现代数学译丛 1

椭圆曲线

及其在密码学中的应用—导引

〔德〕Andreas Enge 著

吴铤　董军武　王明强 译

科学出版社
www.sciencep.com

(O-2913.0101)

ISBN 978-7-03-020034-1

9 787030 200341 >

定 价：38.00 元

现代数学译丛 1

椭圆曲线及其在密码学中的应用——导引

〔德〕Andreas Enge 著
吴 铤 董军武 王明强 译

科学出版社
北京

图字：01-2006-3955 号

内 容 简 介

本书以介绍椭圆曲线在密码学中的应用为目标，用浅显易懂的语言全面讲述了椭圆曲线公钥密码的相关知识，包括公钥密码学概述、有限域上椭圆曲线的算术理论、椭圆曲线上离散对数的求解算法以及有限域上椭圆曲线的求解算法等.

本书最突出的特点在于只利用近世代数等基础知识来揭示椭圆曲线内在的代数和几何结构，所以特别适合作为研究生和高年级本科生等初学者了解、掌握椭圆曲线公钥密码理论的入门书籍，也可供相关研究人员参考.

图书在版编目(CIP)数据

椭圆曲线及其在密码学中的应用——导引/(德)恩格(Enge, A)著；吴铤，董军武，王明强译. —北京：科学出版社，2007
(现代数学译丛)
ISBN 978-7-03-020034-1

I. 椭… II. ①恩… ②吴… ③董… ④王… III. 椭圆曲线-应用-密码-理论 IV. TN918.1

中国版本图书馆 CIP 数据核字(2007) 第 174763 号

责任编辑：陈玉琢 莫单玉 / 责任校对：陈玉凤
责任印制：赵德静 / 封面设计：王 浩

科 学 出 版 社 出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

源海印刷有限责任公司 印刷
科学出版社发行 各地新华书店经销
*
2007 年 12 月第 一 版 开本：B5(720×1000)
2007 年 12 月第一次印刷 印张：11 1/2
印数：1—3 000 字数：212 000

定价：38.00 元

(如有印装质量问题，我社负责调换<长虹>)

译者的话

随着计算机技术、网络技术，特别是 Internet 技术的飞速发展和广泛普及，人类社会已经进入了信息化时代. 如何在开放的网络环境中实现信息的保密性、完整性、可用性、可控性以及抗抵赖性已经成为了人们关注的焦点问题. 可以说信息安全问题已经成为制约信息化进程的主要瓶颈之一. 现代密码学作为解决信息安全问题的主要手段，其重要性得到了广泛认同.

自 1949 年, Shannon 发表的《保密通信的信息理论》将密码学研究纳入科学轨道以来，现代密码学基本可分为以 DES 为代表的对称密码学和以 RSA 为代表的公钥密码学. 由于公钥密码体制的非对称结构，从而在一定程度上克服了对称密码体制需要利用秘密通道来传递密钥这一弱点. 同时也正是由于其非对称结构，使公钥密码学不仅可以应用于数据加密，还可以被广泛地应用于身份识别、数字签名、密钥协商和电子支付等诸多领域，所以自 1976 年 Diffie 和 Hellman 提出公钥密码思想以来，公钥密码就引起了人们的广泛重视，在现代密码学中占据重要地位.

根据公钥密码体制所基于的数学难题来分类，目前有以下三类系统被认为是安全有效的：基于大整数分解问题的 RSA 型公钥密码；基于有限域上离散对数问题的 ElGamal 型公钥密码; 基于椭圆曲线离散对数问题的椭圆曲线公钥密码. 与其他公钥密码体制相比，椭圆曲线公钥密码体制突出的优势在于：对于其所基于的数学难题 —— 椭圆曲线离散对数问题 —— 目前并不存在亚指数时间算法，从而能够以更小的密钥尺寸来满足相同的安全性要求. 通常认为，为得到合理的安全性，RSA 应当使用 1024 比特的模长，而对于椭圆曲线密码体制，只需要使用 160 比特的模长. 而较小的密钥尺寸又带来了运行速度快、存储空间小、传输带宽要求少等诸多优点，这些优点使其特别适用于计算能力、存储能力、带宽受限，但又要求高速实现的应用领域，例如智能卡、无线通讯等. 椭圆曲线密码体制的以上优点使其受到国际上的广泛关注，这已经对 RSA, ElGamal 等公钥密码体制形成强劲的挑战.

