第16卷 第3期 电子科学学刊 Vol. 16, No. 3
1994 年 5 月 JOURNAL OF ELECTRONICS May, 1994

关于实多项式型公钥密码体制的破译和有关问题的探讨*

李 大 兴
(山东大学数学系 济南 250100; 信息安全国家重点实验室 北京 100039)
李 大 为
(中国银行锦西支行 锦西 125001)

摘要 本文通过计算等价密钥彻底破译了实多项式型公钥密码体制，同时讨论了与密码安全性有关的一些计算问题.

关键词 密码分析,公钥密码,计算

1 引 言

文献[1]提出了一种建立在实多项式上的概率公钥密码体制. 这种构筑在连续理论上的密码体制引起了我们的兴趣,并对其强度进行分析,结果表明该体制可以通过比较系数法求出一个等价密钥,从而被彻底破译. 本文将给出攻击方法的详细过程.

为什么其安全性已被充分"论证"的密码体制却又如此轻易地被破译?透过对文献[1]体制的分析，可以清楚地看到在公钥密码设计与安全性论证中存在着常被人们忽视的误区,正是这些误区使一些不安全的体制被轻率地认为十分安全,公钥密码研究人员应对此有所注意. 这是本文将要探讨的一个问题.

为完整起见,本文首先介绍文献[1]的体制,其中某些符号做了改换,这样做仅仅是为了容易分辨而已,丝毫没有改变原符号的意义.

2 实多项式型公钥密码体制介绍

任取实数域上 $k(>3)$ 次实多项式 $R(x)$、实数 $\pi_1, \pi_2, \alpha_1, \alpha_2, \beta_1, \beta_2, \eta$ 和正奇数 m, n (这里 $m, n > 10^3$),做两个实数域上的多项式 $P(x)$ 和 $Q(x)$ 为

$$P(x) = \alpha_1[R(x) + \eta]^m + \beta_1,$$
$$Q(x) = \alpha_2[\pi_1 R(x) + \pi_2\eta]^n + \beta_2,$$

1992-12-17, 收到 1993-03-30 定稿
* 国家自然科学基金及教委博士点基金资助项目
李大兴 男，1963 年生，讲师，博士生，现从事密码学、计算复杂性及计算数论的教学和科研工作.
李大为 男，1971 年生，助理工程师，现从事银行微机的开发与管理和计算机安全的研究工作.

这里要求 $\alpha_1 \cdot \alpha_2 \neq 0$，$\pi_1 \neq \pi_2$，$\pi_1 \neq 0$。

于是

$$\eta = \frac{[(Q(x)-\beta_2)/\alpha_2]^{1/n} - \pi_1[(P(x)-\beta_1)/\alpha_1]^{1/n}}{\pi_2 - \pi_1}.$$

新体制的各个参数为

加密密钥　多项式 $P(x), Q(x)$。

解密密钥　实数 $\pi_1,\pi_2,\alpha_1,\alpha_2,\beta_1,\beta_2,\eta$ 和正奇数 m,n。

加密方法　设明文为二进制数列 $(m_1,m_2,\cdots,m_l)$，$m_i \in \{0,1\}$，$i=1,2,\cdots,l$，任取一实数数列 $(r_1,r_2,\cdots,r_l)$，计算 $P(r_i)$ 和 $Q(r_i)$ 并记

$$d_i = Q(r_i),\quad c_i = \begin{cases} P(r_i), & m_i = 1; \\ P(r_i)+\Delta_i, & m_i = 0; \end{cases}$$

（这里 Δ_i 为任意非零实数），则密文为二元序列：

$$[(c_1,d_1),(c_2,d_2),\cdots,(c_l,d_l)]. \tag{1}$$

解密方法　当合法用户收到密文(1)式后，用其解密密钥计算

$$\delta_i = \frac{[(d_i-\beta_2)/\alpha_2]^{1/n} - \pi_1[(c_i-\beta_1)/\alpha_1]^{1/n}}{\pi_1 - \pi_2}.$$

