A Study of Hyperelliptic-Curve Cryptography

Synopsis Seminar

Anindya Ganguly

Under the supervision of

Prof. Abhijit Das
Prof. Dipanwita Roy Chowdhury

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

December 24, 2020

>>> Outline

>>> Motivation

* Curve based cryptography takes a lot of attention from Crypto community
  * Elliptic curve cryptography proposed by Koblitz and Miller
  * Hyperelliptic curves cryptography proposed by Koblitz
* Hyperelliptic curves are less frequently studied than schemes based on RSA, DSA and ECDSA
* Lesser bit required to achieve the same security level as elliptic curve
* Arithmetic of hyperelliptic curve is less efficient than elliptic curve
* Although subfield curve admit faster Jacobian arithmetic
* Faster algorithm exist for large-genus curve
* For genus $g \leq 3$, no such subexponential algorithm exist

>>> Overview

* Generating a cryptographically suitable hyperelliptic curves is a major issue
* Subfield curves over $\mathbb{F}_q$ to be considered, $q = p^5$, p is a single-precision prime
* Choose a curve $\mathcal{C}$ over $\mathbb{F}_p$ and compute the order of $\mathbb{J}_p$ using Baby steps Giant steps method
* Using Newton-Girard formula derive the order of $\mathbb{J}_q$
* Implement the Jacobian arithmetic over $\mathbb{F}_q$
* Set the security levels 80, 96, 112, and 128 bits
* Comparative performance analysis is tabulated
* A variant of ElGamal encryption scheme is proposed
* Strong mathematical proof has been established for adopted scheme

>>> Cryptography

* Cryptography is a science that applies mathematics and logic to design strong encryption methods.
* Symbol replacement, the most basic form of cryptography, appears in ancient time.
* Thomas Jefferson's wheel cipher is the basis for American military cryptography until as late as the World War-II.
* In computer age, 128-bit mathematical encryption, far stronger than any ancient or medieval cipher.
* In 1970, Whitfield Diffie and Martin Hellman introduced the first Public Key Cryptography Standard(PKCS).
* In digital era, it helps to secure e-business, e-mail, smart card system, AADHAR, electronic voting machine.
* Five primary functions are privacy, authentication, integrity, non-repudiation, and key exchange.

>>> Public Key Cryptography

Modern cryptographic algorithms are designed around computational hardness assumptions.

- **Discrete logarithm problem (DLP)**
  Let $(G, \cdot)$ be an Abelian group. Given $a, b \in G$, find x (if it exists) such that $a^x = b$.
  e.g. DSA, ElGamal encryption, DH key exchange etc.
- **Integer factorization problem (IFP)**
  Let p and q be two large prime. It is infeasible to factorize $N = pq$ in polynomial time.
  e.g. RSA, Rabin Cryptosystem, BBS generator etc.

$\Rightarrow$ It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means.

$\Rightarrow$ These schemes are therefore termed *computationally secure*.

$\Rightarrow$ These problems are used as a *trapdoor one-way* function.

$\Rightarrow$ For DLP, Group G: fast group arithmetic, large order, cyclic, infeasible DLP

>>> Discrete Log Crypto

Discrete Logarithm Problem

Let $(G, \cdot)$ be an Abelian group. Given $a, b \in G$, find x (if it exists) such that $a^x = b$.

Groups must satisfy the following properties.

For practicality:

- Compact group elements
- Fast group operations

For security:

- Large order
- Cyclic or almost cyclic (some other restrictions on the order)
- Infeasible discrete logarithm problem (DLP)

>>> Proposed Groups

Difficulty of DLP depends on the group G.

* Very easy: Polynomial time algorithm exists
  e.g. $G = (\mathbb{Z}_n, +)$.
* Hard: Sub-exponential time algorithm exists
  e.g. $G = (\mathbb{F}_p, \cdot)$ proposed by Diffie-Hellman, 1976.
* Very hard: Exponential time algorithm exists
  e.g.
  * Elliptic curves over finite fields proposed by Koblitz 1985, Miller 1985.
  * Hyperelliptic curves over finite fields proposed by Koblitz 1989.

DLP on curve based cryptography

Given a group $G = < P >$ and some $Q \in G$, it is hard to determine the integer k such that $Q = [k]P$ (where P, Q are the points for elliptic curves and divisors for hyperelliptic curves with genus $g \geq 2$).

>>> Why Hyperelliptic Curves?