另一方面，椭圆曲线公钥密码是以有限域上椭圆曲线的深刻理论为基础，涉及了包括代数数论、代数几何等许多数学分支，这给椭圆曲线公钥密码的普及与应用带来了一定的困难. 本书作者以椭圆曲线的密码应用为目标，通过浅显易懂的语言全面介绍了椭圆曲线公钥密码的相关理论. 本书的一个显著特点是：学习本书只需要大学里介绍的近世代数知识，而利用这些基础知识, 本书揭示了椭圆曲线内在的代数和几何结构，同时通过本书的学习就可以了解目前最新的研究进展，从而适合

作为信息安全研究人员，特别是研究生和高年级本科生等初学者了解、掌握椭圆曲线公钥理论的入门书籍.

本书的第 1 章对公钥密码学进行了概述性的描述；第 2 章介绍了椭圆曲线上的群运算法则，并揭示了其与除子类群之间的内在联系；第 3 章详细介绍了有限域上的椭圆曲线；第 4 章介绍了离散对数问题的各种求解方法，包括针对椭圆曲线离散对数问题的 MOV, Xedni 等多个攻击方法；第 5 章在重点介绍椭圆曲线点数计算方法 ——Schoof 算法的同时，描述了 SEA 算法的基本思想以及计算流程.

原书的序言、前言和第 1 章由王明强翻译，第 2 章、第 3 章、第 4 章由吴铤翻译，第 5 章由董军武翻译. 同时译者根据原书勘误说明对原书进行了相应的修改. 在本书的翻译过程中，得到了清华大学王小云教授以及译者同学的帮助，特此感谢！由于译者的专业知识和外语水平所限，书中错误与不妥之处在所难免，敬请读者批评指正.

本书的翻译和出版得到了国家“973”项目 (项目编号：2007CB807900, 课题编号：2007CB807902) 的资助，特此感谢！

译　者

2006.10.24

序　言

自 1976 年 Diffie 和 Hellman 提出公钥密码算法以来，许多公钥密码方案应运而生. 在通常情况下，几乎所有方案的安全性都是基于数学上的 "困难" 问题. 特别是大整数分解以及离散对数问题是几个最著名方案的安全核心.

公钥密码技术正广泛地应用于网上电子支付、无线股票交易以及在智能卡上的应用等多个商业安全领域. 其中比较著名的是 RSA 方案与 DSA(数字签名算法). RSA 的安全性基于大整数分解问题，而 DSA 的安全性则基于有限域乘法群中的离散对数问题. 以上这两个问题都存在亚指数时间算法. 这意味着在实际应用中为了获得足够的安全性，所使用的密钥长度必须超过 1000 比特. 由此可见，在能耗、存储空间以及带宽受限的许多应用领域中，就无法利用以上的技术来建立实用的公钥密码体制.

1985 年，Neal Koblitz 与 Victor Miller 独立地提出了椭圆曲线密码算法. 该算法的安全性是基于有限域上椭圆曲线点群中的离散对数问题. 到目前为止，求解椭圆曲线离散对数问题的最佳算法是指数时间算法. 由此就可以用较小的密钥长度来达到与原来相同的安全性要求，从而可以将椭圆曲线密码算法应用于上述的受限应用领域. 经过过去十几年的发展，椭圆曲线密码算法已经得到广泛的应用，并且诸如 ANSI, IEEE 与 ISO 等机构正在将椭圆曲线密码算法进一步地加以标准化. 在 1999 年 1 月，DSA 的椭圆曲线版本 (ECDSA) 成了美国金融机构的 ANSI9.62 标准.

