Computer Science and Application 计算机科学与应用, 2023, 13(4), 789-798
Published Online April 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.134078

Hans汉斯

基于箝位矩阵的公钥密码

陆成刚

浙江工业大学理学院，浙江 杭州

收稿日期：2023年3月18日；录用日期：2023年4月17日；发布日期：2023年4月23日

摘　要

箝位矩阵是一类以一定范围内的非负整数作为构成元素的矩阵，且在运算过程中这些元素始终被钳制在该范围内。构造两组在箝位运算下互逆的方阵，由此生成一对密钥。在一阶矩阵情形下，箝位设计的密码系统只具多项式复杂度的破解强度，当二阶以上时，原先的破解成因不复存在，从而陡然升级为NP复杂度的强度，且随着矩阵阶数的增大而呈指数增强。箝位矩阵的公钥密码具有独特的批处理加解密形式，虽然涉及矩阵乘法，但对每个明文处理单位而言，平均计算占有并不高。和传统的RSA、Elgamal和椭圆密码等基于大素数（或基于由大素数参与构造的有限域）不同，箝位矩阵的公钥密码的密钥获取极其容易，不需要耗费资源作素数检验，而且箝位矩阵的公钥密码系统具有自然的随机性效应，不像RSA需要专门设计的附加随机填充的模式，或者不像Elgamal引入临时性的随机密钥那样的协议交互的设计。由此可以证明，箝位矩阵密码系统具有像离散对数困难性那样的归约到枚举未定矩阵形式的NP复杂度强度。

关键词

箝位矩阵，公钥密码，RSA算法，Elgamal算法，辗转相除法

Public Key Cryptography Based on Clamping Matrix

Chenggang Lu

College of Science, Zhejiang University of Technology, Hangzhou Zhejiang

Received: Mar. 18th, 2023; accepted: Apr. 17th, 2023; published: Apr. 23rd, 2023

Abstract

Clamping matrix was a kind of matrix whose elements were non-negative integers within a certain range, and these elements were always clamped within this range during the operation. Be constructed to two sets of square matrices that were mutually inverse under the clamp operation, the-

文章引用: 陆成刚. 基于箝位矩阵的公钥密码[J]. 计算机科学与应用, 2023, 13(4): 789-798.
DOI: 10.12677/csa.2023.134078

reby generating a pair of keys. In the case of a first-order matrix, the clamp-designed cryptosystem only had the cracking strength of polynomial complexity. When it was higher than the second order, the original cracking causes no longer exists, so it was suddenly upgraded to the strength of NP-complexity, and with the matrix, it increased exponentially as the order increases. The public key cryptography of the clamping matrix had a unique form of batch encryption and decryption. Although matrix multiplication was involved, the average computational occupancy for each plaintext processing unit was not high. Different from traditional RSA, Elgamal and elliptic ciphers based on large prime numbers (or based on finite fields constructed by large prime numbers), the key acquisition of public key cryptography with a clamping matrix was extremely easy, and it did not need to spend resources for prime number checking. Moreover, the public key cryptosystem of the clamping matrix had a natural randomness effect, unlike RSA which required a specially designed pattern of additional random padding, or the design of protocol interaction like Elgamal introduced a temporary random key. Clamping-matrix cryptosystems could be shown to have NP-complexity strengths like discrete log difficulty reduction to enumerated indeterminate matrix form.

Keywords

Clamp Matrix, Public Key Cryptography, RSA Algorithm, Elgamal Algorithm, Euclid's Tossing and Dividing

1. 问题背景

经典的非对称密钥系统 RSA (Rivest-Shamir-Adleman)、Elgamal 和椭圆密码[1] [2] [3] [4] [5]是基于大素数来构造的：例如 RSA 是通过两个大素数相乘获得一个半素数，通过基于半素数的欧拉数的同余运算构造一对互逆指数实现加解密运算；Elgamal 是使用一个大素数构造的有限域上的循环群的离散对数计算困难性实现保密机制；椭圆密码则是利用有限域上的椭圆曲线的点加运算实现加解密，本质上还是一类离散对数计算困难性问题，相对于一般有限域循环群的离散对数问题，椭圆密钥构造需要的素数可以更小一些。

以上算法的构造无一例外都需要大素数的支持，而大素数是稀缺资源，寻找合适的大素数是和素数检验有密切的关系的。Miller-Rabin 素数检验是概率素数检验，一种确定给定数是否可能为素数的算法[6] [7] [8]。可见，这些算法的应用在安全性较高的前提下，其本身的密钥生成的不易程度也是一个制约因素。此外，这些密码系统都是基于数域可交换运算来实现的，对未来量子计算而言，具有一定的无法抵御攻击的风险。专利[9]是一类基于箝位矩阵结构而设计的非对称密码，专利阐述了该方法的步骤并给出应用的示例。本文分析该专利的公钥密码系统的原理和安全性，以及数值实验比较，它突出的特点是密钥生成极其简单，另外，由于矩阵乘法的不可交换性，对于抵御未来量子计算攻击也是具一定的趋好因素。