若 $\delta_i = \eta$，则 $m_i = 1$；否则 $m_i = 0$，从而求出明文 $(m_1,m_2,\cdots,m_l)$。

3　等价密钥的求解方法

3.1　等价密钥的形式

上节介绍的公钥体制存在一种等价密钥，亦即若能求得正整数 k'，m'，n' 和实数 r'，$\alpha_1',\alpha_2',\beta_1',\beta_2',\eta'$ 使

$$\left.\begin{aligned} P(x) &= \alpha_1'[R'(x)+\eta']^{m'} + \beta_1', \\ Q(x) &= \alpha_2'[R'(x)+r'\eta']^{n'} + \beta_2', \end{aligned}\right\} \tag{2}$$

其中 $R'(x)$ 为 k' 次多项式，$r' \neq 1$。

令 $\delta = (r'-1)\eta'$，则

$$\delta = [(Q(x)-\beta_2')/\alpha_2']^{1/n'} - [(P(x)-\beta_1')/\alpha_1']^{1/m'}.$$

当截收到密文(1)式时，计算

$$\delta_i' = [(d_i-\beta_2')/\alpha_2']^{1/n'} - [(c_i-\beta_1')/\alpha_1']^{1/m'}.$$

与上节的道理一样，若 $\delta_i = \delta$，则 $m_i = 1$；否则 $m_i = 0$，从而恢复出明文 $(m_1,m_2,\cdots,m_l)$。这表明该体制已彻底破译了。可见，满足(2)式的七元组 $(m',n',\alpha_1',\alpha_2',\beta_1',\beta_2',(r'-1)\eta')$ 可构成一个等价密钥。

下面便来讨论如何求出一个等价密钥。分两步来考虑。

3.2　已知 k 时如何求等价密钥

因 $P(x)$ 和 $Q(x)$ 是公开的，所以它们的次数 km 和 kn 也是公开的。k 已知意味着 m 和 n 已知。

令 $P(x) = A \cdot \hat{P}(x)$，$Q(x) = B \cdot \hat{Q}(x)$，$R(x) = D \cdot \hat{R}(x)$，其中 A,B,D 分别为

$P(x),Q(x),R(x)$ 的首项系数,因此 $\hat{P}(x),\hat{Q}(x),\hat{R}(x)$ 均为首一多项式,令

$$\hat{P}(x)=x^{km}+\sum_{i=1}^{km}A_ix^{km-i},$$

$$\hat{Q}(x)=x^{kn}+\sum_{i=1}^{kn}B_ix^{kn-i},$$

$$\hat{R}(x)=x^{k}+\sum_{i=1}^{k}D_ix^{k-i},$$

因此有(注意 $\hat{P}(x)$ 和 $\hat{Q}(x)$ 都是已知多项式)

$$\begin{aligned}\hat{P}(x)&=P(x)/A=(\alpha_1/A)[R(x)+\eta]^m+\beta_1/A\\&=(\alpha_1D^m/A)[\hat{R}(x)+\eta/D]^m+\beta_1/A.\end{aligned}$$

由于 $\hat{P}(x)$ 和 $\hat{R}(x)$都是首一多项式,从而 $\alpha_1D^m=A$,上式又可化简为(只对次数不小于 $k(m-1)$ 的项感兴趣),即

$$\begin{aligned}\hat{P}(x)&=[\hat{R}(x)+\eta\cdot D^{-1}]^m+\beta_1\cdot A^{-1}\\&=(\hat{R}^m(x)+m\cdot\eta D^{-1}\cdot\hat{R}^{m-1}(x)+\cdots)+\beta_1A^{-1}\\&=(\hat{R}^m(x)+m\cdot\eta D^{-1}x^{k(m-1)}+\cdots)+\beta_1\cdot A^{-1}.\end{aligned}\tag{3}$$

下面递归地使用比较系数法逐一求出 $\hat{R}(x)$ 的系数 $D_1,D_2,\cdots,D_{k-1}$。

基础步 $i=1$

只考虑(3)式两边的 $km-1$ 次项,有

$$\begin{aligned}\hat{P}(x)=\hat{R}^m(x)+\cdots&=(x^k+D_1x^{k-1}+\cdots)^m+\cdots\\&=x^{km}+mD_1x^{km-1}+\cdots,\end{aligned}$$

