基于同余简单半环上循环矩阵作用的密钥交换协议

Otero Sanchez, Alvaro*[1]

[1]Department of Mathematics, University of Almería 04120 , Almería, Spain

2025 年 5 月 3 日

摘要

我们提出了一种基于循环矩阵在同余简单半环上的矩阵操作的新密钥交换协议。我们描述了如何计算具有实施该协议所需特性的矩阵。此外，我们提供了对其计算成本的分析以及其针对已知攻击的安全性分析。

1 介绍

[1] 的基础论文被认为是现代密码学的开端。在其原始表述中，一个循环群 $\mathbb{Z}_n$ 被用作密钥交换协议的代数设置。后来，基于椭圆曲线的类似协议由 [3] 和 [5] 独立提出。

这些协议已成为密码学的标准，并且其使用范围广泛。然而，在 [13] 中，Shor 提出了一种能够在代数环境中解决离散对数问题的量子算法。由于这些密码协议的安全性依赖于这个问题的难度，一台足够大的量子计算机将能够破解大多数今天的密钥交换协议。因此，科学界已经启动了寻找新的后量子密码学协议的研究，即那些能够抵御量子攻击的协议。

一种这样的提议在 [4] 中提出，作者引入了交换半群的使用。该协议的安全性依赖于以下问题：

*Autor de correspondencia: aos073@ual.es

问题 1.1. 令 $(G,\cdot)$ 是一个半群，S 是一个集合，并且令 $\varphi : G \times S \to S$ 是 G 在 S 上的作用，即，φ 满足 $\varphi(g \cdot h, s) = \varphi(g, \varphi(h, s))$。半群作用问题表示，在给定 $x \in S$ 和 $y \in \varphi(G, x)$ 的情况下，找到 $g \in G$ 使得 $\varphi(g, x) = y$。

在他们的工作中，作者定义了一个使用半模的新密钥交换协议的一般框架。半模类似于模，但作用的环被半环替换，底层结构是一个交换幺半群。作为一个明确的例子，他们使用了同余单半环。在这种情况下，半环中心上的多项式通过在两个固定的矩阵上进行求值和乘法来作用于半环上的矩阵。他们提供了一个使用六个元素的半环的例子。

然而，在 [14] 中，作者通过对半环的操作表得出的方程组进行求解，对这个特定示例进行了密码分析。使用其他半环的协议的有效性一直是一个开放问题，直到 [11]，作者开发了一种针对任何同余简单半环上该协议的一般攻击方法，仅假设了在协议定义中使用的多项式的次数有一个界限。

在本文中，我们提出了一种基于同余简单半环的新密钥交换协议。为此，我们将指数视为 $\mathbb{N}$ 在该半环上的矩阵上的一个操作。此密钥交换可以被视为 [4] 中协议的一个特例，但具有不同的操作。此外，我们分析了其抵抗已知攻击的能力并评估了其计算成本。

2 数学设定

首先，我们将介绍半群的概念

定义 2.1: A 半群是一个配备内部二元运算 $\cdot : S \times S \to S$ 的集合 S，使得该运算是结合的；也就是说，对于每一个 $a, b, c \in S$，我们有 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$。

我们说 S 是交换的，如果除此之外，该运算对于所有 $a, b \in S$ 满足 $a \cdot b = b \cdot a$。

现在，我们将回顾众所周知的循环矩阵

定义 2.2: 令 R 为一个环。矩阵 $C \in \mathrm{Mat_n(R)}$ 被称为循环矩阵，如果存在 $c_0, c_1, \cdots, c_{n-1} \in R$ 使得

$$C = \begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \cdots & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ c_2 & c_1 & c_0 & \cdots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & c_{n-3} & \cdots & c_0 \end{pmatrix}$$

我们将 C 表示为 $C = Circ(c_0, \cdots, c_{n-1})$

一个关于循环矩阵结构的著名结果是

定理 2.3. 集合 $Circ_n(R)$ 由在 R 上的 $n \times n$ 圆形矩阵组成，形成 $\mathrm{Mat_n(R)}$ 的一个交换子环。

由于这个条件，我们可以定义一个新的动作

定理 2.4. 令 S 为一个半群，$T \subset S$ 是一组交换元素，即：

$$ab = ba \forall a, b \in T \tag{1}$$

则

$$\begin{aligned} Circ_n(\mathbb{N}) \times T^n &\longrightarrow T^n \\ (C, (v_i)_{i=0}^{n-1}) &\longmapsto (\prod v_j^{a_{j-i}})_{i=0}^{n-1} \end{aligned}$$