后文首先介绍该密钥的设计原理和加解密模式，然后再证明密钥的安全性，最后是加解密数据实验。

2. 密钥设计原理

引入记号 $z_N z_{N-1}\cdots z_2 z_1$ 表示一 N 位的整数，其中 z_i 为 $0,1,2,\cdots,9$ $(\forall 1\le i\le N)$的任意数字，即

$$z_N z_{N-1}\cdots z_2 z_1 = \sum_{i=1}^{N-1} z_i \cdot 10^{i-1} \tag{1}$$

对任意 $1 \le 2k+1 \le N$ 显然有拆分 $z_N z_{N-1}\cdots z_2 z_1 = z_N z_{N-1}\cdots z_{2k+2}\cdot 10^{2k+1} + z_{2k+1}z_{2k}\cdots z_2 z_1$ (对 $2k+1=N$，有 $z_N z_{N-1}\cdots z_{2k+2}=0$)。再定义取该整数末 $2k+1$ 位数的箝位运算，即 $\underline{z_N z_{N-1}\cdots z_2 z_1} = z_{2k+1}z_{2k}\cdots z_2 z_1$ (k 为任意大于等于零的整数)，参数 $2k+1$ 为箝位指数。下面先从整数开始探讨箝位运算的一些性质，最后引申到矩阵也具备这些箝位特点。

引理 1. 设正整数 a，b，k，则高斯取整函数满足性质 $\left[\frac{a\cdot b}{10^k}\right] \ge \left[\frac{a}{10^k}\right]\cdot b$。

证明：设 $a=\sum_{i=0}^{N} a_i\cdot 10^i, b=\sum_{i=0}^{M} b_i\cdot 10^i$ 其中 N、M 为非负整数，于是 $a\cdot b = \sum_{i+j=0}^{i+j=N+M}\left(a_i b_j\right)10^{i+j}$，$\left[\frac{a\cdot b}{10^k}\right] = \sum_{i+j=k}^{i+j=N+M}\left(a_i b_j\right)10^{i+j-k}$。而 $\left[\frac{a}{10^k}\right]\cdot b = \sum_{i=k}^{N} a_i\cdot 10^{i-k}\cdot\sum_{i=0}^{M} b_i\cdot 10^i = \sum_{i\ge k, i+j=k}^{i+j=N+M}\left(a_i b_j\right)10^{i+j-k}$。对于每个固定的幂次 $i+j-k$，均有 $\sum_{i+j\ge k} a_i b_j \ge \sum_{i\ge k} a_i b_j$ (因为 $i\ge k \Rightarrow i+j\ge k$ 故右边的组合选项数少于左边)，例如对于 $i+j-k=2$，左边为 $\left(a_0 b_{k+2}+\cdots+a_k b_2+a_{k+1}b_1+a_{k+2}b_0\right)$，右边为 $\left(a_k b_2+a_{k+1}b_1+a_{k+2}b_0\right)$。故 $\left[\frac{a\cdot b}{10^k}\right] \ge \left[\frac{a}{10^k}\right]\cdot b$。

引理 2. 低位保持原理。设整数 u、v、a 的乘积为 N 位整数，u、a 乘积为 M 位整数，v、a 乘积为 T 位整数，即

$$uva = z_N z_{N-1}\cdots z_2 z_1 \tag{2}$$

$$ua = x_M x_{M-1}\cdots x_2 x_1 \tag{3}$$

$$va = y_T y_{T-1}\cdots y_2 y_1 \tag{4}$$

并且 $N\ge 2k+1$，$M\ge 2k+1$ 以及 $M\ge 2k+1$，k 为某确定的正整数。则有

$$\underline{uva} = \underline{\underline{uav}} = \underline{\underline{vau}} \tag{5}$$

证明：首先由式(2)得

$$\underline{uva} = z_{2k+1}z_{2k}\cdots z_2 z_1 \tag{6}$$

由式(3)得到

$$\underline{ua} = x_{2k+1}x_{2k}\cdots x_2 x_1 \tag{7}$$

于是

$$\begin{aligned}
\underline{\underline{uav}} &= x_{2k+1}x_{2k}\cdots x_2 x_1\cdot v \\
&= \left(x_M x_M\cdots x_{2k+2}x_{2k+1}x_{2k}\cdots x_2 x_1 - x_M x_M\cdots x_{2k+2}\cdot 10^{2k+1}\right)\cdot v \\
&= x_M x_M\cdots x_{2k+2}x_{2k+1}x_{2k}\cdots x_2 x_1\cdot v - x_M x_M\cdots x_{2k+2}\cdot 10^{2k+1}\cdot v \\
&= uav - x_M x_M\cdots x_{2k+2}\cdot 10^{2k+1}\cdot v \\
&= z_N z_{N-1}\cdots z_2 z_1 - x_M x_M\cdots x_{2k+2}\cdot 10^{2k+1}\cdot v \\
&= z_N z_{N-1}\cdots z_{2k+3}z_{2k+2}\cdot 10^{2k+1} + z_{2k+1}z_{2k}\cdots z_2 z_1 - x_M x_M\cdots x_{2k+2}\cdot 10^{2k+1}\cdot v \\
&= \left(z_N z_{N-1}\cdots z_{2k+3}z_{2k+2} - x_M x_M\cdots x_{2k+2}\cdot v\right)\cdot 10^{2k+1} + z_{2k+1}z_{2k}\cdots z_2 z_1
\end{aligned}$$