Advantages

* Lesser bit required to achieve same security
* Abelian group structure
* Field arithmetic cost: $O((\log q)^2)$ (over $\mathbb{F}_q$)
* Cryptographic protocols can be implemented based on the hardness of DLP

But

Limitations

* Implementation of the arithmetic isn't efficient as elliptic curves, takes $O(g^2)$ field operations
* Few hyperelliptic curves are used for cryptographic purpose

>>> Hyperelliptic Curves

* Let GF(q) be a finite field.
* $\mathcal{C} : y^2 = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ with $a_i \in \mathrm{GF}(q)$ is a hyperelliptic curve defined over GF(q).
* The Jacobian $\mathbb{J}_q$ is an Abelian group associated with $\mathcal{C}$.
* The elements of $\mathbb{J}_q$ has a unique representation (Mumford representation) as a reduced divisor (u, v).
* Let (x_1, y_1) and (x_2, y_2) be two points on $\mathcal{C}$. Then
    * $u(x) = (x - x_1)(x - x_2)$ and $v(x) = \left(\frac{x-x_2}{x_1-x_2}\right) y_1 + \left(\frac{x-x_1}{x_2-x_1}\right) y_2$.
* Divisor with single point (x_1, y_1) on $\mathcal{C}$
    * $u(x) = (x - x_1)$ and $v(x) = y_1$.
* The inverse of $(u(x), v(x))$ is $(u(x), -v(x))$.
* The additive identity is $(1, 0)$.
* The Jacobian arithmetic follows Cantor's addition algorithm.

>>> Optimized formulas for Jacobian Arithmetic

Algorithms	Addition	Doubling
Elliptic Curve Arithmetic	$I+2M+S$	$I+2M+2S$
Cantor's Algorithm	$2I+44M+4S$	$2I+42M+8S$
Harley's Formula	$2I+24M+3S$	$2I+24M+6S$
Matsuo's Improvement	$2I+22M+S$	$2I+23M+2S$
Lange's Explicit Version	$I+22M+3S$	$I+22M+5S$
Projective Coordinate	$47M+4S$	$38M+6S$
Weighted Coordinate	$47M+7S$	$34M+7S$
Costello and Lauter	$43M+4S$	$30M+9S$
Hisil and Costello	$41M+7S$	$28M+8S$

Table : Divisor-Class Addition Algorithms

[0] I: Inversion, M: Multiplication, S: Squaring

>>> Related Work

Curve-based cryptographic library

* Gaudry: $m_p\mathbb{F}_q$ library used for curve-based public key cryptography
* Pelzl: includes genus two and three HECC
* Avanzi: nuMONGO includes ECC and HECC

No implementation of subfield curve is reported

Existing hyperelliptic curves

* Furukawa: $y^2 = x^5 + ax$ and $y^2 = x^5 + a$ over prime field
* Satoh: $y^2 = x^5 + ax^3 + bx$ over $\mathbb{F}_p$, $|\mathbb{J}_p|$ has large prime divisor
* Buhler and Koblitz: $y^2 + y = x^n$ over $\mathbb{F}_p$, n is an odd prime with $n|(p-1)$

All curves are defined over large prime field.

>>> Our Curve

* Fix a prime field $\mathbb{F}_p$ and extension field $\mathbb{F}_q$. Start with the simple hyperelliptic curve :

$$y^2 = x^5 + x + a.$$

  Vary a to generate different curves.
* Set the security levels l to 80, 96, 112, and 128 bits
* The curves offer groups of prime orders of size 160, 192, 224, 256 bits
* Consider quintic extension
* Take a prime of size $l/4$
* Size of extension field is $5l/4$
* Order of $\mathbb{J}_p \approx p^2$, $\mathbb{J}_q \approx q^2$
* $n = |G| = |\mathbb{J}_q|/|\mathbb{J}_p|$
* If n is a prime then store the curve

>>> Example

l-size	p-size	q-size	$\|\mathbb{J}_p\|$-size	$\|\mathbb{J}_q\|$-size	$\|G\|$-size
80	20	100	40	200	160
96	24	120	48	240	192
112	28	140	56	280	224
128	32	160	64	320	256

Table : Relation between security level and group size (in bits)

>>> Why Quintic Extension?

* Best choice is to work over prime fields at the desired security level
* Point counting algorithms over large prime fields are difficult and inefficient

---

* Point counting is efficient for prime fields of size ≤ 32 bits
* Curves $\mathcal{C}$ defined over $\mathbb{F}_p$ are also defined over $\mathbb{F}_q$, $q = p^d$
* It is easy to derive $|\mathbb{J}_q|$ from $|\mathbb{J}_p|$
* $\mathbb{J}_p$ is a subgroup of $\mathbb{J}_q$, so $|\mathbb{J}_p|$ divides $|\mathbb{J}_q|$
* A curve is suitable if the cofactor $n = |\mathbb{J}_q|/|\mathbb{J}_p|$ is a prime
* d should be small and prime to avoid intermediate subgroups
* For $d = 5$, point counting is doable over $\mathbb{F}_p$
* Loss of efficiency: Theoretically no more than 50%