椭圆曲线密码算法是以有限域上椭圆曲线的深刻理论为基础. 据我所知，很少有书介绍这些基本理论，而以密码学应用为目的来介绍这些基本理论的书就更少了. 在本书中，Andreas Enge 用简单透彻但是易懂的语言介绍了这些基本理论. 在为高年级的本科生讲授椭圆曲线密码课时，我就用了这本书的初稿作为教材我也曾鼓励他将书稿出版. 现在，我为 Andreas 出版这本书而感到高兴. 我坚信对于那些想研究有限域上的椭圆曲线理论及其在密码学上的应用的人来说，本书是一本非常好的入门书籍.

S. A. Vanstone

前　言

在过去的 20 年里，公钥密码体制的诞生连同计算机科学技术的出现为一直以来都被看作是“纯粹”数学分支的数论和代数几何开创的一个新的应用领域. 椭圆曲线是现代密码学中最有发展前途的工具之一. 这进一步引起了数学工作者以及关注新密码算法实现的工程师和计算机科学家对这一领域的研究兴趣.

我们的目标是为那些学习椭圆曲线一般理论的读者提供一本入门教科书，为进一步学习更深刻的理论知识打下基础. 学习这本书只需要大学里讲授的近世代数知识. 读者只需了解多项式环、域的扩张和有限域的基本理论，通过本书的学习就能了解到目前最前沿的研究课题，如椭圆曲线上点的个数问题. 这个问题直到最近几年才被完全解决.

虽然椭圆曲线的其他应用，如大整数分解或素性证明，只涉及素域上的椭圆曲线，但是在密码学中特别感兴趣的是特征值为 2 的情况. 本书着重于对椭圆曲线进行一般性的描述，兼顾了特征值为奇数或偶数的情况，并且只有当需要的时候才对特征值的不同情况加以区分.

这里要非常感谢的是 Reinhard Schertz，正是他在大学里精彩的讲座引起了我对椭圆曲线的兴趣. 还要感谢 Dieter Jungnickel，是他建议我研究这个课题并且指导我完成了论文，并以那篇论文为基础最终形成了本书. 感谢 Leonard Charlap 与 David Robbins，他们精彩的报告是这本书的基础. 同时非常感谢 Marialuisa de Resmini 与 Scott Vanstone，正是由于他们的鼓励，才使本书得以出版. 在此还要对 Drik Hachenberger, Dieter Jungnicked, Charles Lam 与 Berit Skjernaa 说声谢谢，他们阅读了本书的初稿并且提出了很多有益的建议.

希望读者在阅读本书时能像我在写本书时那样获得很多乐趣.

Anderas Enge

目　录

第 1 章　公钥密码算法

随着电子网络在现代经济社会中的广泛应用，密码学的应用已不局限于专门的军事和保密机构，而成为了一个公众关注的话题，诸如 UNO ([UNCITRAL, 1998a] 和 UNCITRAL [1998b]) 与 EU ([Commission of the European Communities, 1998]) 等国际组织也对密码学的应用表示出高度的关注. 与传统的密码相比，公钥算法应用范围更加广泛. 利用公钥算法大体上就可以实现世界上任何两人之间安全、可信的通信. 在下面的章节中，我们简要介绍公钥密码学的许多概念与基本思想，并给出一些具体算法. 我们将着重介绍加密和数字签名的体制. 这些体制可以被推广到任意群上，特别是可以推广到椭圆曲线的点群上. 文献 [Stinson, 1995] 和 [Menezes et al., 1997] 对密码算法有综合全面的论述.