又根据引理 1 得

$$z_N z_{N-1}\cdots z_{2k+3}z_{2k+2} = \left[\frac{uav}{10^{2k+1}}\right] \ge \left[\frac{ua}{10^{2k+1}}\right]v = x_M x_M\cdots x_{2k+2}\cdot v$$

即式中 $z_N z_{N-1}\cdots z_{2k+3}z_{2k+2}-x_M x_M\cdots x_{2k+2}\cdot v$ 为非负整数，

所以

$$\underline{\underline{uav}}=z_{2k+1}z_{2k}\cdots z_2 z_1 \tag{8}$$

类似地

$$\begin{aligned}\underline{\underline{vau}}&=y_{2k+1}y_{2k}\cdots y_1\cdot u\\&=y_T y_{T-1}\cdots y_{2k+2}y_{2k+1}y_{2k}\cdots y_1\cdot u-y_T y_{T-1}\cdots y_{2k+2}\cdot 10^{2k+1}\cdot u\\&=vau-y_T y_{T-1}\cdots y_{2k+2}\cdot 10^{2k+1}\cdot u\\&=z_N z_{N-1}\cdots z_2 z_1-y_T y_{T-1}\cdots y_{2k+2}\cdot 10^{2k+1}\cdot u\\&=z_N z_{N-1}\cdots z_{2k+3}z_{2k+2}\cdot 10^{2k+1}+z_{2k+1}z_{2k}\cdots z_2 z_1-y_T y_{T-1}\cdots y_{2k+2}\cdot 10^{2k+1}\cdot u\\&=\left(z_N z_{N-1}\cdots z_{2k+3}z_{2k+2}-y_T y_{T-1}\cdots y_{2k+2}\cdot u\right)\cdot 10^{2k+1}+z_{2k+1}z_{2k}\cdots z_2 z_1\end{aligned}$$

根据引理 1 得到

$$\underline{\underline{vau}}=z_{2k+1}z_{2k}\cdots z_2 z_1 \tag{9}$$

由式(6)、(8)、(9)得式(5)的结论。证毕。

该引理说明任意三个整数乘积的末连续几位的数字可以由仅仅取中间结果的末连续几位来乘积得到，而根据整数乘法满足交换律，中间结果有两种可能形式，均符合这种性质。我们称这种性质为低位保持原理，很明显多于三个数的乘积也满足这种原理。

引入对 n 阶非负整数方阵的箝位运算，$\underline{\boldsymbol{A}}$ 表示 $\boldsymbol{A}$ 的每个元素钳制到箝位指数控制的范围内，而 $\underline{\boldsymbol{AB}}$ 表示矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 先做矩阵乘法再做箝位运算，类似地，可以证明：

$$\underline{\boldsymbol{AB}}=\underline{\boldsymbol{A}}\,\underline{\boldsymbol{B}}\ \text{且}\ \underline{\underline{\boldsymbol{ABC}}}=\underline{\boldsymbol{ABC}}=\underline{\boldsymbol{A}}\,\underline{\boldsymbol{BC}} \tag{10}$$

定理 2. 加密解密原理。由从 0 到 $10^{2k+1}-1$ 整数的随机数构成的 n 阶方阵 $\boldsymbol{A}$，设 $\boldsymbol{E}$ 为单位矩阵，根据

$$\begin{aligned}\left[\boldsymbol{A}^{2k+1}10^{2k+1}+\boldsymbol{E}\right]&=\left[(10\boldsymbol{A})^{2k+1}+\boldsymbol{E}\right]\\&=(10\boldsymbol{A}+\boldsymbol{E})\left[(10\boldsymbol{A})^{2k}-(10\boldsymbol{A})^{2k-1}+\cdots+(10\boldsymbol{A})^{2}-(10\boldsymbol{A})+\boldsymbol{E}\right]\\&=(10\boldsymbol{A}+\boldsymbol{E})\left\{(10\boldsymbol{A}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{A})^{2i-1}+\boldsymbol{E}\right\}\end{aligned} \tag{11}$$

在箝位运算机制下有

$$\underline{(10\boldsymbol{A}+\boldsymbol{E})}\underline{\left\{(10\boldsymbol{A}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{A})^{2i-1}+\boldsymbol{E}\right\}}=\underline{\left\{(10\boldsymbol{A}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{A})^{2i-1}+\boldsymbol{E}\right\}}\underline{(10\boldsymbol{A}+\boldsymbol{E})}=\boldsymbol{E} \tag{12}$$