是乘法半群 $Circ_n(\mathbb{N})$ 在 T^n 上的作用。它将被表示为 Cv

证明. 显然，如果 $C \in Circ_n(\mathbb{N}), v \in T^n$，则 $Cv \in T^n$，因为 Cv 的元素是彼此交换的元素的乘积。我们必须证明对于所有 $A, B \in Circ_n(\mathbb{N}), v \in T^n$ 都有 $A(Bv) = (AB)v$。

$$A = \begin{pmatrix} a_0 & a_{n-1} & a_{n-2} & \cdots & a_1 \\ a_1 & a_0 & a_{n-1} & \cdots & a_2 \\ a_2 & a_1 & a_0 & \cdots & a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_0 \end{pmatrix}$$

$$b = \begin{pmatrix} b_0 & b_{n-1} & b_{n-2} & \cdots & b_1 \\ b_1 & b_0 & b_{n-1} & \cdots & b_2 \\ b_2 & b_1 & b_0 & \cdots & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n-1} & b_{n-2} & b_{n-3} & \cdots & b_0 \end{pmatrix}$$

然后，

$$AB = \begin{pmatrix} \sum_{i=0}^{n-1} a_i b_{n-i} & \sum_{i=0}^{n-1} a_i b_{n-1-i} & \sum_{i=0}^{n-1} a_i b_{n-2-i} & \cdots & \sum_{i=0}^{n-1} a_i b_{n+1-i} \\ \sum_{i=0}^{n-1} a_i b_{n+1-i} & \sum_{i=0}^{n-1} a_i b_{n-i} & \sum_{i=0}^{n-1} a_i b_{n-1-i} & \cdots & \sum_{i=0}^{n-1} a_i b_{n+2-i} \\ \sum_{i=0}^{n-1} a_i b_{n+2-i} & \sum_{i=0}^{n-1} a_i b_{n+1-i} & \sum_{i=0}^{n-1} a_i b_{n-i} & \cdots & \sum_{i=0}^{n-1} a_i b_{n+3-i} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n-1} a_i b_{n-1-i} & \sum_{i=0}^{n-1} a_i b_{n-2-i} & \sum_{i=0}^{n-1} a_i b_{n-3-i} & \cdots & \sum_{i=0}^{n-1} a_i b_{n-i} \end{pmatrix}$$

其中所有下标都取模 n。因此，$AB = Cir(c_0, \cdots, c_{n-1})$ 与 $c_k = \sum_{i=0}^{n-1} a_i b_{k-i}$。现在，我们有

$$\begin{aligned} A(Bv) =& A\left(\prod_{j=0}^{n-1} v_j^{b_{-i+j}}\right)_{i=0}^{n-1} = \left(\prod_{i=0}^{n-1}\left(\prod_{j=0}^{n-1} v_j^{b_{-i+j}}\right)^{a_{i-k}}\right)_{k=0}^{n-1} = \left(\prod_{i=0}^{n-1}\prod_{j=0}^{n-1} v_j^{b_{-i+j}a_{i-k}}\right)_{k=0}^{n-1} \\ =& \left(\prod_{j=0}^{n-1}\prod_{i=0}^{n-1} v_j^{b_{-i+j}a_{i-k}}\right)_{k=0}^{n-1} = \left(\prod_{j=0}^{n-1} v_j^{\sum_{i=0}^{n-1} b_{-i+j}a_{i-k}}\right)_{k=0}^{n-1} = \\ =& \left(\prod_{j=0}^{n-1} v_j^{\sum_{i=0}^{n-1} a_i b_{j-k-i}}\right)_{k=0}^{n-1} = \left(\prod_{j=0}^{n-1} v_j^{c_{k-j}}\right)_{k=0}^{n-1} = (AB)v \end{aligned}$$

□

之前的动作是 [4] 的一个特例，在这里我们考虑由 T 生成的 $\mathbb{N} - module$ M，并对所有 $n \in \mathbb{N}, t \in M$ 施加 $\mathbb{N}-$ 动作 $n \cdot t = t^n$。我们将使用半环上的矩阵的交换子集作为 T。

定义 2.5: 一个集合 R 带有两个内部运算 $+$ 和 $\cdot$ 被称为半环，如果这两个运算是结合的并且满足对于每一个 $a, b, c \in R$ 都有 $a \cdot (b + c) = a \cdot b + a \cdot c$ 和 $(b + c) \cdot a = b \cdot a + c \cdot a$。我们说 S 是加法上（分别乘法上）交换的，当且仅当加法（分别乘法）满足交换律。我们说当两个运算都满足交换律时，R 是可交换的。

关于用于密码学目的的半环研究是由 [4] 开始的，该研究重点在于用于密码学应用的同余简单半环。

定义 2.6: 半环 R 上的同余关系是一个等价关系 $\sim$，使得

$$a \sim b \Rightarrow \begin{cases} a + c & \sim & b + c \\ c + a & \sim & c + b \\ a \cdot c & \sim & b \cdot c \\ c \cdot a & \sim & c \cdot b \end{cases}$$

对于每一个 $a, b, c \in R$。一个仅允许平凡同余关系 id_R 和 $R \times R$ 的半环 R 被称为同余单的。

以下结果首次出现在 [6] 中，建立了同余简单半环的分类。

定理 2.7. *([6, Theorem 4.1] 令 R 为一个有限的、加法交换的、同余单半环。则以下之一成立：)*

1. $|R| \leq 2$.

