# Cryptographic Applications of Bilinear Pairings

*A Hands-on Introduction*

Paulo S. L. M. Barreto
University of São Paulo

# Outline

- Motivation
- Pairings on curves
- Divisors
- Miller’s algorithm and its variants
- Some protocols

# Caveat

- The *hands-on* qualifier in the subtitle means that the discussion will be *informal* on occasion (in other words, I won't be telling the whole truth – it's up to *you* to find out what I'm omitting).
- Some of the questions you make may become *your* homework if working out the answer is particularly insightful – or if answering it now would take too long ☺

# A Motivation

- Discrete logarithm problem (DLP): given a cyclic group $\langle G \rangle$ and a point $P = \alpha G$ for some $\alpha$, compute $\alpha$.
- The DLP in some elliptic curve groups is conjectured to be intractable: best algorithm known runs in time exponential in $\#\langle G \rangle$.

# A Motivation

- Consider two cyclic groups $\langle G_1 \rangle$ and $\langle G_2 \rangle$ such that:
  - The DLP is easy in $\langle G_2 \rangle$;
  - There is an efficiently computable isomorphism $\varphi$: $\langle G_1 \rangle \rightarrow \langle G_2 \rangle$.
- How hard is the DLP in $\langle G_1 \rangle$?

# A Motivation

- The DLP in $\langle G_1 \rangle$ is no harder than the DLP in $\langle G_2 \rangle$!
- Let $P_1 = \alpha G_1$ for some (unknown) $\alpha$.
- Compute (efficiently) $P_2 = \varphi(P_1)$, and note that $P_2 = \varphi(\alpha G_1) = \alpha\varphi(G_1) = \alpha G_2$.
- Solving the DLP for $P_2 = \alpha G_2$ is easy, and gives the solution $\alpha$ to the DLP for $P_1$.
- *MOV-FR reduction.*

# A Motivation

- Are there any groups $\langle G_1 \rangle$ and $\langle G_2 \rangle$ where the MOV-FR reduction is feasible?
- Well, yes, of course! ☺
- But then, what are these groups, and what is the isomorphism $\varphi$?
- To answer this question, we need to define *pairings*.

# Pairings on curves

- Let:
  - E be a curve defined over a field $\mathbf{F}_q$,
  - $r$ be coprime to char($\mathbf{F}_q$),
  - K be a “suitable” extension of $\mathbf{F}_q$,
  - G be a “suitable” subgroup of E(K), and
  - $\mu_r$ be the subgroup of $\mathbf{F}_q$* consisting of all $r$-th roots of unity.

# Pairings on curves

- Definition: a *pairing* on E is a function

$$e: E(K)[r] \times G \rightarrow \mu_r$$

- satisfying:
  - [*bilinearity*]: $\forall$ P, $P_1$, $P_2 \in E(K)[r]$, $\forall$ Q, $Q_1$ , $Q_2 \in G$:
    $e(P_1 + P_2, Q) = e(P_1, Q)\, e(P_2,Q)$,
    $e(P, Q_1 + Q_2) = e(P, Q_1)\, e(P, Q_2)$.
  - [*non-degeneracy*]: $\forall P \in E(K)[r]$, $\exists Q \in G$: $e(P, Q) \neq 1$.

# Pairings on curves

- Weil pairing:
  - K is the (smallest) extension of $\mathbf{F}_q$ containing all coordinates of points of $r$-torsion of E.
  - G = E(K)[$r$].
- (Reduced) Tate pairing:
  - K is the (smallest) extension of $\mathbf{F}_q$ containing all $r$-th roots of unity.
  - G = E(K).
- Note that $K \subseteq \mathbf{F}_{q^k}$ for the smallest positive $k$, called the *embedding degree* of E(K)[$r$], such that $r \mid q^k - 1$. We will always assume $k > 1$.

# Pairings on curves

- Cautionary notes:
  - The pairings are only efficiently computable if the embedding degree $k$ is of manageable size (but not *too* small).
  - In general, $k$ is enormous, so that special curves are needed to implement pairings.