即 $10\boldsymbol{A}+\boldsymbol{E}$ 与 $(10\boldsymbol{A}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{A})^{2i-1}+\boldsymbol{E}$ 在箝位运算机制下互逆。

证明：由式(11)和式(10)箝位运算的机制，结论是显然的。证毕。

定理 3. 密钥生成。由从 0 到 $10^{2k+1}-1$ 整数的随机数构成的 n 阶方阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，令

$$\boldsymbol{U}=(10\boldsymbol{A}+\boldsymbol{E})\left\{(10\boldsymbol{B}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{B})^{2i-1}+\boldsymbol{E}\right\} \tag{13}$$

及

$$\boldsymbol{V}=(10\boldsymbol{B}+\boldsymbol{E})\left\{(10\boldsymbol{A}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{A})^{2i-1}+\boldsymbol{E}\right\} \tag{14}$$

则 $\underline{\boldsymbol{UV}}=\underline{\boldsymbol{VU}}=\boldsymbol{E}$，即 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 在箝位运算下是一对互逆矩阵。

证明：由矩阵乘法和箝位运算的结合律，结论是显然的。

3. 加解密方法

由定理 2 和 3，得到

$$\underline{(\boldsymbol{VU})\boldsymbol{X}}=\underline{\boldsymbol{V}(\boldsymbol{UX})}=\boldsymbol{X} \tag{15}$$

利用 $\boldsymbol{U}$、$\boldsymbol{V}$ 的互逆性，可以实现加解密。密钥 $\boldsymbol{U}$、$\boldsymbol{V}$ 的角色是对称的，如果 $\boldsymbol{U}$ 作为公钥，那么 $\boldsymbol{V}$ 作为私钥，或者反之；用作加密与用作数字签名，在计算上没有差异，只是由收、发方加密处理的不同。而且明文(在签名时就是明文的摘要)、密钥和明文均是 n 阶箝位矩阵，所有元素可以取到 0 到 $10^{2k+1}-1$ 的整数，密钥是随机整数。

又由于密钥

$$\begin{aligned}\boldsymbol{U}&=(10\boldsymbol{A}+\boldsymbol{E})\left\{(10\boldsymbol{B}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{B})^{2i-1}+\boldsymbol{E}\right\}\\&=10\boldsymbol{A}\left\{(10\boldsymbol{B}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{B})^{2i-1}+\boldsymbol{E}\right\}+10\left\{(10\boldsymbol{B}-\boldsymbol{E})\sum_{i=1}^{k}\left(10^{2i-2}\boldsymbol{B}^{2i-1}\right)\right\}+\boldsymbol{E}\end{aligned} \tag{16}$$

前两项都含因子 10，导致该矩阵除对角线外所有元素末位为 0，于是直接用明文矩阵 $\boldsymbol{X}$ 计算，得到

$$\boldsymbol{UX}=10\boldsymbol{A}\left\{(10\boldsymbol{B}-\boldsymbol{E})\sum_{i=1}^{k}(10\boldsymbol{B})^{2i-2}+\boldsymbol{E}\right\}\boldsymbol{X}+10\left\{(10\boldsymbol{B}-\boldsymbol{E})\sum_{i=1}^{k}\left(10^{2i-2}\boldsymbol{B}^{2i-1}\right)\right\}\boldsymbol{X}+\boldsymbol{X} \tag{17}$$

则密文矩阵 $\underline{\boldsymbol{UX}}$ 的所有元素的末尾是 $\boldsymbol{X}$ 的元素的末位数字。为此，$\boldsymbol{X}$ 在加密时先作一种预处理，每个元素的数字右移一位，将个位数设置一个随机数, 然后加密，即与密钥作箝位机制下的矩阵乘法。这样处理一举两得，一方面可以避免末位泄露，另一方面自然地引入了随机性，即对相同的明文使用同一密钥生成的密文不同。

由此每个明文单位可对应取到的整数在 0 到 $10^{2k}-1$ 之间。每次进行加密的文本单位的规模是矩阵的阶数平方 n^2，这是一种批处理模式，它和 RSA、Elgamal 等算法的点处理模式明显不同。由于矩阵乘法的复杂度是 $O(n^3)$，平均到每个文本单位的计算复杂度是 $O(n)$，即依密钥尺度呈线性增长，这样的加解密计算的消耗与 RSA 之类算法相仿。图 1 给出箝位矩阵加密和签名的算法流程。

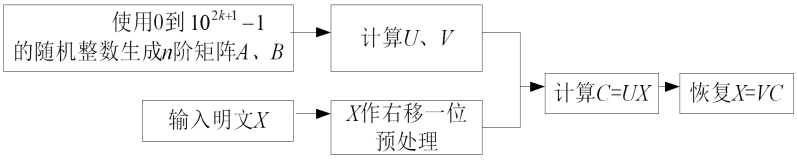

Figure 1. Application of public key cryptography with clamping matrix
图 1. 箝位矩阵的公钥密码的运用