>>> Construction of quintic extension

Group size l	Prime p	Irreducible polynomial $f(x)$
20	1048571	$x^5 - 2$ or $x^5 + 2$
24	16777199	$x^5 + x - 3$ or $x^5 - 4x - 1$
28	268435399	$x^5 - x - 2$
32	4294836163	$x^5 + 2x - 1$

Table : Constructing a suitable extended fields

>>> The Algorithm at a Glance

* Choose a curve $\mathcal{C} : y^2 = x^5 + x + a$ over a medium-sized prime field $\mathbb{F}_p$
* Count $|\mathbb{J}_p|$ using the baby-step-giant-step method
* Exhaustively enumerate the number of rational points on $\mathcal{C}$ over $\mathbb{F}_p$
* Use the Newton-Girard formula to compute $|\mathbb{J}_q|$, $q = p^5$
* Compute $n = |\mathbb{J}_q|/|\mathbb{J}_p|$
* If n is not prime, repeat
* Implement $\mathbb{F}_q$ arithmetic
* Implement $\mathbb{J}_q$ arithmetic (in Mumford representation)
* Choose a random point $Q \in \mathbb{J}_q$ and compute $P = (|\mathbb{J}_q|/n)Q$
* If $P \neq \mathcal{O}$, it is a point of order n
* Use P as the base point for designing cryptosystems

>>> The Order-Finding Procedure

1. Set $w_l = \left\lceil (\sqrt{p}-1)^4 \right\rceil$, $w_h = \left\lfloor (\sqrt{p}+1)^4 \right\rfloor$, $W = w_h - w_l$, and $S = \left\lceil \sqrt{W} \right\rceil$.
2. Precompute $-jP$ for $j = 0, 1, 2, \ldots, S-1$, and store the pairs $(-jP, j)$ in a list L.
3. If some $j > 0$ is found such that $-jP = (1, 0)$, return j as the order of P.
4. Sort the list L with respect to $-jP$.
5. Compute $Q = w_l P$ and $SP = -[-(S-1)P + (-P)]$.
6. For $i = 0, 1, 2, \ldots, S-1$, repeat
   - 6.1 Search the list for Q using the binary search algorithm.
   - 6.2 If some entry (Q, j) is found in the list, store $k = w_l + iS + j$.
   - 6.3 Update $Q = Q + SP$.
7. If there is only one match k, then return this k as the order of P.
8. If there are multiple matches, return the difference between any two consecutive matches as the order of P.

>>> The Order-Lifting Procedure

- Zeta function of a curve
  $Z_{\mathcal{C}}(T) = 1 + N_1 T + \frac{1}{2}(N_1^2 + N_2)T^2 + \cdots$
- Alternative expression $Z_{\mathcal{C}}(T) = \frac{L(T)}{(1-T)(1-pT)}$
- L-function $L(T) = 1 + s_1 T + s_2 T^2 + s_1 p T^3 + p^2 T^4$
- $L(T)$ is related to Jacobian $L(1) = |\mathbb{J}_p|$, and $L(-1) = |\widetilde{\mathbb{J}_p}|$
- $Z_{\mathcal{C}}(T) = 1 + (p + s_1 + 1)T + (p^2 + s_2 + 1 + s_1 + s_1 p + p)T^2 + \cdots$
- $N_1 = p + s_1 + 1$, and $N_2 = p^2 - s_1^2 + 2s_2 + 1$
- $L^{(opp)}(T) = T^4 + s_1 T^3 + s_2 T^2 + s_3 T + s_4$, $,\alpha_i$ are roots
- Define $L_d(T) = (1 - \alpha_1^d T)(1 - \alpha_2^d T)(1 - \alpha_3^d T)(1 - \alpha_4^d T)$
- Connection between L-polynomials and the Jacobian orders:
  $|\mathbb{J}_{p^d}| = L_d(1) = (1 - \alpha_1^d)(1 - \alpha_2^d)(1 - \alpha_3^d)(1 - \alpha_4^d)$
- If we can compute $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ with sufficient precision, we readily obtain the Jacobian orders in extension fields.

