Efficient Implementation of Bilinear Pairings on ARM Processors

Reza Azarderakhsh

Centre for Applied Cryptographic Research
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, ON, N2L 3G1, Canada
razarder@math.uwaterloo.ca

Joint work with: Gurleen Grewal (U Waterloo), Patrick Longa (Microsoft Research), Shi Hu (Stanford University), and David Jao (U Waterloo)

SAC 2012
Windsor, ON, Canada

Outline

Introduction

- Pairing-based cryptography (PBC) is the use of a pairing between elements of two cryptographic groups to a third group.
- PBC relies on finite fields and is a function of two points on the elliptic curve groups to construct cryptographic systems.

D. Boneh and M. Franklin (2001)

- Applications of PBC:
  - Identity-based encryption (Boneh-Franklin)
  - Short signatures (Boneh-Lynn-Shacham)
  - Identity-based key agreement
  - Certificateless encryption
  - Blind signature
  - ...

Speed Records To Date

- Recent High Performance Implementation on Ordinary Curves on PC: Less than a millisecond for 128-bit security level [1]. (U Waterloo)

[1] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, Julio López: *Faster Explicit Formulas for Computing Pairings over Ordinary Curves*. EUROCRYPT 2011: 48-68.

- Recent Resource-Constrained Implementation on Ordinary Curves on a ARM Processor: 55 milliseconds for 128-bit security level [2]. (Microsoft Research)

[2] Tolga Acar, Kristin Lauter, Michael Naehrig, and Daniel Shumow, "*Affine Pairings on ARM*", Pairing-Based Cryptography – Pairing 2012, Lecture Notes in Computer Science, Springer-Verlag (2012), to appear.

Elliptic Curves and Pairings

- (Hyper)Elliptic curves: A group of points (group law)
  - Only known mathematical setting where desirable pairings exist.
  - Attacks on elliptic curves are slower in comparison to the one on finite fields (for a fixed field size).
- Pairings: are a function of two points in the elliptic curves
  - Very few elliptic curves admit a usable pairing.

Pairings

Definition

$e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$

- $\mathbb{G}_1$ is a subgroup of $E(\mathbb{F}_q)$ (additively)
- $\mathbb{G}_2$ is a subgroup of $E(\mathbb{F}_{q^k})$ (additively)
- $\mathbb{G}_T$ is the multiplicative subgroup of a finite field $\mathbb{F}_{q^k}$
  - $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, then $e(P,Q) \in \mathbb{G}_T$.
  - $\#E(\mathbb{F}_q) = q+1-t \approx q$ and $\#E(\mathbb{F}_q) = nh$ (h small and n is a big prime).
  - $k > 1$ is the smallest integer, $n | q^k - 1$ and called the embedding degree.
- Symmetric pairing: η_T pairing (supersingular elliptic curves).
- Asymmetric pairing: ate pairing, R-ate pairing, and optimal-ate pairing (general elliptic curves).
- Finding pairing friendly curves is important.

Barreto-Naehring Curves

- Barreto and Naehrig describe a family of pairing friendly curves called BN curves.
- BN curves $E: y^2 = x^3 + b$ ($b \neq 0$, $\#E = n$ and embedding degree $k = 12$)
- Defined over a prime field $\mathbb{F}_q$ where q and n are given by the polynomials:

$$\begin{aligned} q &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \\ n &= 36x^4 + 36x^3 + 18x^2 + 6x + 1, \end{aligned}$$

for some integer x such that both q and n are prime and $b \in \mathbb{F}_q^*$ such that $b+1$ is a quadratic residue.

- Short signature applications (ideally) require pairing friendly curves of $k > 6$ at high security levels, e.g., $k \geq 6$ for 80-bit, $k \geq 12$ for 128-bit, and $k \geq 18$ for 192-bit security level.

Optimal Ate Pairing

- $\mathbb{G}_1 = E[n] \cap \ker(\Pi_q - [1])$ (the base field)
- $\mathbb{G}_2 = E[n] \cap \ker(\Pi_q - [q])$ (the extension field)
- Then, points in $\mathbb{G}_1$ have coordinates in $\mathbb{F}_q$, and points in $\mathbb{G}_2$ have coordinates in $\mathbb{F}_{q^k}$, e.g., $\mathbb{F}_{q^{12}}$.
- The Optimal Ate (O-Ate) pairing is defined by

$$a_{\text{opt}} : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_n, \;\; (Q, P) \to f_{6x+2,Q}(P) \cdot h(P)$$

where $h(P) = l_{[6x+2]Q,qQ}(P) l_{[6x+2]Q+qQ,-q^2Q}(P)$, $l_{Q_1,Q_2}(P)$ is the line arising in the addition of Q_1 and Q_2 at point P. Also, $f_{6x+2,Q}(P)$ is the Miller function.