1.1　私钥密码学与公钥密码学

密码学是一门在一个公开信道上传递信息的艺术，其应当至少满足保密性与真实性这两个基本安全性要求. 保密性意味着即使攻击者能够窃听到所传送的信息，他也不能恢复出相应的明文消息；而真实性意味着消息发送者的身份与所发送消息的完整性是可以验证的. 我们首先讨论如何实现保密性，在 1.4 节我们对消息的真实性加以论述.

一般来说消息都以数字化的形式通过电子网络进行传送，如电子邮件、协议以及技术计划书、软件甚至人的声音等. 为了防止攻击，确保发送消息内容的安全，原始数据必须首先通过数学运算将其变成随机的形式，然后再加以传送.

因此，第一步是将消息转化成某种数学形式. 一般地，它们将被分成固定长度的组，每一组又被转化成一个整数或者一个比特串. 这一编码过程只是一种技术上的处理，它不能确保消息的安全. 对这一过程我们不做详细的描述，我们仍然称转化后的组为 “消息”.

第二步是用某种方式将每组数据进行进一步的转化，使得未经授权者不能恢复出原始数据. 转化后的数据传送至指定的接受者，该接受者进行逆转化并按一定的规律重新组合恢复出原始数据. 粗略地来说，这种数据转化的算法就形成了一个密码系统.

严格来讲，一个*密码系统*由三个有限集合 $\mathcal{M},\mathcal{C}$ 和 $\mathcal{K}$ 以及一族*加密函数* $f_k:\mathcal{M}\to\mathcal{C}$ 构成，其中 $\mathcal{M},\mathcal{C},\mathcal{K}$ 分别表示明文空间、密文空间以及密钥空间，$k\in\mathcal{K}$. 换句话说，$\mathcal{M}$ 就是前面所说的原始数据集合，$\mathcal{C}$ 就是转化后的数据集合. 为了使加密和解密成为可能，f_k 必须能够有效计算并且是单射. 在许多情况下，$\mathcal{M}=\mathcal{C}$ 且 f_k 是双射.

当 Kevin 想给 Laura 发送一个秘密消息 $m\in\mathcal{M}$ 时，他首先选择一个密钥 $k\in\mathcal{K}$，然后计算密文 $c=f_k(m)$ 并且将该密文通过一个可能不安全的信道发送给 Laura. Laura 必须对密文 c 应用加密函数的逆函数 ——*解密函数* f_k^{-1}，以获得原始消息 $m=f_k^{-1}(c)$.

在通常的*私钥密码系统*中，f_k 或者 f_k^{-1} 与密钥 k 是等价的. 这意味着一个有能力加密的人也能完成解密，反之亦然. 因此上面的方法有两个方面的主要缺陷：

密钥分发问题　任何两个想进行秘密通信的成员必须事先确定一个共同的密钥. 为此就必须事先通过一个安全信道来交换该共同密钥，例如他们可以事先见面商量，抑或使用一个可信赖的信使或者其他安全的方法. 该安全信道的建立往往要比用来传递后续消息的不安全信道要昂贵得多. 同时为了保证通信有较高的安全性，就必须经常改变这个密钥. 而这就增加了整个系统的运行成本. 与此同时，在一个有多个成员的网络中，整个系统的密钥数量大致是该系统成员数量的平方. 这样就会给密钥管理带来极大的麻烦.

签名问题　为了确保通过电子网络达成的协议的合法性，秘密消息的接收者必须能够向第三者 (如法官) 证明发送者的身份. 由于在传统的密码系统中，一个能解密密文的人也能加密任意的消息，因此对接受者来说，伪造一个自主选择消息的密文是没有任何问题的.