>>> The Order-Lifting Procedure

* The elementary symmetric polynomials in four variables $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are
    * $e_0 = 1$,
    * $e_1 = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$,
    * $e_2 = \alpha_1\alpha_2 + \alpha_1\alpha_3 + \alpha_1\alpha_4 + \alpha_2\alpha_3 + \alpha_2\alpha_4 + \alpha_3\alpha_4$,
    * $e_3 = \alpha_1\alpha_2\alpha_3 + \alpha_1\alpha_2\alpha_4 + \alpha_1\alpha_3\alpha_4 + \alpha_2\alpha_3\alpha_4$,
    * $e_4 = \alpha_1\alpha_2\alpha_3\alpha_4$,
    * $e_k = 0$ for $k \geq 5$
* $L^{(opp)}(T) = T^4 + s_1T^3 + s_2T^2 + s_3T + s_4 = (T - \alpha_1)(T - \alpha_2)(T - \alpha_3)(T - \alpha_4)$
* $e_0 = 1, e_1 = -s_1, e_2 = s_2, e_3 = -s_3, e_4 = s_4, e_k = 0$ for $k \geq 5$
* Define $p_k = \alpha_1^k + \alpha_2^k + \alpha_3^k + \alpha_4^k$ for all $k \geq 1$
* By Newton--Girard formula $ke_k = \sum_{i=1}^{k} (-1)^{i-1} e_{k-i} p_i$

>>> The Order-Lifting Procedure

* We know e_k values so compute p_ks
  * $p_1 = e_1$,
  * $p_2 = e_1 p_1 - 2e_2$,
  * $p_3 = e_1 p_2 - e_2 p_1 + 3e_3$,
  * $p_4 = e_1 p_3 - e_2 p_2 + e_3 p_1 - 4e_4$,
  * $p_k = e_1 p_{k-1} - e_2 p_{k-2} + e_3 p_{k-3} - e_4 p_{k-4}$ for all $k \geq 5$
* Put $\beta_i = \alpha_i^d$; $L_d^{(opp)}(T) = (T - \beta_1)(T - \beta_2)(T - \beta_3)(T - \beta_4)$
* Power sum $P_k = \beta_1^k + \beta_2^k + \beta_3^k + \beta_4^k = \alpha_1^{dk} + \alpha_2^{dk} + \alpha_3^{dk} + \alpha_4^{dk} = p_{dk}$
* Using N-G formula compute E_i's
* $L_d(T) = E_0 - E_1 T + E_2 T^2 - E_3 T^3 + E_4 T^4$
* $|\mathbb{J}_{p^d}| = L_d(1) = E_0 - E_1 + E_2 - E_3 + E_4$

>>> The Order-Lifting Procedure

1. Compute $|\mathbb{J}_p|$ and the count N_1 of rational points over $\mathbb{F}_p$.
2. Compute $s_1 = N_1 - p - 1$ and $s_2 = |\mathbb{J}_p| - 1 - s_1 - s_1 p - p^2$.
3. Take $e_0 = 1$, $e_1 = -s_1$, $e_2 = s_2$, $e_3 = -s_1 p$, and $e_4 = p^2$.
4. Compute $p_1 = e_1$, $p_2 = e_1 p_1 - 2e_2$, $p_3 = e_1 p_2 - e_2 p_1 + 3e_3$,
   $p_4 = e_1 p_3 - e_2 p_2 + e_3 p_1 - 4e_4$, and
   $p_k = e_1 p_{k-1} - e_2 p_{k-2} + e_3 p_{k-3} - e_4 p_{k-4}$ for $5 \leq k \leq 20$.
5. Take $P_i = p_{5i}$ for $i = 1, 2, 3, 4$.
6. Compute $E_0 = 1$, $E_1 = P_1$, $E_2 = \frac{1}{2}(E_1 P_1 - P_2)$,
   $E_3 = \frac{1}{3}(E_2 P_1 - E_1 P_2 + P_3)$, and $E_4 = \frac{1}{4}(E_3 P_1 - E_2 P_2 + E_1 P_3 - P_4)$.
7. Then, $|\mathbb{J}_q| = E_0 - E_1 + E_2 - E_3 + E_4$.
8. Compute the cofactor $n = |\mathbb{J}_q| / |\mathbb{J}_p|$.
9. If n is prime, store the curve.

>>> Successful Attempt

- $\mathcal{C}_1 : y^2 = x^5 + x + 47$
- $|\mathbb{J}_p| = 1099928953312 = 2^{40} + 417325536$
- Count of rational points on $\mathcal{C}_1$ over $\mathbb{F}_p$ is 1048979
- This gives

$$
\begin{aligned}
& |\mathbb{J}_q| \\
= \;& 1606861421126112580388908685296656425664857224973157020278432 \\
= \;& 2^{200} - 766231328776951530534070445061768573457688096358150229440
\end{aligned}
$$