# Pairings on curves

- For cryptographic purposes, it is convenient for efficiency reasons to restrict the first pairing argument to $E(\mathbf{F}_q)[r]$.
- The Tate pairing is usually faster than the Weil pairing, and hence preferred in practice.
- On supersingular curves (only), there exist distortion maps $\psi : E(\mathbf{F}_q)[r] \rightarrow G$ which enable the use of modified pairings $\hat{e} : E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r] \rightarrow \mu_r$ defined by $\hat{e}(P, Q) = e(P, \psi(Q))$.

# Another application of pairings

- The MOV-FR reduction is feasible for the groups $E(\mathbf{F}_q)[r]$ and $\mu_r$. The efficiently computable isomorphism is $\varphi$: $E(\mathbf{F}_q)[r] \to \mu_r$ defined by $\varphi(P) = e(P, Q)$ for some Q.
- Are pairings useful for anything else?
- Yes, they are – but it took quite a while for cryptographers to notice this.

# Solving the DDHP

- Discrete logarithm problem (DLP):
  - Given P and $a$P, compute $a$.
- Computational Diffie-Hellman problem (CDHP):
  - Given P, $a$P, and $b$P, compute ($ab$)P.
- Decision Diffie-Hellman problem (DDHP):
  - Given P, $a$P, $b$P, and $c$P, decide if $c \equiv ab \pmod{r}$.
- There are groups (called *gap groups*) where the DDHP is easy even though the CDHP is (conjectured to be) hard. Currently, the only known gap groups are those where we can compute pairings: $c \equiv ab \pmod{r} \Leftrightarrow e(aP,bP) = e(cP,P)$.

# Further motivation

- As it turns out, pairings are an amazingly flexible tool to construct cryptographic protocols (often based on new security assumptions).

- But first, we need to know how to compute pairings effectively. To begin with, we need curves with small $k$. Then we need to discuss divisors.

# Pairing-friendly curves

- Supersingular curves over $\mathbf{F}_{p^m}$ always have small $k$:
  - large char $p$, $m = 1 \Rightarrow k = 2$.
  - large char $p$, $m = 2 \Rightarrow k = 3$.
  - char 2, odd $m \Rightarrow k = 4$.
  - char 3, odd $m \Rightarrow k = 6$.
- MNT curves (see [Miyaji-Nakabayashi-Takano]) constructed using the CM method are attractive as a non-supersingular alternative over $\mathbf{F}_p$: $k \in \{3, 4, 6\}$.

# Pairing-friendly curves

- It is actually possible to construct curves containing a subgroup of any desired $k$, but that subgroup's order is usually small compared to the underlying finite field.
- More precisely, we know how to build $E(\mathbf{F}_q)[r]$ so that $r \mid q^k - 1$ for any chosen $k$, but in general $\log q \sim 2 \log r$ except for MNT curves. See [Dupont-Enge-Morain], [Cocks-Pinch], [Brezing-Weng], [Barreto-Lynn-Scott].

# Divisors in a nutshell

- Let E be an elliptic curve over a finite field A, and let $\bar{A}$ be the algebraic closure of A.
- A *divisor* over E is a formal sum

$$D = \Sigma_{P \in E(\bar{A})}\, n_P(P)$$

  where $n_P \in Z$ and only a finite number of coefficients is nonzero.
- The *support* of D is the set $\{P \in E(\bar{A}) : n_P \neq 0\}$.
- The *degree* of D is the value $\deg(D) = \Sigma_{P \in E(\bar{A})}\, n_P$.

