Uma comparação do custo computacional dos emparelhamentos bilineares Ate, R-Ate e Optimal Ate sobre curvas Barreto-Naehrig

Leandro Aparecido Sangalli[1], **Marco Aurélio Amaral Henriques**[1]

[1]Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (UNICAMP)
Av. Albert Einstein - 400, Cidade Universitária Zeferino Vaz - Barão Geraldo,
Campinas-SP, Brasil - 13083-853

Abstract. *This work presents a detailed theoretical evaluation and compares the computational costs of Ate, R-Ate and Optimal Ate bilinear pairings defined over Barreto-Naehrig curves. The evaluation confirmed some experimental results present in the literature, showing a slightly better performance (around* $0,5\%$*) of R-Ate pairings over Optimal Ate. Moreover, a virtual generic processor, with a restricted instruction set, was used to measure and compare the costs of the mentioned pairings under different conditions, such as different word sizes (including 128 and 256 bits), different multipliers and different coordinates. The change in word size was found to have the largest impact in the pairings computational costs when compared to other parameters.*

1. Introdução

Emparelhamentos bilineares são funções que mapeiam dois elementos, que podem ser de conjuntos distintos ou não, em um terceiro elemento de outro conjunto. Os emparelhamentos bilineares sobre curvas elípticas pertencem a uma classe desses mapeamentos, que recebem como argumentos elementos (pontos) de certas curvas elípticas.

O cálculo de emparelhamentos bilineares é dividido em duas etapas: o laço de Miller, cujo custo de execução representa a parte mais significativa do custo total, e a exponenciação final, que garante a unicidade do emparelhamento (esta segunda etapa não é necessária em todos os tipos de emparelhamentos). O cálculo do laço de Miller é efetuado tradicionalmente por meio do algoritmo de Miller, um algoritmo iterativo que computa o valor do emparelhamento em tempo polinomial, porém com um alto custo computacional. A eficiência de quaisquer protocolos baseados em emparelhamentos depende diretamente do custo de execução do algoritmo de cálculo destes emparelhamentos.

Para minimizar o custo de cálculo dos emparelhamentos, diversas formas de otimização foram propostas, entre elas a utilização de diferentes tipos de curvas, como as de Miyaji-Nakabayashi-Takano [Miyaji et al. 2001] e Barreto-Naehrig [Barreto and Naehrig 2002]. Outra tentativa de facilitar a utilização de emparelhamentos bilineares em protocolos criptográficos, foi o desenvolvimento de algoritmos de cálculo de emparelhamentos bilineares alternativos ao de Miller, como o BKLS [A. J. Devegili and Dahab 2007], que é uma otimização do algoritmo de Miller por meio de escolhas apropriadas dos parâmetros utilizados para definir a função de emparelhamento.

 ©2015 SBC — Soc. Bras. de Computação

Os tipos de emparelhamentos bilineares inicialmente utilizados em criptografia foram os de Weil e Tate. No decorrer da primeira década do século 21, em busca de maior eficiência dos criptossistemas baseados em emparelhamentos, foram propostos novos tipos, como o emparelhamento Eta [Barreto et al. 2004], Ate [Hess et al. 2006], R-Ate [E. Lee and Park 2008], $\chi-$Ate [Galbraith and Paterson 2008] e o Optimal Ate [Vercauteren 2010], um dos mais utilizados atualmente.

Com base na literatura é possível encontrar trabalhos que apresentam implementações de diferentes tipos de emparelhamentos sob as mesmas condições, utilizando como métricas de comparação medidas de tempo, número de ciclos, entre outras. Porém, os resultados dessas implementações podem variar de acordo com a arquitetura do processador.

Este trabalho tem os seguintes objetivos:

- comparar o custo de cálculo dos emparelhamentos Ate, R-Ate e Optimal Ate no nível de operações aritméticas sobre corpos finitos, quando definidos sobre curvas do tipo Barreto-Naehrig;
- avaliar o impacto que o tamanho da palavra e outras características de um processador têm sobre o custo destes emparelhamentos.

O desenvolvimento deste trabalho possibilitou: confirmar de forma teórica que o emparelhamento R-ate pode ser mais eficiente que o Optimal Ate; avaliar o custo de cálculo destes emparelhamentos em situações não existentes na prática, como por exemplo, com processadores de 128 e 256 bits; constatar que o tamanho da palavra do processador é a característica que mais impacta o desempenho no cálculo dos emparelhamentos.

