ElGamal 数字签名方法

大连财贸职工学院 邓国民
海军大连舰艇学院 薛化文

本文介绍了一种新的数字签名方法——ElGamal 公钥密码体制，阐述了实现的方法，分析了其安全性及可能的攻击方法，与其它公钥密码体制的数字签名方法进行了比较，最后介绍了实现中注意的问题。

一、概 述

众所周知，一个人想证明自己身份可以用他的印章或手写签名，一个单位呢可以用公章，在信息高度电子数字化的今天，难道我们可以在一篇数据文件上盖印章吗？回答是肯定的，这就是要使用数据签名技术。数字签名技术作为计算机数据安全的一项重要手段，现在正被广泛应用，电子邮件(E-mail)、电子资金转帐(EET)、电子数据交换(EDI)和软件分发等方面，都要使用数字签名技术，随着计算机网络的应用越来越普及，网络对等实体的识别、通信保密和数据完整性显得越来越重要，而确实解决这一问题则必须要使用数字签名技术。

在1976年公钥密码体制没有发明以前，人们使用传统的密码技术解决数据交换中的安全问题，一个人能使用密码加密一个文件给另外一个人，那么另外的那个人必须要利用解密密钥才能读懂加密过的文件，这时通信双方的身份和文件的完整性都随着文件的解密是否成功而不言而喻。1976年，W. Diffie 和 M. Hellman 发明了公钥密码体制，美国公布了数据加密标准(DES)，标志现代密码技术到达了一个新的阶段，数字签名技术也得到了飞速地发展，采用私钥密码体制(DES)技术和公钥算法(如 RSA 等)都可实现数字签名。由于私钥密码体制本身的特性，使用私钥的数字签名体制在存储和通信上花费都较大，故一般不考虑使用。而由于公钥密码体制的特点，其非常适合应用于签名技术。因此现在公布的几种数字签名建议标准中都使用的是公钥密码体制。如 ISO 建议过的 RSA 算法；美国国家标准与技术研究所(NIST)的建议标准 DSS (DigitalSignature Standard)，也是一种从下面我们介绍的 ElGamal 公钥密码体制演变而来的。

二、ElGamal 数字签名方法

ElGamal 公钥密码体制是1984年 Stanford 大学的 Tather ElGamal 提出的，这是基于离散对数运算基础上的公钥体制，它采用的是 Diffie-Hellman 密钥分配体制的思想，综合了其它一些加密体制的优点。利用 ElGamal 公钥密码体制设计出的数字签名方法，具有与一般公钥密码体制签名方法的不同之处，具有高安全性和实用性。

1. ElGamal 数字签名算法

下面叙述 ElGamal 数字签名算法；讨论其安全性，列举出可能的破译方法和伪造手段；最后，分析该方法与其它公钥签名方法的主要的相同点和不同点。

ElGamal 公钥系统是基于 Diffie-Hellman 的密钥分配体制思想。Diffie-Hellman 体制是指：在基于素数的有限域 GF(p)上，其中 p 是一素数，g 是其本原元，A、B 两方想要通信，首先要得到双方都知道的一个通信密钥 KAB，假设 A、B 各持有一私用密钥 XA、XB，A 可计算 $YA=g^{XA} \bmod p$，将 YA 发往 B；B 可计算 $YB=g^{XB} \bmod p$，将 YB 发往 A。然后双方同时计算 $KAB=YA^{XB} \bmod p=YB^{XA} \bmod p(=g^{XAXB} \bmod p)$。

ElGamal 密码体制也是通信双方各持有一私用密钥 XA、XB，并将 $YA(=g^{XA} \bmod p)$、$YB=(g^{XB} \bmod p)$作为公开密钥公布。双方加解密的方法是通过生成一组随机数混入明文块中以产生一二维密文数组，以达到随机密码的效果。

ElGamal 数字签名方法则主要是让接收者验证一个等式：$g^m=y^r r^s \bmod p$，这里(m,r,s)是签名者发给接收者的，y 是签名者的公开密钥。(设其私用密钥为 x，则 $y=g^x \bmod p$)。具体算法如下：

1. A 签发一条明文 m 给 B：

a、A 选择一随机数 k，$0<k<p-1$，且 $\gcd(k,p-1)=1$ (kmp−1)互素。

b、A 计算 $r=g^k \bmod p$

c、A 计算 $s=k^{-1}(m-xr) \bmod p$、该式由 $g^m=y^r r^s \bmod p$ 推导而来。

2. B 验证签名，B 接收到(m,r,s)

a、B 计算 $left=g^m \bmod p$

b、B 计算 $right=y^r r^s \bmod p$

c、比较 left=right?相等，说明签名合法；否则为无效。

很明显，ElGamal 加密算法的安全性等价于 Diffie-Hellman 体制的安全性。虽然还没有完全证明破译 ElGamal 签名体制等价于求解 GF(p)上的离散对数问题。但从可能的攻击方法来看，其破译方法大多都等价于求解 GF(p)上的离散对数问题。

