Wireless communication Security
無線通訊安全

Lecture II-1

April 30, 2009

洪國寶

Outline

Part II:

(d) **橢圓曲線密碼技術**

- 基本代數概念
- 橢圓曲線簡介
- 基本橢圓曲線密碼協定
- 橢圓曲線之其他性質與應用

(e) **無線感測網路安全**

- 無線感測網路簡介
- 無線感測網路的安全議題
  - Key distribution/management
  - Secure routing

(f) **相關論文討論**

Some references

- http://**mathworld**.wolfram.com/topics/Algebra.html
- http://en.**wikipedia**.org/wiki/Main_Page
- Buttyán and Hubaux, **Security and Cooperation in Wireless Networks** **http://secowinet.epfl.ch/**
- Haenselmann, **An FDL'ed Textbook on Sensor Networks** **http://www.informatik.uni-mannheim.de/~haensel/sn_book/**

A (simple) problem

- **What is the solution of the equation**

  $$4x = 3$$

A (simple) problem

- **What is the solution of the equation**

  $$4x = 3$$

- The answer depends on what "things" we allow x to be.
  - If we are doing all our arithmetic using the integers then there is no solution--there is no integer that gives 3 upon being multiplied by 4.
  - On the other hand if we are doing our arithmetic in $Z/5$ ("Integers mod 5") then $x = 2$ is a solution.
  - If we are using the more usual rational number system Q, then the solution is $x = 3/4$.

A (simple) problem

- We can gain insight by considering the generalized equation $a \bullet x = b$ and then bringing up the questions:
  - What objects are a and b?
  - To what class of objects is x allowed to belong?
  - What is the operation symbolized by the dot (•)?

Algebraic Structures

- An algebraic structure consists of one or more **set**s closed under one or more **operation**s, satisfying some axioms.
  - **Groups** – structures of a set with a single binary operation
  - **Rings** and **fields**—structures of a set with two particular binary operations, (+) and (×)

Finite fields

- Finite fields are of increasing importance in cryptography
  - AES, **Elliptic Curve**, Public Key
- They concern operations on "numbers"
  - where what constitutes a "number" and the type of operations varies considerably
- We will start with concepts of groups, rings, fields from abstract algebra. ■

Group

Group – structure of a set with a single binary operation

- a set of elements or “numbers”
- with some operation whose result is also in the set (closure)
- obeys: (axioms)
  - 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** ■

Additive groups

- An additive group is a group where the operation is called **addition** and is denoted +. In an additive group, the identity element is called **zero**, and the inverse of the element *a* is denoted - *a* (**minus** *a*).
  - The symbols and terminology are borrowed from the additive groups of numbers: the ring of integers Z, the field of rational numbers Q, the field of real numbers R, and the field of complex numbers C are all additive groups.
  - In general, every ring and every field is an additive group.
- Scalar multiplication

Multiplicative groups

- A group whose group operation is identified with **multiplication**. As with normal multiplication, the multiplication operation on group elements is either denoted by · a raised dot or omitted entirely, giving the notation $g \cdot h$ or gh. In a multiplicative group, the identity element is denoted 1, and the inverse of the element is written as g^{-1}, voiced "*g* **inverse**."
  - This notation and terminology is borrowed from the multiplicative groups formed by numbers, where the operation is the usual arithmetical product, the identity element is the number 1, and the inverse coincides with the multiplicative reciprocal.
- Exponentiation

Cyclic Group

- define **exponentiation** as repeated application of multiplication operation
  - 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 ■

Some examples

Group	Not a Group
$(\boldsymbol{R}, +)$	$(\boldsymbol{R}, \bullet)$
$(\boldsymbol{Z}, +)$	$(\boldsymbol{N}, +)$
$(\boldsymbol{Q}, +)$	$(\boldsymbol{Q}, \bullet)$
$(\boldsymbol{R}\backslash\{0\}, \bullet)$	$(\boldsymbol{R}\backslash\{0\}, +)$
$(\{0\}, -)$	$(\boldsymbol{Z}, -)$
$(\boldsymbol{R}^{+}, \bullet)$	$(\boldsymbol{R}^{-}, \bullet)$
(all polynomials, +)	(all polynomials, •)
$(\boldsymbol{Q}\backslash\{0\}, \bullet)$	$(\boldsymbol{Q}\backslash\{0\}, +)$

A non-commutative Groups

- The set of N-by-N non-singular matrices form a group under matrix multiplication.
  - The product of two N-by-N nonsingular matrices is an N-by-N nonsingular matrix; (closure)
  - matrix multiplication is associative,
  - the set contains the identity matrix and
  - since the matrices are non-singular they have inverses which are also non-singular.
- This is a non-commutative group as matrix multiplication does not generally commute.