2. Conceitos preliminares

Uma curva elíptica E definida sobre um conjunto $\mathbb{F}_p$ é dada pela equação de Weierstrass $E : y^2 + axy + by \equiv x^3 + cx^2 + dx + e \ (mod\ p)$ onde $a, b, c, d, e \in \mathbb{F}_p$. Esta pode ser escrita também na forma reduzida de Weierstrass $E : y^2 \equiv x^3 + ax + b \ (mod\ p)$ onde $a, b \in \mathbb{F}_p$ e $4a^3 + 27b^2 \neq 0$. O conjunto de pontos da curva, juntamente com um ponto especial ∞ (ponto no infinito) é denominado grupo elíptico, sendo denotado por $E(\mathbb{F}_p)$. Ou seja, $E(\mathbb{F}_p) = \{(x, y) \in E\} \cup \{\infty\}$. Seja $P \in E(\mathbb{F}_p)$, então a ordem de P é o menor inteiro n tal que $n \cdot P = \infty$. Seja o conjunto $E(\mathbb{F}_{p^k})[n]$. Neste caso, k é chamado de grau de imersão de E, sendo definido como o menor valor inteiro, tal que $n|(p^k - 1)$ e $n \nmid (p^s - 1)$ para qualquer $0 < s < k$.

Existem curvas elípticas que são mais adequadas para emparelhamentos, sendo denominadas curvas amigáveis a emparelhamentos (Pairing-Friendly Elliptic Curves). O tipo de curva amigável mais utilizado atualmente é a curva Barreto-Naehrig, ou BN. As curvas BN são definidas sobre corpos finitos primos, e possuem ordem prima. Nestas curvas, os parâmetros necessários para configuração do sistema são obtidos da seguinte forma: seja $u \in \mathbb{Z}^*$ escolhido de tal modo que a característica p do corpo $\mathbb{F}_p$ sobre o qual a curva elíptica $E(\mathbb{F}_p)$ é definida, seja dada pela equação $p(u) = 36u^4+36u^3+24u^2+6u+1$, onde p é primo. A ordem da curva E é dada por $n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1$, sendo n primo, e o traço de Frobenius da curva obtido a partir de $t(u) = 6u^2 + 1$. Vale ressaltar que o parâmetro u configura o nível de segurança de um sistema definido sobre uma curva BN.

 ©2015 SBC — Soc. Bras. de Computação

3. Emparelhamentos bilineares

Definição 3.1. *Sejam dois grupos cíclicos aditivos $\mathbb{G}_1$, $\mathbb{G}_2$ e um grupo cíclico multiplicativo $\mathbb{G}_3$, todos de ordem prima n. Um emparelhamento bilinear, $\boldsymbol{e}$, é um mapa bilinear não-degenerado e eficientemente computável,*

$$\boldsymbol{e}\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$$

que satisfaz as seguintes propriedades:

(i) ***Bilinearidade:*** *$\forall (P,Q) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\forall a,b \in \mathbb{F}_p$, $\boldsymbol{e}(aP,bQ) = \boldsymbol{e}(-aP,-bQ) = \boldsymbol{e}(bP,aQ) = \boldsymbol{e}(abP,Q) = \boldsymbol{e}(P,abQ) = \boldsymbol{e}(P,bQ)^a = \boldsymbol{e}(bP,Q)^a = \boldsymbol{e}(aP,Q)^b = \boldsymbol{e}(P,aQ)^b = \boldsymbol{e}(P,Q)^{ab}$.*

(ii) ***Não-degeneração:*** *$\boldsymbol{e}(P,Q) \neq I$, onde I é o elemento identidade do grupo multiplicativo $\mathbb{G}_3$, sendo $P \in \mathbb{G}_1$ e $Q \in \mathbb{G}_2$ geradores destes grupos.*

(iii) ***Computabilidade:*** *O emparelhamento $\boldsymbol{e}(P,Q)$ deve ser computado eficientemente em tempo polinomial.*

3.1. Tipos de emparelhamentos

A busca por eficiência no cálculo dos emparelhamentos possibilitou a descoberta de outros tipos de emparelhamentos, derivados a partir dos emparelhamentos básicos de Weil e de Tate, os dois precursores em aplicações criptográficas. Esta seção apresenta os emparelhamentos Ate, R-Ate e Optimal Ate que são alguns dos obtidos a partir das otimizações no emparelhamento de Tate.

