Comp. Appl. Math. (2017) 36:1273–1297
DOI 10.1007/s40314-015-0281-9

Maximal cyclic subgroups of the groups of units of Galois rings: a computational approach

Tariq Shah[1] · **Nasir Mehmood**[2] · **Antonio Aparecido de Andrade**[3] · **Reginaldo Palazzo Jr.**[4]

Received: 21 April 2015 / Revised: 5 August 2015 / Accepted: 28 September 2015 /
Published online: 5 November 2015
© SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional 2015

Abstract We present a computational approach based on algorithmic techniques to obtain maximal cyclic subgroups of the groups of units in Galois rings. The objective of this work was to provide an automated methodology to obtain such maximal cyclic subgroups for minimizing the human effort in such calculations. Necessity of getting a stock of maximal cyclic subgroups is due to their novel role in the formation of cyclic codes over finite commutative rings and prophesied shifting of S-box construction through binary field extensions $\mathrm{GF}(2^h)$, $1 \leq h \leq 8$ to these maximal cyclic subgroups of the groups of units of finite Galois rings $\mathrm{GR}(2^k, h)$.

Keywords Galois field · Galois ring · Group of units · Maximal cyclic subgroup · Cayley table

Mathematics Subject Classification 11T71 · 13M99 · 94B15

Communicated by Jinyun Yuan.

✉ Tariq Shah
stariqshah@gmail.com

Nasir Mehmood
nasir.chandio@gmail.com

Antonio Aparecido de Andrade
andrade@ibilce.unesp.br

Reginaldo Palazzo Jr.
palazzo@dt.fee.unicamp.br

1 Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan

2 Department of Electronics, Quaid-i-Azam University, Islamabad, Pakistan

3 Department of Mathematics, IBILCE, UNESP, São José do Rio Preto, SP, Brazil

4 Department of Telecommunication, FEEC-UNICAMP, Campinas, SP, Brazil

1 Introduction

Recently, the maximal cyclic subgroup G_s of the group of units of a Galois ring has assumed a valued role in algebraic coding theory. Initially Shankar (1979) has given a construction technique of BCH codes over local commutative ring $\mathbb{Z}_{p^k}$ based on the maximal cyclic subgroup G_s of the group of units of a Galois extension ring $\mathrm{GR}(p^k, h)$ of $\mathbb{Z}_{p^k}$. The obtainability of G_s is due to a *mod-p* reduction map from the ring $\mathbb{Z}_{p^k}$ to its residual field $\mathbb{Z}_p$ (see Shankar 1979). Although the exponential sums and an upper bound on the amalgam sum over the Galois rings using their maximal cyclic subgroups of the groups of units are given by Cohen (2009) and Shanbhag et al. (1996), Andrade and Palazzo (1999, 2005) have also given a construction method including the decoding procedure of BCH codes over Galois rings by use of their corresponding maximal cyclic subgroups of the groups of units. As a result, Shah et al. (2012b, a) present a construction and decoding procedures for a sequence of BCH codes using the corresponding sequence of maximal cyclic subgroups of the given sequence of the groups of units in the sequence of finite Galois rings and finite unitary commutative rings.

On the lines of Andrade and Palazzo (1999, 2005), Andrade et al. (2010a, b, 2011), Andrade and Shah (2011a, b), Shah et al. (2011a, b) and Shah and Andrade (2012a, b, c), respectively, give the decoding and construction methods of cyclic, BCH, Goppa and Srivastava codes over finite Galois rings by use of different monoid rings. In fact in this construction again the role of maximal cyclic subgroups is essential.

Currently, Shah et al. (2013) introduced a construction technique of substitution boxes using maximal cyclic subgroups of the groups of units in Galois extension rings $\mathrm{GR}(2^2, 2)$ and $\mathrm{GR}(2^2, 4)$. Accordingly, the complexity of these newly obtained S-boxes increases the strength of encryption and decryption. Obtaining a maximal cyclic subgroup G_s of the group of units of the Galois ring extension $\mathrm{GR}(p^k, h)$ of an arbitrary local ring $\mathbb{Z}_{p^k}$ is quite difficult. Thus there is a necessity to introduce a computational method to investigate higher order maximal cyclic subgroups. In this paper we introduce algorithms based on a computational technique to find a maximal cyclic subgroup G_s of the group of units in the Galois ring $\mathrm{GR}(p^k, h)$. The purpose of this study is to propose an automated method to reduce the human effort in such calculations. The demand of getting a stock of maximal cyclic subgroups is due to their innovative role in constructing classes of cyclic codes over finite commutative rings, and the anticipated conversion of S-box construction over binary field extensions $\mathrm{GF}(2^h)$, $1 \leq h \leq 8$, into these maximal cyclic subgroups of the groups of units of finite Galois rings $\mathrm{GR}(2^k, h)$.

This paper is organized as follows: In Sect. 2, the basic notion of a finite local ring and its Galois ring extension are given. The algebraic structure of the maximal cyclic subgroup of the group of units of a Galois ring extension with examples is also included. Section 3 consists in describing the proposed algorithms to find maximal cyclic subgroups of the groups of units of Galois rings. In Sect. 4, a complete description of the computational method is given. However, examples of computationally finding of maximal cyclic subgroups are given in Sect. 5. The utility of higher order computationally obtained maximal cyclic subgroups in coding theory and cryptography is, respectively, given in Sects. 6 and 7. Finally, in Sect. 8 the conclusions and future directions are drawn.

2 Maximal cyclic subgroup of the group of units of a Galois ring

In this section we discuss some basic concepts such as local ring, Galois extension ring of local ring, unit elements, and maximal cyclic subgroup of the group of units of a Galois ring which are relevant to the development of the paper.

2.1 Galois ring

An element u in a commutative ring R with identity is a unit if there exists an element $v \in R$ such that $u \cdot v = 1$, where 1 is the identity in the ring. A commutative ring R with identity is said to be a *local* ring if it has only one maximal ideal. A commutative ring R with identity is a *local* ring if and only if the set of all its non-unit elements forms an additive Abelian group. For a prime p and a positive integer k, $\mathbb{Z}p^k$, the ring of integers modulo p^k, is an example of a local ring. A nonzero element a of a commutative ring R with identity is called a *zero divisor* in R if there exists a nonzero element b in R such that $a \cdot b = 0$.

Let R be a local commutative ring with identity having maximal ideal M. An irreducible polynomial $f_0 + f_1x + f_2x^2 + \cdots + f_nx^n \in R[x]$ is said to be a *basic irreducible* polynomial over R if the polynomial $\bar{f}_0 + \bar{f}_1x + \bar{f}_2x^2 + \cdots + \bar{f}_nx^n$ is irreducible over the residue field R/M, where $\bar{f}_i$ denotes the class of R/M.

Take the finite local ring $\mathbb{Z}_{p^k}$, where p is prime and k is a positive integer, with corresponding residue field $\mathbb{Z}_p$. Let $\mathbb{Z}_{p^k}[x] = \{f_0 + f_1x + f_2x^2 + \cdots + f_nx^n\colon f_i \in \mathbb{Z}_{p^k}, n \in \mathbb{Z}^+\}$ be the polynomial extension of $\mathbb{Z}_{p^k}$ in the indeterminate x and $\mathbb{Z}_p[x] = \{f_0 + f_1x + f_2x^2 + \cdots + f_nx^n\colon f_i \in \mathbb{Z}_p n \in \mathbb{Z}^+\}$ be the polynomial extension of $\mathbb{Z}_p$ also in the indeterminate x. Let $f(x) \in \mathbb{Z}_{p^k}[x]$ be a basic irreducible polynomial with degree h. The ideal generated by $f(x)$ is denoted by $\langle f(x)\rangle$ and defined as $\langle f(x)\rangle = \{a(x) \cdot f(x)\colon a(x) \in \mathbb{Z}_{p^k}[x]\}$.