A (simple) problem (again)

- Now let's go back to our original question: what is the solution of $a \bullet x = b$
- In "solving" this equation we will assume that a and b are elements of a group with the group operation symbolized by •.
- We are looking for the member of the group that x could be replaced by to satisfy the equation.
- We'll use the group axioms to "solve" the equation in any group. (continued in next slide)

A (simple) problem (again)

- Using the closure axiom and the axiom for inverses we operate on both sides of the equation by the inverse of a.
- The inverse axiom says that a^{-1}, the inverse of a exists and the closure axiom says that the product of a^{-1} and any other group element exists and is still in the group.

  $$a^{-1} \bullet (a \bullet x) = a^{-1} \bullet b$$

- Now applying the associative axiom, $(a^{-1} \bullet a) \bullet x = a^{-1} \bullet b$
- The axiom of inverses gives $e \bullet x = a^{-1} \bullet b$
- Finally using the axiom of identity we get, $x = a^{-1} \bullet b$

A (simple) problem (again)

- So we "solved" the equation without answering the questions about *a*, *b* or *x* actually were or even what the operation indicated by • was. **This is the power of abstraction.**
  - **The emphasis is not on the particular group we might be interested in for some given application.**
  - **The emphasis is on the basic quality of *groupness* that all groups have in common.**

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that James Ellis (UK CESG) secretly proposed the concept in 1970
- is a practical method for public exchange of a secret key
- used in a number of commercial products

Diffie-Hellman Setup

- all users agree on global parameters:
  - large prime integer q
  - α a primitive root mod q
- each user (eg. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - compute their **public key**: $y_A = \alpha^{x_A} \bmod q$
- each user makes public that key y_A

Diffie-Hellman Key Exchange

- shared session key for users A & B is K_{AB}:

  $K_{AB} = \alpha^{x_A x_B} \bmod q$

  $= y_A^{x_B} \bmod q$ (which **B** can compute)

  $= y_B^{x_A} \bmod q$ (which **A** can compute)
- K_{AB} is used as session key in private-key encryption scheme between Alice and Bob
- if Alice and Bob subsequently communicate, they will have the **same** key as before, unless they choose new public-keys
- attacker needs an x, must solve discrete log

Diffie-Hellman Example

- users Alice & Bob who wish to swap keys:
- agree on prime $q=353$ and $\alpha=3$
- select random secret keys:
  - A chooses $x_A=97$, B chooses $x_B=233$
- compute public keys:
  - $y_A=3^{97} \bmod 353 = 40$ (Alice)
  - $y_B=3^{233} \bmod 353 = 248$ (Bob)
- compute shared session key as:

$K_{AB}= y_B^{x_A} \bmod 353 = 248^{97} = 160$ (Alice)

$K_{AB}= y_A^{x_B} \bmod 353 = 40^{233} = 160$ (Bob)

Abstract (generalized) Diffie-Hellman Key Exchange

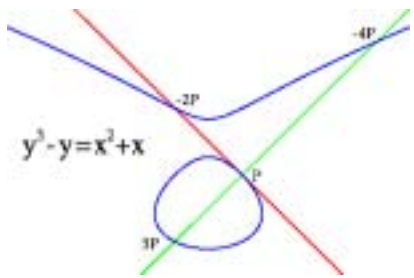
- Here's a more general description of the Diffie-Hellman Key Exchange protocol:

1. Alice and Bob agree on a finite cyclic group G and a generating element g in G.
   - This is usually done long before the rest of the protocol; g is assumed to be known by all attackers.)
   - We will write the group G multiplicatively.
2. Alice picks a random natural number a and sends g^a to Bob.
3. Bob picks a random natural number b and sends g^b to Alice.
4. Alice computes $(g^b)^a$.
5. Bob computes $(g^a)^b$.