O emparelhamento Ate é uma versão generalizada do emparelhamento Eta para curvas elípticas ordinárias amigáveis a emparelhamentos [Hess et al. 2006].

Definição 3.2 (Ate)**.** *Seja $P \in E(\mathbb{F}_p)[n]$ e $Q \in E(\mathbb{F}_{p^k})[n]$. O emparelhamento Ate pode ser definido de forma genérica como*

$$e_{Ate}(P,Q) = f_{j,P}(Q)^{(p^k-1)/n}.$$

sendo $j = t-1$, onde t representa o traço de Frobenius de E e $f_{j,P}$ a função de Miller que é obtida no decorrer do cálculo de jP.

O Algoritmo 1 apresenta um método de cálculo do emparelhamento Ate, utilizando o algoritmo BKLS [A. J. Devegili and Dahab 2007]. A função $g_{T,Q}(P)$ será detalhada mais adiante.

O emparelhamento R-Ate é uma generalização do emparelhamento Ate, e pode ser definido sobre curvas elípticas e hiperelípticas [E. Lee and Park 2008].

Definição 3.3 (R-Ate)**.** *Seja $P \in E(\mathbb{F}_p)[n]$, $Q \in E(\mathbb{F}_{p^k})[n]$. O emparelhamento R-Ate é dado por*

$$e_{R-Ate}(Q,P) = (f \cdot (f \cdot l_{jQ,Q}(P))^p \cdot l_{\phi(jQ+Q),jQ}(P))^{\frac{p^k-1}{n}}.$$

onde $j = -6u-3$, $\phi(Q) = \phi(x_Q, y_Q) = (x_Q^p, y_Q^p)$ e $l_{\phi(jQ+Q),jQ}$ representa a equação da reta que passa pelos pontos $\phi(jQ+Q)$ e jQ.

 ©2015 SBC — Soc. Bras. de Computação

Algoritmo 1 Emparelhamento Ate

1: **function** ATE
2: **Entrada:** $P \in E(\mathbb{F}_p)[n], Q \in E(\mathbb{F}_{p^k})$ e $j = t - 1 = \sum_{i=0}^{\lceil log_2 j \rceil} 2^i j_i$.
3: **Saída:** $e_{Ate}(Q, P)$
4: $f \longleftarrow 1, T \longleftarrow Q$
5: Para i de $\lfloor log_2 j \rfloor - 2$ até 0
6: $\quad f \longleftarrow f^2 \cdot g_{T,T}(P); T \longleftarrow 2T$
7: $\quad$ Se $j_i = 1$
8: $\quad\quad f \longleftarrow f \cdot g_{T,Q}(P); T \longleftarrow T + Q$
9: $e_{Ate}(Q, P) \longleftarrow f^{\frac{p^k-1}{n}}$
10: Retorna $e(Q, P)$
11: **end function**

Algoritmo 2 Emparelhamento R-Ate

1: **function** R-ATE
2: **Entrada:** $P \in E(\mathbb{F}_p)[n], Q \in E(\mathbb{F}_{p^k})$ e $j = -6u - 3 = \sum_{i=0}^{\lceil log_2 j \rceil} 2^i j_i$.
3: **Saída:** $e_{R-Ate}(Q, P)$
4: $f \longleftarrow 1, T \longleftarrow Q$
5: Para i de $\lfloor log_2 j \rfloor - 2$ até 0
6: $\quad f \longleftarrow f^2 \cdot g_{T,T}(P); T \longleftarrow 2T$
7: $\quad$ Se $j_i = 1$
8: $\quad\quad f \longleftarrow f \cdot g_{T,Q}(P); T \longleftarrow T + Q$
9: $f \longleftarrow f \cdot (f \cdot g_{T,Q}(P))^p \cdot g_{\phi(T+Q),T}(P)$
10: $e_{R-Ate}(Q, P) \longleftarrow f^{\frac{p^k-1}{n}}$
11: Retorna $e_{R-Ate}(Q, P)$
12: **end function**

O Algoritmo 2 mostra o cálculo do emparelhamento R-Ate [Aranha and López 2009].

O emparelhamento Optimal Ate é uma generalização do emparelhamento Ate que pode ser calculado com aproximadamente $log_2 n/\varphi(k)$ iterações no laço de Miller, onde φ é a função de Euler. Emparelhamentos que obedecem a esta condição são chamados Emparelhamentos Ótimos ou *Optimal Pairings* [Vercauteren 2010].

