操作系统。如果要保证系统管理员有合法的权限操作密码文件，那么这个保护就将更加复杂。只容许合法操作，禁止其它非法操作，是几乎不可能做到的。

这给系统提出了一个几乎不可能实现的功能要求：在不知道正确密码的情况下，对用户输入的密码进行判断。这个要求似乎在逻辑上是行不同的，但是想法还是相当令人满意的。当用户第一次创建密码 PW 时，系统自动计算函数 $f(PW)$，将其存储到系统密码文件中去，而不是存储 PW 。每次用户登录时，根据用户输入的密码 X，自动计算 $f(X)$，将其与密码文件中的 $f(PW)$ 比较。如果相同，则容许用户进入，否则拒绝。用户每次登录时，系统都要计算函数 f，因此 f 的计算量要小，运算量要限制在 1000 000 次之内。如果我们能够确保函数 f 的逆函数 f^{-1} 的计算量在 10^{30} 以上，那么攻击者即使成功获得了系统密码文件，也不能利用 $f(PW)$ 计算出密码 PW，也就不能非法登录系统。要知道 $f(PW)$ 不是系统的合法登录密码，因为 $f(f(PW))$ 不等于系统密码文件中的 $f(PW)$。

函数 f 可以是公开的，因为通过 f 求解 f^{-1} 是困难的。这类函数叫做单向函数。R.M.Needham 是第一个将它使用到系统登录程序中去的人[9 , p . 91]。在最近的两篇文章中[10]，[11]讨论并给出了设计单向函数的方法。

对定义域中的任意 x， $f(x)$ 是容易计算的，但对几乎所有的值域中的 y，求满足 $y=f(x)$ 的 x 在计算上是不可行的。

我们要知道，函数在计算上是不可逆的不等同于数学中函数不可逆。通常数学中的函数不可逆是指 y 的逆元不是唯一的（即存在 x^1， x^2，且 $x^1 \neq x^2$，但是 $f(x^1)=y=f(x^2)$）。我们要强调的是在知道 y 的值和函数 f 的情况下，求解 $f(x)=y$ 中的 x 的值不是一般的困难，而是几乎不可能的事情。如果函数 f 是通常意义下的函数不可逆，那么求解 x 的工作就变的简单的多。极端的，如果对于任何定义域中的 x，都有 $f(x)\equiv y_0$，即值域为 $\{y_0\}$，那么我们可以任取一个 x 作为 $f^{-1}(y_0)$。因此， f 不能为退化函数。如果 f 是小程度退化的，那么是可以接受的。在稍后的讨论中，我们会知道，现在比较实用的单向函数大部分都是小程度退化函数。

多项式可以用来构造初等的单向函数。对于多项式 $p(x)$，已知 x_0，求 $y=p(x)$ 是容易的，但若已知 y 求出 x_0 是困难的。Purdy[11]建议在有限域上使用高程度的疏多项式，因为它看起来攻击需要更高的时间代价。在第四部分，我们还将详细讨论单向函数的理论依据。在第五部分，我们会知道在现实中，单向函数很容易设计的。

多用户系统中的登录单向函数只解决了部分安全问题。它防止了对系统中不正在使用的鉴别数据的破坏，但是它仍然需要登录的用户输入正确的密码。此外，还需要额外的密码机制来防止抵赖，解决双方的冲突。

公钥密码算法可用来产生一个真正的单向认证体系。当用户 A 要发信息 M 给用户 B 时，他用其保密的解密密钥 D_A "解密" M 并将结果 $D_A(M)$ 传给 B，B 收到时用 A 公布的加密密钥 E_A "加密"此消息从而得到信息 M 。用户 B 保存消息 $D_A(M)$，作为消息 M 是来源于用户 A 的证据。因为解密密钥是保密的，只有 A 发送的消息才具有这样的性质，从而确认