Abstract (generalized) Diffie-Hellman Key Exchange

Works on any finite cyclic group G

- The multiplicative group Z^*_p of the integers modulo a prime p.
- The multiplicative group $F^*_{2^m}$ of the finite field $F^*_{2^m}$ of characteristic 2
- **The group of points on an elliptic curve over a finite field**
- Braid group based on Knot theory
- Ideal class group of number fields
- Jacobian varieties of curves over finite fields. ■



Further abstraction (Diffie-Hellman)

- We can further minimize the requirements needed for key exchange.
- We will come back to this in a moment.

Subgroup

- In group theory, given a group G under a binary operation *, we say that some subset H of G is a subgroup of G if H also forms a group under the operation *.
- Given a subgroup H and some a in G, we define the left coset aH = {ah : h in H}.
- Every element of G is contained in precisely one left coset of H; the left cosets are the equivalence classes corresponding to the equivalence relation $a_1 \sim a_2$ if and only if $a_1^{-1}a_2$ is in H.
- The number of left cosets of H is called the *index* of H in G and is denoted by $[G : H]$.
- Lagrange's theorem states that for a finite group G and a subgroup H,

$$[G : H] = \frac{|G|}{|H|}$$

where $|G|$ and $|H|$ denote the order of G and H, respectively.

Normal subgroup

- Right cosets are defined analogously: Ha = {ha : h in H}.
- If aH = Ha for every a in G, then H is said to be a normal subgroup.
- A subgroup, *N*, of a group, *G*, is called a **normal subgroup** if it is invariant under conjugation; that is, for each element, *n*, in *N* and each *g* in *G*, the element gng^{-1} is still in *N*. We write

$$N \triangleleft G \iff \forall n \in N, \forall g \in G\ , gng^{-1} \in N$$

Quotient group

- A **quotient group** (or factor group) is a group obtained by identifying together elements of a larger group using an equivalence relation.
- For example, the cyclic group of addition modulo n can be obtained from the integers by identifying elements that differ by a multiple of n.

Homomorphism

- In abstract algebra, a **homomorphism** is a **structure-preserving map** between two algebraic structures (such as groups, rings, or vector spaces).
  - An **isomorphism** is a bijective homomorphism.
  - An epimorphism is a surjective homomorphism.
  - An **endomorphism** is a homomorphism from an object to itself.
  - An **automorphism** is an endomorphism which is also an isomorphism.

Homomorphism

- Any homomorphism f : X $\rightarrow$ Y defines an equivalence relation ~ on X by a ~ b if and only if f(a) = f(b). The relation ~ is called the kernel of f. It is a congruence relation on X. The quotient set X/~ can then be given an object-structure in a natural way, i.e. [x] * [y] = [x * y]. In that case the image of X in Y under the homomorphism f is necessarily isomorphic to X/~; this fact is one of the isomorphism theorems.
- If G and H are groups, a **homomorphism** from G to H is a function f: $G \rightarrow H$ such that for any elements $g_1, g_2 \in G$, where $\square$ denotes the respective binary operations (the first $\square$ denoting the operation in G, and the second $\square$ denoting the operation in H).

Isomorphism

Informal: Two groups are isomorphic if they are "essentially the same".

Definition: Let $(\boldsymbol{G}, \bullet)$ and $(\boldsymbol{H}, *)$ be groups. $\boldsymbol{G}$ is *isomorphic* to $\boldsymbol{H}$ if there exists a bijective function $\phi: \boldsymbol{G} \rightarrow \boldsymbol{H}$ such that $\forall a, b \in \boldsymbol{G}$,

$\phi(a \bullet b) = \phi(a) * \phi(b)$.

- Because the function is bijective, we know the groups are the same size.
- Because of the equation $\phi(a \bullet b) = \phi(a) * \phi(b)$, we know that the operation "works the same" in each group.

Kernel

- Let G and H be groups and let f be a group homomorphism from G to H. If e_H is the identity element of H, then the ***kernel*** of f is the preimage of the singleton set $\{e_H\}$; that is, the subset of G consisting of all those elements of G that are mapped by f to the element e_H. The kernel is usually denoted "ker f" (or a variation). In symbols:

$$\ker f := \{g \in G : f(g) = e_H\}.$$

- Since a group homomorphism preserves identity elements, the identity element e_G of G must belong to the kernel. The homomorphism f is injective if and only if its kernel is only the singleton set $\{e_G\}$.
- It turns out that ker f is not only a subgroup of G but in fact a normal subgroup. Thus, it makes sense to speak of the quotient group G /(ker f).
- The first isomorphism theorem for groups states that this quotient group is naturally isomorphic to the image of f (which is a subgroup of H).