2.2 Maximal cyclic subgroup

Let $R = \frac{\mathbb{Z}_{p^k}[x]}{\langle f(x)\rangle} = \{f_0 + f_1x + f_2x^2 + \cdots + f_{h-1}x^{h-1}\colon f_i \in \mathbb{Z}_{p^k}\}$ represent the set of residue classes of polynomials in x over $\mathbb{Z}_{p^k}$ modulo the polynomial $f(x)$. This ring, denoted by $\mathrm{GR}(p^k, h)$, is a commutative ring with identity and is called *Galois extension ring* of $\mathbb{Z}_{p^k}$. It is noticed that $\mathrm{GR}(p^k, 1)$ is isomorphic to$\mathbb{Z}_{p^k}$, and $\frac{\mathbb{Z}_p[x]}{\langle \bar{f}(x)\rangle} = \mathrm{K}$ is isomorphic to the Galois field $\mathrm{GF}(p^h)$, an extension of $\mathbb{Z}_p$having p^h elements, where $\bar{f} = r_p(f)$is the polynomial with coefficient modulo p.

Let K^* and R^* be the multiplicative group of units of the field K and ring R, respectively. Then R^* is a multiplicative Abelian group and can be written as a direct product of cyclic subgroups (McDonald 1974). As rings have zero divisors, so naturally we are interested in cyclic subgroup of R^*, whose elements are the roots of $x^n - 1$, for some positive integer n. The next Theorem 2.2.1, which follows directly from McDonald (1974), Theorem XVIII.2, provides the order of the cyclic subgroup of interest.

Theorem 2.2.1 *There is only one maximal cyclic subgroup of R^* having order relatively prime to p. This cyclic subgroup has order $p^h - 1$.*

The maximal cyclic subgroup is denoted by G_n, where $n = p^h - 1$, *and it is isomorphic to K^*.*

2.2.1 Constructing Galois field GF(2, 2) *and Galois ring* GF(4, 2)

The corresponding residue field of the ring $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ is $\mathbb{Z}_2 = \mathrm{GF}(2) = \{0, 1\}$. The polynomial $f(x) = x^2 + 3x + 1$ is a monic and basic irreducible polynomial over $\mathbb{Z}_4$. Thus, the polynomial $\bar{f}(x) = x^2 + x + 1$ is irreducible over $GF(2)$. Hence,

Table 1 The elements of GF (2^2) and $G_3 \bigcup\{0\}$

Exp	Polynomial	Binary string	Exp	Polynomial	String mod 4
$-\infty$	0	00	$-\infty$	0	00
0	1	10	0	1	10
1	$\bar{u}$	01	2	$u^2 = 3 + u$	31
2	$\bar{u}^2 = 1 + \bar{u}$	11	4	$u^4 = 3u$	03

$\langle\overline{f}(x)\rangle = \{a(x).\overline{f}(x)\colon a(x) \in \mathrm{GF}(2)[x]\}$ is the ideal generated by the polynomial $\overline{f}(x) \in \mathrm{GF}(2)[x] = \{\sum_{i=0}^{n} a_i x^i : a_i \in \mathrm{GF}(2), n \in \mathbb{Z}^+\}$. Accordingly, for the polynomial ring $\mathbb{Z}_4[x] = \{\sum_{i=0}^{n} a_i x^i : a_i \in \mathbb{Z}_4, n \in \mathbb{Z}^+\}$, the Galois ring of order 16 becomes $\mathbf{R} = \frac{\mathbb{Z}_4[x]}{\langle x^2+3x+1\rangle}$ and the corresponding Galois field of order 4 is $\mathbf{K} = \frac{\mathrm{GF}(2)[x]}{\langle x^2+x+1\rangle}$. The multiplicative group of units of $\mathbf{K}$ is $\mathbf{K}^* = \{1, \bar{u}, \bar{u}+1\}$, where $\bar{u}$ denotes the residue class containing $\{x\}$. Similarly, the multiplicative group of units of $\mathbf{R}$ is $\mathbf{R}^*$ and its maximal cyclic subgroup is $G_3 = \{1, 3u, u+3\}$, where u is the residue class containing $\{x\}$. The elements of the Galois field $\mathrm{GF}(2^2)$ and the maximal cyclic subgroup G_3 of the Galois ring $\mathrm{GR}(4, 2)$ are shown in Table 1.

2.2.2 Constructing Galois field $\mathrm{GF}(2^4)$ *and Galois ring* $\mathrm{GR}(4, 4)$

For the ring $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, GF(2) is the residue field. Since $f(x) = x^4+x+1$ is a monic and irreducible polynomial over $\mathbb{Z}_4$ and $\bar{f}(x) = x^4 + x + 1$ is irreducible over $\mathbb{Z}_2$, $f(x)$ is a basic irreducible polynomial. Consequently, the Galois ring $\mathbf{R} = \frac{\mathbb{Z}_4[x]}{\langle x^4+x+1\rangle} = \mathrm{GR}(4, 4)$ of order 256 is obtained and its corresponding Galois field of order 16 is $\mathbf{K} = \frac{\mathrm{GF}(2)[x]}{\langle x^4+x+1\rangle} = \mathrm{GF}(2^4)$. The elements of GR (2, 4) can be obtained using the identity $\bar{u}^4 + \bar{u} + 1 = 0$ and the reduction modulo 2 (see Table 2).

Since the multiplicative groups of units of $\mathbf{K}$ and $\mathbf{R}$ are $\mathbf{K}^* = \mathbf{K}\backslash\{0\}$ and $\mathbf{R}^*$, respectively, the cyclic subgroup of $\mathbf{R}^*$ is G_{15}. However, the corresponding element of $\bar{u} + 1$ is $u + 1$ and its order is 30. Thus the generator of the cyclic group G_{15} is $(u + 1)^2$,where $\bar{u}$ is the residue class containing $\bar{u}$ in $\mathbf{K}$ and u is the corresponding residue class containing u in $\mathbf{R}$. The elements of the cyclic subgroup of order 15 are shown in Table 2.

3 Computational approach to find the maximal cyclic subgroup $\mathbf{G_s}$

We propose a computational approach to find the maximal cyclic subgroups of the groups of units in Galois rings. The approach is based on algorithmic procedures listed in Table 3. Although the complexity of some of the proposed algorithms is high, the objective is to give an automated computational approach as opposed to an analytical approach to reduce human effort in computing maximal cyclic subgroups of the group of units in a Galois ring.

3.1 Computational procedures

A polynomial of degree $n - 1$ is represented by an array consisting of n indices. That is, a n-index based array consists of an element in the position 0 representing the coefficient of x_0, an element in the position 1 representing the coefficient of x_1, up to an element at position $(n - 1)$ representing the coefficient of x_{n-1}. The procedure Add(poly1, poly2) adds

Table 2 The elements of GF(2^4) and $G_{15} \bigcup \{0\}$

Exp	Polynomial	Binary string	Exp	Polynomial	String mod 4
$-\infty$	0	0000	$-\infty$	0	0000
0	1	1000	0	1	1000
1	$1+\bar{u}$	1100	2	$1+2u+u^2$	1210
2	$1+\bar{u}^2$	1010	4	$3u+2u^2$	0320
3	$1+\bar{u}+\bar{u}^2+\bar{u}^3$	1111	6	$2+u+3u^3$	2103
4	$\bar{u}$	0100	8	u^2	0010
5	$\bar{u}+\bar{u}^2$	0110	10	$3+3u+u^2+2u^3$	3312
6	$\bar{u}+\bar{u}^3$	0101	12	$2+2u+3u^3$	2203
7	$1+\bar{u}^2+\bar{u}^3$	1011	14	$u+3u^2+u^3$	0131
8	$\bar{u}^2$	0010	16	$3+3u$	3300
9	$\bar{u}^2+\bar{u}^3$	0011	18	$3+u+u^2+3u^3$	3113
10	$1+\bar{u}+\bar{u}^2$	1110	20	$u+3u^2+2u^3$	0132
11	$1+\bar{u}^3$	1001	22	$1+3u^2+u^3$	1031
12	$\bar{u}^3$	0001	24	$3u^2+3u^3$	0033
13	$1+\bar{u}+\bar{u}^3$	1101	26	$3+u^3$	3001
14	$\bar{u}+\bar{u}^2+\bar{u}^3$	0111	28	$1+3u+2u^2+u^3$	1321

two polynomials in a Galois ring and returns the sum. All arithmetic is done in $\mathbb{Z}_n$, where n is represented by variable 'base' in the algorithms. We implemented the computational algorithms in C^{++} language. The algorithms are presented in pseudo-code form.