2. $R \cong \mathrm{Mat_n}(\mathbb{F_q})$ 对于某个有限域 $\mathbb{F}_q$ 和某个 $n \geq 1$。

3. R 是一个素数阶的零乘环。

4. R 是加法幂等的，即对于每一个 $x \in R$ 都有 $x + x = x$。

5. R 包含一个吸收元 ∞，即对于每一个 $x \cdot \infty = \infty \cdot x = \infty$ 和 $x \in R$ 及 $R + R = \{\infty\}$。

为了获得一个交换集，我们将使用以下来自 [6] 的结果

定义 2.8: 令 R 为具有中心

$$C_R = \{r \in R \mid rs = sr \text{ for all } s \in R\},$$

的半环，并令 $A \in \mathrm{Mat}_{n\times n}(R)$ 为一个矩阵。定义 $C_R[A]$ 为在 A 上且系数在 C_R 中的多项式的集合。

引理 2.9: 令 R，C_R 和 A 如上所述。集合 $C_R[A]$ 是矩阵半环 $\mathrm{Mat}_{n\times n}(R)$ 的一个交换子半环。

因此，我们将使用半环上矩阵的乘法半群，并将作为交换子集使用的元素为 $C_R[A]$，带有 $A \in Mat_n(R)$ 和 $n \in \mathbb{N}$。此外，在 [6] 中也证明了

引理 2.10: 令 R 是一个具有加法交换性的含 1 和 0 的半环，并且令 $\sim$ 是 $\mathrm{Mat}_n(R)$ 上的一个同余关系。则存在一个在 R 上的同余关系 $\sim_0$，使得

$$A \sim B \in \mathrm{Mat}_n(R) \quad \Leftrightarrow \quad a_{ij} \sim_0 b_{ij}, \quad \forall 0 \leq i, j \leq n.$$

因此我们可以确保新的半环也将是简单的。在这种情况下，我们可以引入以下密钥交换协议

协议 2.11. 爱丽丝和鲍勃同意使用一个简单的半环 R，一个高阶的矩阵 $M \in \mathrm{Mat_n(R)}$，以及 $v = (M_i)_{i=0}^{n-1} \in C_R[M]^n$，并将它们公开。

1. 爱丽丝选择了 $A \in Circ_n(\mathbb{N})$ 并公开了它 $pk_1 = Av$

2. 鲍勃选择了 $B \in Circ_n(\mathbb{N})$ 并将其公开 $pk_2 = Bv$

3. 爱丽丝计算 Apk_2，鲍勃计算 Bpk_1

公共密钥是

$$Apk_2 = A(Bv) = (AB)v = (BA)v = B(Av) = Bpk_1$$

为了计算目的，Alice 和 Bob 可以采用矩阵 A, B，其中 $a_i, b_i \leq m$ 被视为双方可计算假设的上界 $m \in \mathbb{N}$。

协议的安全性依赖于以下问题

问题 2.12. 令 R 为一个简单的半环和 $M \in \mathrm{Mat_n(R)}$。给定 $v, Av, Bv \in C_R[M]^n$ 和 $A, B \in Circ_n(\mathbb{N})$，计算 ABv。

一个更容易的问题暗示了解决前一个问题的方法是

问题 2.13. 令 R 为一个简单半环和 $M \in \mathrm{Mat_n(R)}$。给定 $v, Av \in C_R[M]^n$ 和 $A \in Circ_n(\mathbb{N})$，找到 $C \in Circ_n(\mathbb{N})$ 使得 $Av = Cv$。

为了避免暴力攻击，我们需要确保集合 $C[A]$ 足够大。为此，我们需给出以下定义

定义 2.14: 令 $a = \{a_k\}_{k\in\mathbb{N}}$ 是有限集中的一个序列，满足

$$a_n = a_m \Rightarrow a_{n+1} = a_{m+1}.$$

顺序 $\mathrm{ord}(a)$ 的 a 是最小的正整数 m，使得存在 $k < m$ 有 $a_k = a_m$。预备周期 $\mathrm{pr}(a)$ 的 a 是最大的非负整数 m，使得对于所有的 $k > m$ 我们有 $a_k \neq a_m$。周期 $\mathrm{per}(a)$ 的 a 是最小的正整数 m，使得存在一个整数 N，满足

$$a_{m+k} = a_k \quad \text{for all } k > N.$$

如果 g 是半群的元素，则我们设定

$$\mathrm{ord}(g) = \mathrm{ord}(\{g^n\}_{n\in\mathbb{N}}), \quad \mathrm{per}(g) = \mathrm{per}(\{g^n\}_{n\in\mathbb{N}}), \quad \mathrm{pr}(g) = \mathrm{pr}(\{g^n\}_{n\in\mathbb{N}}).$$