Definição 3.4 (Optimal Ate)**.** *Seja $P \in E(\mathbb{F}_p)[n]$, $Q \in E(\mathbb{F}_{p^k})[n]$, onde E é uma curva elíptica Barreto-Naherig e $j = |6u + 2|$. Seja $\phi^2(x, y) = (x^{p^2}, y^{p^2})$. Então, o emparelhamento Optimal Ate é dado por*

$$e_{opt-Ate}(Q, P) = (f_{j,Q}(P) \cdot f_{jQ,\phi(Q)}(P) \cdot g_{jQ+\phi(Q),-\phi^2(Q)}(P))^{\frac{p^k-1}{n}}.$$

O emparelhamento Optimal Ate é calculado pelo Algoritmo 3 [Aranha et al. 2013].

 ©2015 SBC — Soc. Bras. de Computação

Algoritmo 3 Emparelhamento Optimal Ate

1: **function** OPTIMAL ATE
2: **Entrada:** $P \in E(\mathbb{F}_p)[n]$, $Q \in E(\mathbb{F}_{p^k})$ e $j = |6u+2| = \sum_{i=0}^{\lceil log_2 j \rceil} 2^i j_i$
3: **Saída:** $e_{Opt-Ate}(Q, P)$
4: $d \leftarrow g_{Q,Q}(P)$, $T \leftarrow 2Q$, $e \leftarrow 1$
5: Se $j_{\lceil log_2 j \rceil - 1} = 1$
6: $\quad e \leftarrow g_{T,Q}(P)$, $T \leftarrow T + Q$
7: $f \leftarrow d \cdot e$
8: Para i de $\lfloor log_2 j \rfloor - 2$ até 0
9: $\quad f \longleftarrow f^2 \cdot g_{T,T}(P)$, $T \longleftarrow 2T$
10: $\quad$ Se $j_i = 1$
11: $\quad\quad f \longleftarrow f \cdot g_{T,Q}(P)$, $T \longleftarrow T + Q$
12: $Q_1 \leftarrow \phi_p(Q)$, $Q_2 \leftarrow \phi_{p^2}(Q)$
13: Se $u < 0$
14: $\quad T \leftarrow -T$, $f \leftarrow f^{p^6}$
15: $d \leftarrow g_{T,Q_1}(P)$, $T \leftarrow T + Q_1$
16: $e \leftarrow g_{T,-Q_2}(P)$, $T \leftarrow T - Q_2$, $f \leftarrow f \cdot (d \cdot e)$
17: $e_{Opt-Ate}(Q, P) \leftarrow f^{(p^6-1)(p^2+1)(p^4-p^2+1)/n}$
18: Retorna $e_{Opt-Ate}(Q, P)$
19: **end function**

Observando os Algoritmos 1, 2, 3 pode-se perceber que aparentemente há mais operações envolvidas no cálculo dos emparelhamentos R-Ate e Optimal Ate que no do emparelhamento Ate, o que é verdade, mas há uma diferença bem mais significativa entre estes emparelhamentos: o comprimento do laço de Miller, que é a parte mais significativa no cálculo do emparelhamento.

4. Análise de custo do cálculo dos emparelhamentos

No decorrer do cálculo dos emparelhamentos Ate, R-Ate e Optimal Ate são exigidas operações sobre corpos de extensão que requerem operações sobre $\mathbb{F}_{p^{12}}$. Buscando maior eficiência na implementação destas operações, são utilizadas extensões de corpos finitos em torre (Tower Extension) que é uma forma de representar elementos de $\mathbb{F}_{p^{12}}$ como polinômios de primeiro ou segundo grau e coeficientes em extensões de $\mathbb{F}_p$ utilizando aritmética sobre binômios irredutíveis. Neste caso, as extensões em torre utilizadas foram:

$$\begin{aligned} \mathbb{F}_{p^2} &= \mathbb{F}_p[i]/(i^2 - \beta);\ \text{onde } i \in \mathbb{F}_{p^2} \text{ e } \beta \in \mathbb{F}_p \\ \mathbb{F}_{p^6} &= \mathbb{F}_{p^2}[v]/(v^3 - \varepsilon);\ \text{onde } v \in \mathbb{F}_{p^6} \text{ e } \varepsilon \in \mathbb{F}_{p^2} \\ \mathbb{F}_{p^{12}} &= \mathbb{F}_{p^6}[z]/(z^2 - v);\ \text{onde } z \in \mathbb{F}_{p^{12}} \end{aligned} \tag{1}$$

Deve ser ressaltado que β, ε e v devem ser não-quadrados e não-cubos, sendo utilizados para representar extensões de corpos finitos.