此信息来源于 A，也就建立了一个单向认证体系。

马萨诸塞州计算机联合组织的 Leslie Lamport 还提出另一种单向消息认证方法，它是应用在 k 维二进制向量空间上的单向函数 f 到其自身的映射来实现的。若发送者发送 N 位比特的信息 M，他要产生 $2N$ 个随机 k 维二进制向量 x_1，X_1，x_2，X_2，…，x_N，X_N，并保密，随后把这些向量在 f 下的像 y_1，Y_1，y_2，Y_2，…，y_N，Y_N 发送给接收者。当发送信息 $M=(m_1,m_2,\cdots,m_N)$ 时，当 m_1=0 发送 x_1，m_1=1 发送 X_1，依次类推。接收者把收到的信息用 f 映射之，若为 y_1 则 m_1=0，若为 Y_1 则 m_1=1，如此下去便得到了 M。中途传输的消息 M 中即使只有一个比特被改变，接收者也对此无能为力。

这种解决方案大约需要传输 100 倍额外数据。当 N 接近或者大于 1000 000 时，有一种改进方法用来消除这种额外空间开销。让我们引入单向映射 g，它的功能是将 N 位比特的空间映射成 n 位的比特空间，其中 n 在 50 左右。N 位比特消息 m 经 g 作用后变为 n 位比特消息 m'。用前述方案将 m' 传输。如果 $N=10^6$，$n=50$，$k=100$，那么需要在这个消息中添加 $kn=5000$ 比特的认证消息。这样一来，消息传输只含有 5% 的额外数据（如果算入需要传输消息 y_1，Y_1，y_2，Y_2，…，y_N，Y_N，那也只需要传输 15% 的额外数据）。虽然平均有 2^{N-n} 个消息的认证消息相同，但是密码分析者想寻找到一个满足要求的消息在计算上是不可行的，也就不能伪造消息了。这里要求 g 具有比一般的单向函数更强的性质，它要求即使在知道消息 m 的情况下，也很难找到一个不同的消息 m'，使它们有共同的认证消息。这样的函数 f 看起来似乎很难找到（具体请看第五部分）。

还有一种单向认证问题的解决方案。使用者产生一个保密的密码 X，通过单向函数 f 变换，将结果 $f^T(X)$ 提交给系统。在时刻 t，系统利用 $f^t(X)$，验证鉴别消息 $f^{T-t}(X)$。这种解决方法的缺点是每次登录系统时都要进行大量计算（计算量的数量级比破解小的多）。例如，如果 t 是随时间增加的，系统在每个密码上运行一个月，那么 T =2600 000。那么每次登录时用户和系统都要得平均迭代运算 f 1300 000 次。如果不能解决运算量问题，那么这种解决方案就没有实际意义。如果能找到一种简单的方法快速计算 $f^{2\uparrow n}$，其中 n =1,2,…，形如 $X^8=((X^2)^2)^2$，那么这个问题就能解决。通过二进制分解 T 和 t，就能快速计算 f^{T-t} 和 f^t。不幸的是，如果能快速计算 f^n，那么 f 就可能不是单向的。

5 问题的相关性和陷门技术

在这个部分，我们将讨论一些迄今为止还无法转化为其它相关问题的密码学问题。我们根据这些问题的难度进行一个初步的排序。此外，我们也讨论一个更难的密码学问题——陷门问题。

在第二部分，我们已经知道了用于保密的密码算法也可用于认证。这种算法也可用于完成其它密码学功能。

1、一个对已知明文攻击安全的密码算法能产生一个单向函数。

设 $S_k:\{P\}\to\{C\},k\in\{K\}$，是一个对已知明文攻击安全的算法，取 $P=P_0$，考虑映射

$$f:\{K\}\to\{C\} \qquad (14)$$

定义为

$$f(x)=S_X(P_0) \qquad (15)$$

则 f 是一个单向函数，因为要由 $f(x)$ 得到 x 和已知明文攻击是等价的。公开函数 f 等价于公开 $\{S\}$ 和 P_0。

理论 1 反过来就不一定成立了。起初在寻找单向函数的过程中产生的某个函数也许就能产生一个好的密码算法。在第三部分讨论的离散指数函数就能做到这一点[8]。

单向函数是分组密码和密钥生成算法的基础。一个密钥产生器就是一个伪随机比特流产

生器。伪随机比特流产生器产生的结果叫做密钥流，效仿“一次一密”算法的做法，将其模2 加到二进制明文消息中。密钥作为伪随机比特流的“种子”。已知明文攻击问题就变为从密钥流中确定密钥。如果一个系统是安全的，那么要求从密钥流中计算出密钥是计算上不可行的。相反，如果一个系统是有用的，那么一个密钥要很容易计算出它的密钥流。因此，从定义上看，一个好的密钥产生器就是一个单向函数。