现在，为了给出一些界限，我们需要兰道函数。

$$g(n) = \max\{\mathrm{ord}(\sigma) \mid \sigma \in S_n\} = \max\{\mathrm{lcm}\{a_1, a_2, \ldots, a_m\} \mid a_i > 0, a_1 + \cdots + a_m = n\}.$$

在 [9] 中证明了

$$n\ln(n) \leq \ln(g(n)) \leq \sqrt{n}\ln(n)\left(1 + \frac{\ln\ln(n)}{2\ln(n)}\right)$$

最后，在 [4] 中建立了这一点

定理 2.15. 令 $n \in \mathbb{N}$ 和 R 是一个具有 *0* 和 *1* 的半环，并且中心是 C。然后存在一个 $n \times n$ 矩阵 M，其元素在 R 中，使得 M 的阶大于 $g(n)$。特别是，$C[M]$ 的大小也大于 $g(n)$。

为了获得合适的矩阵，我们将使用以下结果

引理 2.16: 令 $P, A \in Mat_{n\times n}(R)$ 且 P 可逆。则 $ord(A) = ord(PAP^{-1})$。

证明. 我们有 $PAP^{-1} \cdot PAP^{-1} = PA^2P^{-1}$。因此，$(PAP^{-1})^n = PA^nP^{-1}$。此外，

$$(PAP^{-1})^n = (PAP^{-1})^k \iff PA^nP^{-1} = PA^kP^{-1} \iff A^n = A^k.$$

在最后一个等价关系中，我们将两边都乘以了 P 和 P^{-1}。 □

为了继续进行，我们需要理解我们半环上的可逆矩阵 R。为此，我们首先引入半群的和与直和的概念。

定义 2.17: 令 $(R, +)$ 为一个有限半群，并且令 $A, B \leq R$ 子半群。半群 A, B 的和被定义为

$$A + B := \{a + b : a \in A, b \in B\}$$

定义 2.18: 设 $(R, +)$ 是一个半群，并且设 $A, B \leq R$ 是两个子半群，使得 $A \cap B = \emptyset$ 并且对于所有 $c \in A + B$，存在唯一的元素 $a \in A$ 和 $b \in B$ 使得 $a + b = c$。然后和 $A + B$ 被称为直和，并用 $A \oplus B$ 表示。

由于直接和，我们可以定义不可约和可约半群的概念

定义 2.19: 令 $(R, +)$ 是一个有限半群。我们说 R 是可约的，如果存在适当的子半群 $A, B \lneq R$ 使得 $R = A \oplus B$ 成立。否则，我们说 R 是不可约的。

现在我们将展示一个具有中性元素的半群不可约性的充分条件。为此，我们将使用以下简单的证明事实

引理 2.20: 令 $(R, +)$ 是一个具有基数 n 的有限可约半群，带有 $R = A \oplus B$。那么 $1 < |A|, |B| < n$。

下列定理是一个新的结果，给出了有限半群不可约的一个充分条件

定理 2.21. 令 $(R, +)$ 为一个有限半群。如果存在一个元素 $x \in R$ 使得

$$z + y = x \Longrightarrow z = x \text{ ó } .y = x$$

$$x + y = y + x = x \; \forall y \in R$$

然后 R 是不可约的。

证明. 证明通过矛盾进行。假设存在子半群 A, B，使得 $A \oplus B = R$，并令 x 为前面提到的元素。那么，由于 $R = A \oplus B$，存在 $(a, b) \in A \times B$，使得 $a + b = x$。由第一个性质可知，这表明要么是 $a = x$ 要么是 $b = x$。不失一般性，假设 $x \in A$。

由之前的引理，我们有 $|B| \geq 2$，所以存在不同的元素 $d, b \in B$ 与 $d \neq b$。然而，这暗示了 $x + d = x + b = x$，这与表示的唯一性相矛盾。因此，假设必须是错误的。 □