- The cofactor

$$
\begin{aligned}
n &= |\mathbb{J}_q|/|\mathbb{J}_p| \\
&= 1460877465119621059080883122151454896336021166011 \\
&= 2^{160} - 624172211281859122801710564828123319911376965
\end{aligned}
$$

  is prime

>>> An Unsuccessful Attempt

* $\mathcal{C}_2 : y^2 = x^5 + x + 46$
* $|\mathbb{J}_p| = 1097744558000 = 2^{40} - 1767069776$
* Count of rational points on $\mathcal{C}_2$ over $\mathbb{F}_p$ is 1046895
* This gives:

$$\begin{aligned} & |\mathbb{J}_q| \\ = \; & 1606861421126118518527811084904153739543257852153511445450000 \\ = \; & 2^{200} - 76623132871757014151007437008862978945141629281389851376 \end{aligned}$$

* The cofactor

$$\begin{aligned} n &= |\mathbb{J}_q|/|\mathbb{J}_p| \\ &= 1463784456425534398803014685411133451998636874275 \\ &= 2^{160} + 2282819094631480599329852694850432342704331299 \end{aligned}$$

is not prime

>>> Another ‘‘Successful’’ Attempt

* $\mathcal{C}_3 : y^2 = x^5 + x + 60$
* $|\mathbb{J}_p| = 1098401972048 = 2^{40} - 1109655728$
* Count of rational points on $\mathcal{C}_3$ over $\mathbb{F}_p$ is 1047522
* This gives

$$\begin{aligned} & |\mathbb{J}_q| \\ = & \ 1606861421126117326279311266898329713697223055120690303050128 \\ = & \ 2^{200} - 766231328729492626508254428328882497993866210253225124 8 \end{aligned}$$

* The cofactor

$$\begin{aligned} n &= |\mathbb{J}_q|/|\mathbb{J}_p| \\ &= 1462908354152060576672027642006156546558828957461 \\ &= 2^{160} + 1406716821157658468342809289873526902896414485 \end{aligned}$$

is prime but larger than 2^{160}

>>> Some Good Curves

We take curves $y^2 = x^5 + x + a$ with $1 \leq a \leq 1000$.

* $p_{20} = 2^{20} - 5$
  $a = 47, 52, 125, 135, 343, 360, 385, 436, 488, 523, 673, 718, 755, 769, 925$

* $p_{24} = 16777199 = 2^{24} - 17$
  $a = 182, 268, 497, 577, 742, 805, 966$

* $p_{28} = 268435399 = 2^{28} - 57$
  $a = 10, 167, 170, 194, 303, 331, 368, 421, 502, 622, 623, 668, 837, 844, 902, 911, 992$

* $p_{32} = 4294836163 = 2^{32} - 2^{17} - 61$
  $a = 23, 43, 64, 67, 144, 155, 212, 269, 363, 412, 417, 503, 620$

>>> Discrete Logarithm Problem

Generic Square Roots Attack

* Pollard Rho, Lambda, Pohlig-Hellman are example of such attacks
* Possess a complexity of $O(\sqrt{|G|})$
* For 128 bit security we choose $|G| \approx 256$

Transfer Discrete log to $\mathbb{F}_q$ vector space

* $\mathbb{J}_q$ be the Jacobian of a genus g hyperelliptic curve over $\mathbb{F}_{p^d}$ with $p \mid |\mathbb{J}_q|$
* There exist a morphism from $\mathbb{J}_q$ to the $\mathbb{F}_q$ vector space of holomorphic differentials of the curve.
* This vector space is isomorphic to $\mathbb{F}_q^{2g-1}$.
* Time complexity is $O((2g-1)\log q^k)$ for small constant k.
* For our family $p \mid |\mathbb{J}_q|$ does not hold

>>> Discrete Logarithm Problem

Transfer DL via Weil descent technique

* It reduces DLP from $E_{\mathbb{F}_{p^d}}$ to $\mathbb{J}_p$ of curve C_p.
* Gaudry, Hess and Smart develop Weil descent method for elliptic curves over $\mathbb{F}_{2^d}$
* Galbraith generalizes this to hyperelliptic curves over even binary extension fields
* Diem studies elliptic and hyperelliptic curves over finite extension fields of odd characteristics
* He shows that for $d = 5$, there exist potentially vulnerable elliptic curves
* Not for our family of hyperelliptic curve
* Hess generalizes this attack to arbitrary Artin-Schreier extensions
* Concentrates only on small prime $p = 2, 3$

>>> Discrete Logarithm Problem

Cover Decomposition Attack

* Gaudry invented for elliptic curves
* Nagao generalizes to hyperelliptic curves over extension fields
* Time complexity is $O(q^{2-\frac{2}{dg}})$, $d:$ degree of the extension
* Joux and Vitse proposed this attack for elliptic curve over $\mathbb{F}_{p^6}$

Quantum Attack

* Proos shows that Shor's algorithm can solve ECDLP with $O(l)$ qubits and $O(l^3)$ Toffoli gates
* Huang extends this algorithm for HECDLP
* Replacing prime field arithmetic to extension field arithmetic makes our curve is vulnerable against quantum attacks.