1976 年, Diffie 与 Hellman 提出了解决这些问题的一个方法. 该方法的提出给密码学带来了革命性的影响 [Diffie and Hellman, 1976]. Diffie-Hellman 的方法是以所谓的*单向函数*为基础，或者更精确地说是*单向陷门函数*. 假如每一个密钥 k 对应的加密函数 f_k 满足 "即使别人知道 f_k，他也是不可能计算 f_k^{-1}"，那么 Kevin 就能公布他的加密函数，即所谓的*公钥*. 这样任何人 (包括 Laura) 都能发送秘密的消息给 Kevin. 但是此时即使是 Kevin 也不能解密他收到的密文，这就限制了该密码体制的用途 (但是这种体制广泛应用在身份鉴别的协议中，身份鉴别一般存储一个加密口令). 这就是需要引入陷门函数的原因：对于单向陷门函数 f_k 来说，只要知道*秘密密钥*或者是*陷门* k，就很容易从 f_k 出发确定 f_k^{-1}. 如果 Kevin 知道密钥 k，那么他就能解密他所得到的密文.

这种处理办法解决了上面所述的两个问题：分发密钥时不再需要秘密信道. 相

反我们须将公钥发布在一个显著的地方以便两个陌生人之间实现保密通信. 如果 Kevin 想对一个发送给 Laura 的消息 m 进行签名 (这个消息可能是他们之间合同的一部分), Kevin 首先将解密函数 f_k^{-1} 作用在 m 上, 然后将 $(m, f_k^{-1}(m))$ 发送给 Laura. 由于 Kevin 是唯一知道解密函数 f_k^{-1} 的人, 因此 Laura 能通过比较 m 是否与 $f_k(f_k^{-1}(m))$ 相等来向任何第三方证明 $(m, f_k^{-1}(m))$ 的确来自于 Kevin. 如果所发送的消息还需要保密的话, 就可以用 $c = f_l(m)$ 来代替 m, 其中 f_l 是 Laura 的公钥.

文献 [Diffie and Hellman, 1976], p.648 中用诸如 "易于计算" 或者 "计算不可行" 这种非正式的语言来描述单向陷门函数, 除此以外对于单向陷门函数似乎不存在一个一般的可以接受的定义. 当我们系统地阐述这种函数所要满足的最低要求时, 就会看到产生这一问题的原因是明显的. 因为在复杂度理论中我们假设 "易于计算" 就是 "可以利用确定性算法在多项式时间内完成计算", 那么知道密钥 k 就可以在多项式时间内确定 f_k 和 f_k^{-1}. 同时如果已知函数 f_k 和 f_k^{-1}, 就可以在多项式时间内完成 $f_k(m)$ 和 $f_k^{-1}(c)$ 的计算. 另一方面, 即使已知 f_k(由此对任意的 m, $f_k(m)$ 就是已知的), 也不可能在多项式时间内确定 f_k^{-1} 或 $f_k^{-1}(c)$. 但是满足这些合理需求的单向陷门函数是否存在还是不确定的: 很明显存在一个非确定性的多项式时间算法来求 $f_k^{-1}(c)$, 也就是猜测一个 m 并且验证 $f_k(m) = c$ 是否成立. 如果利用确定性算法在多项式时间内可解的问题与利用非确定性算法在多项式时间可解的问题是一致的话, 即 $P = NP$, 那么就不存在单向陷门函数. 而 $P = NP$ 是否成立, 这是复杂性理论的一个重要的公开问题.

在实际应用中, 单向陷门函数满足即使在计算能力非常有限的情况下也能在 "合理的" 时间完成计算的函数, 而利用目前已知的最佳算法, 在计算能力非常强的情况下也不能在合理的时间内计算出它的逆函数. 当然这个定义服从于用户需要. 保密机构可以采用与个人用户不同的标准. 目前已经设计出许多单向陷门函数, 我们将在下面几节中介绍其中的几个.

在应用加密函数时, 经常会对 $\mathcal{M}$ 或者 $\mathcal{C}$ 中的数学对象进行运算. 由于指定消息的接受者在接到消息时必须对其进行逆变换, 因此数学对象的这些基本运算也应具有类似的性质. 一般地, 我们选择 $\mathcal{M} = \mathcal{C}$ 为群或者是只有极少数元素不可逆的幺半群 (通常来说, 元素的不可逆性将导致密码体制被破解, 因此这种情况发生的概率必须是可忽略的). 这里我们只对群的情况进行讨论, 并且在下面各节中用一般乘法群中的符号对算法进行描述, 因此椭圆曲线上的点群只是其中一种特殊情况.