示例 2.1: 在 [4] 中介绍了一个用于其密码学应用的含有 20 个元素的半环。

+	0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	1
0	0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	1
a	a	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	1
b	b	b	b	c	e	e	f	g	h	i	k	k	l	m	n	o	p	q	r	1
c	c	c	c	c	f	f	f	h	h	i	l	l	l	1	n	p	p	q	r	1
d	d	d	e	f	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	1
e	e	e	e	f	e	e	f	g	h	i	k	k	l	m	n	o	p	q	r	1
f	f	f	f	f	f	f	f	h	h	i	l	l	l	1	n	p	p	q	r	1
g	g	g	g	h	g	g	h	g	h	i	m	m	1	m	n	o	p	q	r	1
h	h	h	h	h	h	h	h	h	h	i	1	1	1	1	n	p	p	q	r	1
i	i	i	i	i	i	i	i	i	i	i	n	n	n	n	n	q	q	q	r	n
j	j	j	k	l	j	k	l	m	1	n	j	k	l	m	n	o	p	q	r	1
k	k	k	k	l	k	k	l	m	1	n	k	k	l	m	n	o	p	q	r	1
l	l	l	l	l	l	l	l	1	1	n	l	l	l	1	n	p	p	q	r	1
m	m	m	m	1	m	m	1	m	1	n	m	m	1	m	n	o	p	q	r	1
n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	q	q	q	r	n
o	o	o	o	p	o	o	p	o	p	q	o	o	p	o	q	o	p	q	r	p
p	p	p	p	p	p	p	p	p	p	q	p	p	p	p	q	p	p	q	r	p
q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	q	r	q
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
1	1	1	1	1	1	1	1	1	1	n	1	1	1	1	n	p	p	q	r	1

$\cdot$	0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0	0	0	0	a	a	a	a	a	b	b	b	c	a
b	0	0	0	0	a	a	a	b	b	c	a	a	a	b	c	b	b	c	c	b
c	0	a	b	c	a	b	c	b	c	c	a	b	c	b	c	b	c	c	c	c
d	0	0	0	0	0	0	0	0	0	0	d	d	d	d	d	g	g	g	i	d
e	0	0	0	0	a	a	a	b	b	c	d	d	d	e	f	g	g	h	i	e
f	0	a	b	c	a	b	c	b	c	c	d	e	f	e	f	g	h	h	i	f
g	0	0	0	0	d	d	d	g	g	i	d	d	d	g	i	g	g	i	i	g
h	0	a	b	c	d	e	f	g	h	i	d	e	f	g	i	g	h	i	i	h
i	0	d	g	i	d	g	i	g	i	i	d	g	i	g	i	g	i	i	i	i
j	0	0	0	0	0	0	0	0	0	0	j	j	j	j	j	o	o	o	r	j
k	0	0	0	0	a	a	a	b	b	c	j	j	j	k	l	o	o	p	r	k
l	0	a	b	c	a	b	c	b	c	c	j	k	l	k	l	o	p	p	r	l
m	0	0	0	0	d	d	d	g	g	i	j	j	j	m	n	o	o	q	r	m
n	0	d	g	i	d	g	i	g	i	i	j	m	n	m	n	o	q	q	r	n
o	0	0	0	0	j	j	j	o	o	r	j	j	j	o	r	o	o	r	r	o
p	0	a	b	c	j	k	l	o	p	r	j	k	l	o	r	o	p	r	r	p
q	0	d	g	i	j	m	n	o	q	r	j	m	n	o	r	o	q	r	r	q
r	0	j	o	r	j	o	r	o	r	r	j	o	r	o	r	o	r	r	r	r
1	0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	1

此半环是不可约的，因为 r 满足前一个引理的条件。

定义 2.22: 令 R 是一个半环，令 $A \in \mathrm{Mat}_{n\times n}(R)$。我们说 A 是一个广义置换矩阵，如果每一行和每一列恰好包含一个非零项，并且这个项是可逆的。

在文章 [10] 中，建立了半环的加法半群结构与其上矩阵可逆性之间的有趣关系。

定理 2.23. 令 $(R,+,\cdot)$ 为一个有限半环，使得 $(R,+)$ 不可约，并令 $A \in \mathrm{Mat}_{n\times n}(R)$。则 A 可逆当且仅当 A 是一个广义置换矩阵。

现在，我们将看到如何获得所需的矩阵。首先，我们取一个无同余半环 R 和一个自然数 n。我们选择元素 $a_i \in \mathbb{N}$，对于 $i = 1,\ldots,k$，使得 $\mathrm{LCM} = \mathrm{lcm}(a_1,\ldots,a_k)$ 足够大，并且满足约束条件 $\sum_{i=1}^{n} a_i \le n$。然后，我们按照定理 2.15构建与该划分相关的矩阵。这个矩阵将具有以下形式：

$$\left(\begin{array}{c|c|c|c|c} T_{a_1} & 0 & \cdots & 0 & 0 \\ \hline 0 & T_{a_2} & \cdots & 0 & 0 \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline 0 & 0 & \cdots & T_{a_r} & 0 \\ \hline 0 & 0 & \cdots & 0 & Id_s \end{array}\right)$$