Com o esquema de representação em torre (1) é possível representar elementos de $\mathbb{F}_{p^{12}}$ como polinômios de grau máximo 1 e coeficientes em $\mathbb{F}_{p^6}$. Da mesma forma, é possível representar elementos de $\mathbb{F}_{p^6}$ como polinômios de grau máximo 2 e coeficientes

 ©2015 SBC — Soc. Bras. de Computação

em $\mathbb{F}_{p^2}$. E um elemento de $\mathbb{F}_{p^2}$ é representado como um polinômio de grau máximo 1 e coeficientes em $\mathbb{F}_p$. Por meio destas representações em torre é possível efetuar operações aritméticas sobre corpos de extensão de forma mais eficiente.

4.1. Custos das operações sobre as extensões em torre

Considere as notações (a, m, s, r, i) e ($\tilde{a}$, $\tilde{m}$, $\tilde{s}$, $\tilde{r}$, $\tilde{i}$) denotando as operações de adição, multiplicação, quadrado, redução modular e inversão em $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$, respectivamente. Sejam (m_u, s_u) e ($\tilde{m_u}$, $\tilde{s_u}$) as operações de multiplicação sem redução e quadrado sem redução sobre $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$, respectivamente. Sejam m_β, m_ϵ e m_v multiplicações pelas constantes β, ϵ e v, as quais possuem custo inferior ao de uma multiplicação sobre $\mathbb{F}_p$ devido aos valores específicos assumidos por tais constantes. A Tabela 1 descreve o custo das operações de adição, multiplicação, quadrado e inversão sobre $\mathbb{F}_{p^2}, \mathbb{F}_{p^6}$ e $\mathbb{F}_{p^{12}}$, respectivamente, que foram derivadas utilizando operações simples de adição e multiplicação de polinômios com a técnica de Karatsuba. Esta abordagem fez com que tais resultados difiram ligeiramente de outras contagens presentes na literatura como em [Aranha et al. 2013], entre outras.

Tabela 1. Custo das operações aritméticas sobre extensões em torre

$\mathbb{F}_{p^2}$	
Operação	Custo
Adição/Sub.	$\tilde{a} = 2a$
Multiplicação	$\tilde{m} = 3m_u + 8a + 2r + m_\beta$
Quadrado	$\tilde{s} = m_u + 2s_u + 4a + 2r + m_\beta$
Inversão	$\tilde{i} = 2m_u + 2s_u + 3a + i + 2r + m_\beta$
$\mathbb{F}_{p^6}$	
Operação	Custo
Adição/Sub.	$3\tilde{a}$
Multiplicação	$6\tilde{m_u} + 24\tilde{a} + 3\tilde{r} + 2m_\epsilon$
Quadrado	$3\tilde{m_u} + 3\tilde{s_u} + 12\tilde{a} + 3\tilde{r} + 2m_\epsilon$
Inversão	$10\tilde{m_u} + 3\tilde{s_u} + 8\tilde{a} + \tilde{i} + 6\tilde{r} + 4m_\epsilon$
$\mathbb{F}_{p^{12}}$	
Operação	Custo
Adição/Sub.	$6\tilde{a}$
Multiplicação	$18\tilde{m_u} + 96\tilde{a} + 6\tilde{r} + 6m_\epsilon + m_v$
Quadrado	$12\tilde{m_u} + 6\tilde{s_u} + 60\tilde{a} + 6\tilde{r} + 4m_\epsilon + m_v$
Inversão	$28\tilde{m_u} + 9\tilde{s_u} + 89\tilde{a} + 16\tilde{r} + \tilde{i} + 12m_\epsilon + m_v$

5. Aritmética básica de emparelhamentos sobre curvas elípticas

5.1. Configurações dos emparelhamentos

Para estimar o custo e posteriormente efetuar uma comparação dos emparelhamentos Ate, R-Ate e Optimal-Ate, serão utilizados os seguintes parâmetros: uma curva do tipo BN dada pela equação $E : y^2 = x^3 + 2$ definida sobre um corpo finito $\mathbb{F}_p$ de 254 bits, onde p é parametrizado a partir de $u = -(2^{62} + 2^{55} + 1)$, parâmetros estes que foram adotados buscando melhor eficiência no cálculo dos emparelhamentos [Aranha et al. 2013].