破译签名体制本质上就是试图从(m,r,s)中得到签名者的私用密钥 x。假设给定一组明文$\{m_i:i=1,2,\cdots,l\}$及其相关的签名$\{(r_i,s_i):j=1,\cdots,n\}$，那么破译者可根据下式列出 n 个方程组来：

$m=(xr+ks) \bmod (p-1)$(从 s 的产生式导出)

但是 n 个方程有 n+1 个未知数，$k_1,k_2,\cdots,k_n,X$，故有无数组解。因此，在 k 互不相同的情况下，用此方法破译 ElGamal 签名体制是不行的。

若从其它途径来攻击，如求解 $g^m=y^r r^s \bmod p$ 方程来得到 x，其等价于求解 GF(p)上的离散对数问题。

另外一种威胁 ElGamal 签名体制的方法是假冒签名。这分为两种情况，一种是假冒者有任一组明文 m，他想假冒别人的签名，但不知道别人的私用密钥，若想满足 $g^m=y^r r^s \bmod p$ 这一关系式，只有构造一对(r,s)来，但现在还没有有效的算法解决这一问题。

另一种是假冒者已知一组合法明文及其对应的签名，那么他就能伪造另外一组合法的明文及签名；假设伪造者已知的明文及签名为(m,r,s)，那么他可构造另外一组(m',r',s',)如下：

$r'=g^A y^B \bmod p$

$s'=-r'/B \bmod (p-1)$

$m'=-r'A/B \bmod (p-1)$这里 A、B 是二整数，且 $\gcd(B,p-1)=1$；

这样，$y^{r'} r'^{m'}=y^{r'}(g^A y^B)^{-r'/B} \bmod p$

$=y^{r'} g^{-Ar'/B} y^{-r'} \bmod p$

$=g^{-Ar'/B} \bmod p$

$=g^{m'} \bmod p$

这一情况在其它签名体制中也存在，但假冒者要想伪造任意一组明文是不可能的。这可以用限制明文的结构和采用均匀性较好的 HASH 函数压缩明文来克服这一弱点。

2. ElGamal 算法与 RSA 签名算法比较

公钥密码体制的安全强度一般等价于求解数学中某一难题，现在已知最好的求解离散对数问题和大数分解问题的算法时间复杂性为：$O(\exp \sqrt{0.69m\ln m})$，这里 m 为公钥系统的密钥长度。因此要保证 ElGamal 体制的安全性，故其密钥长度(大素数 p 的长度)，不能低于 RSA 中所用的密钥长度。一般要100位上的十进制整数。在 RSA 加密系统中，每两个用户之间就需要一对私用密钥和公开密钥，即需要产生两个大素数和运行密钥产生程序。假设系统中有 N 个用户，那么公钥表就需要2N 个大素数。而在 ElGamal 系统中，全系统的用户可共用统一的 p 和 g，每个用户保留自己的私用密钥 x，然后将其公开密钥 y 公布即可。这不仅减少了系统密钥生成、分配和管理的工作量，而且也提高了系统的维护性。这一体制适合于用户较多的安全系统，例如银行资金转帐系统等。

同 RSA 签名算法比较，ElGamal 算法的安全性更强一些，因为 ElGamal 签名算法两次对相同的明文签名会得到不同的签名结果，这是因为签名时选取了不同的 k 的原因。但也存在与 ElGamal 体制加解密算法同样的问题，即签名后代码量增加了一倍，计算量增加了四倍以上。

需要指出的是 ElGamal 签名算法的实现比 RSA 签名算法和 ElGamal 密码体制的实现都要困难，除了要寻找本元以外，还要有高效的 gcd 算法。因为不是任意两个数都是互素的，为了加快寻找与 p−1互质的随机数 k 的速度，一般二进制 gcd 算法来进行判断，然后再用扩展欧几里德算法来计算 k^{-1}。因为 p−1是偶数。因此只考虑 k 为奇数的情形一般来说，k 与 p−1互素的概率是非常高的，约为81%。只要试一、两个奇数就可以找到使得 $\gcd(k,p-1)=1$的 k。

三、结　语

总之，ElGamal 公钥密码体制的数字签名方法是可以作为一个系统中数据安全考虑采用的一种方法，我们已经在 PC 机上实现了384比特长的 ElGamal 公钥数字签名系统，在大素数生成、查找本元和大数计算方面都是使用了自己研制的大数计算软件包，其速度是较满意的。