接下来，我们对块对角线以上的元素进行修改，得到如下形式的矩阵：

$$M = \left(\begin{array}{c|c|c|c|c|c} T_{a_1} & A_{1,2} & A_{1,3} & \cdots & A_{1,r} & A_{1,r+1} \\ \hline 0 & T_{a_2} & A_{2,2} & \cdots & A_{2,r} & A_{2,r+1} \\ \hline 0 & 0 & T_{a_3} & \cdots & A_{2,r} & A_{2,r+1} \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline 0 & 0 & 0 & \cdots & T_{a_r} & A_{r,r+1} \\ \hline 0 & 0 & 0 & \cdots & 0 & Id_s \end{array}\right)$$

其中矩阵 $A_{i,j} \in Mat_{a_i \times a_j}(R)$ 是半环 R 中的矩形矩阵。由于对角线及其下方的零块保持不变，该矩阵的幂将具有如下形式：

$$M^s = \left(\begin{array}{c|c|c|c|c|c} T_{a_1}^s & B_{1,2} & B_{1,3} & \cdots & B_{1,r} & B_{1,r+1} \\ \hline 0 & T_{a_2}^s & B_{2,2} & \cdots & B_{2,r} & B_{2,r+1} \\ \hline 0 & 0 & T_{a_3}^s & \cdots & B_{2,r} & B_{2,r+1} \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline 0 & 0 & 0 & \cdots & T_{a_r}^s & B_{r,r+1} \\ \hline 0 & 0 & 0 & \cdots & 0 & Id_s \end{array}\right)$$

使用 $B_{i,j} \in \mathrm{Mat}_{a_i \times a_j}(R)$。

由先前的结果可知，唯一的可逆矩阵是广义置换矩阵。因此我们可以改变矩阵的行和列，使其幂次计算变得更加困难。由于这些改变与 S_n 建立了一一对应关系，共有 $n!$ 种可能的变化。对于足够大的 n，暴力破解以逆转这一过程在计算上是不可行的。

示例 2.2: 我们将采用在 [4] 中介绍的 20 个元素的半环，并令 $n = 6$。我们取分区 $[2, 3]$。然后，$\mathrm{lcm}(2, 3) = 6$，我们得到分块矩阵：

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

在这个矩阵中，我们随机修改主对角线以上的元素，得到

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & r & b & l \\ 0 & 0 & 0 & 1 & 0 & e \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

现在，我们通过一个广义置换矩阵对该矩阵进行共轭。在进行计算后，我们得到

Output

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \mathrm{e} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & \mathrm{r} & 0 & \mathrm{b} & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Generalized permutation matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

示例 2.3: 现在，我们将展示一个至少具有 280 个不同幂次且大小为 20×20 的示例。为了做到这一点，只需注意到集合 $[8,5,7]$ 满足其和为 20 并且它们两两互质，因此它们的最小公倍数是 $8\cdot5\cdot7 = 280$。由此划分得出的矩阵将是：

$$
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

在这个矩阵中，我们随机修改对角线以上的元素，得到

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & f & k & r & 0 & c & g & f & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & o & 0 & p & 0 & h & 0 & g & 0 & 0 & 0 & 0 & e \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & m & 0 & g & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & l & l & j & b & 0 & 0 & 0 & p & 0 & 0 & q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & k & 0 & o & 0 & 0 & i & 0 & 0 & 0 & n & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & j & 0 & 0 & k & b & 0 & 0 & p & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & d & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & q & g & 0 & p & d & d \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & a & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & o & b & 1 & 0 & r & 0 & l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & c & 0 & o & m & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & o & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

现在，我们将这个矩阵通过一个广义置换矩阵进行共轭。在进行计算后，我们得到

输出

$$\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & d & a & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n & 0 & k & 0 & 0 & i & 1 & 0 & 0 & 0 & 0 & 0 & 0 & o & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
p & 1 & 0 & 0 & 0 & k & 0 & 0 & 0 & 0 & 0 & 0 & b & j & 0 & 0 & 0 & 0 & 0 & 0 \\
f & 0 & 0 & 0 & 0 & r & 0 & 0 & 0 & 1 & f & k & 0 & 0 & 0 & 0 & 0 & g & 0 & c \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b & 0 & 1 & m & 0 & g & 0 \\
0 & 0 & 0 & e & 0 & g & 0 & 0 & 1 & 0 & h & 0 & 0 & 0 & 0 & 0 & o & 0 & p & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & o & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & c & 0 & 0 & 0 & 0 & o & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
q & 0 & l & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p & b & 0 & 0 & l & 0 & j & 0 \\
d & 0 & 1 & d & 0 & q & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & p & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & l & 0 & b & 0 & 0 & 0 & 0 & 0 & o & 1 & 1 & 0 & 0 & 0 & r & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}$$

