The Security of Elliptic Curve Cryptosystems -

Motorola Project 16

PI's

Dick Blahut
Iwan Duursma
Andreas Stein

Graduate Student RA's

Samuel Kadziela
Timothy Kilbourn
Eric Landquist
Jonathan Webster
Qingquan Wu

Motorola Contact

Doug Kuhlman
Tom Messerges
Larry Puhl

Overview

- Motivation
- Elliptic Curve Cryptography
  - Discrete Logarithm Problem
  - Subexponential Attacks
  - Implementation and Results
  - Elliptic Curve Construction
  - Weil-Pairing and Tate-Pairing
- Future Work
- Activities

Motivation

- The **Illinois Center for Cryptography and Information Protection** is a multidisciplinary center with affiliated faculty in Mathematics, Computer Science, and Electrical Engineering.

- The Center's expertise is in **watermarking** and **public-key cryptography**.

- The goals of the public-key cryptography group are to analyze and improve the **security and efficiency of public-key cryptosystems**.

(continues)

- **Security and efficiency** depend on
  - abstract mathematics (number theory, algebraic geometry)
  - computational feasibility of the underlying problems (subexponential attacks, parallelizable attacks)
  - subtle implementation issues.
- We study the **discrete logarithm problem**
  - for various class groups (number fields, elliptic and hyperelliptic curves)
  - using state of the art solutions (random walks, sieving methods, iterative methods in linear algebra)
  - implementing and optimizing the solutions in a combination of C, NTL and GMP.

Discrete Logarithm Problem

- Given elements g and h of a group G with $h \in \langle g \rangle$, find $x \in \mathbb{Z}$ such that $g^x = h$.

- Subexponential attack of the discrete logarithm in the class group of binary quadratic forms, $Cl(\Delta)$ for fixed discriminant Δ.

- First approach is a **random walk** method by Enge and Gaudry.

- A much quicker approach applies the Self-Initializing Quadratic Sieve **(SIQS)**.

- Both methods involve finding **smooth elements** of the group, putting the resulting information into a matrix, and solving the matrix to find a relation among the smooth elements to find the discrete logarithm.

Subexponential Attacks

- Elements in $Cl(\Delta)$ are binary quadratic forms, BQF's, of the form $aX^2 + bXY + cY^2$, where $a, b, c \in \mathbb{Z}, c = (b^2 - \Delta)/(2a)$. Denote the form (a, b).

- The class number, $h_\Delta = |Cl(\Delta)|$ is known.

- Create a **factor base** FB: prime elements (p_i, b_{p_i}), where $p_i \in \mathbb{Z}$ is prime, and $p_i \leq B$, for some bound B.

- Fill a matrix A with relations of **B-smooth** elements, i.e. that factor over FB.

- When there are more relations than $\#FB$, we find a nontrivial $\vec{k} \in Ker(A)$ using the **Lanczos Algorithm**, and determine x.

Finding Relations with a Random Walk

- Create say 16 multipliers, $m_i = g^{a_i} h^{b_i}$, where $a_i, b_i \in_R \{0, ..., h_\Delta - 1\}$.

- Perform a random walk through the group $w_j = w_{j-1} m_{H(w_{j-1})}$, testing elements $w_j = g^{\alpha_j} h^{\beta_j}$ on step j for smoothness.

- Record smooth factorizations in A, and exponents α_j and β_j in separate vectors.

- The value x such that $g^x = h$ is $x = -\left(\sum \alpha_i k_i\right)\left(\sum \beta_i k_i\right)^{-1} \pmod{h_\Delta}$, where $\vec{k} = (k_1, ..., k_n) \in Ker(A)$.

Finding Relations using SIQS

- SIQS used for factoring integers, and applies very naturally to working with BQF's.

- Vastly reduces the time to fill A by selecting elements which are likely to be smooth, replaces division with addition.

- Create a sieving interval $[-M, M]$ initialized to all 0's.

- Sieving polynomial: $f = \prod_{i=q}^{q+t-1} (p_i, b_i)^{e_i} = aX^2 + bXY + cY^2, e_i \in \{\pm 1\}$. Any form $(n, -2ax-b)$ where $n = f(x,1)$ is equivalent to f, so $(n, -2ax-b)f^{-1}$ is the identity element of the class group $Cl(\Delta)$.

SIQS (cont.)

- Find the roots of $f \pmod{p_i}, p_i \in FB$. If r_1 and r_2 are roots of $f \pmod{p}$, step along the interval adding $\lg p$ to all the spots $x \in [-M, M]$ in which $x \equiv r_1, r_2 \pmod{p}$.

- Pick out values close to $\lg f(x)$, and test $f(x)$ for $B-$smoothness.