从而有 $A_1=mD_1$, 亦即 $D_1=A_1/m$。

归纳步 设 $D_1,D_2,\cdots,D_i$ 已求出 $(1\leqslant i<k-1)$。令 $\hat{R}_i(x)=x^k+D_1x^{k-1}+\cdots+D_ix^{k-i}$, 则有 $\hat{R}(x)=\hat{R}_i(x)+D_{i+1}x^{k-i-1}+\cdots$。只考虑 $km-i-1$ 次以上各项,有

$$\begin{aligned}\hat{R}^m(x)&=(\hat{R}_i(x)+D_{i+1}x^{k-i-1}+\cdots)^m\\&=\hat{R}_i^m(x)+mD_{i+1}\hat{R}_i^{m-1}x^{k-i-1}+\cdots\\&=\hat{R}_i^m(x)+mD_{i+1}x^{km-i-1}+\cdots.\end{aligned}$$

将上式代入(3)式并比较两边的 $km-i-1$ 次项系数,有

$$A_{i+1}=mD_{i+1}+(\hat{R}_i^m(x)\text{ 的 }km-i-1\text{ 次项系数}).$$

因 $\hat{R}_i(x)$ 是已知多项式,所以求 $\hat{R}_i^m(x)$ 的 $km-i-1$ 次项系数可以在 $O(k^2\log m)$ 次实数运算时间内完成(该计算比展开 $\hat{R}_i^m(x)$ 要容易),所以 D_{i+1} 也是易求的。

归纳完毕。这样便可一直求到 D_{k-1}。

D_k 的计算与前面稍有不同。令 $\hat{R}_{k-1}(x)=x^k+\sum_{i=1}^{k-1}D_ix^{k-i}$,类似前面的讨论,并注意右边较前增加了 $m\eta D^{-1}x^{k(m-1)}$ 这一项,有

$$A_k=m\eta D^{-1}+mD_k+(\hat{R}_{k-1}^m(x)\text{的 }k(m-1)\text{次项系数}).$$

由此可解出 $(D_k+\eta D^{-1})$ 而不是 D_k,但这并不影响破译工作,因为以后起关键作用的既

不是 D_k 也不是 ηD^{-1}，而是 $D_k+\eta D^{-1}$。这样多项式 $\hat{R}(x)+\eta D^{-1}$ 便已求出。再比较(3)式两边的常数项，有

$$A_{km}=(D_k+\eta D^{-1})^m+\beta_1 A^{-1},$$

亦即 $\beta_1=A(A_{km}-(D_k+\eta D^{-1})^m)$。

至于 $Q(x)$，可用类似的办法处理。我们写出

$$\hat{Q}(x)=Q(x)/B=(\alpha_2\pi_1^n D^n/B)[\hat{R}(x)+(\pi_2/\pi_1)\cdot\eta D^{-1}]+\beta_2/B,$$

因而有 $\alpha_2\pi_1^n D^n=B$，且

$$\hat{Q}(x)=[\hat{R}_{k-1}(x)+D_k+(\pi_2/\pi_1)\eta D^{-1}]+\beta_2/B,$$

由此可求出 $(D_k+(\pi_2/\pi_1)\eta D^{-1})$ 和 β_2。

令 $k'=k$，$m'=m$，$n'=n$，$\alpha_1'=A$，$\alpha_2'=B$，$\beta_1'=\beta_1$，$\beta_2'=\beta_2$，$R'(x)=\hat{R}(x)$，$\eta'=\eta D^{-1}$，$r'=\pi_2/\pi_1$，且

$$\delta=(r'-1)\eta'=(\pi_2/\pi_1)\eta D^{-1}-\eta D^{-1}=[D_k+(\pi_2/\pi_1)\eta\cdot D^{-1}]-(D_k+\eta D^{-1})$$

是已知的。不难验证如此构成的七元组是一个等价密钥。

3.3 未知 k 时如何求等价密钥

因 $km=\deg(P(x))$，$kn=\deg(Q(x))$，所以 $P(x)$和 $Q(x)$的存贮空间(单元)和计算时间以及体制的加密、解密时间都是关于 km 和 kn 的多项式(视一个实数占一个单元，两个实数的一次四则运算为一个计算单位)，所以 km 和 kn 都不能太大，k 也就更不能太大。因此，我们可以对 k 从 4 开始试算，这样至多试算 k 次就达到 k，每次试算都假设所取的值就是真正的 k，并使用上节的方法求出一个七元组，验证该七元组是否满足(2)式，若满足则已得到一等价密钥(无论该值是否为真正的 k)；若不满足则将猜测值加 1 并重复上述过程。因猜对 k 时(2)式必满足，所以对 k 的猜测在不超过 k 次之内就可使(2)式满足。由于每次循环的计算时间都是关于 k,m,n 的多项式，因此整个计算过程仍是 k,m,n 的多项式时间。