4. 密钥安全性分析

破解箝位矩阵的公钥密码，并不意味着直接去计算箝位运算限制下密钥矩阵的逆，因为那意味着关于整数矩阵求解整数形式的箝位逆。更一般的等价形式是求如下的方程

$$\boldsymbol{F}\cdot\boldsymbol{U}=\boldsymbol{G}\cdot 10^{2k+1}+\boldsymbol{E} \tag{18}$$

式(18)中，$\boldsymbol{U}$ 是已发布的受攻击的密钥，$\boldsymbol{F}$ 就是待定的在箝位运算限制下的 $\boldsymbol{U}$ 的整数逆。方程右边是一个未定的适配矩阵 $\boldsymbol{G}$，而之所以写成 $\boldsymbol{G}\cdot 10^{2k+1}+\boldsymbol{E}$，是由于右端在箝位限制下变成单位阵的需要。显然，式(18)已经包含了直接求 $\boldsymbol{U}$ 的箝位逆的情形。事实上，式(18)是一个典型的整系数不定方程如何求取整数解的问题。

首先考虑一个特例，当箝位矩阵的阶数 $n=1$ 时，由于方程(18)中 U 是末位 1 的整数，与 10^{2k+1} 互素，因而可以使用欧几里德辗转相除法，得到确定的整数 F、G 满足

$$F\cdot U=G\cdot 10^{2k+1}+1 \tag{19}$$

辗转相除法的计算复杂度是在 $O\left(\log 10^{2k+1}\right)=O(k)$ 级别。但当箝位矩阵的阶数大于 1 时，对于整数矩阵没有带余除法，且矩阵乘法具不可交换性，因此也没有辗转相除法。式(18)只有从一般的解线性方程组的角度去分析。

一般整系数线性不定方程求取整数解在理论上是可解的[10]，通常把一部分变量视作已知量，其余变量写成这些视作已知的变量的表达式。但显然，这样的形式解在这里没有用处，式(18)中必须确定 $\boldsymbol{F}$ 的具体数字(因而也可确定 $\boldsymbol{G}$ 的数字)才能破解

$$\underline{\boldsymbol{FU}}\underline{\boldsymbol{X}}=\underline{\underline{\boldsymbol{FU}}}\underline{\boldsymbol{X}}=\boldsymbol{X} \tag{20}$$

由于式(18)是不定方程，要确定具体的数字解，必须令视作已知的变量取作具体的数字，由于箝位运算导致变量 $\boldsymbol{F}$、$\boldsymbol{G}$ 的元素取值在箝位指数控制的范围内(0 到 $10^{2k}-1$ 之间)，所以尝试取值的可能数目是 10^{2k}。又由于矩阵的元素数目为 n^2，所以总的取值尝试次数为 10^{2kn^2}。当 $k=3$、$n=10$ 时，这个数目就是 10^{600}。显然，箝位矩阵公钥的破解难度是依参数 k、n 呈指数增加的。这里把破解算法归约到枚举该未定矩阵 $\boldsymbol{G}$ 的所有元素，得出破解强度是 NP 的结论，就像 Elgamal 之类的算法是基于离散对数计算归结为枚举所有对数值的困难性一样。

顺便指出，在式(18)中如果直接使用 $\boldsymbol{U}$ 的常规逆，则 $\boldsymbol{F}$ 未必为整数矩阵，如此 $\boldsymbol{F}\cdot\boldsymbol{U}$ 的元素的取值必受高于 $2k+1$ 位数数字的影响。于是 $\boldsymbol{F}\cdot\boldsymbol{U}$ 虽然为单位阵，但显然 $\underline{\boldsymbol{F}\cdot\boldsymbol{U}\cdot\boldsymbol{X}}\neq\underline{\underline{\boldsymbol{F}\cdot\boldsymbol{U}}\cdot\boldsymbol{X}}=\boldsymbol{X}$ 即不满足结合律式(10)。

综上所述，箝位矩阵公钥系统满足计算复杂性意义下的强度和安全性。

5. 与 RSA、Elgamal 密码的比较

首先，一个重要的不同是密钥生成的难易程度的差异，RSA 需要寻找两个大素数，Elgamal 的有限域依赖大素数来构造；其次，RSA 的一对公私钥是关于欧拉数模余运算互逆的，当已知一个密钥，求另外一个密钥需要使用辗转相除法来计算。而本文算法在密钥生成时，不需要依赖大素数，只要生成一个指定位长的随机数方阵作为种子，进行若干次矩阵乘法即可获得。最后，RSA、Elgamal 对明文加密处理是逐点式的，而本文算法是批处理模式。