1.2 Diffie-Hellman 密钥交换协议

密钥交换协议是 Diffie 和 Hellman 设计的. 该算法还不完全是公钥密码算法,

Diffie 和 Hellman 将其归类为 "公钥分配协议"([Diffie and Hellman, 1976], p.468). 这个协议的思想是在不安全的信道上只交换部分信息，以使后来双方能共享一个共同的密钥. 同时即使攻击者碰巧获得这个部分信息，其也不能构造出这个共享的密钥. 该共享密钥能被用在传统的密码体制之中. 具体描述如下：

1. Kevin 与 Laura 公开地选择一个循环群 G 及其生成元 α . (在原始文献里 G 就是有限域的乘法群.)
2. Kevin 与 Laura 分别随机地选择整数 k 和 l 作为各自的密钥. 然后分别计算 α^k 和 α^l 并交换计算结果.
3. Kevin 与 Laura 利用各自获取的信息以及各自的密钥计算

$$\alpha^{kl} = (\alpha^k)^l = (\alpha^l)^k,$$

α^{kl} 就是共享的密钥.

注意到 α 的幂次甚至是较高的幂次都能够通过 "平方–乘" 算法得以有效地计算.

算法1.1 ("平方–乘" 算法)　设 α 是群中的一个元素并且 k 是一个自然数. 下面算法用 $O(\log k)$ 次群运算就能计算出 $\gamma = \alpha^k$.

1. 令 $\gamma = 1$.

2. 重复下面的步骤直到 $k = 0$.

3. 如果 k 是奇数，则用 $k-1$ 代替 k, $\gamma\alpha$ 代替 γ, 这样总可假设 k 是偶数. 用 $\frac{k}{2}$ 代替 k, α^2 代替 α.

证明　在这个算法的运行过程中，值 $\gamma\alpha^k$ 是一个不变量，因此当 $k = 0$ 时 γ 包含所要的结果. 这就证明了这个算法的正确性. 如果 k 的两进制表示长度是 r, 也就是 $2^{r-1} \leqslant k < 2^r$, 并且 $s \leqslant r$ 表示非零比特的个数，那么该算法恰好需要 $r-1$ 次平方运算和 $s-1$ 次乘法运算. 由此我们就给出了该算法的复杂度证明. □

当这个群是交换群时，在上面算法中我们可以用加法符号代替乘法符号，那么 "平方 - 乘" 算法就变成了双倍加算法. 这种算法早在古埃及时期就在整数乘法计算中使用, 参见 [Gillings, 1972].

一个窃听者在获取 Diffie-Hellman 密钥交换协议中传递的信息之后，如果想要恢复出密钥的话，就要从 α, α^k 和 α^l 中计算出 α^{kl}. 这个问题是著名的 Diffie-Hellman问题 . 求解该问题的一个明显方法是从 α^k 中计算 k, 这就是计算以 α 为底的 α^k 的离散对数 . 注意到 k 是在模 G 的阶的意义下是确定的，只要知道满足 $\alpha^{k'} = \alpha^k$ 的 k' 就能计算 $\alpha^{kl} = (\alpha^l)^{k'}$.

虽然到目前为止，不知道 Diffie-Hellman 问题与离散对数问题计算上是否等价，但是人们广泛认为这一结论应该是正确的. (实际上, 对于很大一部分有限群来说, Maurer 与 Wolf 已经证明它们之间的等价性, 参见 [Maurer and Wolf, 1996].) 如果等价的话，Diffie-Hellman 体制的安全性就是建立在离散对数问题困难性的基础上. 离散对数问题的困难性也与群的表示方法有关：如果模 n 的循环群 $G=(\mathbb{Z}_n,+)$ 被表示成 $\{0,\cdots,n-1\}$ 并且 $\alpha=1$，那么离散对数问题的求解是容易的. 我们将在第 4 章介绍离散对数问题. 要注意的是对于利用多项式或者正规基表示的有限域的乘法群，目前还不存在求解离散对数问题的多项式算法.