下面用稍加形式化的方式来表述上面的思想：

算法 AL（输入 $P(x),Q(x)$）

第 0 步(初始化)　置 $\hat{k}:=4$，$g:=GCD(\deg(P(x)),\deg(Q(x)))$。

第 1 步　若 $\hat{k}\nmid g$，则执行第 4 步；否则令 $\hat{m}:=\deg(P(x))/\hat{k}$，$\hat{n}:=\deg(Q(x))/\hat{k}$，若 $\hat{m},\hat{n}$ 中无偶数，则执行第 2 步，否则执行第 4 步。

第 2 步　将 $\hat{k}$ 视为 k 并用第 3.2 节的方法求出 $\hat{D}_1$，$\hat{D}$，$\cdots$，$\hat{D}_{\hat{k}-1}$ 和 $\hat{D}_{\hat{k}}+\hat{\eta}$ 以及 $\hat{\beta}_1$，$\hat{\beta}_2$，$\hat{D}_{\hat{k}}+\hat{r}\cdot\hat{\eta}$。

第 3 步　验证下列等式是否成立

$$\left.\begin{aligned}P(x)&=A\Big[x^{\hat{k}}+\sum_{i=1}^{\hat{k}-1}\hat{D}_i x^{\hat{k}-i}+\hat{D}_{\hat{k}}+\hat{\eta}\Big]^{\hat{m}}+\hat{\beta}_1,\\ Q(x)&=B\Big[x^{\hat{k}}+\sum_{i=1}^{\hat{k}-1}\hat{D}_i x^{\hat{k}-i}+\hat{D}_{\hat{k}}+\hat{r}\hat{\eta}\Big]^{\hat{n}}+\hat{\beta}_2,\\ (\hat{D}_{\hat{k}}+\hat{r}\cdot\hat{\eta})&-(\hat{D}_{\hat{k}}+\hat{\eta})\neq 0,\end{aligned}\right\}$$

若成立则执行第 5 步；否则执行第 4 步。

第 4 步　$\hat{k}:=\hat{k}+1$，退回执行第 1 步。

第 5 步 输出 $(\hat{m},\hat{n},A,B,\hat{\beta}_1,\hat{\beta}_2,(\hat{r}-1)\hat{\eta})$,并停止!

说明 1 第 3 步中的第三个不等式是用来判别 $\hat{r}\neq 1$,这还需要假设体制构造中的 $\eta\neq 0$. $\eta=0$ 的破译方法完全类似于 $\eta\neq 0$,而且要简单些,为简明起见,只考虑 $\eta\neq 0$ 时的破译.

说明 2 设第 3.2 节求等价密钥方法的时间为 Poly(k,m,n), 其中 Poly 记某一多项式,则算法 AL 的时间至多为 $k\cdot$Poly(k,m,n).

4 有关公钥密码学中的一些计算问题的讨论

公钥密码的安全性建立在容易计算的加密函数之逆函数的计算困难性上,因此,公钥密码学从一诞生就注定与计算理论密切相关.

一个问题是容易计算的,是指存在求解该问题的多项式时间算法(允许是概率的),否则称难于计算的. 须特别注意的是这里所指的多项式是关于问题描述长度 (即问题的二进制输入长度)的. 例如,文献[1]的实多项式体制对 l 比特进行加密运算时,其输入长度为 $l+(m+n)kL$,其中 L 表示一个实数的描述长度, 亦即信息长度加密钥长度. 有时输入长度用一些具有代表性的参数的长度代替而不必用真正的描述输入的长度, 但必须要求输入长度≤Poly (各参数长度)(Poly 代表某个多项式,下同).

4.1 模乘和通常乘法之间计算上的差异

众所周知,计算 $a^m(\mathrm{mod}N),0<a<N$,只须做 Poly$(\log m,\log N)$ 次比特运算, 因此 m 和 N 都可取得很大,如 $m,N\approx 10^{200}$. 但计算 a^m 与之有着本质区别, 因 a^m 的长度为 $\lceil m\log a\rceil$,将 a^m 的计算结果输出便至少需要 $\lceil m\log a\rceil$ 次比特操作(在二进制图灵机模型下),因此无论何种算法计算 a^m 至少需要 $\lceil m\log a\rceil$ 个比特操作,所以是关于 $\log m$ 的指数而不是多项式.实际上 a^m 的计算时间为 Poly$(m,\log a)=$ Poly$(2^{\log m},\log a)$.