使用任何一种加密算法产生一个单向函数都存在一个小问题。如果函数 f 不是单射的，那么原象就不唯一了。只有找到一个满足条件的就可以了。如果映射 S_K 是双射的，那么函数 f 的原象就是唯一的，必须找到这个惟一的密钥。事实上，想确保加密算法的函数变换是双射的是困难的。一个好的加密算法中的函数映射应该具有这样特性：对于定义域中的任何一个值，映射的结果随机的（即值域中任何一个 y 都是等概率等于 $f(X_i)$ 的）。在这种情况下，如果密钥 X 和消息 Y 可能取值个数是相同的，那么随机选择一个密钥 X_i，$Y_i = f(X_i)$ 可能有 $k+1$ 个原象的概率近似为 $e^{-1}/k!$，其中 $k = 0,1,2,3,...$。这是一个 $\lambda = 1$ 的泊松分布。因此逆元个数的理想期望值为 2。函数 f 可以更大程度的退化，但是一个好的加密系统是不容许这种大程度退化的。因为这样有可能出现加密密钥无效的现象。更有甚者，如果 f 退化到 $f(X) \equiv Y_0$，即 $S_K(P_0) \equiv C_0$，那么加密消息 P_0 根本就不依赖于选择哪个密钥。

我们经常讨论的函数 f 的定义域和值域的可能取值个数大体是相同的，但是我们要知道，有的函数是例外的。比如，之前，我们就讨论过一个将长消息映射成短消息的单向函数。使用密钥长度大于块大小的分组密码算法，借助于以上介绍的技术，就可以设计这种例外函数。

Evans 等人[10]还提出过另一种方法从分组密码中构造一个单向函数。这种方法不用选择一个已经知道存在于值域中的 P_0 作为输出，他使用的映射是

$$f(X) = S_X(X) \qquad (16)$$

这种方法很吸引人，因为这个方程很难求解，即使 S 是一个相当简单的函数。这确实增加了破解的难度，但是我们也得看到这种方法的单向函数破坏了对已知明文攻击安全的要求。

2、一个公钥密码算法可用来产生一个单向认证体系。

反之，则不一定成立，因为设计一个公钥密码算法比设计一个单向认证算法要复杂的多。相似的，一个公钥密码算法可以用作一个公钥分配算法，反之则不成立。

在公钥密码算法中，加密算法 E 和解密算法 D 是公开的，而且公开 E 的使用方法，即怎样将明文变换成密文。这样的加密系统是一套真正的陷门单向函数组合。通过简单计算就可以得到它的逆函数的函数，不是真正意义上的单向函数。寻找单向函数的逆在计算上是不可行的，除非攻击者知道了陷门函数创建过程信息（也就是说，知道了用于创建 E-D 的随机比特字符串）。

陷门技术已经在先前介绍的陷门单向函数中使用了，但是它被用在了别的地方。一个含有陷门的密钥可以抵制任何密码分析者的强烈的攻击，除非是知道此密钥使用的陷门信息的人。因而，如果密钥创建者在不顾自己名声地情况下把这个陷门信息卖给了其他用户，那么这个密钥就不再是安全的了。我们得清楚的知道，密钥创建者能够完成其他密钥用户不能完成的功能的做法不是一个好的做法。如果他不小心泄露了陷门信息，那么他就和其他用户一样了。陷门就好比是一个号码锁。任何人只要知道了这个锁的号码，那么他就可以在几秒之内将其打开；如果不知道，那么即使是技术最好的锁匠，也可能需要几个小时才能将其打开。同样，如果密钥创建者忘记了陷门构造信息，那么他也和其他用户一样了，再无什么特权可言。

