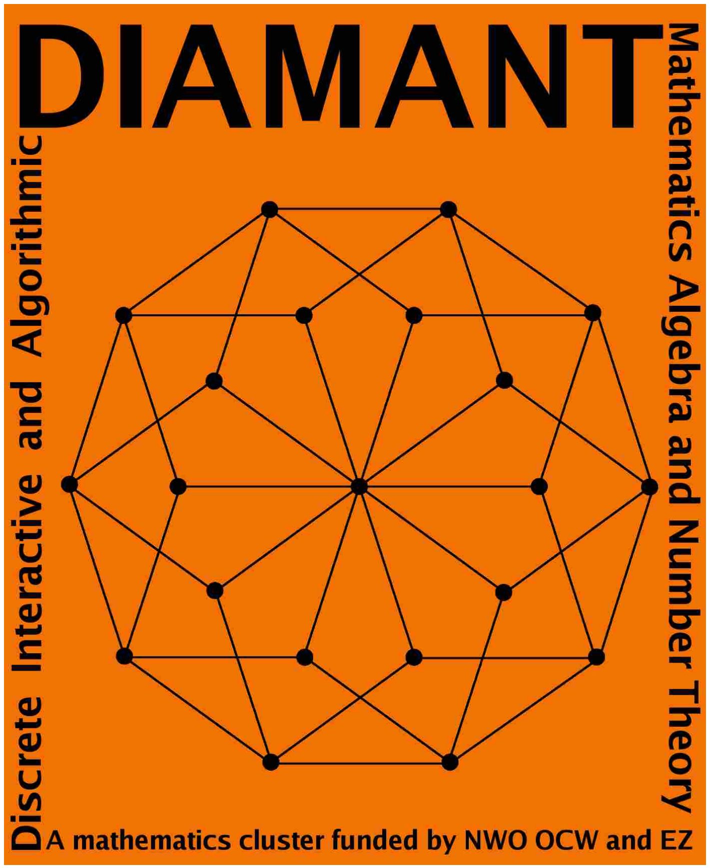
Welcome!



DIAMANT-Summer School on Elliptic and Hyperelliptic Curve Cryptography.

DIAMANT is one of three "mathematics clusters" in the Netherlands.
It is funded by NWO and the ministries of EZ and OCW.

Slides and exercise sheets will appear on the summer school home page.

Get online: user: tlange, password tue000

Background (Alice, Bob, and finite fields)

Tanja Lange

Department of Mathematics and Computer Science

Technische Universiteit Eindhoven

tanja@hyperelliptic.org

15.09.2008

Different methods for encryption

- Partners share a secret (key)
  $\Rightarrow$ can use symmetric encryption methods
  - classical methods (Caesar, ...) belong to this class
  - one-time pad
  - AES, stream ciphers
- Partners have no initial secret
  $\Rightarrow$ use of public key cryptography
  (topic of this school and most of ECC'08 falls in this category).

Symmetric methods are faster than public key methods for encryption. Most applications use a combination of both (e.g. public key cryptography is used to define a session key). This is called hybrid encryption.

Public key cryptography

Each user U has *two* keys

- a private key s_U, chosen by him which *never* leaves the secure environment,
- a public key p_U, which is made public in a kind of phone book.
- These keys are linked by a (computational) one-way function; the general principles are public.

- Use public key of recipient to encrypt
  $\Rightarrow$ only recipient knows secret key and can decrypt.
- Sender uses his private key to sign message
  $\Rightarrow$ signature must be due to him as only he knows the secret. Everybody can verify signature using public key.

Tasks of the cryptographer

- Find good candidates for computational one-way functions.
- Look for variants having better performance.
- Analysis of attacks.

Proposed candidates

- Multiplying two integers is easy while factoring is hard (RSA).

  primes $p, q \Rightarrow N = p \cdot q \longleftrightarrow$ integer $N \Rightarrow p, q$.
- Computing scalar multiples in some group is easy while it is hard to find the corresponding scalar given two group elements $\Rightarrow$ discrete logarithm problem (DLP).

  $P \in G$, integer $a \Rightarrow Q = [a]P \longleftrightarrow Q, P \in G \Rightarrow a$.

Signatures in discrete logarithm systems

Public parameters:
Group G, generator P, ord$(P) = \ell$, some function $H : G \to \mathbb{Z}/\ell\mathbb{Z}$.
Signer has secret key s_S and public key $p_S = [s_S]P$.

- Sign message $m \in \mathbb{Z}/\ell\mathbb{Z}$
  - choose random integer k
  - compute $R = [k]P$ and $S = s_S m + kH([k]P) \bmod \ell$
  - send (R, S)
- Validate signature (R, S) on m
  - Check whether $[S]P == [m]p_S \oplus [H(R)]R$

(This should agree as

$$[S]P = [s_S m]P \oplus [kH([k]P)]P = [m]([s_S]P) \oplus H(R)[R]).$$

In practice, $h(m)$ for a hash function h is used instead of m.