# Divisors in a nutshell

- Let $f$ be a function $E(\bar{A}) \to \bar{A}$. Thus $f(P) = f(x, y) = n(x, y) / d(x, y)$ for polynomials $n$, $d$ in $\bar{A}[x, y]$ such that $\gcd(n, d) = 1$.
- We denote the order (multiplicity) of $f$ at P by $\mathrm{ord}_P(f)$:
  - if P is a zero of $f$ (i.e. a zero of $n$), then $\mathrm{ord}_P(f) > 0$.
  - if P is a pole of $f$ (i.e. a zero of $d$), then $\mathrm{ord}_P(f) < 0$.
  - otherwise, $\mathrm{ord}_P(f) = 0$.

# Divisors in a nutshell

- We define the *divisor of function f* as:

$$(f) = \Sigma_{P \in E(\bar{A})} \operatorname{ord}_P(f)\,(P).$$

- A divisor D is called *principal* if $D = (f)$ for some function $f$.
- Properties:
  - $(fg) = (f) + (g), \ (f/g) = (f) - (g).$
  - $(f) = 0 \Leftrightarrow f$ is constant.
  - $\deg((f)) = 0.$

# Divisors in a nutshell

- Consequence: if $(f) = (g)$, then $(g) - (f) = (g/f) = 0$, i.e. $g$ is a constant multiple of $f$. Thus $(f)$ determines $f$ up to a nonzero factor.
- We say that two divisors D and D' are *equivalent*, $D \sim D'$, if $D - D' = (f)$ for some function $f$.
- Function of a divisor: for a divisor D such that $\deg(D) = 0$, we define:

$$f(D) = \prod_{P \in E(\bar{A})} f(P)^{n_P}.$$

# Reduced Tate pairing

- Let $P \in E(\mathbf{F}_q)[r]$ and $Q \in E(\mathbf{F}_{q^k})$, let $f$ be a function such that $(f) \sim r(P) - r(O)$, and let $D \sim (Q) - (O)$ with support disjoint from the support of $(f)$, e.g. $D = (Q + R) - (R)$ for some $R \in E(\mathbf{F}_q)[r]$.
- The reduced Tate pairing is the map $e(P, Q) = f(D)^z$, where $z = (q^k - 1)/r$. Note that raising to $z$ is necessary to ensure that $e(P, Q) \in \mu_r$.
- Miller's algorithm computes $f(D)$ in polynomial time (*see appendix*).
- Faster variants were first described by [Galbraith-Harrison-Soldera] and [Barreto-Kim-Lynn-Scott].

# Line functions

- $g_{U,V}$: line through points U, V $\in E(\mathbf{F}_q)$.
- Notation:

  $U = (x_U, y_U)$, $V = (x_V, y_V)$, $Q = (x, y)$,

  $\lambda_1 = (3x_V{}^2 + a)/(2y_V)$,

  $\lambda_2 = (y_U - y_V)/(x_U - x_V)$.
- Properties (exercise!):

  $g_{U,V}(O) = g_{U,O}(Q) = g_{O,V}(Q) = 1$,

  $g_{V,V}(Q) = \lambda_1(x - x_V) - y + y_V$, $Q \neq O$,

  $g_{U,V}(Q) = \lambda_2(x - x_V) - y + y_V$, $Q \neq O$, $U \neq \pm V$,

  $g_{V,-V}(Q) = x - x_1$, $Q \neq O$.

# Miller's algorithm

$// \; r = (r_t, r_{t-1}, \ldots, r_1, r_0)_2; \; r_t = 1; \; \mathrm{P}, \mathrm{Q} \neq \mathrm{O}.$

$f \leftarrow 1, \;\; \mathrm{V} \leftarrow \mathrm{P}$

**for** i $\leftarrow$ t – 1 **downto** 0 **do**

$\quad f \leftarrow f^{\,2} \cdot g_{\mathrm{V,V}}(\mathrm{Q+R}) \cdot g_{\mathrm{2V,-2V}}(\mathrm{R}) \,/\, g_{\mathrm{2V,-2V}}(\mathrm{Q+R}) \cdot g_{\mathrm{V,V}}(\mathrm{R}),$ $\quad \mathrm{V} \leftarrow 2\mathrm{V}$