- Computation of Miller function (Miller's Algorithm)
- Computation of the Final Exponentiation ($f^{\frac{q^k-1}{n}}$)
- Arithmetic over BN, $\mathbb{F}_q$, $\mathbb{F}_{q^2}$, and $\mathbb{F}_{q^{12}}$.

Representation of Extension Fields (1)

- Computation of the Miller function and final exponentiation involves arithmetic over $\mathbb{F}_{q^2}$ and $\mathbb{F}_{q^{12}}$ respectively.
- Efficient implementation of the underlying extension fields is crucial to achieve fast pairing results.
- The IEEE P1363.3 standard recommends using towers to represent $\mathbb{F}_{q^k}$ which is employed in the previous works as

Fact

For approximately 2/3rds of the BN-primes $q \equiv 3 \bmod 8$, *the polynomial* $y^6 - \alpha$, $\alpha = 1 + \sqrt{-1}$ *is irreducible over* $\mathbb{F}_{q^2} = \mathbb{F}_q(\sqrt{-1})$.

- This gives the following towering scheme:

$$\begin{cases} \mathbb{F}_{q^2} = \mathbb{F}_q[i]/(i^2 - \beta), & \text{where } \beta = -1. \\ \mathbb{F}_{q^6} = \mathbb{F}_{q^2}[v]/(v^3 - \xi), & \text{where } \xi = 1 + i. \\ \mathbb{F}_{q^{12}} = \mathbb{F}_{q^6}[w]/(w^2 - v). \end{cases}$$

Representation of Extension Fields (2)

- In practice, desirable BN-curves are rare, and it is sometimes necessary to use primes $q \equiv 7 \bmod 8$ (e.g., BN-446 and BN-638) in order to optimize other aspects such as the Hamming weight of x. Then, we propose that

Fact

For approximately 2/3rds of the BN-primes $q \equiv 7 \bmod 8$, the polynomial $y^6 - \alpha$, $\alpha = 1 + \sqrt{-2}$ is irreducible over $\mathbb{F}_{q^2} = \mathbb{F}_q(\sqrt{-2})$.

- This gives the following towering scheme:

$$\begin{cases} \mathbb{F}_{q^2} = \mathbb{F}_q[i]/(i^2 - \beta), & \text{where } \beta = -2. \\ \mathbb{F}_{q^6} = \mathbb{F}_{q^2}[v]/(v^3 - \xi), & \text{where } \xi = 1 + i. \\ \mathbb{F}_{q^{12}} = \mathbb{F}_{q^6}[w]/(w^2 - v). \end{cases}$$

Finite Field Operations and Lazy Reduction

- Lazy reduction reduces the number of reductions in the multiplication over extension fields.
- In [1] a Lazy reduction scheme for computation of pairing in tower fields and curve arithmetic (projective) is proposed.
- We extend lazy reduction to field inversion and curve arithmetic over affine coordinates.
  - We saved one $\mathbb{F}_q$-reduction in $\mathbb{F}_{q^2}$-inversion, and 36 $\mathbb{F}_q$-reductions in $\mathbb{F}_{q^{12}}$-inversion.
- The line function in the Miller loop evaluates to a sparse $\mathbb{F}_{q^{12}}$ element containing only three of the six basis elements over $\mathbb{F}_{q^2}$. Thus, when multiplying the line function output with $f_{i,Q}(P)$, the sparseness property is utilized to avoid full $\mathbb{F}_{q^{12}}$ arithmetic.
  - This requires 13 fewer additions over $\mathbb{F}_{q^2}$ compared to the one used in [1].

[1] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, Julio López: *Faster Explicit Formulas for Computing Pairings over Ordinary Curves*. EUROCRYPT 2011: 48-68.