4 Description of the computational procedures

All arithmetic related to the coefficients is done in $\mathbb{Z}_n$ (n = base). Description of the computational procedures used in finding maximal cyclic subgroups is shown in Table 4.

4.1 EvaluatePolynomialAt(poly, x)

The loop at lines 2–4 computes the appropriate power of the input variable 'x' and multiplies the resulting power by the corresponding coefficient. The remainder is then applied to the resulting value at line 5. Value after taking remainder is returned.

4.2 Add(poly1, poly2)

This procedure returns the sum of the two polynomials 'poly1' and 'poly2'. The loop at lines 3–5 adds the coefficients and saves the result in the new array which is returned.

4.3 Subtract(poly1, poly2)

This procedure returns the difference of the two polynomials 'poly1' and 'poly2'. The loop at lines 3–8 performs subtraction of the coefficients and saves the result in the new array which

Table 3 Algorithms

1

```
EvaluatePolynomialAt(poly, x)
Begin
        1.  Set val=0
        2.  For i=0 to poly.length-1
        3.  Begin
        4.    val = val + power(x,i)*poly[i]
        5.  End
        6.  val = val%base
        7.  return val
End
Add(poly1, poly2)
Begin
        1.  length_max
            =max(poly1.length,poly2.length)
        2.  p= ZeroPolynomial
        3.  For i=0 to length_max-1
        4.  Begin
        5.    p[i] = (poly1 [i]+poly2 [i])%base
        6.  End
        7.  return p
End
Subtract(poly1, poly2)
Begin
        1.  length_max
            =max(poly1.length,poly2.length)
        2.  p= ZeroPolynomial
        3.  For i=0 to length_max-1
        4.  Begin
        5.  p[i] = (poly_min[i]-poly_max[i])
        6.    While p[i]<0
        7.      Begin
        8.        p[i] = p[i]+base
        9.      End
       10.  End
       11.  return p
End
Multiply(poly1, poly2)
Begin
        1.  p = ZeroPolynomial
        2.  For i=0 to poly1.length-1
        3.  Begin
        4.    c1 = poly1[i];
        5.    For j=0 to poly2.length-1
        6.    Begin
        7.      c2 = poly2[j];
        8.      index = i+j;
        9.      temp = (p[index]+(c1*c2));
       10.      p[index] = temp%base;
       11.    End
       12.  End
       13.  return p
End
Remainder(poly1, poly2)
Begin
        1.  Set a = p1;
        2.  Set b = p2;
        3.  While(a.length>=b.length)AND
            (a!=zero_poly)
        4.  Begin
        5.    c1 = a[a.length-1]
        6.    c2 = b[b.length-1]
        7.    Set temp = 0
        8.      while ((temp*c2)%base)!=c1)
        9.        Begin
       10.          temp++
       11.          if temp>=p1.base
       12.            Begin
       13.              break
       14.            End
       15.        End
```

2

```
       16.  if temp>=base
       17.    Begin
       18.      break
       19.    End
       20.    Set q=ZeroPolynomial
       21.  q.coefficients[a.length-b.length]= temp;
       22.  q.length = a.size-b.size+1;
       23.  a = Subtract(a,Multiply(q,b))
       24.  End
       25.  return a
End
power(poly, pwr)
begin
        1.  p = ZeroPolynomial
        2.  p[0] = 1
        3.  for i=0 to pwr-1
        4.  Begin
        5.  p=remainder(product(p,poly),
            g_divisor_polynomial)
        6.  End
        7.  return p;
End
IsIrreducible(poly)
Begin
        1.  if(poly.length<=2)
        2.  Begin
        3.    return true;
        4.  End
        5.  while i<total_polynomials
        6.    Begin
        7.    p = GetPolynomial(i);
        8.    if p.length>1
        9.    Begin
       10.    if Remainder(poly,p)==ZeroPolynomial
       11.      Begin
       12.        return false;
       13.      End
       14.    End
       15.    i++
       16.  End
       17.  return true;
End
IsUnit(poly)
Begin
        1.  For i=0 to poly.size-1
        2.  Begin
        3.    For j=0 to base-1
        4.      Begin
        5.        if( ((p.coefficients[i]*j)%base)==1)
        6.          Begin
        7.            return true;
        8.          End
        9.      End
       10.  End
       11.  return false;
End
GetOrder(poly)
Begin
        1.  If!IsUnit(poly)
        2.  Begin
        3.    return 0;
        4.  End
        5.  For i=1 to ring_exponent
        6.  Begin
        7.  temp1 = power(poly,i)
        8.    temp=remainder(temp1,divisor_polynomial)
        9.  if(temp==IdentityPolynomial)
       10.  Begin
       11.    return i;
       12.  End
       13.  End
       14.  return 0;
End
```

3

```
ComputeMaximalCyclicSubgroup()
Begin
        1.  For i=0 to total_polynomials-1
        2.  Begin
        3.  poly= GetPolynomial(i)
        4.  If IsUnit(poly)
        5.  Begin
        6.    order = GetOrder(poly);
        7.  if  order==(power(P,divisor_polynomial.size-
            1)-1)
        8.  Begin
        9.  mcsg_array[0] = poly
       10.  mcsg_array[1] = UnitPolynomial
       11.  For j=1 to order-1
       12.  Begin
       13.  mcsg_array[j+1]=
            ((poly^j)%divisor_polynomial)
       14.  End
       15.  break
       16.  End
       17.  End
       18.  End
       19.  Return mcsg_array
End
PrintMaximalCyclicSubgroup(mscg_array)
Begin
        1.  print "Generator: "
        2.  print mscg_array[0]
        3.  For j=1 to mscg_array.size-1
        4.  Begin
        5.    print "Polynomial "
        6.    print j
        7.    print mscg_array[j]
        8.  End
End

PrintMCSGCaleyTable(mscg_array)



Begin
        1.  For i=1 to mscg_array.size-1
        2.  Begin
        3.    Print mscg_array[i]
        4.  End
        5.  for i=1 to mscg_array.size-1
        6.  Begin
        7.  Print mscg_array[i]
        8.  For j=1 to mscg_array.size-1
        9.  Begin
       10.  poly=  Remainder(Multiply(array[i],array[j]),
            divisor_polynomial)
       11.  Print poly
       12.  End
       13.  End
End
```

is returned. Since the subtraction is performed in $\mathbb{Z}_n$, the negative answer of the subtraction is handled in the *while loop* at lines 5–7. Base is added to the negative answers until the result becomes positive.

4.4 Multiply(poly1, poly2)

Multiply returns the product of the two polynomials 'poly1' and 'poly2'. As the coefficient of the power x^{i+j}, for each fixed $i + j$, with i and j varying from 0 to n and from 0 to m, respectively, is the addition of the product of the coefficients of the corresponding elements with the power adding to $i + j$, therefore, at line 6, the index is the sum of i and j. The product of the coefficients is then added up in the resulting polynomial at line 8.

Table 4 Description of the algorithms

Procedure	Description
4.1. EvaluatePolynomialAt(poly, x)	Returns value of polynomial at a particular value 'x'
4.2. Add(poly1, poly2)	Returns sum of two polynomials 'poly1' and 'poly2'
4.3. Subtract(poly1, poly2)	Returns difference of two polynomials 'poly1' and 'poly2'
4.4. Multiply(poly1, poly2)	Returns product of two polynomials 'poly1' and 'poly2'
4.5. Remainder(poly1, poly2)	Returns remainder polynomial when polynomial 'poly1' is divided by polynomial 'poly2'
4.6. power(poly, pwr)	Returns polynomial 'poly' raise to the power 'pwr'
4.7. IsIrreducible(poly)	Returns 'true"if polynomial 'poly' is irreducible otherwise returns 'false'
4.8. IsUnit(poly)	Returns 'true' if polynomial 'poly' is unit polynomial otherwise returns 'false'
4.9. GetOrder(poly)	Returns order of the unit polynomial 'poly'
4.10. ComputeMaximalCyclicSubgroup ()	Prints all the maximal cyclic subgroups along with their generators
4.11. PrintMaximalCyclicSubgroup(mscg_array)	Prints maximal cyclic subgroup mscg_array
4.12. PrintMCSGCaleyTable(mscg_array)	Prints caley table for the maximal cyclic subgroup mscg_array

4.5 Remainder(poly1, poly2)

It returns the remainder polynomial when 'poly2' is divided by 'poly1'. In the iteration realized in the *while loop* at line 3, one higher order coefficient is removed by multiplying the divisor polynomial with the appropriate integer and then subtracting from the dividend polynomial. This continues until the difference polynomial becomes zero or its degree is less than that of the divisor polynomial.