**if** $r_i = 1$ **then**

$\quad\quad f \leftarrow f \cdot g_{\mathrm{V,P}}(\mathrm{Q+R}) \cdot g_{\mathrm{V+P,-V-P}}(\mathrm{R}) \,/\, g_{\mathrm{V+P,-V-P}}(\mathrm{Q+R}) \cdot g_{\mathrm{V,P}}(\mathrm{R}),$

$\quad\quad \mathrm{V} \leftarrow \mathrm{V} + \mathrm{P}$

**end if**

**end for**

$z \leftarrow (q^k - 1) \,/\, r$

**return** $f^{\,z}$ $\quad // \; e(\mathrm{P}, \mathrm{Q})$

# BKLS algorithm

- Curves with even $k$.
  - Property: $q^{k/2}-1 \mid (q^k-1)/r$.
- Choose Q = (x, y) so that $x \in \mathbf{F}_{q^{k/2}}$, $y \notin \mathbf{F}_{q^{k/2}}$.
  - Property: $\Phi^{k/2}(Q) = -Q$.
  - Property: $g_{U,-U}(Q) \in \mathbf{F}_{q^{k/2}}$, $\forall U \in E(\mathbf{F}_q)$.
- Choose $R \in E(\mathbf{F}_q)[r]$.
  - Property: $g_{U,V}(R) \in F_q$, $\forall U, V \in E(\mathbf{F}_q)$.
- Therefore, factors $g_{2V,-2V}(R)$ and $g_{V+P,-V-P}(R)$, and all denominators are wiped out by the $z$ powering and can be omitted.

# BKLS algorithm

// $r = (r_t, r_{t-1}, \ldots, r_1, r_0)_2$: $r_t = 1$; P, Q $\neq$ O.

$f \leftarrow 1$, $V \leftarrow P$

**for** i $\leftarrow$ t – 1 **downto** 0 **do**

$\quad f \leftarrow f^2 \cdot g_{V,V}(Q)$, $V \leftarrow 2V$

$\quad$ **if** $r_i = 1$ **then**

$\quad\quad f \leftarrow f \cdot g_{V,P}(Q)$, $V \leftarrow V + P$

$\quad$ **end if**

**end for**

$z \leftarrow (q^k - 1) / r$

**return** $f^z$ // e(P, Q)

# Duursma-Lee algorithm

- The BKLS algorithm is currently the fastest way to compute the Tate pairing on MNT curves, and also works on supersingular curves.
- There is a faster way for supersingular curves in characteristic 3: the Duursma-Lee algorithm [Duursma-Lee]:
  - Simpler step for Miller's algorithm.
  - Simpler final powering.
- Generalization to other characteristics and genera is possible (ECC'2004 talk).

# Pairing-based protocols

- Pairings enable many protocols with novel properties (check the Pairing-Based Crypto Lounge for a long list of research papers).
- New security assumptions, e.g. intractability of the Bilinear Diffie-Hellman problem (BDHP): given P, $a$P, $b$P, and $c$P, compute $e(P,P)^{abc}$.

# BLS signatures

- More properly, perhaps, OP-BLS. See [Okamoto-Pointcheval], [Boneh-Lynn-Shacham].
- One of the shortest signatures known.
- Security assumption does *not* involve the intractability of the BDHP.
- Parameters: P $\in E(\mathbf{F}_q)[r]$, Q $\in E(\mathbf{F}_{q^k})$.
- Hash function H: $\{0,1\}^* \to E(\mathbf{F}_q)[r]$. Thus, $H(m) = \alpha P$ for some (unknown) $\alpha$.
- Signer's key pair: $(s, V = sQ)$.