Mapping from the Twisted Curve to the Original Curve

- BN curves admit a sextic twist.
- Previously, the M-type sextic twist has been ignored for pairing computations, most likely due to the inefficient untwisting map.
- In a D-type twist, the pairing is computed on points on the original curve E. The untwisting map is almost free.
- In an M-type twist, we compute the pairing on points on the twisted curve E'. The inverse map is almost free.
- We demonstrate that by computing the pairing on the twisted curve, we can bypass the inefficient untwisting.
- This allows the pairing to be computed optimally regardless of the type of twist involved.
- Our work indicates that D-type and M-type twists achieve equivalent performance for point/line evaluation computation.

Final Exponentiation (1)

- Final Exponentiation is the computation of $f^{\frac{q^k-1}{n}}$.
- First $\frac{q^{12}-1}{n}$ is factored into q^6-1, q^2+1, and $\frac{q^4-q^2+1}{n}$.
  - q^6-1 requires conjugation and inversion, q^2+1 requires q^2-power Frobenius and a multiplication (simple part).
  - $\frac{q^4-q^2+1}{n}$ is performed in the cyclotomic subgroup (hard part).
- We then perform the following exponentiations as

$$f \mapsto f^x \mapsto f^{2x} \mapsto f^{4x} \mapsto f^{6x} \mapsto f^{6x^2} \mapsto f^{12x^2} \mapsto f^{12x^3}$$

  - Which requires 3 exponentiations by x, 3 squarings and 1 multiplication in $\mathbb{F}_{q^{12}}$.

Final Exponentiation (2)

- We then compute the terms

$$a = f^{12x^3} f^{6x^2} f^{6x}$$
$$b = a(f^{2x})^{-1}$$

which require 3 multiplications. f^{2x} is in the cyclotomic subgroup, so the inversion required for computing b is just a conjugation.

- The final pairing value is obtained as

$$af^{6x^2} fb^p a^{p^2} (bf^{-1})^{p^3}$$

which costs 6 multiplications and 6 Frobenius operations.

- In total, this method requires 3 exponentiations by x, 3 squarings, 10 multiplications, and 3 Frobenius operations.
- Note that [1] requires 3 additional multiplications and an additional squaring, which is slower.

[1] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, Julio López: *Faster Explicit Formulas for Computing Pairings over Ordinary Curves*. EUROCRYPT 2011: 48-68.

Curve Arithmetic

- We optimize the number of addition operations required for point doubling, addition, and line function computation in affine and homogeneous projective coordinates.

Coordinates		Cost of PA	Cost of PD
Affine		$\tilde{i}+3\tilde{m}+2\tilde{s}+8\tilde{a}+2m$	$\tilde{i}+3\tilde{m}+1\tilde{s}+7\tilde{a}+2m$
Projective	Jacobian	$6\tilde{m}+4\tilde{s}+13\tilde{a}+4m$	$10\tilde{m}+3\tilde{s}+6\tilde{a}+4m$
	Homogeneous	$2\tilde{m}+7\tilde{s}+22\tilde{a}+4m$	$11\tilde{m}+2\tilde{s}+8\tilde{a}+4m$

- For PD, homogeneous coordinates are faster.
- For PA, Jacobian coordinates are slightly faster (not sufficient).
- With cheap inversion, affine coordinates will be faster.
  - Also leads to a faster dense-sparse multiplication algorithm.

Operation Counts

- Cost of the computation of O-Ate pairings using various coordinates

Curve	Operation	Cost
BN-254	Proj. ML	$1841\tilde{m}_u + 457\tilde{s}_u + 1371\tilde{r} + 9516\tilde{a} + 284m + 3a$
	Affine ML	$70\tilde{i} + 1658\tilde{m}_u + 134\tilde{s}_u + 942\tilde{r} + 8292\tilde{a} + 540m + 132a$
	FE	$386\tilde{m}_u + 1164\tilde{s}_u + 943\tilde{r} + 4\tilde{i} + 7989\tilde{a} + 30m + 15a$
BN-446	Proj. ML	$3151\tilde{m}_u + 793\tilde{s}_u + 2345\tilde{r} + 18595\tilde{a} + 472m + 117a$
	Affine ML	$118\tilde{i} + 2872\tilde{m}_u + 230\tilde{s}_u + 1610\tilde{r} + 15612\tilde{a} + 920m + 230a$
	FE	$386\tilde{m}_u + 2034\tilde{s}_u + 1519\tilde{r} + 4\tilde{i} + 13374\tilde{a} + 30m + 345a$
BN-638	Proj. ML	$4548\tilde{m}_u + 1140\tilde{s}_u + 3557\tilde{r} + 27198\tilde{a} + 676m + 166a$
	Affine ML	$169\tilde{i} + 4143\tilde{m}_u + 330\tilde{s}_u + 2324\tilde{r} + 22574\tilde{a} + 1340m + 333a$
	FE	$436\tilde{m}_u + 2880\tilde{s}_u + 2143\tilde{r} + 4\tilde{i} + 18528\tilde{a} + 30m + 489a$

ARM Platforms

- A Marvell Kirkwood 6281 ARMv5 CPU processor (Feroceon 88FR131) @ 1.2 GHz.