Cyclic Groups and Subgroups

- The main point about cyclic groups and isomorphism is that all finite cyclic groups of the same order are isomorphic. Thus there is only one cyclic group (up to isomorphism) of order, say, 6. If we let a be a generator then the elements of the cyclic group of order 6 are

  $\boldsymbol{a, a^2, a^3, a^4, a^5}$**, and** $\boldsymbol{a^6 = e}$**.**

Cyclic Groups and Subgroups

- The canonical example of a cyclic group of order n is the additive group of integers mod n: Z/nZ.
- The set of integers mod n is not a group under multiplication because 0 has no inverse – there is nothing that multiplies 0 to give 1, the identity element for multiplication.
- If n is a prime number and we throw out zero the remaining elements of Z/nZ does form a group – a cyclic group.

Further abstraction (Diffie-Hellman)

- We can further minimize the requirements needed for key exchange.
- **Use blackboard**

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`
- if multiplication operation is commutative, it forms a **commutative ring**
- if multiplication operation has identity and no zero divisors, it forms an **integral domain** ■

Characteristic

- In mathematics, the **characteristic** of a ring R, often denoted char(R), is defined to be the smallest number of times one must add the ring's multiplicative identity element (1) to itself to get the additive identity element (0); the ring is said to have characteristic zero if this repeated sum never reaches the additive identity.
- That is, char(R) is the smallest positive number n such that

$$\underbrace{1 + \cdots + 1}_{\text{n summands}} = 0$$

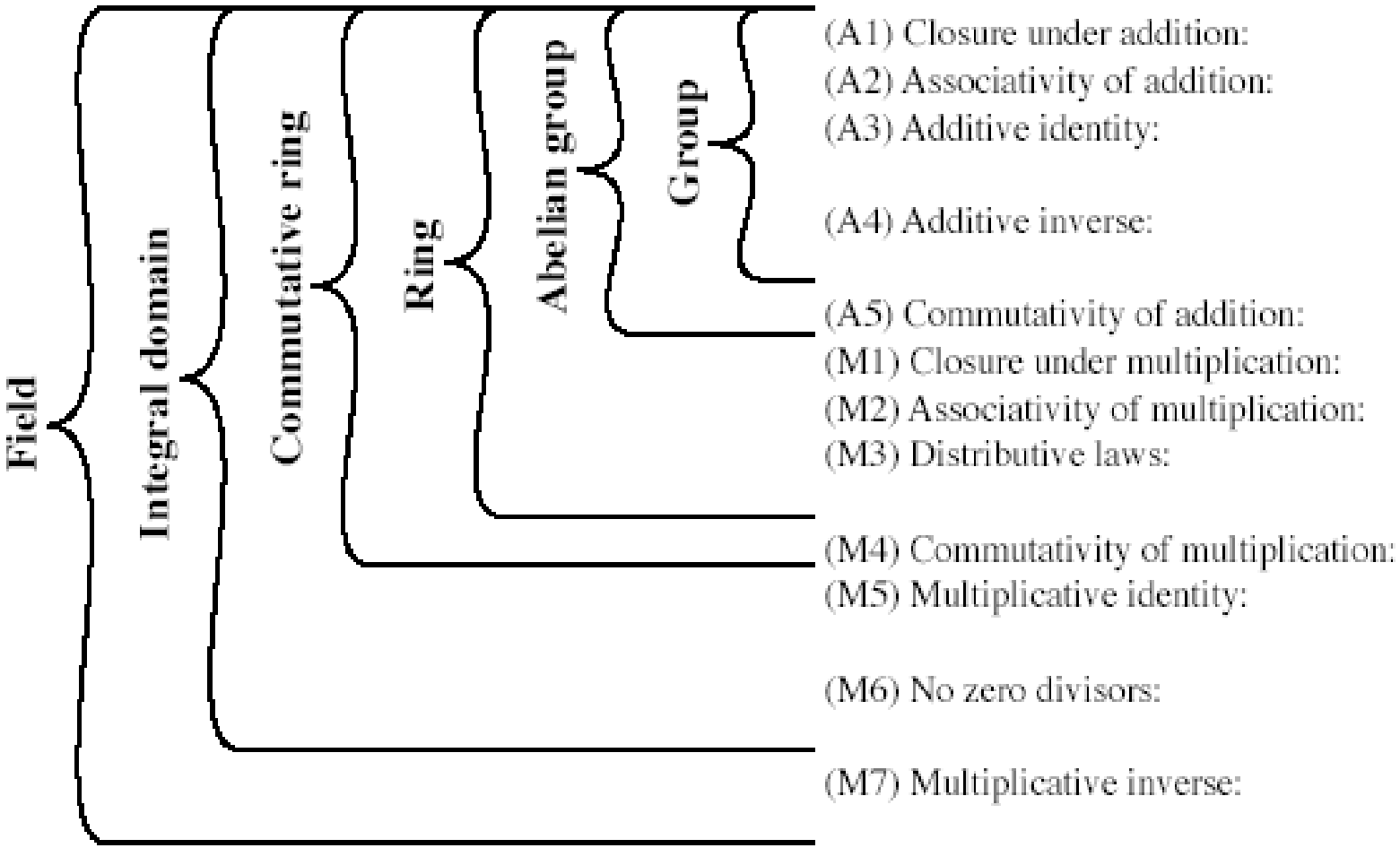
if such a number n exists, and 0 otherwise.