# BLS signatures

- Signing: compute $\Sigma = s\mathrm{H}(m)$; the signed message is $(m, \Sigma)$.
- Verification: accept $(m, \Sigma) \Leftrightarrow \mathrm{e}(\Sigma, \mathrm{Q}) = \mathrm{e}(\mathrm{H}(m), \mathrm{V})$.
- This works because:

$$\mathrm{e}(\Sigma, \mathrm{Q}) = \mathrm{e}(s\alpha\mathrm{P}, \mathrm{Q}) = \mathrm{e}(\mathrm{P}, \mathrm{Q})^{s\alpha}.$$

$$\mathrm{e}(\mathrm{H}(m), \mathrm{V}) = \mathrm{e}(\alpha\mathrm{P}, s\mathrm{Q}) = \mathrm{e}(\mathrm{P}, \mathrm{Q})^{s\alpha}.$$

# BF identity-based encryption

- First practical instance of an identity-based cryptosystem.
- Security based on the intractability of the BDHP.
- Key Generation Centre (KGC), *aka* Trust Authority (TA), *aka* Private Key Generator (PKG): ($s$, $T = sP$).
- Hash function H: $\{0,1\}^* \rightarrow E(\mathbf{F}_{q^k})$.
- Symmetric cipher $\mathcal{E}$: $\mu_r \times \{0,1\}^* \rightarrow \{0,1\}^*$.

# BF identity-based encryption

- Key extraction: $Q_{id} = H(id)$, $D_{id} = sQ_{id}$.
- Encryption: to encrypt a message $m$, choose random $u \in Z^*_r$ and compute $N = uP$, $K = e(T, Q_{id})^u$, $c = \mathcal{E}_K(m)$. The ciphertext is the pair $(N, c)$.
- Decryption: to decrypt $(N, c)$, compute $K = e(N, D_{id})$ and $m = \mathcal{E}^{-1}{}_K(c)$.
- This works because $e(T, Q_{id})^u = e(sP, Q_{id})^u = e(uP, sQ_{id}) = e(N, D_{id})$.

# Other schemes

- One can do id-based signatures (lots of different kinds), authenticated key agreement, threshold encryption, ...
- Conventional (non-id-based) schemes with quite unconventional properties are possible, including signatures (many more different kinds), hierarchical systems, access control, certificateless PKC, ...
- Your contribution to the list is welcome.

# Thanks!

# References


- [Barreto-Kim-Lynn-Scott]
  - P. S. L. M. Barreto, H. Y. Kim. B. Lynn, M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems," Advances in Cryptology -- Crypto'2002, Lecture Notes on Computer Science 2442, Springer-Verlag (2002), pp. 354--368. See also Cryptology ePrint Archive, Report 2002/008.
- [Barreto-Lynn-Scott]
  - P. S. L. M. Barreto, B. Lynn, M. Scott, "Constructing Elliptic Curves with Prescribed Embedding Degrees," Security in Communication Networks -- SCN'2002, Lecture Notes on Computer Science 2576, Springer-Verlag (2003), pp. 257--267. See also Cryptology ePrint Archive, Report 2002/088.
- [Boneh-Franklin]
  - D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," Advances in Cryptology -- Crypto'2001, Lecture Notes on Computer Science 2139, Springer-Verlag (2001), pp. 213--229.
- [Boneh-Lynn-Shacham]
  - D. Boneh, B. Lynn, H. Shacham, "Short signatures from the Weil pairing," Advances in Cryptology -- Asiacrypt'2001, Lecture Notes on Computer Science 2248, Springer-Verlag (2002), pp. 514--532.
- [Brezing-Weng]
  - F. Brezing, A. Weng, "Elliptic curves suitable for pairing based cryptography," Cryptology ePrint Archive, Report 2003/143.
- [Cocks-Pinch]
  - C. Cocks, R. G. E. Pinch, "Identity-based cryptosystems based on the Weil pairing," unpublished manuscript, 2001.