- An iPad 2 (Apple A5) using an ARMv7 Cortex-A9 MPCore processor @ 1.0 GHz.

- A Samsung Galaxy Nexus TI OMAP 4460 ARM Cortex-A9 @ 1.2 GHz.

Implementations

- The proposed method for O-Ate pairing is implemented for different security levels.
- Platform-independent C code (BN-254, BN-446, and BN-638.)
  - runs unmodified on all the mentioned ARM platforms
- Hand-optimized Assembly (BN-254)
  - Loop unrolling (to avoid conditional branches and reorder the instructions)
  - Instruction re-ordering (to minimize the number of pipeline stalls)
  - Register allocation (to eliminate the need to access memory)
  - Multiple stores (to minimize the memory access instructions)

Implementation Results

Feroceon at 1.2 GHz [This work]					Galaxy (ARM v7) Cortex-A9 at 1.2 GHz [This work]			
Field Size	Operation Timing [μs]				Operation Timing [μs]			
	ML	FE	O-A(a)	O-A(p)	ML	FE	O-A(a)	O-A(p)
254 (asm)	9,722	6,176	16,076	15,898	6,147	3,758	10,573	9,905
254 (C)	11,877	7,550	19,427	19,509	6,859	4,382	11,839	11,241
446 (C)	42,857	23,137	65,994	65,958	25,792	13,752	39,886	39,544
638 (C)	98,044	51,351	149,395	153,713	65,698	33,658	99,356	99,466

iPad 2 1.0 GHz [This work]					Tegra 2 (ARM v7) Cortex-A9 at 1.0 GHz [1]			
Field Size	Operation Timing [μs]				Operation Timing [μs]			
	ML	FE	O-A(a)	O-A(p)	ML	FE	O-A(a)	O-A(p)
254 (C)	8,338	5,483	14,604	13,821	26,320	24,690	51,010	55,190
446 (C)	32,087	17,180	49,365	49,267	97,530	86,750	184,280	195,560
638 (C)	79,056	40,572	119,628	123,410	236,480	413,370	649,850	768,060

[1] Tolga Acar, Kristin Lauter, Michael Naehrig, and Daniel Shumow, "*Affine Pairings on ARM*", Pairing-Based Cryptography – Pairing 2012, Lecture Notes in Computer Science, Springer-Verlag (2012), to appear.

Comparison

- I/M ratio is lower in larger base fields on all platforms.
- In [1], affine coordinates is always faster than projective.
  - It is true when I/M ratio is < 9.
  - Faster results for projective coordinates for BN-254 and BN-446.
- In [1], T_{FE} goes up at the higher security levels.
  - $T_{ML} > T_{FE}$ in all platforms for all security levels.

Curve	I/M Ratio			
	Feroceon	iPad2	Galaxy Nexus	Tegra 2 [1]
BN-254	10.0	10.5	11.3	10.6
BN-446	9.1	9.3	9.8	8.9
BN-638	7.7	7.8	7.9	6.8

[1] Tolga Acar, Kristin Lauter, Michael Naehrig, and Daniel Shumow, "*Affine Pairings on ARM*", Pairing-Based Cryptography – Pairing 2012, Lecture Notes in Computer Science, Springer-Verlag (2012), to appear.

Conclusions

- Efficiently implemented the O-Ate pairing on BN curves for different security levels.
- Extended the concept of lazy reduction to inversion in extension fields and optimized the sparse multiplication algorithm in the degree 12 extension.
- Efficiently performed final exponentiation and reduced its computation time.
- Homogeneous projective coordinates are unambiguously faster than affine coordinates for O-Ate pairings at the 128-bit security level when higher levels of optimization are used.
- Our timing results are over three times faster than the previous fastest results.

Thank You!