RSA 密钥长度与选定的大素数数字长短有关，且为增加破解强度而延长密钥长度的代价是搜寻更大的素数；本文算法的密钥长度增长有两个维度，其中以增加矩阵的阶数的方式扩充密钥，增加破解强度，是较容易做到的。在加密解密运算方面，RSA、Elgamal 都是运算可交换的，而本文算法基于矩阵乘法，运算有左乘右乘之分。在量子计算里，由于量子的物理性态决定了量子适合对称作用的并发模拟，对非交换的矩阵乘法的并行性模拟反而是其弱项，因此基于矩阵乘法的密码有适应后量子密码时代的特征[11] [12] [13] [14] [15]。表 1 总结了与 RSA、Elgamal 比较的分析。

Table 1. Comparison with RSA and Elgamal
表 1. 与 RSA、Elgamal 的比较

算法	密钥生成难易	密钥长度扩充	明文处理模式	加解密运算
RSA	需要 2 个大素数	不易	点处理	可交换
Elgamal	大素数	不易	点处理	可交换
本文算法	随机数方阵	容易	批处理	左乘、右乘

6. 加解密实验

算法的数据试验采用 $k = 3$、$n = 10$ 的参数，即运用箝位指数为 7 的 10 阶矩阵作实验，则定理 3 中的 $\boldsymbol{A}$、$\boldsymbol{B}$ 作为种子是 10 阶 7 位(十进制位)长随机数构成的矩阵。由于 MSVC 系统库里的 rand()算法的随机数上界的限制，实际使用的是 6 位长随机数的种子，而生成的其中一个密钥如图 2 所示。

4840561 3575950 6112360 5241180 6125830 385600 1733800 7910630 7733220 9022910
5746580 8185381 6911040 4709280 6760690 7307130 2032580 4545240 6634190 8324160
1239650 2338340 3086811 561680 8766520 4056270 8141510 4980260 1324210 8621400
9977420 6188510 9516110 8079301 9835890 4530970 5592030 3880620 5728710 9117550
3984430 438810 9150170 5069880 5920491 1309220 7785400 5114430 5648660 519310
9590260 9090560 7931900 4593920 4400900 6939631 6536430 1911010 4766510 6955110
830800 2329340 9154510 3297990 6898510 4880480 5721641 8251390 2574210 9996500
7984580 9117110 5653540 3353650 3724880 5471350 9922830 719111 8711080 9087260
7901580 1769700 2407440 8312220 3994730 918220 1078060 9922470 9304781 8932180
3196020 7423230 3228060 3942230 3980070 1758560 620000 3350440 7873430 363881

Figure 2. Key example
图 2. 密钥示例

对如图 3 所示的明文进行加密，加密结果如图 4 所示。

85591 25431 44627 110083 30663 113049 210117 313168 258096 90416
60175 319738 130163 274143 270229 72693 63908 281369 124094 6467
125301 163259 44522 102152 21856 154337 212693 10904 73793 214935
181529 194119 286317 53109 292081 261238 52272 148086 294868 24214
195374 201644 245205 104714 52630 126952 57757 76612 153744 1414
167549 154745 246943 160771 10630 152792 279769 154854 48366 82815
310363 35932 116080 51256 76448 78550 148326 219822 95695 108308
320711 124269 109571 108635 107333 85374 21203 6598 175009 73438
122236 306554 59433 70914 188752 161979 132523 243766 139509 125016
150205 50995 198126 265632 268657 173420 303275 244732 95192 165315

Figure 3. Plaintext example
图 3. 明文示例

4541001 2146471 1562177 8360023 9576633 3175559 6855387 1265058 8628366 9103276
708865 8350628 6929313 5155773 4918069 984383 3989158 5720509 8805884 8528337
9420741 623669 2579672 2607812 5311586 4031827 7329893 3501464 4221903 1517635
6061219 9183849 9944617 3277569 1211761 3136708 3545202 1119016 9911358 1771574
7570224 8272524 1458825 1018084 5463690 1411962 8400947 4055282 2905224 9723744
8134369 1718055 2705073 9082931 1798830 1504362 419569 1381334 4052056 2555555
8777813 6162822 7931160 9894356 751168 5746990 8558686 5015062 9802525 8942738
8712081 6175279 6849151 9091375 5426303 5893824 2874523 2298788 2445749 6576658
5196066 1420864 7289313 2955344 4974762 1292339 4224943 8439446 923019 8768716
152345 5133725 9485126 7811692 4491517 872040 1478435 709722 1343782 856545

Figure 4. Encrypted result
图 4. 加密结果

对该密文解密后的结果如图 5 所示。

85591 25431 44627 110083 30663 113049 210117 313168 258096 90416
60175 319738 130163 274143 270229 72693 63908 281369 124094 6467
125301 163259 44522 102152 21856 154337 212693 10904 73793 214935
181529 194119 286317 53109 292081 261238 52272 148086 294868 24214
195374 201644 245205 104714 52630 126952 57757 76612 153744 1414
167549 154745 246943 160771 10630 152792 279769 154854 48366 82815
310363 35932 116080 51256 76448 78550 148326 219822 95695 108308
320711 124269 109571 108635 107333 85374 21203 6598 175009 73438
122236 306554 59433 70914 188752 161979 132523 243766 139509 125016
150205 50995 198126 265632 268657 173420 303275 244732 95192 165315