# References


- [Dupont-Enge-Morain]
  - R. Dupont, A. Enge, F. Morain, "Building curves with arbitrary small MOV degree over finite prime fields," Cryptology ePrint Archive, Report 2002/094.
- [Galbraith-Harrison-Soldera]
  - S. D. Galbraith, K. Harrison, D. Soldera, "Implementing the Tate pairing," Algorithmic Number Theory Symposium -- ANTS-V, Lecture Notes on Computer Science 2369, Springer-Verlag (2002), pp. 324--337.
- [Granger-Page-Stam]
  - R. Granger, D. Page, M. Stam, "On Small Characteristic Algebraic Tori in Pairing-Based Cryptography," Cryptology ePrint Archive, Report 2004/132.
- [Miyaji-Nakabayashi-Takano]
  - A. Miyaji, M. Nakabayashi, S. Takano, "New Explicit Conditions of Elliptic Curve Traces for FR-Reduction," IEICE Transactions on Fundamentals E84-A(5) (2001), pp. 1234--1243.
- [Okamoto-Pointcheval]
  - T. Okamoto, D. Pointcheval, "The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes," Practice and Theory in Public Key Cryptography -- PKC'2001 Lecture Notes on Computer Science 1992, Springer-Verlag (2001), pp. 104--118.
- And of course don't forget the Pairing-Based Crypto Lounge: `<http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html>`

# Appendix

# Miller's iterative formula

- Define a family of functions $f_{n,\mathrm{P}}$ such that $(f_{n,\mathrm{P}}) = n(\mathrm{P}) - (n\mathrm{P}) - (n-1)(\mathrm{O})$.
- We need to compute $f_{r,\mathrm{P}}$ with divisor $(f_{r,\mathrm{P}}) = r(\mathrm{P}) - r(\mathrm{O})$ for the Tate pairing.
- Solution: recurrent definition.

# Miller's iterative formula

- $(f_{0,\mathrm{P}}) = (f_{1,\mathrm{P}}) = 0$, i.e. $f_{0,\mathrm{P}}$ and $f_{1,\mathrm{P}}$ are constant functions: take $f_{0,\mathrm{P}} = f_{1,\mathrm{P}} = 1$.
- $(f_{a+b,\mathrm{P}}) = (a+b)(\mathrm{P}) - ([a+b]\mathrm{P}) - (a+b-1)(\mathrm{O}) =$
  $a(\mathrm{P}) - (a\mathrm{P}) - (a-1)(\mathrm{O}) +$
  $b(\mathrm{P}) - (b\mathrm{P}) - (b-1)(\mathrm{O}) +$
  $(a\mathrm{P}) + (b\mathrm{P}) + (-[a+b]\mathrm{P}) - 3(\mathrm{O})$
  $- \{([a+b]\mathrm{P}) + (-[a+b]\mathrm{P}) - 2(\mathrm{O})\} =$
  $(f_{a,\mathrm{P}}) + (f_{b,\mathrm{P}}) + (g_{a\mathrm{P},b\mathrm{P}}) - (g_{[a+b]\mathrm{P},-[a+b]\mathrm{P}})$.

# Miller's iterative formula

- Particular cases:
  - $(f_{2a,\mathrm{P}}) = 2(f_{a,\mathrm{P}}) + (g_{a\mathrm{P},a\mathrm{P}}) - (g_{2a\mathrm{P},-2a\mathrm{P}})$, hence $f_{2a,\mathrm{P}} = f_{a,\mathrm{P}}{}^2 \cdot g_{a\mathrm{P},a\mathrm{P}} \,/\, g_{2a\mathrm{P},-2a\mathrm{P}}$.
  - $(f_{a+1,\mathrm{P}}) = (f_{a,\mathrm{P}}) + (g_{a\mathrm{P},\mathrm{P}}) - (g_{[a+1]\mathrm{P},-[a+1]\mathrm{P}})$, hence $f_{a+1,\mathrm{P}} = f_{a,\mathrm{P}} \cdot g_{a\mathrm{P},\mathrm{P}} \,/\, g_{[a+1]\mathrm{P},-[a+1]\mathrm{P}}$.
- All we need is to compute the line functions at the specified multiples of P, namely, those that appear during the computation of $r$P.