类似地,计算 $f^m(x)\mathrm{mod}g(x)(\deg f<\deg g)$ 的时间为 Poly$(\log m,\deg g)$个数域上的算术运算,但 $f^m(x)$的计算时间通常至少为 $m\cdot\deg f$, 因 $f^m(x)$有 $m\cdot\deg f$ 个系数须求出(象 $f(x)=x^k$ 这样的个别情形除外),所以 $f^m(x)$的计算时间只可能表示成 Poly$(m,\deg f)$.

上述两种计算问题之间的实质性差异, 对于用通常幂次设计密码体制的设计者来说须特别加以注意, 否则就容易造成误解, 认为增大幂次或多项式次数会使密码强度增加[1-3], 但相应的攻击方法的时间同加密、解密时间一样都是幂次或多项式次数的多项式[4,5],因此与加密、解密时间多项式相关,所以攻击是有效的.

4.2 公钥密码中哪些参数无法保密

保护解密密钥是任何密码安全性的前提. 但在分析和论证安全强度时, 必须分清哪些密钥参数能被认为是可保密的, 哪些是无法保密的. 若某参数可选取的个数受限于输入长度的多项式,则该参数可以像第 3.3 节那样用穷举法试算出来, 所以该参数对攻击者来说与公开参数没有什么不同. 如果所论证出的"安全性"是建立在这类参数保密前提上的,那么这种"安全性"就很值得怀疑. 实多项式型公钥密码的"安全性"严重地依赖于 m, n,k 的保密.

上述原则对密码分析同样适用．攻击某种密码只须在那些无法保密的密钥参数为已知的情形下找到一种有效攻击方法，然后再像第 3.3 节那样对这些参数进行穷举．

4.3 关于连续型密码体制的一点注记

一些密码体制和安全系统被构筑在实数理论之上[1,6,7]，但真实数密码是无法实现的．因为若想实现这类体制，首先面临的问题是如何用有限长字符串表示实数(更实际的问题是如何用定长字符串表示实数)，能被这样表示的实数只有可数个，因此这类体制本质上还是建立在离散结构上．所面临的第二个问题是如何进行实数计算，实数的计算是通过有限位小数来逼近的，而且计算时间受精度要求的影响，因此在精度要求给定的情况下，计算出的实数只是有效位受限的小数，与其说它们是实数还不如直接说它们是有效位给定的有理数．实数密码体制只有经如此还原之后再去讨论其加密、解密的计算和其安全性才有实际意义．例如，实多项式型体制经还原之后便成为有理多项式体制，求有理多项式的有理根是存在多项式时间算法的[8]，这是一种非常直观的威胁．

关于实数的计算理论请参见文献[9]的介绍．

参　考　文　献

[1] 刘　锐．电子学报，1992，20(8)：101—103．
[2] 曹珍富．电子学报，1988，16(4)：120—121．
[3] Yang Yixian. Electron. Lett., 1987, 23(11): 560—561.
[4] Li Daxing. Electron. Lett., 1991, 27(3): 228—229.
[5] 李大兴．关于一些公钥密码分析的综合报告．中国密码学会成立大会．北京：1990．
[6] 黄国祥，刘　健．计算机学报，1987，10(6)：321—327．
[7] 曹珍富，刘　锐．高校应用数学学报，1989，4(1)：1—5．
[8] Lenstra A K, Lenstra H W, Lovász L. Math. Ann. 1982, 26(4): 515—534.
[9] Ker-I Ko. Studies in Complexity Theory. New York: John Wiley & Sons, 1986, 1—62.

ATTACKS ON REAL POLYNOMIAL TYPE PUBLIC-KEY CRYPTOSYSTEMS AND DISCUSSION ON RELATED PROBLEMS

Li Daxing

(Shandong University, Jinan 250100; State key Lab. of Information Security, Beijing 100039)

Li Dawei

(Jinxi Branch of Bank of China, Jinxi 125001)

Abstract　This paper completely breaks up the real type polynomial public-key cryptosystems by computing the equivalent secure keys. And some computational problems related to securities of cryptosystems are discussed.

Key words　Cryptanalysis, Public-key cryptosystems, Computation