Figure 5. Decryption result
图 5. 解密结果

为验证末位随机化预处理的效果，考虑上面明文的另一预处理版本(即仅末位不同)，见图 6。

85599 25439 44620 110086 30663 113042 210112 313164 258096 90416
60172 319731 130165 274148 270226 72699 63908 281367 124099 6463
125307 163251 44522 102151 21858 154336 212699 10905 73798 214936
181521 194116 286315 53109 292088 261234 52279 148088 294864 24218
195375 201645 245205 104710 52634 126952 57758 76612 153746 1416
167544 154740 246941 160776 10637 152794 279766 154856 48365 82818
310363 35931 116089 51255 76446 78553 148320 219826 95696 108302
320712 124262 109570 108633 107336 85374 21203 6598 175001 73430
122239 306557 59438 70913 188755 161978 132528 243765 139506 125014
150202 50997 198127 265636 268656 173428 303278 244737 95194 165312

Figure 6. Another version of the plaintext example with randomization at the end
图 6. 末尾随机化的明文示例的另一版本

使用上面相同的密钥加密后的密文，如图 7 所示。

5521749 9551989 8959170 2877136 9405323 4092912 6316062 6433364 1266176 9062006
5617732 536111 4284285 1443658 6490606 4793539 474708 424217 4081529 179483
7924207 4490231 2878682 5206281 5870908 4234846 3818169 537725 4147308 9995896
672441 5162756 1467905 5335779 7820288 1642234 5996869 5796128 6854104 8708328
2348085 2437875 5404015 7293710 6982424 1204582 3950638 7237802 2064966 2683746
3472764 5568250 8391791 5164646 8733997 6971244 1278416 1992936 3509885 7216578
5459963 104161 6528229 6874675 7952516 5884633 3205450 2839486 4927246 4931422
8288132 6662162 2770330 1711263 5414226 334474 8939443 1846328 1074031 3066360
1490389 2696407 2292928 7214263 1665995 8166438 5439378 3830405 7279396 6836134
2347692 6550347 2518457 421576 6571486 4457038 7905488 4134797 5956844 6034402

Figure 7. Ciphertext encrypted using the key in Figure 2
图 7. 使用图 2 密钥加密后的密文

可见上面两段仅末位不同的明文在相同密钥处理后的密文差异极大，这个反映了箝位密码具有一定的随机性效应。将上面这段密文解密后的明文如下，见图 8。

85599 25439 44620 110086 30663 113042 210112 313164 258096 90416
60172 319731 130165 274148 270226 72699 63908 281367 124099 6463
125307 163251 44522 102151 21858 154336 212699 10905 73798 214936
181521 194116 286315 53109 292088 261234 52279 148088 294864 24218
195375 201645 245205 104710 52634 126952 57758 76612 153746 1416
167544 154740 246941 160776 10637 152794 279766 154856 48365 82818
310363 35931 116089 51255 76446 78553 148320 219826 95696 108302
320712 124262 109570 108633 107336 85374 21203 6598 175001 73430
122239 306557 59438 70913 188755 161978 132528 243765 139506 125014
150202 50997 198127 265636 268656 173428 303278 244737 95194 165312

Figure 8. Decryption after encryption of another version of the plaintext example
图 8. 明文示例另一版本加密后的解密

使用 RSA 与本文算法作运算时效对比实验。RSA 算法运行不包含大素数选取、密钥生成，只从明文加密再解密恢复的一轮运算时间着手，并且代码经过优化[16]；而本文算法包含生成两个随机种子矩阵，再进行密钥生成，然后对明文加密、再解密的一轮运算的时间总和，但代码未经过优化。由于双方对明文处理的长度都不一样，所以本文算法处理时间除以 n^2，这样可以显示在某个相同基准上衡量两个算法的运算占时(图 9 的纵轴)。从图 9 可以看出，对选取的 5 个算例(图 9 的横轴)，本文算法比 RSA 运算占时少了三十多倍。实验结果表明，本文算法在单位明文长度的计算效率上比 RSA 算法高三十倍。

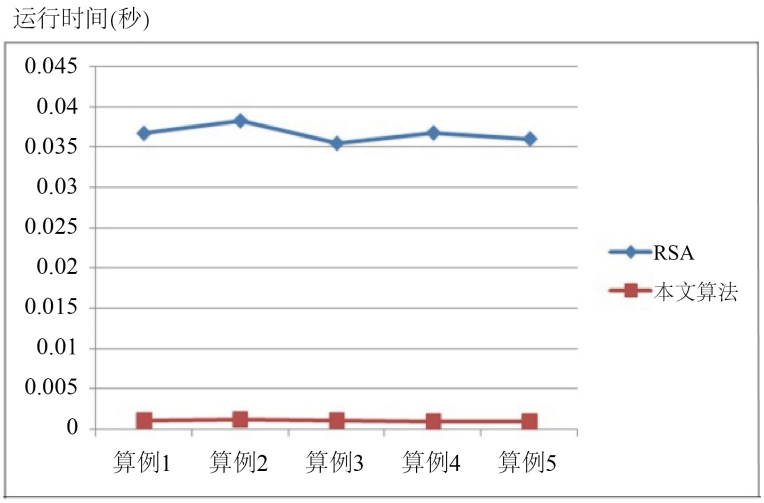

Figure 9. Comparison of running time between RSA and the algorithm in this paper
图 9. RSA 和本文算法的运行时效比较