 ©2015 SBC — Soc. Bras. de Computação

O cálculo dos emparelhamentos Ate, R-Ate e Optimal Ate está dividido em duas etapas: o laço de Miller, que representa parte considerável no custo total de cálculo destes emparelhamentos e a exponenciação final [Aranha et al. 2013].

5.2. Custo de operações básicas do laço de Miller

Emparelhamentos bilineares definidos sobre curvas elípticas podem ser calculados utilizando tanto coordenadas afins como projetivas. As coordenadas projetivas têm proporcionado melhor eficiência em relação às coordenadas afins. A seguir são apresentadas as equações de soma e duplicação de pontos tanto para coordenadas afins como para projetivas que foram utilizadas para avaliar o custo do laço de Miller nos emparelhamentos.

5.2.1. Coordenadas Afins

Para o cálculo do emparelhamento bilinear $e(Q, P)$, é fundamental a execução da função linha $g_{T,Q}(P)$. Esta função tem o seguinte significado: $g_{T,Q}$ representa a equação da reta (função linha) que passa por T e Q. Já, $g_{T,Q}(P)$ representa o valor dessa equação quando aplicado a ela o ponto P, onde $g_{T,Q}(P) \in \mathbb{F}^*_{p^{12}}$. Então, $g_{T,Q}(P)$ é denominada função linha avaliada a partir de P. Se $T = Q$, então calcular $g_{T,T}(P)$ consiste em encontrar no ponto P o valor da equação da reta tangente à curva elíptica E' no ponto T, onde E' é um *twist* de E [Aranha et al. 2013]. Sejam $P = (x_P, y_P) \in E(\mathbb{F}_p)$ e $T = (x_T, y_T) \in E'(\mathbb{F}_{p^2})$. Então,

$$g_{T,T}(P) = y_P - \lambda x_P z + (\lambda x_T - y_T)z^3. \tag{2}$$

Fazendo, $\overline{x_P} = -x_P$, $x'_P = \dfrac{\overline{x_P}}{y'_P}$ e $y'_P = \dfrac{1}{y_P}$, a Equação 2 pode ser reescrita como:

$$y'_P \cdot g_{T,T}(P) = 1 + \lambda x'_P z + y'_P(\lambda x_T - y_T)z^3. \tag{3}$$

Desta forma, a Equação 3 pode ser calculada da seguinte forma:

$$y'_P \cdot g_{T,T}(P) = 1 + Fz + Gz^3;$$

$$\begin{gathered} A = \frac{1}{2y_T}; \quad B = 3x_T^2; \quad C = A \cdot B; \quad D = 2x_T \quad x_R = C^2 - D; \\ E = Cx_T - y_T; \quad y_R = E - Cx_R; \quad F = Cx'_P; \quad G = Ey'_P. \end{gathered} \tag{4}$$

Considere que $\overline{x_P}$, y'_P e x'_P sejam pré-computados. O custo para calcular a Equação 4 é de $3\tilde{m} + 2\tilde{s} + 7\tilde{a} + \tilde{i} + 5\tilde{r} + 4m$. Neste, está incluído o custo de calcular $g_{T,T}(P)$ (Equação 2) e de duplicar $R = 2T$. O custo da pré-computação de $\overline{x_P}$, x'_P y'_P é de $m + i + a$. Se $T \neq Q$, então

$$\begin{gathered} y'_P \cdot g_{T,Q}(P) = 1 + Fz + Gz^3; \\ A = \frac{1}{x_Q - x_T}; \quad B = y_Q - y_T; \quad C = A \cdot B; \quad D = x_T + x_Q \quad x_R = C^2 - D; \\ E = Cx_T - y_T; \quad y_R = E - Cx_R; \quad F = Cx'_P; \quad G = Ey'_P. \end{gathered} \tag{5}$$

O custo de cálculo da Equação 5 é de $3\tilde{m} + \tilde{s} + 6\tilde{a} + \tilde{i} + 4\tilde{r} + 4m$. Neste custo, está incluído o custo de calcular $g_{T,Q}(P)$ (Equação 5) e de somar os pontos $R = T + Q$.