Frobenius endomorphism

- Let R be a commutative ring of positive and prime characteristic p (the characteristic is always prime when R is an integral domain, for example). The Frobenius endomorphism F is defined by $F(r) = r^p$ for all r in R.
- Let **F** be a field of field characteristic p. Then the Frobenius automorphism on **F** is the map $\phi : F \to F$ which maps a^p to a for each element a of .

Field

- a set of numbers with two operations:
  - abelian group for addition
  - abelian group for multiplication (ignoring 0)
  - ring
- The characteristic of any field is either 0 or a prime number.


Field
Integral domain
Commutative ring
Ring
Abelian group
Group
(A1) Closure under addition:
(A2) Associativity of addition:
(A3) Additive identity:
(A4) Additive inverse:
(A5) Commutativity of addition:
(M1) Closure under multiplication:
(M2) Associativity of multiplication:
(M3) Distributive laws:
(M4) Commutativity of multiplication:
(M5) Multiplicative identity:
(M6) No zero divisors:
(M7) Multiplicative inverse:

(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 $a \quad 0$, there is an element a^{-1} in S such that $aa^{-1} = a^{-1}a = 1$

Modular Arithmetic

- define **modulo operator** `a mod n` to be remainder when a is divided by n
- use the term **congruence** for: $a \equiv b \bmod n$
  - when divided by *n*, a & b have same remainder
  - eg. 100 = 34 mod 11
- b is called the **residue** of a mod n
  - since with integers can always write: $a = qn + b$
- usually have $0 \leq b \leq n-1$
  - $-12 \bmod 7 \equiv -5 \bmod 7 \equiv 2 \bmod 7 \equiv 9 \bmod 7$ ■

