Chapter 4 – Finite Fields

Introduction

- will now introduce finite fields
- of increasing importance in cryptography
  - AES, Elliptic Curve, IDEA, Public Key
- concern operations on “numbers”
  - what constitutes a “number”
  - the type of operations and the properties
- start with concepts of groups, rings, fields from abstract algebra

Group

- a set of elements or "numbers"
  - A generalization of usual arithmetic
- obeys:
  - closure: `a.b also in G`
  - associative law: $(a.b).c = a.(b.c)$
  - has identity e: $e.a = a.e = a$
  - has inverses a^{-1}: $a.a^{-1} = e$
- if commutative $a.b = b.a$
  - then forms an **abelian group**
- Examples in P.105

Cyclic Group

- define **exponentiation** as repeated application of operator
  - example: $a^3 = a.a.a$
- and let identity be: $e=a^0$
- a group is cyclic if every element is a power of some fixed element
  - ie $b = a^k$ for some a and every b in group
- a is said to be a generator of the group
- Example: positive numbers with addition

Ring

- a set of "numbers" with two operations (addition and multiplication) which are:
- an abelian group with addition operation
- multiplication:
  - has closure
  - is associative
  - distributive over addition: `a(b+c) = ab + ac`
- In essence, a ring is a set in which we can do addition, subtraction [a – b = a + (–b)], and multiplication without leaving the set.
- With respect to addition and multiplication, the set of all *n*-square matrices over the real numbers form a ring.

Ring

- if multiplication operation is commutative, it forms a **commutative ring**
- if multiplication operation has an identity element and no zero divisors ($ab=0$ means either $a=0$ or $b=0$), it forms an **integral domain**
- The set of Integers with usual + and x is an integral domain

Field

- a set of numbers with two operations:
  - Addition and multiplication
  - F is an integral domain
  - F has multiplicative reverse
    - For each a in F other than 0, there is an element b such that ab=ba=1
- In essence, a field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set.
  - Division is defined with the following rule: $a/b = a\,(b^{-1})$
- Examples of fields: rational numbers, real numbers, complex numbers. Integers are NOT a field.

Definitions

Field Integral domain Commutative ring Ring Abelian group Group

(A1) Closure under addition:	If a and b belong to S, then $a + b$ is also in S
(A2) Associativity of addition:	$a + (b + c) = (a + b) + c$ for all a, b, c in S
(A3) Additive identity:	There is an element 0 in R such that $a + 0 = 0 + a = a$ for all a in S
(A4) Additive inverse:	For each a in S there is an element $-a$ in S such that $a + (-a) = (-a) + a = 0$
(A5) Commutativity of addition:	$a + b = b + a$ for all a, b in S
(M1) Closure under multiplication:	If a and b belong to S, then ab is also in S
(M2) Associativity of multiplication:	$a(bc) = (ab)c$ for all a, b, c in S
(M3) Distributive laws:	$a(b + c) = ab + ac$ for all a, b, c in S $(a + b)c = ac + bc$ for all a, b, c in S
(M4) Commutativity of multiplication:	$ab = ba$ for all a, b in S
(M5) Multiplicative identity:	There is an element 1 in S such that $a1 = 1a = a$ for all a in S
(M6) No zero divisors:	If a, b in S and $ab = 0$, then either $a = 0$ or $b = 0$
(M7) Multiplicative inverse:	If a belongs to S and a 0, there is an element a^{-1} in S such that $aa^{-1} = a^{-1}a = 1$

Figure 4.1 Group, Ring, and Field

Modular Arithmetic

- define **modulo operator** `a mod n` to be remainder when a is divided by n
  - e.g. 1 = 7 mod 3 ; 4 = 9 mod 5
- use the term **congruence** for: $a \equiv b \pmod{n}$
  - when divided by *n,* a & b have same remainder
  - eg. $100 \equiv 34 \pmod{11}$
- b is called the **residue** of a mod n
  - since with integers can always write: $a = qn + b$
- usually have `0 <= b <= n-1`

  `-12 mod 7 = -5 mod 7 = 2 mod 7 = 9 mod 7`

Modulo 7 Example

...						
-21	-20	-19	-18	-17	-16	-15
-14	-13	-12	-11	-10	-9	-8
-7	-6	-5	-4	-3	-2	-1
0	**1**	**2**	**3**	**4**	**5**	**6**
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
...						

all numbers in a column are equivalent (have same remainder) and are called a **residue class**

Divisors

- say a non-zero number `b` **divides** `a` if for some `m` have `a=mb` (`a,b,m` all integers)
  - $0 \equiv$ `a mod b`
- that is `b` divides into `a` with no remainder
- denote this `b|a`
- and say that `b` is a **divisor** of `a`
- eg. all of 1,2,3,4,6,8,12,24 divide 24