 ©2015 SBC — Soc. Bras. de Computação

5.2.2. Coordenadas Projetivas Homogêneas

Seja $T = (X_T, Y_T, Z_T) \in E'(\mathbb{F}_{p^2})$ um ponto representado em coordenadas projetivas homogêneas. Portanto, pode-se efetuar a duplicação de T, ou seja, $R = 2T = T + T = (X_R, Y_R, Z_R)$, mediante utilização das seguintes equações:

$$X_R = \frac{XY}{2}(Y^2 - 9b'Z_T^2); \;\; Y_R = \left[\frac{1}{2}(Y_T^2 + 9b'Z_T^2)\right]^2 - 27b'^2 Z_T^4 \;\text{ e }\; Z_R = 2Y_T^3 Z_T. \quad (6)$$

Seja $P = (x_P, y_P) \in E(\mathbb{F}_p)$. Quando E' é um *twist* do tipo D [Aranha et al. 2013], a função $g_{T,T}(P)$ é calculada da seguinte forma:

$$g_{T,T}(P) = -2Y_T Z_T y_p + 3X_T^2 x_p z + (3b'Z_T^2 - Y_T^2)z^3. \quad (7)$$

Fazendo $\overline{y_p} = -y_p$ e $x'_p = 3x_p$, então, a Equação 7 pode ser reescrita como

$$g_{T,T}(P) = H\overline{y_p} + 3X_T^2 x'_p z + (E - B)z^3;$$

$$A = \frac{X_T Y_T}{2}; \quad B = Y_T^2; \quad C = Z_T^2; \quad E = 3b'C \quad F = 3E; \quad X_R = A(B - F); \quad (8)$$
$$G = \frac{B + F}{2} \quad Y_R = G^2 - 3E^2; \quad H = (Y_T + Z_T)^2 - (B + C); \quad Z_R = BH.$$

A multiplicação por b' tem custo de uma adição sobre $\mathbb{F}_{p^2}$, fato que pode ser verificado via utilização de extensões em torre. A operação de divisão por 2, também apresenta custo de uma adição sobre $\mathbb{F}_{p^2}$, pois pode ser efetuada por uma operação de shift para a direita. Considerando que $\overline{y_p}$ e x'_p na Equação 8 sejam pré-computados, o custo de cálculo da Equação 8 é $3\tilde{m} + 6\tilde{s} + 17\tilde{a} + 8\tilde{r} + 4m$. Neste custo, está incluído o custo de encontrar $g_{T,T}(P)$ (Equação 8) e da duplicação $R = 2T$ (Equação 6). Sejam $T = (X_T, Y_T, Z_T)$, $Q = (x_Q, y_Q) \in E'(\mathbb{F}_{p^2})$, pontos representados em coordenadas projetivas e afins, respectivamente. Então, a soma $R = T + Q = (X_R, Y_R, Z_R)$ pode ser determinada por meio das seguintes equações:

$$\begin{aligned} X_R &= \lambda(\lambda^3 + Z_T\theta^2 - 2X_T\lambda^2) \\ Y_R &= \theta(3X_T\lambda^2 - \lambda^3 - Z_T\theta^2) - Y_T\lambda^3 \\ Z_R &= Z_T\lambda^3. \end{aligned} \quad (9)$$

Seja $P = (x_P, y_P) \in E(\mathbb{F}_p)$. Quando E' é twist do tipo D, então $g_{T,Q}(P)$ pode ser calculado da seguinte forma:

$$g_{T,Q}(P) = -\lambda y_P - \theta x_P z + (\theta x_Q - \lambda y_Q)z^3. \quad (10)$$

Fazendo, $\overline{x_P} = -x_P$. Então, a Equação 10 pode ser reescrita como:

$$g_{T,Q}(P) = \lambda\overline{y_P} + \theta\overline{x_P}z + Jz^3;$$

$$A = y_Q Z_T; \quad B = x_Q Z_T; \quad \theta = Y_T - A; \quad \lambda = X_T - B \quad C = \theta^2; \quad D = \lambda^2;$$
$$E = \lambda^3 \quad F = Z_T C; \quad G = X_T D; \quad H = E + F - 2G; \quad X_R = \lambda H; \quad (11)$$
$$I = Y_T E; \quad Y_R = \theta(G - H) - I; \quad Z_R = Z_T E; \quad J = \theta x_Q - \lambda y_Q.$$

 ©2015 SBC — Soc. Bras. de Computação

O custo de cálculo da Equação 11 é $11\tilde{m} + 2\tilde{s} + 8\tilde{a} + 11\tilde{r} + 4m$. Neste custo, está incluído o custo de calcular $g_{T,Q}(P)$ (Equação 11) e de somar $R = T + Q$ (Equação 9). Na utilização de coordenadas projetivas homogêneas, o custo das pré-computações necessárias para tornar as funções de linha mais eficientes é de $4a$.