1.3 ELGAMAL 密码体制

在 Diffie-Hellman 密钥交换协议的基础上，ElGamal 在文献 [ElGamal, 1985] 中设计了一个真正意义上的公钥密码体制. 设 G 是循环群，α 为其生成元；$\mathcal{M}=G$, $\mathcal{C}=G\times G$. 每一个成员各自选择一个私钥 $a\in\mathbb{Z}$ 并公布 α^a. 假设 Kevin 希望传送一个消息 m 给 Laura.

1. Kevin 随机选择一个整数 k 并查看 Laura 的公钥 α^l.
2. Kevin 计算 $\alpha^{kl}=(\alpha^l)^k$，并发送 $(\alpha^k, m\alpha^{kl})$ 给 Laura.
3. Laura 利用自己的私钥 l 计算 $\alpha^{kl}=(\alpha^k)^l$ 并由此恢复出消息 m.

这个体制显然与 Diffie-Hellman 密钥交换协议等价. 这里以通过公开 α^l 的方式减少了一次数据的交换. 该体制的一个缺点是*消息扩展率*为 2：为了将群中某个一个元素所蕴涵的信息传送给对方，就必须传递两个群元素. 如果 Kevin 在每次发送消息时都使用同一个公钥 α^k，那么这种情况就可以避免，即 Kevin 只要传送这个公钥 α^k 一次即可. 但是这种简化有一个安全缺陷：如果偷听者已知某一组消息 – 密文对 $(m_1, m_1\alpha^{kl})$，他就可以针对下一个密文 $m_2\alpha^{kl}$ 通过计算

$$m_2=m_1\frac{m_2\alpha^{kl}}{m_1\alpha^{kl}}$$

恢复出明文 m_2. 一个合理的折衷方案是每一次 "会话" 时都采用不同的密钥 k. 而一次会话 (如 e-mail) 内容通常都由几个连续的信息 $m_1, m_2, \cdots, m_n$ 组成，这样加密后的数据由 α^k 和 $m_1\alpha^{kl}, m_2\alpha^{kl}, \cdots, m_n\alpha^{kl}$ 组成，从而将消息扩展率降为 $1+\dfrac{1}{n}$.

注意到与 1.1 节介绍的一般概念相比，在该体制中有一点是不对称的，从而使得签名变得不可能：因为 Laura 可以自己任意选择一个 α^k 或利用 Kevin 的公钥 α^k，来生成任意明文 m 的有效密文. ElGamal 有一个不同的签名方案，我们将在下一节加以阐述.

1.4 签名方案

在这一节里, 我们给出几个重要的签名方案, 其具有前面所述的性质. 假设 Kevin 想发送一个签了名的消息 m 给 Laura, 那么他利用自己的密钥 k 以及明文, 在 m 上附加上一个 "签名". Laura 通过验证这个签名来证明 Kevin 是真正的发送者, 并且消息在发送过程中并没有被篡改过.

ElGamal 签名方案

文献 [ElGamal, 1985] 在给出加密体制的同时, 提出了如下的签名方案. 其安全性依赖于群 G 上的 Diffie-Hellman 问题, 其中 G 是以 α 为生成元的循环群. 假设 $g : \mathcal{M} = G \to \{0, \cdots, |G| - 1\}$ 是一个可有效计算的双射, m 为待签明文. Kevin 的私钥与公钥分别是 k 和 α^k.