4.6 power(poly, pwr)

It computes the power of a polynomial. When considering high-power polynomials the power of the resulting polynomial can grow very fast and we only need the arithmetic done in $\mathbb{Z}[x]/\langle g\rangle$, where $g$is the divisor polynomial. Consequently, we take the remainder resulting from the loop at lines 3–5.

4.7 IsIrreducible(poly)

It returns a Boolean value: *true* if the input polynomial 'poly' is irreducible; otherwise it returns *false*. First of all, the polynomials with degree 0 and 1 are irreducible, and so this check is made at lines 1–3. The *while loop* at lines 4–12 checks if any polynomial in the Galois ring divides the input polynomial 'poly'.

4.8 IsUnit(poly)

It returns a Boolean value: *true* if the input polynomial 'poly' is a unit polynomial, otherwise it returns *false*. The algorithm checks if at least one of the coefficients of the input polynomial

is a unit. If so the input polynomial is a unit and if all of the coefficients of the input polynomial are non-units, the input polynomial is non-unit polynomial.

4.9 GetOrder(poly)

It returns *zero* for a non-unit polynomial, and it returns the *order* of the polynomial in case the input polynomial is a unit polynomial. The order of a polynomial lies between 1 and the ring exponent of the Galois ring, both inclusive. Each iteration in the *for loop* at lines 4–10 takes the power of the polynomial raised to loop index 'i' and applies the remainder. If the resulting polynomial, after applying the remainder, is an identity polynomial, then the corresponding value of the loop index variable 'i' is returned as the order of the input polynomial.

4.10 ComputeMaximalCyclicSubgroup()

It computes the maximal cyclic subgroup and saves it in an array. The array element at position 0 is the generator of the maximal cyclic subgroup and the elements from 1 to array_size-1 are the polynomials of the maximal cyclic subgroup. If the total number of elements in the maximal cyclic subgroup is n, the size of the array is $n + 1$. Then the algorithm returns the maximal cyclic subgroup array.

4.11 PrintMaximalCyclicSubgroup(mscg_array)

It prints the maximal cyclic subgroup mscg_array which is passed as a parameter to the algorithm.

4.12 PrintMCSGCayleyTable(mscg_array)

It prints the Cayley table for the maximal cyclic subgroup mscg_array which is passed as a parameter to the algorithm.

5 Finding maximal cyclic subgroups

The procedure "ComputeMaximalCyclicSubgroup()" computes the maximal cyclic subgroup along with its generator. The variable "divisor_polynomial" used in the procedure is the basic monic irreducible polynomial used in the construction of the Galois ring $\mathbb{Z}_{p^k}[x]/\langle divisor\_polynomial \rangle$. The algorithms "PrintMaximalCyclicSubgroup(mscg_array)" and "PrintMCSGCaleyTable(mscg_array)" are used to print the maximal cyclic subgroup and the Cayley table of the maximal cyclic subgroup, respectively.

5.1 Two examples of maximal cyclic subgroups

We compute the maximal cyclic subgroups using the algorithms given in Sect. 3. Two maximal cyclic subgroups obtained computationally are shown in Table 5.

5.2 Cayley tables of two maximal cyclic subgroups

The Cayley tables of the two Galois rings shown in Table 5 are illustrated in Tables 6 and 7, respectively. These Cayley tables are obtained computationally using the proposed algorithms.

Table 5 The elements in G_7 and G_{26}

Galois ring GR($p^{k,h}$)	Maximal cyclic subgroup (Gs)	Representation
GR (4, 3) = $\frac{\mathbb{Z}_4[x]}{x^3+3x^2+2x+3}$	Polynomial 1: 1	100
	Polynomial 2: x	010
	Polynomial 3: x^2	001
	Polynomial 4: $x^2 + 2x + 1$	121
	Polynomial 5: $3x^2 + 3x + 1$	133
	Polynomial 6: $2x^2 + 3x + 3$	332
	Polynomial 7: $x^2 + 3x + 2$	231
GR (9, 3) = $\frac{\mathbb{Z}_9[x]}{x^3+2x^2+2x+2}$	Polynomial 1 : 1	100
	Polynomial 2 : $4x + 4$	440
	Polynomial 3 : $7x^2 + 5x + 7$	757
	Polynomial 4 : $x^2 + x + 8$	811
	Polynomial 5 : $x + 6$	610
	Polynomial 6 : $4x^2 + x + 6$	614
	Polynomial 7 : $6x^2 + 5x + 1$	156
	Polynomial 8 : $5x^2 + 3x + 1$	135
	Polynomial 9 : $x^2 + 3x$	031
	Polynomial 10: $8x^2 + 4x + 1$	148
	Polynomial 11: $2x^2 + x + 3$	312
	Polynomial 12: $5x^2 + 5$	505
	Polynomial 13: $7x^2 + 7x + 7$	777
	Polynomial 14: 8	800
	Polynomial 15: $5x + 5$	550
	Polynomial 16: $2x^2 + 4x + 2$	242
	Polynomial 17: $8x^2 + 8x + 1$	188
	Polynomial 18: $8x + 3$	380
	Polynomial 19: $5x^2 + 8x + 3$	385
	Polynomial 20: $3x^2 + 4x + 8$	843
	Polynomial 21: $4x^2 + 6x + 8$	864
	Polynomial 22: $8x^2 + 6x$	068
	Polynomial 23: $x^2 + 5x + 8$	851
	Polynomial 24: $7x^2 + 8x + 6$	687
	Polynomial 25: $4x^2 + 4$	404
	Polynomial 26: $2x^2 + 2x + 2$	222

6 Application in coding theory

Recently, a great interest on the structure of the multiplicative group of units of certain finite local commutative rings has been pursued in coding theory due to its amazing applications, especially in the construction of BCH codes. The construction has been addressed from the

Table 6 Cayley table of the maximal cyclic subgroup G_7

	1	x	x^2	x^2+2x+1	$3x^2+3x+1$	$2x^2+3x+3$	x^2+3x+2
1	1	x	x^2	x^2+2x+1	$3x^2+3x+1$	$2x^2+3x+3$	x^2+3x+2
x	x	x^2	x^2+2x+1	$3x^2+3x+1$	$2x^2+3x+3$	x^2+3x+2	1
x^2	x^2	x^2+2x+1	$3x^2+3x+1$	$2x^2+3x+3$	x^2+3x+2	1	x
x^2+2x+1	x^2+2x+1	$3x^2+3x+1$	$2x^2+3x+3$	x^2+3x+2	1	x	x^2
$3x^2+3x+1$	$3x^2+3x+1$	$2x^2+3x+3$	x^2+3x+2	1	x	x^2	x^2+2x+1
$2x^2+3x+3$	$2x^2+3x+3$	x^2+3x+2	1	x	x^2	x^2+2x+1	$3x^2+3x+1$
x^2+3x+2	x^2+3x+2	1	x	x^2	x^2+2x+1	$3x^2+3x+1$	$2x^2+3x+3$

Table 7 Cayley table of the maximal cyclic subgroup G_{26}

	1 $5x^2+3x+1$ $5x+5$ $8x^2+6x$	$4x+4$ x^2+3x $2x^2+4x+2$ x^2+5x+8	$7x^2+5x+7$ $8x^2+4x+1$ $8x^2+8x+1$ $7x^2+8x+6$	x^2+x+8 $2x^2+x+3$ $8x+3$ $4x^2+4$	$x+6$ $5x^2+5$ $5x^2+8x+3$ $2x^2+2x+2$	$4x^2+x+6$ $7x^2+7x+7$ $3x^2+4x+8$	$6x^2+5x+1$ 8 $4x^2+6x+8$
1	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$
	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8
	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$
	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$		
$4x+4$	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$
	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$
	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$
	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1		
$7x^2+5x+7$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x
	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$
	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8
	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$		
x^2+x+8	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$
	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$
	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$
	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$		
$x+6$	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$
	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$
	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$
	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8		