7. 结语

传统的基于有限域或素数的密码系统都有依赖大素数的密钥生成的不易性，本文提出一个密钥生成极其容易的公钥密码系统，且具有因使用矩阵乘法而不易被未来的量子计算产生攻陷危机的优势。由于钳位矩阵与二维码维数匹配的自然特性，下一步拟探索将钳位公钥用于二维码的数字签名的研究。

参考文献

[1] Rivest, R., Shamir, A. and Adleman, L. (1978) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, **21**, 120-126. https://doi.org/10.1145/359340.359342

[2] Ibrahima, A., Chefranov, A. and Hamamreh, R. (2021) Ciphertext-Only Attack on RSA Using Lattice Basis Reduction. *The International Arab Journal of Information Technology*, **18**, 237-247. https://doi.org/10.34028/iajit/18/2/13

[3] Kumar, M. and Susan, S. (2021) Exploration and Implementation of RSA-KEM Algorithm. *Proceedings of ICSCS* 2021: *Soft Computing for Security Applications*, Springer, Cham, 161-179. https://doi.org/10.1007/978-981-16-5301-8_12

[4] Reegan, A.S. and Kabila, V. (2021) Highly Secured Cluster Based WSN Using Novel FCM and Enhanced ECC-ElGamal Encryption in IoT. *Wireless Personal Communications*, **118**, 1313-1329. https://doi.org/10.1007/s11277-021-08076-0

[5] Khoirom, M.S., Laiphrakpam, D.S. and Tuithung, T. (2021) Audio Encryption Using Ameliorated ElGamal Public Key Encryption over Finite Field. *Wireless Personal Communications*, **117**, 809-823. https://doi.org/10.1007/s11277-020-07897-9

[6] Yarmish, G., Yarmish, J. and Yarmish, J. (2021) Finding Large Primes. https://arxiv.org/ftp/arxiv/papers/1709/1709.09963.pdf

[7] Miller, G.L. (1976) Riemann's Hypothesis and Tests for Primality. *Journal of Computer and System Sciences*, **13**, 300-317. https://doi.org/10.1016/S0022-0000(76)80043-8

[8] Rabin, M.O. (1980) Probabilistic Algorithm for Testing Primality. *Journal of Number Theory*, **12**, 128-138. https://doi.org/10.1016/0022-314X(80)90084-0

[9] 陆成刚. 一种基于箝位矩阵的整数可逆性的公私钥密码系统[P]. 中国, CN202210465885. 2022-07-29.

[10] Apostol, T.M. (1967) Calculus, 2nd Ed., Vol. 1: One-Variable Calculus, with an Introduction to Linear Algebra. Blaisdell, Waltham, 143 p.

[11] Moldovyan, A.A, Moldovyan, D.N. and Moldovyan, N.A. (2022) A Novel Method for Developing Post-Quantum Digital Signature Algorithms on Non-Commutative Associative Algebras. *Information Management Systems*, **1**, 44-53. https://doi.org/10.31799/1684-8853-2022-1-44-53

[12] Moldovyan, A.A. and Moldovyan, N.A. (2022) Split Logarithm Problem and a Candidate for a Post-Quantum Signature Scheme. *Computer Science Journal of Moldova*, **30**, 121-142. https://doi.org/10.56415/csjm.v30.14

[13] Moldovyan, A. and Moldovyan, D. (2022) A New Method for Developing Signature Algorithms on Finite Non-Commutative Algebras. *Buletinul Academiei de Ştiinţe a Moldovei*. *Matematica*, **98**, 56-65. https://doi.org/10.56415/basm.y2022.i1.p56

[14] Nguyen, M.H., Hoang, C.N., Moldovyan, A.A., *et al*. (2022) A Novel Version of the Hidden Logarithm Problem for Post-Quantum Signature Algorithms. *Theoretical Computer Science*, **921**, 36-49. https://doi.org/10.1016/j.tcs.2022.03.040

[15] Minh, N.H. and Moldovyan, D.N. (2021) Post-Quantum Blind Signature Protocol on Non-Commutative Algebras. *Journal of Computer Science and Cybernetics*, **37**, 495-509. https://doi.org/10.15625/1813-9663/37/4/16023

[16] Kokke, Dudedsy Brian Evans, Etc. Tiny-Bignum-C. https://github.com/kokke/tiny-bignum-c