3、一个陷门密码算法可用来产生一个公钥分配算法。

比如 A 要和 B 建立公共私钥，A 任选一个密钥，用 B 公布的含有陷门信息的加密密钥

加密之，并将密文发送给 B，B 用保密的陷门信息解密得到此密钥，于是 A 和 B 建立了公共的私钥。

可惜的是，现在几乎没有证据证明陷门密钥的存在，但是我们也得小心它的可能性存在，因而要对从可能的对手处获得的密码系统保持警惕[12]。

从定义上知道，用陷门技术解决陷门问题在计算上是可行的。如果就连单向函数设计者也无法在计算上找到它的逆，那么我们称这种单向函数为准单向函数。因此，本质上说，可用准单向函数代替单向函数，而不损害系统安全性。

公开陷门单向函数的陷门信息，那这个函数就成了准单向函数，当然，这不是准单向函数的唯一获得方式。

将准单向函数从单向函数中划分出来，仅仅是个定义问题。你也可以定义广泛的单向函数将其包含进去。

类似的，一个准安全密钥是一个能抵制任何密钥分析者，包括它的设计者在内的攻击的密钥，并且寻找它的逆也是计算上不可行的。同样，严格区分准安全密钥和安全密钥的实际意义不大。

我们已经知道一个公钥密码系统中包含了一个陷门单向函数。但是，这句话反过来讲就不对了，除非这个陷门单向函数是可逆的（也就是说，它存在唯一的逆元）。

6 计算复杂度

密码学领域不同于其他领域，密码学中会经常出现表面看起来找到一个满足要求的问题解决方法，但通过实际艰苦的努力证明后发现这个解决方法并不满足要求的现象。比如，通过简单的变换就可以将明文变为一段无任何特殊意义的密文，但是，如果一个密码分析者声称他找到了这段密文的明文，那么他就得付出艰苦的努力去证明。事实上，经验告诉我们，从古至今，几乎没有哪个系统可以抵御有经验的密码分析者的攻击，许多声称是很安全的系统最终也被破解掉了。

因此，密码分析者的主要任务就是判断新密码系统的价值。在十六、十七世纪，对加密算法强度分析采用数学方法，主要计算它可能的密钥的个数。这种简单分析方法对密码学这种复杂问题是行不通的，虽然如此，不过就连著名的 Cardano 也犯过这种错误[2 , p . 145]。用这种分析方法证明安全的系统最终都被破解掉了，因此放弃了这种分析方法，取而代之的是使用密码分析者的攻击。

这个世纪，密码算法安全分析的指针又以其它方式摆回来了。在一篇关于信息理论起源的文章中，香农指出最近二十年使用的“一次一密”加密算法具有完全的保密性（即是无条件安全的）。我们要知道没有哪个公开密码算法或单向认证算法是无条件安全的，因为公开的信息就经常能够决定加密信息的明文唯一存在于一个有限的集合中。利用穷尽计算，我们就能找到这个明文。

过去十年，人们越来越加大对计算代价的研究，主要是针对计算复杂度理论和算法分析理论的研究。前者将计算问题按照计算难度分类，后者集中于寻找好的算法和展示算法所使用的资源。通过简单的讨论，我们可将其运用到密码学中，特别运用到对单向函数的分析中去。

能在确定型图灵机上求解，并且计算时间由一些输入的多项式长度决定的函数，属于 P 类复杂度。也许有人觉得 P 类复杂度函数就是很容易计算的函数，这种说法是不准确的。确切的说，不属于 P 类复杂度的函数至少是因为某些输入而很难计算的。我们已经知道了一些不属于 P 类复杂度的函数[13，405-425]。

有很多工程学问题利用任何现有的技术都不能通过多项式时间方式来解决，除非使用高并行度的计算机。这些问题无论是不是 P 类复杂度，但都属于 NP 类复杂度。 NP 类复杂度

是能在非确定型图灵机（也就是说，具有无限并行度）上用多项式时间求解的问题。显然 $P \in NP$，并且在计算复杂度理论中就有 NP 范围是否更大些的问题讨论。

已知的能用 NP 时间解决，而不能用 P 时间解决的问题有：旅行售货商问题，位置计算满足性问题，背包问题，图环问题，还有许多调度和最小化问题[13，pp。363-404]，[14]。我们要知道不是因为我们对这些问题用 P 时间解决不感兴趣或者不够努力。人们深信这些问题中至少有一个是不能用 P 时间解决的，因此 NP 时间范围更大些。