Table 7 continued

	1 $5x^2+3x+1$ $5x+5$ $8x^2+6x$	$4x+4$ x^2+3x $2x^2+4x+2$ x^2+5x+8	$7x^2+5x+7$ $8x^2+4x+1$ $8x^2+8x+1$ $7x^2+8x+6$	x^2+x+8 $2x^2+x+3$ $8x+3$ $4x^2+4$	$x+6$ $5x^2+5$ $5x^2+8x+3$ $2x^2+2x+2$	$4x^2+x+6$ $7x^2+7x+7$ $3x^2+4x+8$	$6x^2+5x+1$ 8 $4x^2+6x+8$
$4x^2+x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$
	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$
	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$
	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$		
$6x^2+5x+1$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$
	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$
	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1
	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$		
$5x^2+3x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8
	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$
	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$
	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$		
x^2+3x	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$
	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$
	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$
	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$		
$8x^2+4x+1$	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$
	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8
	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8
	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x		

Table 7 continued

	1 $5x^2+3x+1$ $5x+5$ $8x^2+6x$	$4x+4$ x^2+3x $2x^2+4x+2$ x^2+5x+8	$7x^2+5x+7$ $8x^2+4x+1$ $8x^2+8x+1$ $7x^2+8x+6$	x^2+x+8 $2x^2+x+3$ $8x+3$ $4x^2+4$	$x+6$ $5x^2+5$ $5x^2+8x+3$ $2x^2+2x+2$	$4x^2+x+6$ $7x^2+7x+7$ $3x^2+4x+8$	$6x^2+5x+1$ 8 $4x^2+6x+8$
$2x^2+x+3$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$
	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$
	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$
	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$		
$5x^2+5$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$
	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$
	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$
	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$		
$7x^2+7x+7$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$
	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$
	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$
	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$		
8	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$
	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1
	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$
	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$		
$5x+5$	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$
	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$
	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x
	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8		

Table 7 continued

	1 $5x^2+3x+1$ $5x+5$ $8x^2+6x$	$4x+4$ x^2+3x $2x^2+4x+2$ x^2+5x+8	$7x^2+5x+7$ $8x^2+4x+1$ $8x^2+8x+1$ $7x^2+8x+6$	x^2+x+8 $2x^2+x+3$ $8x+3$ $4x^2+4$	$x+6$ $5x^2+5$ $5x^2+8x+3$ $2x^2+2x+2$	$4x^2+x+6$ $7x^2+7x+7$ $3x^2+4x+8$	$6x^2+5x+1$ 8 $4x^2+6x+8$
$2x^2+4x+2$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$
	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$
	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$
	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$		
$8x^2+8x+1$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8
	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8
	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$
	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$		
$8x+3$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$
	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$
	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$
	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$		
$5x^2+8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$
	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$
	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$
	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$		
$3x^2+4x+8$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$
	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$
	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8
	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$		

Table 7 continued

	1 $5x^2+3x+1$ $5x+5$ $8x^2+6x$	$4x+4$ x^2+3x $2x^2+4x+2$ x^2+5x+8	$7x^2+5x+7$ $8x^2+4x+1$ $8x^2+8x+1$ $7x^2+8x+6$	x^2+x+8 $2x^2+x+3$ $8x+3$ $4x^2+4$	$x+6$ $5x^2+5$ $5x^2+8x+3$ $2x^2+2x+2$	$4x^2+x+6$ $7x^2+7x+7$ $3x^2+4x+8$	$6x^2+5x+1$ 8 $4x^2+6x+8$
$4x^2+6x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1
	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$
	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$
	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$		
$8x^2+6x$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$
	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x
	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$
	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$		
x^2+5x+8	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$
	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$
	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$
	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$		
$7x^2+8x+6$	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8
	$x+6$	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$
	$5x^2+5$	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$
	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8		
$4x^2+4$	$4x^2+4$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$
	$4x^2+x+6$	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$
	$7x^2+7x+7$	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$
	$3x^2+4x+8$	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$		

Table 7 continued

	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$	$6x^2+5x+1$
	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$	8
	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$	$4x^2+6x+8$
	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$	$2x^2+2x+2$		
$2x^2+2x+2$	$2x^2+2x+2$	1	$4x+4$	$7x^2+5x+7$	x^2+x+8	$x+6$	$4x^2+x+6$
	$6x^2+5x+1$	$5x^2+3x+1$	x^2+3x	$8x^2+4x+1$	$2x^2+x+3$	$5x^2+5$	$7x^2+7x+7$
	8	$5x+5$	$2x^2+4x+2$	$8x^2+8x+1$	$8x+3$	$5x^2+8x+3$	$3x^2+4x+8$
	$4x^2+6x+8$	$8x^2+6x$	x^2+5x+8	$7x^2+8x+6$	$4x^2+4$		

Table 8 The elements of GF (2^6)

Exp	Polynomial	Exp	Polynomial	Exp	Polynomial
0	100000	21	210113	42	130331
1	001000	22	303101	43	100303
2	000010	23	032031	44	012003
3	330000	24	103320	45	011120
4	003300	25	221033	46	220111
5	000033	26	123210	47	321201
6	121000	27	331232	48	032212
7	001210	28	131312	49	312322
8	330012	29	313313	50	201123
9	311300	30	300133	51	233011
10	003113	31	120001	52	321330
11	301031	32	030200	53	113213
12	102010	33	000302	54	302132
13	331020	34	022003	55	131021
14	223310	35	011220	56	210310
15	332233	36	220112	57	332103
16	120322	37	310201	58	010321
17	203203	38	032102	59	213103
18	013032	39	022321	60	013131
19	132130	40	213223	61	103131
20	111321	41	233132	62	100031

approach of specifying a maximal cyclic subgroup of the group of units of an extension ring of a finite commutative ring. The inherent complexity is involved with the factorization of $x^n - 1$ over the group of units of a given Galois extension ring of a local ring to construct the generator polynomial for the BCH codes. There are many classes of Galois extension rings for different local rings having very large cardinality and, consequently, the size of the maximal cyclic subgroups of the corresponding groups of units. Thus there is a need to propose an efficient algorithm to provide such maximal cyclic groups and group of units of Galois extension rings having larger cardinalities.

In the following, we give a computationally obtained example, which serves the purpose in constructing and decoding of BCH codes over a local ring and in its residual field.

For the ring $\mathbb{Z}_4$, the binary field GF(2) is the residue field. Since $f(x) = x^6 + x + 1$ is a monic and irreducible polynomial over $\mathbb{Z}_4$ and $\bar{f}(x) = f(x) = x^6 + x + 1$ is irreducible over GF(2), it is a basic irreducible polynomial. The Galois ring of order 4096 is given by $R = \frac{\mathbb{Z}_4[x]}{\langle f(x)\rangle}$ and the corresponding Galois field of order 64 is $K = \frac{\mathbb{Z}_4[x]}{\langle f(x)\rangle}$. The elements of GF$(2^6)$ can be obtained as shown in Table 8 using the identity $u^6 + u + 1$ modulo 2. Similarly, in GR$(2^2, 6)$, $u^6 + u + 1 = 0$ gives $u^6 = 3u + 3$ and the order of u is 126, therefore, the corresponding maximal cyclic subgroup G_{63} is generated by $u^2 = \alpha$. In this case, using computational algorithm, elements of G_{63} are given in Table 9.