Diffie-Hellman key exchange

Parameters

$P, |\langle P \rangle|$

Alice

1. secretly generates
   $a < |\langle P \rangle|$
2. computes $Q_1 = [a]P$
3. transmits Q_1
4. computes
   $[a]Q_2$

Bob

1. secretly generates
   $b < |\langle P \rangle|$
2. computes $Q_2 = [b]P$
3. transmits Q_2
4. computes
   $[b] \cdot Q_1$

$[a]Q_2 = [ab]P = [b] \cdot Q_1$

Common Key: the group element $k = [(ab)]P \in G$
can be used in symmetric encryption.

Example: key exchange in $\mathbb{Z}$

Public parameter base $P = 2$ and operation $*$.

- First step
  - A chooses a secret number $s_a = 7$
  - B chooses a secret number $s_b = 9$
- second step
  - A computes 2^7 and sends $p_a = 128$
    // for details on computing 2^7 see Christophe's talk
  - B computes 2^9 and sends $p_a = 512$
- third step
  - A computes ${p_B}^7 = 512^7 = 2^{63}$
  - B computes ${p_A}^9 = 128^9 = 2^{63}$

Example: key exchange in $\mathbb{Z}$

Public parameter base $P = 2$ and operation $*$.

- First step
  - A chooses a secret number $s_a = 7$
  - B chooses a secret number $s_b = 9$
- second step
  - A computes 2^7 and sends $p_a = 128$
    // for details on computing 2^7 see Christophe's talk
  - B computes 2^9 and sends $p_a = 512$
- third step
  - A computes ${p_B}^7 = 512^7 = 2^{63}$
  - B computes ${p_A}^9 = 128^9 = 2^{63}$

Problem: Size reveals secret!

What went wrong?

- Actually, $\mathbb{Z}$ does not fit our description since $(\mathbb{Z}, *)$ is not a group and 2 does not have finite order.
- Key exchange only needs semi-group – but finiteness is important.
- Solution: use modular arithmetic.
  Additionally agree on a prime p such that all computations are done mod p. E.g. use $p = 97$. Then
  $$p_A = 128 - 97 = 31,$$
  $$p_B = 512 - 5 \cdot 97 = 27,$$
  and the joint secret key is
  $$p_B^7 = 27^7 \equiv 79 \equiv 31^9 \bmod 97.$$
- Now each value is bounded by 97, so p_A and p_B do not automatically reveal the secret keys s_A and s_B.

Proposed groups

- Integers modulo prime $\mathbb{F}_p^*$.
- Arbitrary finite fields $\mathbb{F}_q^*$.
- Points on elliptic curve over finite field.
- Jacobian of hyperelliptic curve over finite field.

Difference in security:

- first two examples have subexponential algorithms

$$O(\exp(c(\log q)^{1/3}(\log\log q)^{2/3})),$$

- for elliptic and small genus hyperelliptic curves the best algorithms are exponential

$$O(\exp(1/2\log\ell)) = O(\ell^{1/2}).$$

Mathematical background

Groups I

Let G be a set and let $\oplus$ be an operation on G such that

- $P \oplus Q \in G$ for all $P, Q \in G$ (G is closed under $\oplus$).
- $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$ for all $P, Q, R \in G$ ($\oplus$ is associative).
- Neutral element: there exists an element $O \in G$ so that for all $P \in G$ one has $P \oplus O = O \oplus P = P$.
- Inverse: for all $P \in G$ there exists an element $\mathrm{inv}(P) \in G$ with $P \oplus \mathrm{inv}(P) = O$ and $P \oplus \mathrm{inv}(P) = O$.

Then G is a group with respect to $\oplus$. We use $(G, \oplus)$ as shorthand.

$(G, \oplus)$ is commutative or abelian if for all $P, Q \in G$ one has $P \oplus Q = Q \oplus P$.

Groups II

Let $(G, \oplus)$ be a group with neutral element O and let $P \in G$.

- ord$(G) = |G|$ the order of G is the number of elements.
- G is a finite group if it has finitely many elements.
- ord(P) is the smallest positive integer n such that $[n]P = O$.
- Can show ord$(P)|$ord(G) for finite ord(G).
- G is cyclic if there exists a P so that for all $Q \in G$ there exists an $n \in \mathbb{Z}$ with $Q = [n]P$.
  P is called the generator of G.
- $\langle P \rangle = \{\dots [-3]P, [-2]P, [-1]P, O, P, [2]P, [3]P, \dots\}$
  cyclic group generated by P.
- Cyclic groups are always commutative (since $\mathbb{Z}$ is commutative).

Generic attacks I

- If ord$(P) = \ell_1\ell_2$ then ord$([\ell_2]P) = \ell_1$ and ord$([\ell_1]P) = \ell_2$. So instead of solving the DLP in $\langle P\rangle$ one can solve the DLPs in $\langle[\ell_1]P\rangle$ and $\langle[\ell_2]P\rangle$ and combine the results using Chinese Remainder Theorem. (Pohlig-Hellman).
- We usually work in a cyclic (sub-)group of prime order ℓ.
- Brute force attacks take $O(\ell)$ in a group of order ℓ.
- Discrete logarithms are "random self-reducible": can turn one DLP Q, P into many random ones $[r_i]Q, P$ using known integers $r_i \neq 0$. Original DLP is solved once any of these is solved.