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
...
```

Modular Arithmetic Operations

- is 'clock arithmetic'
- uses a finite number of values, and loops back from either end
- modular arithmetic is when do addition & multiplication and modulo reduce answer
- can do reduction at any point, ie
  - `a+b mod n = [a mod n + b mod n] 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 a multiplicative identity
- note some peculiarities
  - 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** ■

Property	Expression
Commutative laws	$(w + x) \bmod n = (x + w) \bmod n$ $(w \times x) \bmod n = (x \times w) \bmod n$
Associative laws	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$ $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
Distributive laws	$[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$ $[w + (x \times y)] \bmod n = [(w + x) \times (w + y)] \bmod n$
Identities	$(0 + w) \bmod n = w \bmod n$ $(1 \times w) \bmod n = w \bmod n$
Additive inverse $(-w)$	For each $w \in Z_n$, there exists a z such that $w + z \equiv 0 \bmod n$

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

Modulo 8 Example (cont.)

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

(b) Multiplication modulo 8

Modulo 8 Example (cont.)

w	$-w$	w^{-1}
0	0	—
1	7	1
2	6	—
3	5	3
4	4	—
5	3	5
6	2	—
7	1	7

(c) Additive and multiplicative inverses modulo 8

Divisors

- say a non-zero number `b` **divides** `a` if for some `m` have `a=mb` (`a,b,m` all integers)
- 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 ■

Greatest Common Divisor (GCD)

- a common problem in number theory
- GCD (a,b) of a and b is the largest number that divides evenly into 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:
  - $\text{GCD}(a,b) = \begin{cases} a & \text{if } b = 0 \\ \text{GCD}(b,\ a \bmod b) & \text{otherwise} \end{cases}$
- **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) ■

Galois Fields

- finite fields play a key role in cryptography
- can show number of elements in a finite field **must** be a power of a prime 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) ■

Example GF(7)

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

(a) Addition modulo 7

Example GF(7) (cont.)

×	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 GF(7) (cont.)

w	$-w$	w^{-1}
0	0	—
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

(c) Additive and multiplicative inverses modulo 7

Finding Inverses

1. Construct a multiplicative table (e.g. Table 4.3)
2. Use extended Euclid algorithm
3. Apply Fermat's little theorem ■

Finding Inverses

Extended 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

Finding Inverses

- Fermat's little theorem
  - If $(A, P) = 1$ then $A^{P-1} \equiv 1 \bmod P$

Inverse of 550 in GF(1759)

- A = 550, P = 1759
  - $A^{P-2} = 550^{1757} \bmod 1757$

Finding Inverses

Algorithm 4 *(Variation on Euclidean Algorithm (VEA)).*
INPUT: Two integers N and A such that $1 < A < N$.
OUTPUT:

$$\begin{cases} \text{modular inverse of } A \bmod N & \text{if } \mathrm{GCD}(N, A) = 1, \\ \mathrm{GCD}(N, A), & \text{otherwise.} \end{cases}$$

1. **if** $N \bmod A = 0$ then return $\mathrm{GCD}(N, A) = A$;
2. $R \leftarrow N^2$; $R' \leftarrow NA + 1$;
3. **while** $R' > N$ **do**
4. $\quad T \leftarrow R'$; $R' \leftarrow R \bmod R'$; $R \leftarrow T$;
5. **if** $N \bmod R' \neq 0$ **then return** modular inverse R';
6. $\quad$ **else return** $\mathrm{GCD}(N, A) = N/R'$.

Reference: Chao-Liang Liu, Gwoboa Horng, and Hsin-Yu Liu, "Computing the Modular Inverses Is as Simple as Computing the GCDs," *Finite Fields and Their Applications,* Vol. 14, issue 1, 65 – 75, 2008.

Inverse of 550 in GF(1759)

- A = 550, N = 1759
  - N^2 = 3094081, NA+1 = 967451 (use blackboard)

Galois Fields

- We have shown how to construct GF(p)
- We will now show how to construct GF(p^n) for $n > 1$ ■

Polynomial arithmetic

- Virtually all encryption algorithms involve arithmetic operations on integers
- **If division is required then we need to work over a field**
- For convenience and for implementation efficiency, we work with integers in the range 0 through 2^n-1 (n-bit word)
- With 8 bit, Z_{256} is not a field,
  - the closest prime is 251
  - Z_{251} is a field (inefficient use of storage)
- How to construct field with 256 elements? (**How to construct GF(p^n) for n > 1 in general?**)
  - We need polynomial arithmetic

Polynomial Arithmetic

- General form of polynomials