6.1 Construction of BCH codes

To construct a BCH code C_1 over GR$(2^2, 6)$ with designed distance 3. Suppose the roots of $g(x)$are α and α^2. Note that $\alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}$ are the conjugates of α which

Table 9 The elements of maximal cyclic group G_{63} of group of units of GR(4, 6)

Exp	Polynomial	Exp	Polynomial	Exp	Polynomial
0	100000	21	110111	42	010111
1	010000	22	101011	43	111011
2	001000	23	100101	44	101101
3	000100	24	100010	45	100110
4	000010	25	010001	46	010011
5	000001	26	111000	47	111001
6	110000	27	011100	48	101100
7	011000	28	001110	49	010110
8	001100	29	000111	50	001011
9	000110	30	110011	51	110101
10	000011	31	101001	52	101010
11	110001	32	100100	53	010101
12	101000	33	010010	54	111010
13	010100	34	001001	55	011101
14	001010	35	110100	56	111110
15	000101	36	011010	57	011111
16	110010	37	001101	58	111111
17	011001	38	110110	59	101111
18	111100	39	011011	60	100111
19	011110	40	111101	61	100011
20	001111	41	101110	62	100001

are also the conjugates of α^2. Therefore, the minimal polynomial $M_1(x)$ is given by $M_1(x) = (x-\alpha)(x-a^2)(x-\alpha^4)(x-\alpha^8)(x-\alpha^{16})(x-\alpha^{32}) = x^6 + 2x^3 + 3x + 1$. In this case the generator polynomial is given by $g(x) = M_1(x) = x^6 + 2x^3 + 3x + 1$. Since $\alpha, \alpha^2$are the consecutive roots of $g(x)$ it follows from Andrade and Palazzo (1999, Definition 2.2), $b = 0$, $t = 1$, and the minimum distance d is greater than 2.

Correspondingly, for $d = 3$, the BCH code over GF(2^6) can be obtained by calculating the minimal polynomial of α^i, where $i = c, c+1, \ldots, c+d-2$. Let $c = 1$, then $i = 1, 2$. Now $u, u^2, u^4, u^8, u^{16}, u^{32}$ are the roots of the minimal polynomial $m_1(x) = x^6 + x + 1 = r_p(M_1(x))$ which is also the generator polynomial of the BCH code C_1' over GF(2^6).

In a similar manner we may construct a BCH code C_2 over GR(2^2, 6) with designed distance 5 by taking the roots α^i, for $i = 1, 2, 3, 4$. $M_1(x)$ is the same minimal polynomial as in the previous case of the BCH code C_1. Let $M_2(x)$ be the minimal polynomial of α^3, then $(\alpha^3)^2 = \alpha^6, (\alpha^3)^4 = a^{12}, (\alpha^3)^8 = a^{24}, (\alpha^3)^{16} = a^{48}, (\alpha^3)^{32} = a^{33}$ are the conjugates of α^3 and the corresponding minimal polynomial is given by $M^2(x) = (x-\alpha^3)(x-\alpha^6)(x-\alpha^{12})(x-\alpha^{24})(x-\alpha^{33})(x-\alpha^{48}) = x^6 + 2x^5 + 3x^4 + 3x^2 + x + 1$. Accordingly, $g(x) = \text{lcm}\{M^1(x), M^2(x)\} = x^{12} + 2x^{11} + 3x^{10} + 2x^9 + 3x^8 + 2x^7 + x^5 + x^4 + 3x^3 + 2x^2 + 1$ is the generator polynomial of the BCH code of length 63 over $\mathbb{Z}_4$. In this case, the consecutive roots of $g(x)$ are $\alpha, \alpha^2, \alpha^3, \alpha^4$. Then by Andrade and Palazzo (1999, Definition 2.2), $b = 0$, $t = 2$, and minimum distance d is greater than 4.

For $d = 5$, the BCH code C_2' over GF(2^6) can be obtained by calculating the minimal polynomial of α^i, for $i = c, c+1, \ldots, c+d-2$. Let $c = 1$, then $i = 1, 2, 3, 4$. Now $u, u^2, u^4, u^8, u^{16}, u^{32}$ are the roots of the minimal polynomial $m_1(x) = x^6 + x + 1$.

Table 10 Modified Berlekamp–Massey algorithm

n	$\sigma^{(n)}(z)$	d_n	l_n	$n - l_n$
-1	1	1	0	-1
0	1	$s_1 = (003230)$	0	0
1	$1 + (001210)z$	(002220)	1	0
2	$1 + (003230)z$	–	1	1

Now, $m_2(x)$ is the minimal polynomial of $u^3, u^6, u^{12}, u^{24}, u^{33}, u^{48}$ which is given by $m_2(x) = (x - u^3)(x - u^6)(x - u^{12})(x - u^{24})(x - u^{33})(x - u^{48}) = x^6 + x^4 + x^2 + x + 1 = r_p(M_2(x))$. Then the generator polynomial of the BCH code of length 63 over $\mathrm{GF}(2^6)$is$g(x) = \mathrm{lcm}\{m_1(x), m_2(x)\} = x^{12} + x^{10} + x^8 + x^5 + x^4 + x^3 + 1 = r_p(g(x))$.

6.2 Decoding of BCH codes

Let C_1 be a (63, 57) BCH code generated by $g(x) = x^6 + 2x^3 + 3x + 1$ over $\mathbb{Z}_4$. Take $R = \mathrm{GR}(2^2, 6) = \frac{Z_4[x]}{(x^6+x+1)}$. The maximal cyclic subgroup of the group of units R^* of the ring R is G_{63}, and $\alpha = (001000)$ is a primitive element of G_{63}. Since the minimum distance $d(C_1) \geq 3$, it follows that the error correction capability is $t \geq 1$. The parity check matrix is given by

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \cdots & \alpha^{62} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \cdots & \alpha^{61} \end{bmatrix}.$$

Suppose the transmitted codeword is the polynomial $v(x) = x + 3x^2 + 2x^4 + x^7$ and the received polynomial is $a(x) = x + 3x^2 + 2x^4$. Now, the syndrome vector is given by $S = aH^T = (s_1, s_2) = (003230, 221130)$. By the modified Berlekamp–Massey algorithm, we obtain the following polynomial: (Table 10).

The root of $\rho(z) = z + (003230)$ [the reciprocal of $\sigma^{(2)}(z)$ is $z_1 = (001210)$]. Among the elements $\alpha^0, \alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^{62}$, we have $x_1 = \alpha^7$ such that $(x_1 - z_1) = 0$. It indicates that in the received polynomial (vector) the error is at position 7. The correct elementary symmetric function $\sigma_1 = (003230)$ is obtained by $x - x_1 = x + \sigma_1$. Now by applying Forney's method, the error magnitude is $y_1 = 3$. Hence the error polynomial (vector) is $e = 3x^7$.

7 Application in cryptography

The natural generalization of the theory of single output Boolean functions is the theory of multiple output Boolean functions, which may be stated as a substitution box (S-box). The connection between the input and the output bits with respect to dimension and exceptionality gives rise to the diversities of S-boxes.

An $n \times m$ S-box is a mapping $\varphi\colon \mathrm{GF}(2)^n \to \mathrm{GF}(2)^m$ from n binary input bits to m binary output bits, whereas there are 2^n and 2^m number of inputs and outputs, respectively. Afterwards, an S-box is nothing but a set of m single output Boolean functions arranged in a fixed order. The dimension of an S-box has an effect on the distinctness of the output and the input which may affect the properties of the S-box. An S-box with dimension $n \times m$, where $m < n$, or equivalently, the number of input bits being greater than the number of output bits, implies that some entries in the S-box must be repeated. While an $n \times n$ S-box may either

contain distinct entries where each input is mapped to different output or replicate several entries of the S-box. The S-boxes are reversible (see Hussain and Shah 2013).

S-box is the central fragment in almost all the cryptosystems which makes the system nonlinear. In the improvement of symmetric, or private-key, cryptosystems which are constructed as substitution–permutation networks (S–P networks), a considerable portion of the time spent on design or on analysis is dedicated to the S-boxes share of the algorithm. This is the reason the rest of the algorithm is linear. Therefore, simple weaknesses in the S-boxes can lead to cryptosystems which are easily broken. In fact, S-boxes are used as a puzzling scheme to authorize the strength of cryptographic algorithms. Thus the construction of S-boxes must be cryptographically hard to shape protected cryptosystems.

In conservative and contemporary cryptography, the S-boxes are typically designed over finite Galois fields ($GF(2)^n$, for $2 \leq n \leq 8$). For instance AES S-box, Residue Prime S-box (Hussain and Shah 2013), Gray S-box (Tran et al. 2008), APA S-box (Cui and Cao 2007), S_8 AES S-box, Skipjack S-box (Kim and Phan 2009), and Xyi S-box (Yi et al. 2002). The power of cryptographic algorithms is based on these nonlinear S-boxes, so the construction of cryptographically strong S-boxes is the core in the scheme of secure cryptosystems. Varied natures of S-boxes were proposed for secure communication which depend on algebraic and real-world structures. S-boxes constructed on algebraic approaches have received a considerable attention due to their robust cryptographic features (Adams and Tavares 1989; Webster and Tavares 1986; Hussain et al. 2011, 2012a, b, 2013).