Modular Arithmetic Operations

- has a finite number of values, and loops back from either end
- modular arithmetic
  - Can perform addition & multiplication
  - Do modulo to reduce the answer to the finite set
- can do reduction at any point, ie
  - `a+b mod n = a mod n + b mod n`

Modular Arithmetic

- can do modular arithmetic with any group of integers: $Z_n = \{0, 1, \ldots, n-1\}$
- form a commutative ring for addition
- with an additive identity (Table 4.2)
- some additional properties
  - if $(a+b) \equiv (a+c) \bmod n$ then $b \equiv c \bmod n$
  - but $(ab) \equiv (ac) \bmod n$ then $b \equiv c \bmod n$ only if a is relatively prime to n

Modulo 8 Example

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(a) Addition modulo 8

Greatest Common Divisor (GCD)

- a common problem in number theory
- GCD (a,b) of a and b is the largest number that divides both a and b
  - eg GCD(60,24) = 12
- often want **no common factors** (except 1) and hence numbers are **relatively prime**
  - eg GCD(8,15) = 1
  - hence 8 & 15 are relatively prime

Euclid's GCD Algorithm

- an efficient way to find the GCD(a,b)
- uses theorem that:
  - `GCD(a,b) = GCD(b, a mod b)`
- **Euclid's Algorithm** to compute GCD(a,b):
  - `A=a, B=b`
  - `while B>0`
    - `R = A mod B`
    - `A = B, B = R`
  - `return A`

Example GCD(1970,1066)

```
1970 = 1 x 1066 + 904   gcd(1066, 904)
1066 = 1 x 904 + 162    gcd(904, 162)
904 = 5 x 162 + 94            gcd(162, 94)
162 = 1 x 94 + 68             gcd(94, 68)
94 = 1 x 68 + 26        gcd(68, 26)
68 = 2 x 26 + 16        gcd(26, 16)
26 = 1 x 16 + 10        gcd(16, 10)
16 = 1 x 10 + 6         gcd(10, 6)
10 = 1 x 6 + 4                gcd(6, 4)
6 = 1 x 4 + 2                 gcd(4, 2)
4 = 2 x 2 + 0                 gcd(2, 0)
```

- Compute successive instances of GCD(a,b) = GCD(b,a mod b).
- Note this MUST always terminate since will eventually get a mod b = 0 (ie no remainder left).

Galois Fields

- finite fields play a key role in many cryptography algorithms
- can show number of elements in any finite field **must** be a power of a prime number p^n
- known as Galois fields
- denoted $GF(p^n)$
- in particular often use the fields:
  - $GF(p)$
  - $GF(2^n)$

Galois Fields GF(p)

- GF(p) is the set of integers {0,1, … , p-1} with arithmetic operations modulo prime p
- these form a finite field
  - since have multiplicative inverses
- hence arithmetic is “well-behaved” and can do addition, subtraction, multiplication, and division without leaving the field GF(p)
  - Division depends on the existence of multiplicative inverses. Why p has to be prime?

Example GF(7)

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

(b) Multiplication modulo 7

Example: 3/2=5
GP(6) does not exist

Finding Inverses

- Finding inverses for large P is a problem
- can extend Euclid's algorithm:

```
EXTENDED EUCLID(m, b)
1. (A1, A2, A3)=(1, 0, m);
   (B1, B2, B3)=(0, 1, b)
2. if B3 = 0
   return A3 = gcd(m, b); no inverse
3. if B3 = 1
   return B3 = gcd(m, b); B2 = b^-1 mod m
4. Q = A3 div B3
5. (T1, T2, T3)=(A1 – Q B1, A2 – Q B2, A3 – Q B3)
6. (A1, A2, A3)=(B1, B2, B3)
7. (B1, B2, B3)=(T1, T2, T3)
8. goto 2
```

Inverse of 550 in GF(1759)

Q	A1	A2	A3	B1	B2	B3
–	1	0	1759	0	1	550
3	0	1	550	1	–3	109
5	1	–3	109	–5	16	5
21	–5	16	5	106	–339	4
1	106	–339	4	–111	355	1

Prove correctness

Polynomial Arithmetic

- can compute using polynomials
- se

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i$$

  -
  - poly arithmetic with coefficients mod p
  - poly arithmetic with coefficients mod p and polynomials mod another polynomial M(x)
- Motivation: use polynomials to model Shift and XOR operations