Generic attacks II

- Let $m = \lceil \sqrt{\ell} \rceil$ and let $Q = [n]P$. Then $n = am + b$ with $0 < a, b < m$, so $Q \oplus [-a]([m]P) = [b]P$.
  One can compute, sort and store $[j]P$ for $0 \leq j < m$ in $O(m)$ group operations. Then one of the m values of i must give equality in $Q \oplus [i]([m]P) = [j]P$ with one j.
  This is Shank's baby-step giant-step algorithm which runs in any cyclic group in time and space $O(\sqrt{\ell})$.
- Attacking multiple targets can reuse first round of computation.
- Pollard's rho and kangaroo method run in time $O(\sqrt{\ell})$ and constant space by using deterministic walk with steps related to P and Q and keeping track of distance.
- More details in Dan Bernstein's talk at last year's tutorial.

Fields

Let K be a set and let two operations $+, \cdot$ with neutral elements $0, 1$ be defined on K such that

- $(K, +)$ is an abelian group.
- $(K^*, \cdot)$ is an abelian group, where $K^* = K \setminus \{0\}$.
- The distributive law holds in K:
  $a \cdot (b + c) = a \cdot b + a \cdot c$ for all $a, b, c \in K$.

Then K is a field with respect to $+, \cdot$, denoted by $(K, +, \cdot)$.

Let L be a field and $K \subseteq L$. If K is a field itself it is a subfield of L and L is an extension field of K.

L can be considered as a vector space over K. The extension degree is defined as $[L : K] = \dim_K(L)$ as vector space.

Polynomials

- Let K be a field. A polynomial in one variable x over K is a finite sum of powers of x with coefficients $f_i \in K$
$$f(x) = \sum_{i=0}^{n} f_i x^i, \; f_i \in K.$$
- We denote the set of polynomials in x over K by $K[x]$.
- Let n be maximal with $f_n \neq 0$, then $\deg(f) = n$ (called the degree of f) and $\mathrm{LC}(f) = f_n$ (called the leading coefficient of f). f is monic if $\mathrm{LC}(f) = 1$.
- One can add and multiply polynomials.
- f is irreducible if $f = g \cdot h$ implies g or h is constant.
- α is a root of f if $f(\alpha) = \sum_{i=0}^{n} f_i \alpha^i = 0$.
- If $f(\alpha) = 0$ then $(x - \alpha) | f(x)$.

Finite fields I

- A field with finitely many elements is a finite field.
- The smallest natural number $n > 0$ such that

$$n \cdot 1 = \underbrace{1 + 1 + \ldots + 1}_{n-\text{times}} = 0$$

  is called the characteristic of K, if no such n exists char$(K) = 0$.
- Finite fields have positive characteristic.
- The characteristic of a field must be 0 or prime.
- Well known example $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$; the integers modulo a prime p, is a finite field of characteristic p.

Finite fields II

- Let F be a finite field with char$(F) = p$, then F contains $\mathbb{F}_p$ as subfield. Let $[F : \mathbb{F}_p] = n$, then $|F| = p^n$. So finite fields have prime-power order. Notation $\mathbb{F}_q$ with $q = p^n$.
- So additive structure of $\mathbb{F}_q$ is that of a vector space of dimension n over $\mathbb{F}_p$.
- Can show that $\mathbb{F}_q^*$ is cyclic. Original version of Diffie-Hellman key exchange was defined over $\mathbb{F}_p^*$.
- Index calculus attacks in $\mathbb{F}_p$ use natural embedding of $\mathbb{F}_p$ into $\mathbb{Z}$.
- Index calculus attacks run in time subexponential in the field size. Security of DLP is weakest of $O(\sqrt{\ell})$ and $O(\exp(c(\log p)^{1/3}(\log\log p)^{2/3}))$.

Computation in finite fields

- Usual construction of $\mathbb{F}_{p^n}$: let $f \in \mathbb{F}_p[x]$ be irreducible of $\deg(f) = n$. Then $\mathbb{F}_{p^n} \cong \mathbb{F}_p[x]/f\mathbb{F}_p[x]$.
  Use polynomials in $\mathbb{F}_p[x]$ of degree $< n$ to represent residue classes.
- Computations are done in $\mathbb{F}_p[x]$, the results are then reduced modulo f.
- Speed of reduction modulo f depends on the number of non-zero coefficients in f; optimal (but not always possible) $f(x) = x^n - a$ for some $a \in \mathbb{F}_p$. See exercises and tomorrow's talk on fast arithmetic in finite fields.
- Note that x is root of f in $\mathbb{F}_{p^n}$.
- Process can be iterated to construct $\mathbb{F}_{q^m}$ as extension of $\mathbb{F}_q$ with an irreducible polynomial $g \in \mathbb{F}_q[y]$ of $\deg(g) = m$ and $\mathbb{F}_{q^m} \cong \mathbb{F}_q[y]/g\mathbb{F}_q[y]$.

Time for coffee break and questions