- Switch polynomials by changing exponents of primes of f above. After using all 2^{t-1} possibilities, switch the primes themselves.

- When A is full, find smooth factorizations for g and h, and put these in A.

- If k_1 and k_2 are the last two elements of $\vec{k}$, representing g and h, then $x \equiv -k_1 k_2^{-1} \pmod{h_\Delta}$.

Linear Algebra

- Matrix properties
  - Sparse Matrix with density less than 2%.
  - Only store nonzero entries.
- Properties of desired algorithm
  - Finds solution $\vec{x}$ to $A\vec{x} = \vec{0}$.
  - Does not destroy sparse structure.
  - Faster than Gaussian Elimination.
- Lanczos algorithm with runtime $O(dn^2)$, where n is the number of columns and d is the density (entries per row).

Implementation

- Originally written in C++ using NTL
  - Used arbitrary-precision integers for calculations.
  - $\Delta = -pq^2 \approx -10^{50}$, $h_\Delta \approx 10^{25}$.
  - Fastest time to solution: 229 seconds on Intel Pentium 4 2.4Ghz.
- Rewritten using GMP in C
  - Optimized several key algorithmic steps.
  - Uses assembly language routines for critical arithmetic.
  - Fastest time to solution: 56 seconds on Intel Pentium 4 2.4Ghz.
  - Can be sped up by using large primes.

Constructing Elliptic Curves with Prescribed Number of Points

- Fix a prime p and $N \in \mathbb{Z}$ in the Hasse-Weil interval. Wish to construct a cryptographically strong elliptic curve $\#E(\mathbb{F}_p) = N$.

- A j-invariant of such a curve E, and hence E, can be obtained as a root of the Hilbert class polynomial $H_D(x)$ reduced mod p, where D is determined by p and N.

- Coefficients of $H_D(x)$ are huge and makes it very hard to compute directly. Compute $H_D(x)$ mod p_i, for enough "small" primes p_i, by searching $\mathbb{F}_{p_i}$ for the correct j-invariants, and lifting these polynomials to $H_D(x)$ via the Chinese Remainder Theorem.

(continues)

- Worked on two methods to optimize this algorithm.

- First, one can narrow down a subset of $\mathbb{F}_{p_i}$ where all the essential j-invariants must lie, hence reducing the search time.

- Secondly, lift to the Weber polynomial $W_D(x)$ mod p instead of $H_D(x)$ mod p. Any root of $W_D(x)$ mod p is sufficient for constructing our elliptic curve E

- The advantage is that the coefficients of $W_D(x)$ are significantly smaller then those of $H_D(x)$.

Analysis and Application of MOV Attack

- Key idea: ECDLP$\rightsquigarrow$DLP over F_{q^l}, where $l = ord_n(q)$ and should be large.

- Main tool: Weil pairing.

- Only effective for supersingular curves.

- Summary:

  – Idea of reduction, since the latter is easier.

  – Use Weil Pairing.

  – Do NOT use supersingular curves.

  – Can assume $End(E)$ is known, if over finite fields.

Tate-Pairing Implementations for Tripartite Key Agreement

Cryptology ePrint Archive 2003/053, 23 pages.

Iwan Duursma (UIUC) and Hyang-Sook Lee (Ewha Womans University, Korea)

- We give a closed formula for the Tate-pairing on the hyperelliptic curve $y^2 = x^p - x + d$ in characteristic p.

- This speeds up recent implementations by Barreto et.al. and by Galbraith et.al. for the special case $p = 3$ by a factor 2.

- As an application, we propose an n-round key agreement protocol for up to 3^n participants by extending Joux's pairing-based protocol to n rounds.

Future Work

- Random walks on elliptic curves (Pollard rho and Kangaroo).

- Index calculus methods on hyperelliptic curves (Enge-Gaudry, Sieving, etc.).

- Implementation, optimization, parallelization for a cluster of PC's in a combination of C, NTL and GMP.

- Other applications of pairings.

Activities in Cryptography

- Illinois Center for Cryptography and Information Protection (ICCIP): Interdisciplinary center involving CS, ECE, and Math. **http://www.iccip.csl.uiuc.edu/**

- **Information Protection Seminar**: Wednesdays, 4 pm.

- **Computational Number Theory Seminar**: Mondays, 3 pm.

- Computer cluster **antfarm** of 17 dual processor PCs. (Supported by NSF SCREMS)

- **Conferences**: Midwest Arithmetical Geometry in Cryptography (MAGC) 1999-2001, UIUC; AMS Special Session on Cryptography and Computational and Algorithmic Number Theory, Bloomington, IN April 2003.