Ordinary Polynomial Arithmetic

- add or subtract corresponding coefficients
- multiply all terms by each other
- eg
  - let $f(x) = x^3 + x^2 + 2$ and $g(x) = x^2 - x + 1$

  $f(x) + g(x) = x^3 + 2x^2 - x + 3$

  $f(x) - g(x) = x^3 + x + 1$

  $f(x) \text{ x } g(x) = x^5 + 3x^2 - 2x + 2$

Polynomial Arithmetic with Modulo Coefficients

- when computing value of each coefficient, modulo some value
- could be modulo any prime
- but we are most interested in mod 2
  - ie all coefficients are 0 or 1
  - eg. let $f(x) = x^3 + x^2$ and $g(x) = x^2 + x + 1$

  $f(x) + g(x) = x^3 + x + 1$

  $f(x) \times g(x) = x^5 + x^2$

Modular Polynomial Arithmetic

- Given any polynomials f,g, can write in the form:
  - $f(x) = q(x)\, g(x) + r(x)$
  - can interpret $r(x)$ as being a remainder
  - $r(x) = f(x) \bmod g(x)$
- if have no remainder say $g(x)$ divides $f(x)$
- if $g(x)$ has no divisors other than itself & 1 say it is **irreducible** (or **prime**) polynomial
- Modular polynomial arithmetic modulo an irreducible polynomial forms a field
  - Check the definition of a field

Polynomial GCD

- can find greatest common divisor for polys
- GCD: the one with the greatest degree
  - $c(x)$ = GCD($a(x)$, $b(x)$) if $c(x)$ is the poly of greatest degree which divides both $a(x)$, $b(x)$
  - can adapt Euclid's Algorithm to find it:
  - EUCLID[$a(x)$, $b(x)$]

1. A(x) = $a(x)$; B(x) = $b(x)$

2. **2. if** B(x) = 0 **return** A(x) = gcd[$a(x)$, $b(x)$]

3. R(x) = A(x) mod B(x)

4. A(x) ¨ B(x)

5. B(x) ¨ R(x)

6. goto 2

Modular Polynomial Arithmetic

- can compute in field $GF(2^n)$
  - polynomials with coefficients modulo 2
  - whose degree is less than n
  - Coefficients always modulo 2 in an operation
  - hence must modulo an irreducible polynomial of degree n (for multiplication only)
- form a finite field
- can always find an inverse
  - can extend Euclid's Inverse algorithm to find

Example GF(2^3)

Table 4.6 Polynomial Arithmetic Modulo ($x^3 + x + 1$)

	+	000 0	001 1	010 x	011 $x+1$	100 x^2	101 x^2+1	110 x^2+x	111 x^2+x+1
000	0	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
001	1	1	0	$x+1$	x	x^2+1	x^2	x^2+x+1	x^2+x
010	x	x	$x+1$	0	1	x^2+x	x^2+x+1	x^2	x^2+1
011	$x+1$	$x+1$	x	1	0	x^2+x+1	x^2+x	x^2+1	x^2
100	x^2	x^2	x^2+1	x^2+x	x^2+x+1	0	1	x	$x+1$
101	x^2+1	x^2+1	x^2	x^2+x+1	x^2+x	1	0	$x+1$	x
110	x^2+x	x^2+x	x^2+x+1	x^2	x^2+1	x	$x+1$	0	1
111	x^2+x+1	x^2+x+1	x^2+x	x^2+1	x^2	$x+1$	x	1	0

(a) Addition

	×	000 0	001 1	010 x	011 $x+1$	100 x^2	101 x^2+1	110 x^2+x	111 x^2+x+1
000	0	0	0	0	0	0	0	0	0
001	1	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
010	x	0	x	x^2	x^2+x	$x+1$	1	x^2+x+1	x^2+1
011	$x+1$	0	$x+1$	x^2+x	x^2+1	x^2+x+1	x^2	1	x
100	x^2	0	x^2	$x+1$	x^2+x+1	x^2+x	x	x^2+1	1
101	x^2+1	0	x^2+1	1	x^2	x	x^2+x+1	$x+1$	x^2+x
110	x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2
111	x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$

(b) Multiplication

Computational Considerations

- since coefficients are 0 or 1, can represent any such polynomial as a bit string
- addition becomes XOR of these bit strings
- multiplication is shift & XOR
  - Example in P.133
- modulo reduction done by repeatedly substituting highest power with remainder of irreducible poly (also shift & XOR)

Summary

- have considered:
  - concept of groups, rings, fields
  - modular arithmetic with integers
  - Euclid’s algorithm for GCD
  - finite fields GF(p)
  - polynomial arithmetic in general and in $GF(2^n)$