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

- Zeroth degree polynomials are called **constant polynomials**
- An nth degree polynomial is said to be a **monic polynomial** if $a_n = 1$
- several alternatives available
  - ordinary polynomial arithmetic
  - poly arithmetic with coords mod p
  - poly arithmetic with coords mod p and polynomials mod M(x) ■

Ordinary Polynomial Arithmetic

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

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

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

  $f(x) \times g(x) = x^5 + 3x^2 - 2x + 2$ ■

Polynomial Arithmetic with Modulo Coefficients

- when computing value of each coefficient do calculation modulo some value
- could be modulo any prime p
- but we are most interested in **mod 2**
  - i.e. **all coefficients are 0 or 1** (i.e. over Z_2)
  - 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

- can write any polynomial 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
- **arithmetic modulo an irreducible polynomial forms a field** ■

Galois Fields GF(p^n)

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

GF(2^n)

- Finite field GF(2^n)
  - polynomials with coefficients modulo 2
  - whose degree is less than n
  - hence must reduce modulo an irreducible poly of degree n (for multiplication only) ■

$GF(2^n)$

- $GF(2^3)$

  $(\{000,001,010,011,100,101,110,111\}, +, \times)$

  – $(\{0, 1, 2, 3, 4, 5, 6, 7\}, +_{\text{mod } 8}, \times_{\text{mod } 8})$

    - Some non-zero elements have no multiplicative inverse
    - Does not form a finite field

Interpreted as binary integers

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

Modulo 8 Example (cont.)

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

(b) Multiplication modulo 8

Modulo 8 Example (cont.)

w	$-w$	w^{-1}
0	0	—
1	7	1
2	6	—
3	5	3
4	4	—
5	3	5
6	2	—
7	1	7

(c) Additive and multiplicative inverses modulo 8

$GF(2^n)$

Interpreted as coefficients of polynomials

- $GF(2^3)$

  $(\{000,001,010,011,100,101,110,111\}, +, \times)$

  – $(\{0, 1, x, x+1, x^2, x^2 + 1, x^2 + x, x^2 + x + 1\}, +_{\text{mod } 2}, \times_{\text{mod } p(x)})$

    - $p(x) = x^3 + x + 1$
    - **Form a field**
    - **Table 4.6**

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

Polynomial GCD

- can find greatest common divisor for polys
  - *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. **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

Finding inverses (polynomials)

- **L.C. Calvez, S. Azou and P. Vilbé, Variation on Euclid's algorithm for polynomials, *Electron. Lett.* 33 (11) (1997), pp. 939–940.**
- **A. Goupil and J. Palicot, Variation on variation on Euclid's algorithm, IEEE Trans. Signal Process. Lett. 11 (5) (2004), pp. 457–458.**

Algorithm *(Euclidean Algorithm for Polynomials (EAP)).*
INPUT: Two polynomials $A(x)$ and $B(x)$ with degree a and b, respectively, where $a, b > 0$.
OUTPUT: $U(x)$, $V(x)$ and $G(x)$ such that $G(x) = \mathrm{GCD}(A(x), B(x)) = A(x)U(x) + B(x)V(x)$.

1. $R(x) \leftarrow x^{a+b}A(x) + x^a$; $R'(x) \leftarrow x^{a+b}B(x) + 1$;
2. **while** $\deg(R'(x)) > a + b$ **do**
3. $\quad T(x) \leftarrow R'(x)$; $R'(x) \leftarrow R(x) \bmod R'(x)$; $R(x) \leftarrow T(x)$;
4. Compute $U(x)$, $V(x)$ and $G(x)$ such that
   $\quad R(x) = x^{a+b}G(x) + x^aU(x) + V(x)$;
5. **return** $U(x)$, $V(x)$ and $G(x)$.

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
  - cf long-hand multiplication
- modulo reduction done by repeatedly substituting highest power with remainder of irreducible poly (also shift & XOR) ■

Computational Example

- in GF(2^3) have (x^2+1) is 101_2 & (x^2+x+1) is 111_2
- so addition is
  - $(x^2+1) + (x^2+x+1) = x$
  - $101 \text{ XOR } 111 = 010_2$
- and multiplication is
  - $(x+1).(x^2+1) = x.(x^2+1) + 1.(x^2+1)$
    $= x^3+x+x^2+1 = x^3+x^2+x+1$
  - $011.101 = (101)\ll 1 \text{ XOR } (101)\ll 0 =$
    $1010 \text{ XOR } 101 = 1111_2$
- polynomial modulo reduction (get q(x) & r(x)) is
  - $(x^3+x^2+x+1) \bmod (x^3+x+1) = 1.(x^3+x+1) + (x^2) = x^2$
  - $1111 \bmod 1011 = 1111 \text{ XOR } 1011 = 0100_2$ ■

Using a Generator

- equivalent definition of a finite field
- a **generator** g is an element whose powers generate all non-zero elements
  - in F have 0, g^0, g^1, …, g^{q-2}
- can create generator from **root** of the irreducible polynomial
- then implement multiplication by adding exponents of generator ■