Shah et al. (2013) have proposed a novel construction technique of S-boxes using the maximal cyclic subgroups of the groups of units of the Galois rings,GR(2^2, 2) and GR(2^2, 4), while their maximal cyclic subgroups G_3 and G_{15} of orders 3 and 15 are, respectively, isomorphic to the cyclic Galois groups GF (2^2)* and GF (2^4)*. The connection of maximal cyclic subgroup with cyclic Galois group is due to the mod-p reduction map from local ring $\mathbb{Z}_{2^k}$ to its residue field GF(2). In fact this relationship motivates the creation of the S-boxes over maximal cyclic subgroup and these new S-boxes have increased encryption and decryption complexities.

From the construction of S-boxes as proposed in Shah et al. (2013), we provide some 4×4 S-boxes using the maximal cyclic subgroups of the groups of units of GR(4, 4), GR(8, 4) and GR(32, 4). The connotation of maximal cyclic subgroups with promising cyclic Galois groups, derived from the mod-2 reduction maps from the rings $\mathbb{Z}_4$, $\mathbb{Z}_8$ and $\mathbb{Z}_{32}$ to their common residue field GF(2), supports in constructing the 4×4 S-boxes.

7.1 Construction of S-boxes based on maximal cyclic subgroups

In the following examples we discuss the construction method of S-boxes with size 4x4. We take the local rings $\mathbb{Z}_4 = \{0, 1, 2, 3\}, \mathbb{Z}_8 = \{0, 1, 2, \ldots, 7\}$, and $\mathbb{Z}_{16} = \{0, 1, 2, \ldots, 15\}, \mathbb{Z}_{32} = \{0, 1, 2, \ldots, 31\}$, while GF(2) is the common residue field. The monic polynomial $f(x) = x^4 + x + 1$ is basic irreducible over the rings $\mathbb{Z}_4$, $\mathbb{Z}_8$, and $\mathbb{Z}_{32}$ such that $\bar{f}(x) = f(x) \mathrm{mod} 2 = x^4 + x + 1$ is an irreducible polynomial over GF(2).

7.1.1 S-box over GF(2^4)

Take the polynomial ring GF(2) $[x] = \{a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n : a_i \in \mathrm{GF}(2), n \in \mathbb{Z}^+\}$ in the indeterminate x over GF(2). Let $\langle \bar{f}(x) \rangle = \{a(x) \cdot \bar{f}(x) : a(x) \in \mathrm{GF}(2)[x]\}$ be the principal ideal in GF(2)$[x]$ generated by $\bar{f}(x)$. Then the elements of the Galois extension field $K = \frac{\mathrm{GF}(2)[x]}{\langle \bar{f}(x) \rangle}$ of order 16 are shown in Table 11.

Table 11 Elements of Galois field GF(2^4)

Exp	Polynomial	Binary string
$-\infty$	0	0000
0	1	1000
1	$1+x$	1100
2	$1+x^2$	1010
3	$1+x+x^2+x^3$	1111
4	x	0100
5	$x+x^2$	0110
6	$x+x^3$	0101
7	$1+x^2+x^3$	1011
8	x^2	0010
9	x^2+x^3	0011
10	$1+x+x^2$	1110
11	$1+x^3$	1001
12	x^3	0001
13	$1+x+x^3$	1101
14	$x+x^2+x^3$	0111

Table 12 S-box over GF(2^4)

0	11	12	6
3	8	4	2
1	9	13	15
14	7	10	5

A basic S-box construction over GF(2^4) (see Table 11) is shown in Table 12. It satisfies all the fundamental properties of being an S-box.

7.1.2 S-box over GR(4, 4)

Let $\mathbb{Z}_4[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in \mathbb{Z}_4, n \in \mathbb{Z}^+\}$ be the polynomial ring in the indeterminate x and $\langle f(x)\rangle = \{a(x).f(x) : a(x) \in \mathbb{Z}_4[x]\}$ be the principal ideal generated by $f(x)$. Thus $R = \frac{\mathbb{Z}_4[x]}{\langle f(x)\rangle} = \left\{a_0 + a_1x + a_2x^2 + \ldots a_{4-1}x^{4-1} : a_i \in \mathbb{Z}_4\right\}$ is the Galois ring extension of order 256 with corresponding Galois field extension $K = \frac{\mathrm{GF}(2)[x]}{\langle \bar{f}(x)\rangle}$ of order 16, whose elements are shown in Table 11.

The maximal cyclic subgroup of R^*, isomorphic to the cyclic Galois group K^* of order 15, is denoted by G_{15} and it is shown in Table 13.

By using the maximal cyclic subgroup shown in Table 13, we obtain the S-box as shown in Table 14.

7.1.3 S-box on GR(8, 4)

Let $\mathbb{Z}_8[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in Z_8, n \in \mathbb{Z}^+\}$ be the polynomial ring in the indeterminate x and $\langle f(x)\rangle = \{a(x).f(x) : a(x) \in \mathbb{Z}_8[x]\}$be the principal ideal

Table 13 Elements of $G_{15} \cup \{0\}$in GR(4, 4)

Exp	Polynomial	String mod4
$-\infty$	0	0000
0	1	1000
2	$1 + 2x + x^2$	1210
4	$3x + 2x^2$	0320
6	$2 + x + 3x^3$	2103
8	x^2	0010
10	$3 + 3x + x^2 + 2x^3$	3312
12	$2 + 2x + 3x^3$	2203
14	$x + 3x^2 + x^3$	0131
16	$3 + 3x$	3300
18	$3 + x + x^2 + 3x^3$	3113
20	$x + 3x^2 + 2x^3$	0132
22	$1 + 3x^2 + x^3$	1031
24	$3x^2 + 3x^3$	0033
26	$3 + x^3$	3001
28	$1 + 3x + 2x^2 + x^3$	1321

Table 14 S-box over GR(4, 4)

0	67	215	159
25	240	15	16
1	113	116	198
109	45	202	44

generated by $f(x)$. Thus $R = \frac{\mathbb{Z}_8[x]}{\langle f(x)\rangle} = \{a_0 + a_1x + a_2x^2 + \ldots a_3x^3 : a_i \in \mathbb{Z}_8\}$is the Galois ring extension of order 4096 with corresponding Galois field extension $K = \frac{\mathrm{GF}(2)[x]}{\langle \bar{f}(x)\rangle}$of order 16, whose elements are shown in Table 11. The cyclic subgroup of R^*, isomorphic to K^*, of order 15, is denoted by G_{15} and is shown in Table 15.

By using the maximal cyclic subgroup shown in Table 15, we obtain the S-box shown in Table 16.

7.1.4 S-box on GR(32, 4)

Let $\mathbb{Z}_{32}[x] = \{a_0+a_1x+a_2x^2+\cdots+a_nx^n : a_i \in \mathbb{Z}_{32}, n \in \mathbb{Z}^+\}$ be the polynomial ring in the indeterminate x and $\langle f(x)\rangle = \{a(x).f(x) : a(x) \in \mathbb{Z}_{32}[x]\}$be the principal ideal generated by $f(x)$. Thus $R = \frac{\mathbb{Z}_{32}[x]}{\langle f(x)\rangle} = \{a_0 + a_1x + a_2x^2 + \ldots a_3x^3 : a_i \in \mathbb{Z}_{32}\}$ is the Galois ring extension of order 1,048,576 with corresponding Galois field extension $K = \frac{\mathrm{GF}(2)[x]}{\langle \bar{f}(x)\rangle}$ of order 16, whose elements are shown in Table 11.

The cyclic subgroup of R^*, isomorphic to K^*, of order 15, is denoted by G_{15} and shown in Table 17.

By using the maximal cyclic subgroup shown in Table 17, we obtain the S-box shown in Table 18.