签名

1. Kevin 随机选择一个与 $|G|$ 互素的整数 k', 并且计算 $r = \alpha^{k'}$.
2. Kevin 求解同余方程

$$g(m) \equiv kg(r) + k's (\mathrm{mod}|G|). \tag{1.1}$$

由于 k' 与 $|G|$ 互素, 因此同余方程存在唯一解 $s \in \{0, \cdots, |G| - 1\}$.

3. 签名就是数对 $(r, s) \in G \times \mathbb{Z}_{|G|}$, 该签名数对与 m 一起被发送给 Laura.

验证签名

1. Laura 根据 Kevin 的公钥 α^k 以及 m, r, s, 计算 $\alpha^{g(m)}$ 和 $\alpha^{kg(r)+k's} = (\alpha^k)^{g(r)} r^s$.
2. 如果第一步计算出的两个值相等, 那么他就认为这个签名是有效的.

对于这个签名方案安全性的分析参见原始文献 [ElGamal, 1985], pp.470–471. 非常关键的一点是: 对于不同的消息应当选择不同的随机数 k', 否则利用两个明文及其对应的签名就能通过求解由 (1.1) 形成的线性方程组, 来获取 k' 以及 Kevin 的私钥 k.

这个方案的一个主要缺陷是传输数据长度的扩展 —— 被签消息的长度是原消息长度的三倍. 这个被扩展的数据可以利用一个 hash 函数来压缩. 在实际应用中, 一次会话中被传送的数据由若干 $m_1, m_2, \cdots, m_n$ 组成. 而 hash 函数就是函数

$$h : \mathcal{M}^{(\mathbb{N})} \to \mathcal{H},$$

这里 $\mathcal{M}^{(\mathbb{N})}$ 表示由 $\mathcal{M}$ 中元素构成的有限序列集合，$\mathcal{H}$ 是一个有限集合. 我们可以对 $h(m_1, m_2, \cdots, m_n)$ 进行签名而不是对每一个 m_i 分别签名. 为了防止伪造签名，h 必须是一个单向函数.

数字签名标准

1994 年，美国国家标准技术局 (NIST)公布了一个数字签名的标准 (DSS)，美国国家机构必须执行这个签名的标准 (参见 [NIST, 1994]). 除了描述特定 hash 函数的使用以外，DSS 给出了一个数字签名算法 (DSA). (在更新后的版本中也允许使用 RSA 签名算法，参见 [NIST, 1998].) 通常的设置如下：

1. p 是一个素数并且满足 $2^{L-1} < p < 2^L$，其中

$$L \in \{512, 576, 640, 704, 768, 832, 896, 960, 1024\}.$$

2. q 是 $p-1$ 的一个素因子，并且满足 $2^{159} < q < 2^{160}$.
3. $\alpha \in \mathbb{F}_p$ 是群 $\mathbb{F}_p^\times$ 中唯一阶为 q 的子群的生成元. 该签名算法是基于群 $\langle\alpha\rangle$ 中的离散对数问题.
4. 函数

$$g : \mathbb{F}_p^\times = \{1, \cdots, p-1\} \to \{0, \cdots, q-1\}$$

表示模 q 约化：对于 $\alpha \in \{1, \cdots, p-1\}$，

$$g(\alpha) \equiv \alpha \pmod{q}.$$

5. $h : \mathcal{M}^{(\mathbb{N})} \to \mathbb{Z}$ 是由安全 hash 函数标准 (SHS) 来确定的 hash 函数，该标准可参见 [NIST, 1995]. Kevin 的私钥是整数 k，$0 < k < q$，其公钥是 α^k. 假设 m 是待签消息.

签名

1. Kevin 随机选择一个整数 k'，$0 < k' < q$，并计算 $r = \alpha^{k'}$ 与 $g(r)$.
2. Kevin 求解同余方程

$$h(m) \equiv -kg(r) + k's \pmod{q} \tag{1.2}$$

的解 s，$0 < s < q$.
3. 签名就是数对 $(g(r), s) \in \mathbb{Z}_q \times \mathbb{Z}_q$.