Karp 还定义了一个 NP 问题的子类，NP 完全，即如果 NP 完全中的任何一个问题属于 P 类，则 NP 中的所有问题都属于 P 类。Karp 列举了 21 个属于 NP 完全问题，集中就包括以上一些问题[14]。

尽管人们认为 NP 完全技术可以被密码学使用，但是目前对它的复杂性的理解还仅限于对最坏情况分析。为了密码学目标，必须考虑典型的计算代价。如果我们用平均或典型计算时间代替最坏计算时间作为我们的密码分析方法，那么当前能够说明 NP 完全问题等价性的证据就不再有效了。这样我们就有了一些新的研究方向。那么为信息理论者熟知的总体性和典型性概念就会有作用了。

现在我们可以确定密码分析问题在计算问题中所占的地位了。

一个加密和解密算法若是能在 P 时间内完成的，那么密码分析的难度不会大于 NP 时间。

我们要知道，任何密码分析是否成功取决于我们是否在一个有限集中找到了密钥、逆象等等。随机取一个可能密钥，用 P 时间验证它是否为正确地密钥，那么，如果有 M 个可能密钥可供选择，那么密码分析者相对于密码使用者就得有 M 倍计算量要做。比如，在一个已知明文攻击中，要使用每一个可能密钥对明文加密，将结果与正确密文比照，寻找到正确的密钥。因此，如果加密需要 P 时间，那么解密仅仅需要 NP 时间。

我们也得到了一般的密码分析问题是一个 NP 完全的问题，这是根据我们对密码学问题定义的宽度知道的。一个属于 NP 完全的单向函数的逆将在下面讨论。

因为 NP 完全问题可以用于加密，那么可以从 NP 复杂度理论中寻找加密算法。特别的，背包问题是一个 NP 完全问题，它有助于我们构建一个单向函数。 设 $Y = f(x) = a \cdots x$，其中 a 是 n 维整型向量 $(a_1, a_2, \cdots, a_n)$， x 是一个 n 维二进制向量。计算 Y 是简单的，包含最多 n 个整型数的求和运算。求解 f 的过程就是个背包问题，它要求寻找一个和为 y 的 $\{a_i\}$ 集合。

穷尽运算去检验这 2^n 个集合，代价是昂贵的，因而当 n 大于 100 时在计算上是不可行的。我们必须知道，想通过选择问题的参数以求简化运算是行不通的。比如，当 n=100，每一个 a_i 都是 32 比特长，那么 y 至多是 39 比特长，而且 f 是高度退化的，而且平均只要求 2^{38} 次运算就能找到一个合适的解。一般地，如果 $a_i = 2^{i-1}$，那么求解 f 就相当于寻找 y 的二进制分解。

这个例子说明了当代计算复杂度理论的有用之处以及还存在着的巨大的不足之处。这个理论仅仅告诉我们背包问题在最坏情况下求解是相当困难的。它没告诉我们它的任何一种特殊的排列方式的复杂度。但是，从 $\{0,1,2,\cdots,2^{n-1}\}$ 选择合适的一组 $\{a_i\}$，当 $n \to \infty$ 时，这的确是一个难题。

算法分析学中的另一个令人感兴趣的潜在的单向函数就是乘幂模 q 运算，它是由斯坦福大学的 Prof . John 提出来的。这个单向函数已经在第三部分讨论过了，这里就不再诉说了。

7 历史回顾

本文所讨论的公钥密码系统和单向认证系统起初看起来不像是古典密码学发展的结果，其实它可能就是密码学几百年来自然发展的趋势。

保密是密码学的核心。在早期的密码学中，我们混淆了密码系统中什么该保密，什么不该保密。像凯撒密码（它的加密过程是将 26 个英文字母循环后移 3 位，即 A 变为 D，B 变为 E，等等）之类的密码系统，安全性是依赖于算法过程的保密性的。随着电报机[2，p.191]的发明,出现了使用特殊密钥的密码系统，它与一般密码系统差别在于：前者容许将其算法泄露，比如，即使不小心将加密工具泄露，那么只需将密钥换成一个新密钥就可以了。Kerchoffs[2，p.235]在 1881 就提出了这个观点，即泄露一个密码系统的加密方法不影响系统的安全性。大约在 1960 年，可以抵御已知明文攻击的密码系统就已经在实际中应用了，这样就无须对以前的信息保密了，减轻了用户负担。公钥密码系统是减少保密内容的系统的自然延续。