广义置换矩阵

$$\begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0
\end{pmatrix}$$

3 密钥交换协议及其安全性

首先，我们将看到密钥交换的计算成本设置的成本

定理 3.1. 矩阵 $M \in Mat_m(R)$ 作为多项式 $h \in \mathbb{N}$ 和 n 基元素的界。设置协议的成本最多为 $O(nH)$ 个矩阵运算，或 $O(nhm^3)$ 个操作在 R

证明. 对于基底的每个元素，我们最多需要计算 h 次幂，并且由于每次都可以通过一个阶为 m 的矩阵乘法来实现，因此开销为 $O(2log_2(h)m^3)$，之后，我们必须进行求和，最多有 H 个矩阵相加，得到在 R 中的 $O(2log_2(h)m^3 + 2log_2(h)) = O(2log_2(h)m^3)$ 次操作。由于基中有 n 个元素，总成本是 $O(n2log_2(h)m^3)$ □

算法设置后的成本

定理 3.2. 若 $A \in Circ_n([0,k]$，则计算 $A(M_i)_{i=1}^n$ 的成本是 $O(n(2log_2k+n)m^3)$ 次操作在 R

证明. 每个产品或幂次具有成本 m^3。我们需要计算直到 k 的幂次，这需要 $O(2log_2k)$ 次幂运算，在 R 中产生 $O(2log_2km^3$，之后还需要乘以所有矩阵，因此在 R 上增加了 $O(nm^3)$ 次操作。需要重复此操作 n 次，得到 $O(n(2log_2k+n)m^3)$ 操作在 R □

现在，我们将看到前一个协议针对已知攻击的安全性。

3.1 暴力攻击和随机攻击

攻击者可能会尝试对协议进行暴力破解或随机攻击。如果 m 是循环矩阵系数的上界，那么攻击者必须在暴力攻击中尝试 m^n 个不同的矩阵。为了避免这种情况，需要确保 m 足够大，以使 m^n 不可行。

然而，这并不足够，因为如果集合 $Pkey = \{C \in Circ_n(\mathbb{N}); Cv = Av\}$ 过大，暴力或随机攻击可以轻易找到一个实例。计算实验表明，对于形式为 2.3的矩阵，随机攻击对其无效。此外，我们呈现了一个结果，确保在某些条件下，我们的私钥是唯一的。

定理 3.3. 如果 $n = 1$ 和 $A = a_1 \leq pord(M_1) - 1$，则 $Xv = Av$ 的解唯一 $X = A$

定理 3.4. 令 $n \in \mathbb{N}$。令 $v = \{M^{b_i}\}_{i=1}^n$ 使得 $B = Cir[b_1, \cdots, b_n]$ 与 $det(B) \neq 0$，并且 A 使得 $\sum_i a_i b_{i+k} \leq pord(M) - 1 \forall k = 0 \cdots, n-1$，则 $Xv = Av$ 的解只有 $X = A$

3.2 Pohlig-Hellman 类型攻击

首先，我们回顾以下众所周知的结果：

定理 3.5. 令 $n \in \mathbb{N}$，则 $U(Mat_n(\mathbb{N}))$ 是置换矩阵。

作为推论，我们有：

定理 3.6. 令 $n \in \mathbb{N}$，则 $U(Circ_n(\mathbb{N}))$ 是置换矩阵。

如 [12] 所示，Shanks 的小步大步方法以及 Pollard-rho 类型的攻击都需要大量逆元素才能成为有效的方法。鉴于我们半群中可逆元素的数量减少，这些攻击将不会是有效的。

对于 [12] 中的 Pohlig-Hellman 还原，利用了半群是有限的事实，并选择了一个元素 $m \in S$，使其在解决 mS 上的问题变得更容易，从而得到原始解。在这种情况下，半群 $\mathrm{Circ}_n(\mathbb{N})$ 是无限的，通过将系数限制为一个正自然数 m 来实现计算机上的适用性。因此，如果攻击者取 $C \in \mathrm{Circ}_n([0,h])$ 并尝试进行还原，他们将面临解决 $\mathrm{Circ}_n([0,th])$ 上的问题，在这个问题中实际上并没有减少找到解的可能候选者的数量。因此，我们看不到这如何能为攻击者提供优势。