Table 15 Elements of $G_{15} \cup \{0\}$ in GR(8, 4)

Exp	Polynomial	String mod8
$-\infty$	0	0000
0	1	1000
2	$1 + 2x + 1x^2$	1210
4	$3x + 6x^2 + 4x^3$	0364
6	$2 + x + 3x^3$	2103
8	$4 + 4x + x^2 + 4x^3$	4414
10	$3 + 7x + x^2 + 2x^3$	3712
12	$6 + 6x + 3x^3$	6603
14	$x + 7x^2 + x^3$	0171
16	$7 + 7x$	7700
18	$7 + 5x + 5x^2 + 7x^3$	7557
20	$4 + x + 7x^2 + 6x^3$	4176
22	$1 + 7x^2 + 5x^3$	1075
24	$4x + 3x^2 + 3x^3$	0433
26	$7 + 5x^3$	7005
28	$5 + 7x + 2x^2 + 5x^3$	5725

Table 16 S-box on GR(8, 4)

0	3	111	123
81	224	63	100
1	193	200	10
189	195	60	152

Table 17 Elements of $G_{15} \cup \{0\}$ in GR(32, 4)

Exp	Polynomial	String mod32
$-\infty$	0	0000
0	1	1000
4	$3x + 6x^2 + 4x^3$	0364
8	$4 + 20x + 9x^2 + 20x^3$	4K9K
12	$30 + 14x + 8x^2 + 19x^3$	UE8J
16	$31 + 7x + 24x^3$	V70O
20	$28 + 9x + 31x^2 + 6x^3$	S9V6
24	$16 + 4x + 11x^2 + 11x^3$	G4BB
28	$13 + 15x + 18x^2 + 13x^3$	DFID
32	$17 + 2x + 17x^2 + 16x^3$	H2HG
36	$2 + 17x + 8x^2 + 3x^3$	2H83
40	$3 + 23x + x^2 + 26x^3$	3N1Q
44	$16 + 25x + 15x^2 + 17x^3$	GPFH
48	$15 + 29x + 5x^2 + 31x^3$	FT5V
52	$17 + 16x + 7x^2 + 29x^3$	HG7T
56	$31 + 8x + 24x^2 + 5x^3$	V8O5

Table 18 S-box in GR(32, 4)

0	17	34	60
96	175	81	255
1	48	237	222
31	227	144	132

These newly designed S-boxes have increasing encryption and decryption complexities as compared to the S-boxes constructed over GF(2^4). However, it would be extremely difficult to construct an S-box based on the Galois rings GR(64, 4), GR(128, 4), and GR(256, 4) by calculating the corresponding maximal cyclic subgroups of these rings. However, this is possible if one uses the computational method proposed in this study.

8 Conclusions

- A computational method was designed to obtain maximal cyclic subgroups of the groups of units of finite Galois rings.
- By this method we obtain an enormous number of maximal cyclic subgroups of the groups of units of finite Galois rings. Consequently, a desired size maximal cyclic subgroup can easily be used.
- A computational algorithm for constructing a Cayley table for any maximal cyclic subgroup was given.
- The stock of maximal cyclic subgroups of the groups of units of finite Galois rings will accelerate the process of construction and decoding of cyclic codes, particularly of BCH codes.
- This stock will help in S-box construction through maximal cyclic subgroups of the groups of units of finite Galois rings instead of the binary field extensions.

References

Adams C, Tavares S (1989) Good S-boxes are easy to find. In: Advances in cryptology: proceedings of CRYPTO_89. Lecture notes in computer science, pp 612–615

Andrade AA, Shah T (2011a) Goppa codes through polynomials of $B[X; (1/(2^2))Z_0]$ and its decoding principle. J Adv Res Appl Math 3(4):12–20

Andrade AA, Shah T (2011b) Goppa codes via $B[X; (1/(3^2))Z_0]$ and its decoding principle. Int J Appl Math 24(4):563–572

Andrade AA, Shah T, Khan A (2010a) Goppa codes through generalized polynomials and its decoding principle. Int J Appl Math 23(3):515–526

Andrade AA, Shah T, Khan A (2010b) Goppa codes through generalized polynomials and its decoding principle. Int J Appl Math 23(3):517–526

Andrade AA, Shah T, Khan A (2011) A note on linear codes over semigroup rings. TEMA Tend Mat Apl Comput 12(2):79–89

Andrade AA, Palazzo R Jr (1999) Construction and decoding of BCH codes over finite rings. Linear Algebra Appl 286:69–85

Andrade AA, Palazzo R Jr (2005) Linear codes over finite rings. TEMA Tend Mat Apl Comput 6(2):207–217

Cohen S (2009) Finite fields and applications. Cambridge University Press, England

Cui L, Cao Y (2007) A new S-box structure named Affine-Power-Affine. Int J Innov Comput I 3(3):45–53

Hussain I, Shah T (2013) Literature survey on nonlinear components and chaotic nonlinear components of block cipher. Nonlinear Dyn. 74:869–904

Hussain I, Shah T, Mahmood H, Gondal MA, Bhatti UY (2011) Some analysis of S-box based on residue of prime number. Proc Pak Acad Sci 48(2):111–115

Hussain I, Shah T, Gondal MA (2012a) Image encryption algorithm based on PGL (2, GF (2^8)) S-boxes and TD-ERCS chaotic sequence. Nonlinear Dyn 70(1):181–187. doi:10.1007/s11071-012-0440-0

Hussain I, Shah T, Gondal MA, Mahmood H (2012b) Generalized majority logic criterion to analyze the statistical strength of S-boxes. Z Naturforsch. 67a:282–288. doi:10.5560/ZNA.2012-0022

Hussain I, Shah T, Mahmood H (2013) A group theoretic approach to construct cryptographically strong substitution boxes. Neural Comput Appl 23:97–104. doi:10.1007/s00521-012-0914-5

Kim J, Phan RCW (2009) Advanced differential-style crypt-analysis of the NSA's skipjack block cipher. Cryptologia 33(3):246–270

McDonald BR (1974) Finite rings with identity. Marcel Dekker. Inc., New York

Shah T, Andrade AA (2012a) Cyclic codes through $B[X]$, $B[X; (1/(kp))Z_0]$: a comparison. J Algebra Appl 11(4):19. doi:10.1142/S0219498812500788

Shah T, Andrade AA (2012b) Cyclic codes through $B[X; (\frac{a}{p})Z_0]$, $(a/b \in Q^+, b = a + 1)$ and encoding. Discret Math Algorithms Appl (DMAA) 04(04). doi:10.1142/S1793830912500590

Shah T, Andrade AA (2012c) Linear codes over finite local rings in a chain. J Adv Res Appl Math 4(4):66–77

Shah T, Khan A, Andrade AA (2011a) Encoding through generalized polynomial codes. Comput Appl Math 30(2):1–18

Shah T, Khan A, Andrade AA (2011b) Constructions of codes through semigroup ring $B[X; (1/(2^2))Z_0]$and encoding. Comput Math Appl 62(4):1645–1654

Shah T, Qamar A, Andrade AA (2012a) Construction and decoding of BCH codes over chain of commutative rings. Math Sci 6:51. doi:10.1186/2251-7456-6-51

Shah T, Qamar A, Andrade AA (2012b) Constructions and decoding of a sequence of BCH codes. Math Sci Res J 16(9):234–250

Shah T, Qamar A, Hussain I (2013) Substitution box on maximal cyclic subgroup of units of a Galois ring. Naturforsch A 68a:567–572. doi:10.5560/ZNA.2013-0021

Shanbhag AG, Kumar PV Helleseth T (1996) Upper bound for a hybrid sum over Galois rings with applications to the aperiodic correlation of some q-ary sequences. IEEE Trans Inf Theory 42(1):250–254

Shankar P (1979) On BCH codes over arbitrary integer rings. IEEE Trans Inf 25(4):480–483

Tran MT, Bui DK, Doung AD (2008) Gray S-box for advanced encryption standard. Int Conf Comput Intell Secur 1:253–256

Webster AF, Tavares S (1986) On the design of S-boxes. In: Advances in cryptology: proceedings of CRYPTO_85. Lecture notes in computer science, pp 523–534

Yi X, Cheng SX, You XH, Lam KY (2002) A method for obtaining cryptographically strong 8×8 S-boxes. Int Conf Inf Netw Appl 2(3):14–20