Software Implementation

* Arithmetic of multiple-precision integers.
* Arithmetic of prime files $\mathbb{F}_p$ ($|p| \leq 32$).
* Polynomial arithmetic over $\mathbb{F}_p$.
* Arithmetic of extension fields $\mathbb{F}_q = \mathbb{F}_{p^5}$.
* Polynomial arithmetic over $\mathbb{F}_q$.
* Jacobian arithmetic over $\mathbb{F}_q$.

>>> Curve Parameters

System Parameters

* Compiler: GNU C compiler (gcc) version 5.5.0
* System: Linux environment on an intel core $i-7$ 3.10 GHz
* Other Library: NTL-11.3.2, GNU multiple precision library (GMP)

Elliptic Curve:

Curve P-256

- Prime $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ of size 256 bits
- Curve $\mathcal{E} : y^2 \equiv x^3 - 3x + b \pmod p$, where $b = 2455155546008943817740293915197451784769108058161191238065$
- Group order: $n = 115792089210356248762697446949407573529996955224135760342422259061068512044369$

>>> Curve Parameters

Hyperelliptic Curve:

Generic-1271

- Prime $p = 2^{127} - 1$ of size 128 bits
- Curve $\mathcal{C}_1 : y^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0 \pmod{p}$, where

  $$\begin{aligned} f_3 &= 34744234758245218589390329770704207149, \\ f_2 &= 132713617209345335075125059444256188021, \\ f_1 &= 90907655901711006083734360528442376758, \\ f_0 &= 6667986622173728337823560857179992816. \end{aligned}$$

- Group order:

  $n = 28948022309329048848169239995659025138451177$
  $9730915513741014757328925803322259$

>>> Curve Parameters

Subfield Curve

- Base prime $p = 4294836163$ of size 32 bits
- Monic irreducible polynomial $f(x) = x^5 + 2x - 1$ over $\mathbb{F}_p$
- Curve $\mathcal{C} : y^2 = x^5 + x + a$, where $a \in \mathbb{F}_p$.
  As a sample, we take $a = 23$.
- Group order:
  $n = 115764326143276219301046410958790255794 5749$
  $684746506164802945703526927706 26891$

>>> Performance Analysis I

Curve (Library)	Doubling	Addition	Scalar Mul
P-256 (NTL)	0.000003	0.000003	0.001375
Generic-1271 (Our work)	0.000191	0.000201	0.038537
Generic-1271 (NTL)	0.000020	0.000022	0.007514
Generic-1271 (GMP)	0.000054	0.000058	0.033367
Subfield curve $\mathcal{C}$ (Our work)	0.000034	0.000038	0.011614
Subfield curve $\mathcal{C}$ (NTL)	0.000100	0.000102	0.034476

Table : Comparison of Cantor's algorithm with elliptic-curve arithmetic

[0] All times are in milliseconds

>>> Performance Analysis II

Coordinate	Curve (Library)	Doubling	Addition	Scalar Mul
Affine	Generic-1271 (NTL)	0.000007	0.000009	0.002439
Affine	$\mathcal{C}$ (Our work)	0.000009	0.0000010	0.003021
Affine	$\mathcal{C}$ (NTL)	0.000028	0.000026	0.008442
Projective	Generic-1271 (NTL)	0.000007	0.000007	0.002466
Projective	$\mathcal{C}$ (Our work)	0.000011	0.000012	0.003167
Projective	$\mathcal{C}$ (NTL)	0.000026	0.000028	0.008604
Weighted	Generic-1271 (NTL)	0.000007	0.000009	0.002576
Weighted	$\mathcal{C}$ (Our work)	0.000008	0.000012	0.002944
Weighted	$\mathcal{C}$ (NTL)	0.000025	0.000031	0.008507

Table : Comparison with different coordinates

[0] $\mathcal{C}: \ y^2 = x^5 + x + a$ is the subfield hyperelliptic curve

>>> ElGamal Encryption

* Taher ElGamal proposed the scheme in 1985
* ElGamal scheme raises an issue, a mapping is required to map a message to a group element
* Virat proposes a new apporach
* In 2006, Mames, Paillier and Pointcheval proposed an encoding free ElGamal
* Joye and Libert modifies and proposes an encoding free ElGamal encryption using elliptic curves
* Fouque, Joux and Tibouchi proposed an injective encoding for elliptic curves.
* Fouque and Tibouchi proposed a nearly bijection encoding map
* Tsiounis and Yung give a IND-CPA proof for the security of ElGamal encryption
* Lipmaa shows that ElGamal encryption is IND-CCA1 secure based on some non standard assumption
* Wu and Stinson also show that ElGamal encryption OW-CCA1 secure under DT-DLA