3.3 O-L 攻击

在 [11] 中，作者们对 [4] 中提出的原始密钥交换协议进行了密码分析。为了做到这一点，给定 $A, B, M, p(A)Mq(B) \in \mathrm{Mat_n(R)}$ 和 $p(x), q(x) \in C_R[X]$，他们计算一个三个非交换变量的多项式 $F[X,Y,Z] \in C_R[X,Y,Z]$，使得 $F[A,M,B] = p(A)Mq(B)$ 并且对于所有 $h(x), l(x) \in C_R[X]$ $F[A, h(A)Ml(B), B] = h(A)p(A)Mq(B)l(B)$ 。

在我们的案例中，攻击者可能使用此算法找到一个 $p_i(x) \in C_r[x]$，使得 $p_i(M) = M_i$。然而，这并不提供关于共享密钥或任何私钥的信息。如果攻击者使用此算法找到 $P_i^A[x_0, \cdots, x_{n-1}] \in C_R[x_0, \cdots, x_{n-1}]$ 多项式在 n 非交换变量中，使得 $(P_i^A(v))_{i=0}^{n-1} = Av$，这并不能断定 $(P_i^A(Bv))_{i=0}^{n-1} = ABv$ 如原始情况中的恒等式，在一般情况下是错误的。

3.4 量子攻击

在 [13] 中，介绍了一种用于计算交换群上离散对数的量子算法。此外，在 [15] 中，作者提出了该算法的一种推广形式，可以计算半群上的离散对数。

在我们的案例中，如果选择矩阵的幂 M 作为换位子集，则上述算法会产生一个线性丢番图方程系统，该系统可以被求解以恢复私钥。

为了防止这种攻击，换位子集应由 M 的多项式组成，避免使用幺半群。在这种情况下，作者并不知道有任何方法可以将现有算法适应来破坏协议的安全性。此外，了解 M 的阶或预阶似乎不会影响协议的安全性。

4 结论

在这项工作中，我们成功开发了一种基于循环矩阵在同余单半环上的矩阵作用的新密钥交换协议。在整个研究过程中，我们有效地解决了具有所需代数性质的矩阵的具体构造问题，以确保协议的正确和安全运行。这涉及对底层数学结构的仔细探索，确保所使用的元素满足保持系统安全所需的条件。

此外，我们对协议每个阶段的计算成本进行了详细分析，使我们能够评估其与现有方法相比的实际可行性。已知攻击也得到了审查，并表明该协议对该类攻击表现出强大的抵抗力。

参考文献

[1] W. Diffie, M. E. Hellman. New directions in cryptography, *IEEE Trans. Inf. Theory* **22** (1976) 644-654.

[2] R. El Bashir, J. Hurt, A. Janaík, T. Kepka. Simple commutative semirings. *J. Algebra* **236** (2001) 277-306.

[3] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.* 48 (1987) 203-209.

[4] G. Maze, C. Monico, J. Rosenthal, Public key cryptography based on semigroup actions, *Adv. Math. Commun.,* **1** (2007) 489-507.

[5] V.S. Miller. (1986). Use of Elliptic Curves in Cryptography n: Williams, H.C. (eds) Advances in Cryptology - CRYPTO ' 85 Proceedings. CRYPTO 1985. Lecture Notes in Computer Science, vol 218. Springer, Berlin, Heidelberg, 417-426.

[6] C. Monico, On finite congruence-simple semirings, *J. Algebra* **271** (2004) 846-854.

[7] R. Steinwandt, A. Suárez. Cryptanalysis of a 2-party key establishment based on a semigroup action problem, *Adv. Math. Commun.* **5** (2011) 87-92.

[8] J. Zumbrägel, Classification of finite congruence-simple semirings with zero, *J. Algebra Appll.* **7** (2008) 363-377.

[9] J.-P. Massias, Majoration explicite de l'ordre maximum d'un élément du groupe symétrique, *Ann. Fac. Sci. Toulouse Math. (5)* **6** (1985), 269–281.

[10] A. Kendziorra, S.E. Schmidt, and J. Zumbrägel, Invertible Matrices over Finite Additively Idempotent Semirings, *Semigroup Forum* **86** (2013) 525–536, https://doi.org/10.1007/s00233-012-9427-x.

[11] A. Otero Sánchez, J. A. López Ramos. Cryptanalysis of a key exchange protocol based on a congruence-simple semiring action, *J. Algebra Appl.*

[12] O. W. Gnilke and J. Zumbrägel, "Cryptographic group and semigroup actions," in: *Journal of Algebra and Its Applications*, vol. 23, no. 07, 2024.

[13] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," in: *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[14] R. Steinwandt and A. Suárez, "Cryptanalysis of a 2-party key establishment based on a semigroup action problem," in: *Advances in Mathematics of Communications*, vol. 5, pp. 87–92, 2011.

[15] A. M. Childs and G. Ivanyos, "Quantum computation of discrete logarithms in semigroups," *J. Math. Cryptol.* **8** (2014), no. 4, 405–416.