直到这个世纪初，密码系统的计算量都被限制在用手工或简单的计算尺之类的工具就能完成的阶段。世界大战时期，密码学开始了变革，到今天已经得到了完善。专用计算机被用来完成加密工作。加密运算由过去的被限制在必须能用简单的电动机械设备完成，到今天的能由通用数字硬件产生。数字计算机的发展将密码学的加密运算从齿轮设备限制中解脱出来，而且容许我们在符合纯粹的密码学标准的前提下去寻找更好的加密算法。

用数学证明去论证密码系统的稳固性的做法被认为是错误的。上个世纪，Kerchoffs[2，p.234]利用这种错误理论，制定了一个密码分析攻击的范例。虽然密码学的一些通用规则已经得到了完善，这有助于防止系统设计者犯一些低级错误，但是，系统最终的测试必须是能抵制一些有经验的分析者在有利的条件下（比如，选择明文攻击）的的攻击。计算机的发展有史以来第一次让人们能够通过算法的数学理论来估计破解一个密码系统所需的计算难度。用数学工具证明系统的安全性又被证明是最好的方法了。

密码学发展史上一个有趣的现象就是区分业余爱好者和专业密码人员。密码分析方法大多是由专业人事发现的，而密码算法则主要是由非专业人员提出的。Thomas Jefferson，一个密码学业余爱好者，在第二次世界大战时期发明了一个至今还在使用的加密算法[2，pp.192-195]；20 世纪最著名的加密算法，轮转机算法，是由四个不同的人同时发明的，这四个人也全是业余爱好者[2，pp.415，420，422-424]。我们希望这些例子能鼓舞其他人也参加到这个迷人的领域的研究中来，而不再像过去的几年一样，几乎成为政府垄断行业。

参考文献

[1] R. Merkle, “Secure communication over an insecure channel,” submitted to *Communications of the ACM.*

[2] D. Kahn, *The Codebreakers, The Story of Secret Writing.* New York: Macmillan, 1967.

[3] C. E. Shannon, “Communication theory of secrecy systems,” *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.

[4] M. E. Hellman, “An extension of the Shannon theory approach to cryptography,” submitted to *IEEE Trans. Inform. Theory*, Sept. 1975.

[5] W. Diffie and M. E. Hellman, “Multiuser cryptographic techniques, presented at National Computer Conference, New York, June 7–10, 1976.

[6] D. Knuth, *The Art of Computer Programming, Vol. 2, Semi-Numerical Algorithms.* Reading, MA.: AddisonWesley, 1969.

[7] ———, *The Art of Computer Programming, Vol. 3, Sort ing and Searching.* Reading, MA.: Addison-Wesley, 1973.

[8] S. Pohlig and M. E. Hellman, “An improved algorithm for computing algorithms in *GF*(*p*) and its cryptographic significance,” submitted to *IEEE Trans. Inform. Theory.*

[9] M. V. Wilkes, *Time-Sharing Computer Systems.* New York: Elsevier, 1972.

[10] A. Evans, Jr., W. Kantrowitz, and E. Weiss, “A user authentication system not requiring secrecy in thecom-puter,” *Communication of the ACM*, vol. 17, pp. 437–442,Aug. 1974.

[11] G. B. Purdy, “A high security log-in procedure,” *Communication of the ACM*, vol. 17, pp. 442–445, Aug. 1974.

[12] W. Diffie and M. E. Hellman, “Cryptanalysis of the NBS data encryption standard” submitted to *Computer*, May 1976.

[13] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Reading, MA.: Addison-Wesley, 1974.

[14] R. M. Karp, “Reducibility among combinatorial problems,” *Complexity of Computer Computations.* R. E. Miller and J. Thatcher, Eds. New York: Plenum, 1972, pp. 85–104.