\# >>> Encoding based ElGamal Encryption Scheme

Key Generation

* Choose $x \in_U [1, n-1]$
* Compute $Y = xP \in G$, P is a base point of G
  x is private key and Y is public key

Encoding Scheme

* Break $m \in \{0,1\}^l$ into two $\frac{l}{2}$-bit chunks: $m = m_0 \,||\, m_1$.
* For each $b \in \{0,1\}$, pad m_b as $x_b = b \,||\, m_b \,||\, r_b$ with $r_b \in_U \{0,1\}^{l'}$.
* Repeat until $x_b^5 + x_b + a$ is a square in $\mathbb{F}_q$.
* Let y_b be a square root of $x_b^5 + x_b + a$ in $\mathbb{F}_q$.
* Take the divisor $(u_2, u_1, u_0, v_1, v_0)$ with the two rational points (x_0, y_0) and (x_1, y_1) as M.
* M is a divisor from $\mathbb{J}_q$.

>>> Encoding based ElGamal Encryption Scheme

Encryption

* Generate $k \in_U \mathbb{Z}_n$ and set $R = kP \in G$
* Compute $S = M + kY \in \mathbb{J}_q$
* Send (R, S) to the recipient

Decryption

* Recover $M = S - xR \in \mathbb{J}_q$

Decoding Scheme

* Form the equations $x_0 + x_1 = -u_1$, and $x_0 x_1 = u_0$.
* Solve these equations (quadratic) to obtain x_0, x_1. Notice that x_b has msb b.
* Recover m_0, m_1 from x_0, x_1 after removing the padding.
* Output $m = m_0 \,||\, m_1$.

>>> Issue

* Jacobian $\mathbb{J}_q$ is the internal direct sum of G with the Jacobian $\mathbb{J}_p$ over the ground field.
* Every divisor D can be split as $D = D_G \oplus D_p$, where $D_G \in G$, $D_p \in \mathbb{J}_p$.
* $D_G = (\epsilon^{-1} \pmod n)(\epsilon D)$, $D_p = (n^{-1} \pmod \epsilon)(nD)$
* Similarly, encoded message $M = M_G \oplus M_p$.
* Eavesdropper can compute $nS = n(M_G \oplus M_p) + nkY = nM_p$
* Random padding strings r_0, r_1 destroy all correlations between m and M_p.
* M_p is fully independent from any other variable like private key x.
* Is this intuitive reason enough for formal security proof?

---

[0] $\epsilon = |\mathbb{J}_p|$, $n = |G|$

>>> Desirable Properties

1. Map is efficiently computable in polynomial time. The inverse of the map is also efficiently computable.
2. It can be applied for all forms of subfield hyperelliptic curves.
3. It is a probabilistic map due to the concatenated pseudorandom bits.
4. It does not preserve arithmetic operation. Let $D_1 = \theta(k_1)$ and $D_2 = \theta(k_2)$. Then, any correlation between k_1 and k_2 does not reflect on D_1 and D_2.
5. Map is well-distributed.

>>> Two theorems

Theorem 1 Let χ be any character of the Abelian group GF(q). The character sum is defined as

$$T(\chi) = \sum_{u \in \mathbb{F}_q} \chi(\theta(u)).$$

Then, for a non trivial character, we have $T(\chi) \leq 2\sqrt{q} + 11$.

Theorem 2 For large enough q, the expected number of iterations in θ on any input message m is less than three.

The image of the encoding map covers almost all reduced divisors of $\mathbb{J}_q$.

>>> Well-distributed

* Character: A multiplicative mapping of the group G into the multiplicative group of all the roots of unity
* θ is B-well distributed if $|T(\chi)| \leq B\sqrt(q)$
* Applying Riemann's hypothesis for the L-function $\left|\sum_{P \in \mathcal{C}_q} \chi(P)\right| \leq 2\sqrt{q}$
* Now, $\sum_{u \in \mathbb{F}_q} \chi(\theta(u)) = |R|\chi((0,0)) + \sum_{P \in \mathcal{C}_p - Wei} \chi(P)$ $= |R|\chi((0,0)) - \sum_{P \in Wei} \chi(P) + \sum_{P \in \mathcal{C}_q} \chi(P)$
* R denotes the set of all zeros of the curve polynomial $f(x)$
* So, we have θ is $(2 + \frac{11}{\sqrt{q}})$ well-distributed
* $\left|\frac{N(D)}{q^2} - \frac{1}{|\mathbb{J}_q|}\right| < (2\sqrt{q} + 11)^2$, where $N(D)$ be the number of preimages of D under θ.
* $\sum_{D \in \mathbb{J}_q} \left|\frac{N(D)}{q^2} - \frac{1}{|\mathbb{J}_q|}\right| \leq \left(2 + \frac{11}{\sqrt{q}}\right)^2$
* The bound on the statistical distance is $\frac{c}{\sqrt{q}} + O(\frac{1}{q})$

>>> Organization of the Thesis

Chapter 1 Introduction

Chapter 2 Mathematical Background

* Briefly introduced notation and terminology

Chapter 3 Jacobian Arithmetic

* Order Computation
* Arithmetic of Divisors
* Discrete Logarithm Problem
* Mathematical Library
* Performance Analysis

Chapter 4 Cryptographic Primitives

* Proposed variant of ElGamal Encryption
* Security Analysis

Chapter 5 Conclusion and Future Scope

Appendix A A list of several cryptographically suitable hyperelliptic curves.

>>> Conclusion and Future Plans

Conclusion

* Narrow the gap between the performances of EC and HEC
* Proposed family of curves are as efficient and practical
* All existing algorithms for solving the discrete logarithm problem have been found to be inefficient
* Designed an encoding scheme
* Security analysis has been established

Future Scope

* Enhance the performance
  1. more efficient point-counting algorithms,
  2. optimized quintic extension fields,
  3. dedicated addition and scalar multiplication formulas
* CCA for ElGamal encryption
* Post-quantum cryptography

>>> Conference

* Anindya Ganguly, Abhijit Das, Dipanwita Roy Chowdhury, and Deval Mehta, *A Family of Subfield Hyperelliptic Curve for Use in Cryptography*, 22^{nd} International Conference on Information and Communications Security (ICICS 2020), Copenhagen, Denmark, 2020.

Thank You

Any Suggestions & Questions

>>> References—1

- G. LOCKE and P. GALLAGHER, *Fips pub 186-3: Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication, 3 (2009), pp. 186-3.
- T. LANGE , *Inversion-free arithmetic on genus 2 hyperelliptic curves*, 2002, p. 147.
- T. LANGE, *Weighted coordinates on genus 2 hyperelliptic curves*
- T. LANGE, *Efficient arithmetic on genus 2 hyperelliptic curves over finite fields via explicit formulae*, IACR Cryptology ePrint Archive, 121 (2002).
- T. LANGE, *Efficient arithmetic on hyperelliptic curves*, IEM, 2002.
- T. GRANLUND, *The GNU multiple precision arithmetic library*, https://gmplib.org/, (1996).
- Avanzi, Roberto Maria. *Aspects of hyperelliptic curves over large prime fields in software implementations.* International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2004.
- N. KOBLITZ, *Hyperelliptic cryptosystems*, Journal of Cryptology, 1 (1989), pp. 139150.

>>> References—2

- Bos, Joppe W., et al. *Fast cryptography in genus 2.* Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2013.
- Buhler, Joe, and Neal Koblitz. *Lattice basis reduction, Jacobi sums and hyperelliptic cryptosystems.* Bulletin of the Australian Mathematical Society 58.1 (1998): 147-154.
- Diffie, Whitfield, and Martin Hellman. *New directions in cryptography.* IEEE transactions on Information Theory 22.6 (1976): 644-654.
- Farashahi, Reza R., et al. *Indifferentiable deterministic hashing to elliptic and hyperelliptic curves.* Mathematics of Computation 82.281 (2013): 491-512.
- Fouque, Pierre-Alain, Antoine Joux, and Mehdi Tibouchi. *Injective encodings to elliptic curves.* Australasian Conference on Information Security and Privacy. Springer, Berlin, Heidelberg, 2013.
- Fouque, Pierre-Alain, and Mehdi Tibouchi. *Deterministic encoding and hashing to odd hyperelliptic curves.* International Conference on Pairing-Based Cryptography. Springer, Berlin, Heidelberg, 2010.
- M. S. VICTOR, Use of elliptic curves in cryptography, in CRYPTO, Springer, 1986, pp. 417426.

>>> References—3

E. FURUKAWA, M. KAWAZOE, AND T. TAKAHASHI, *Counting points for hyperelliptic curves of type y 2= x 5+ ax over finite prime fields*, in International Workshop on Selected Areas in Cryptography, Springer, 2003, pp. 2641.

Gaudry, Pierrick, and Emmanuel Thomé. *The mpFq library and implementing curve-based key exchanges.* 2007.

Pelzl, Jan, et al. *Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves.* International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2003.

Rivest, Ronald L., Adi Shamir, and Leonard M. Adleman. *Cryptographic communications system and method.* U.S. Patent No. 4,405,829. 20 Sep. 1983.

T. SATOH, *Generating genus two hyperelliptic curves over large characteristic finite fields*, in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2009, pp. 536553.

V. SHOUP, *NTL: A library for doing number theory*, http://www. shoup. net/ntl/, (2001)