Outros tipos de operações definidas sobre curvas elípticas necessárias no cálculo dos emparelhamentos são: negação de pontos e potências de Frobenius. Seja $Q = (x_Q, y_Q) \in E'(\mathbb{F}_{p^2})$, então a negação de Q é denotada por $-Q = (x_Q, -y_Q)$ e tem custo de $\tilde{a}$. As funções ϕ_p e ϕ_{p^2} são denominadas $p-$potência de Frobenius e p^2-potência de Frobenius, respectivamente [Hoffstein et al. 2008]. O custo de execução destas funções é de $2\tilde{m} + 2a$ e $2m + \tilde{a}$, respectivamente.

As funções linha representadas tradicionalmente por $g_{A,A}$ e $g_{A,B}$ nos Algoritmos 1, 2 e 3 assumem valores temporários em $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}[z]/(z^6 - \varepsilon)$, sendo um elemento desse conjunto representado de forma genérica por $a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4 + a_5 z^5$, com $a_i \in \mathbb{F}_{p^2}$. Quando são utilizadas curvas com grau de imersão par (por exemplo, $k = 12$) os valores de $g_{A,A}$ e $g_{A,B}$ são chamados de esparsos, pois possuem três de seus seis coeficientes nulos. As multiplicações envolvendo um valor esparso e um completo (não esparso) são denominadas multiplicações do tipo 1-esparsa, como por exemplo as multiplicações $f^2 \cdot g_{A,A}$ e $f \cdot g_{A,B}$. Estas multiplicações do tipo 1-esparsa possuem custo de $10\tilde{m}_u + 26\tilde{a} + 2m_\varepsilon + m_v + 6\tilde{r}$ se forem utilizadas coordenadas afins e de $16\tilde{m}_u + 28\tilde{a} + 2m_\varepsilon + m_v + 6\tilde{r}$ no caso da utilização de coordenadas projetivas. A multiplicação do tipo 2-esparsa consiste em multiplicar dois valores esparsos, como por exemplo a multiplicação $d \cdot e$ (Linhas 7 e 16 no Algoritmo 3). O custo deste tipo de multiplicação é $3\tilde{m}_u + 9\tilde{a} + m_\varepsilon + 5\tilde{r}$ para coordenadas afins e $6\tilde{m}_u + 20\tilde{a} + m_\varepsilon + 5\tilde{r}$ para projetivas.

A partir deste ponto, será considerado o custo das multiplicações pelas constantes ε e v equivalente ao custo de uma adição sobre $\mathbb{F}_{p^2}$, como observado na Equação 1.

5.2.3. Exponenciação final

A exponenciação final, consiste em elevar $f \in \mathbb{F}_{p^{12}}$ que é o resultado da execução do laço de Miller a um valor $\frac{p^k-1}{n}$. Uma forma de tornar mais eficiente o cálculo da exponenciação final é utilizar subgrupos ciclotômicos [Fuentes-Castañeda et al. 2011]. Para o caso $k = 12$ e $d = 6$, características propiciadas pelas curvas BN, tem-se

$$\frac{p^{12}-1}{n} = (p^6 - 1) \cdot ((p^6 + 1)/\phi_{12}(p)) \cdot (\phi_{12}(p)/n) = (p^6 - 1) \cdot (p^2 + 1) \cdot (p^4 - p^2 + 1)/n$$

Via utilização dos subgrupos ciclotômicos, a exponenciação final é dividida em três partes: duas fáceis e uma difícil de calcular. As duas partes fáceis podem ser calculadas como descrito abaixo.

- O custo para calcular $f^{(p^6-1)}$ é de uma conjugação em $\mathbb{F}_{p^{12}}$, uma inversão em $\mathbb{F}_{p^{12}}$ e uma multiplicação em $\mathbb{F}_{p^{12}}$, pois, $f^{(p^6-1)} = f^{p^6} \cdot f^{-1} = \overline{f} \cdot f^{-1}$.
- O custo para calcular $f^{(p^2+1)}$ é de uma p^2-potência de Frobenius em $\mathbb{F}_{p^{12}}$ e uma multiplicação em $\mathbb{F}_{p^{12}}$.

Existem diferentes propostas de como calcular a parte difícil da exponenciação final mas a ideia básica destas propostas é calcular $f^{d'}$ ao invés de f^d, onde $d = (p^4 - p^2 + 1)/n$ e d' é

 ©2015 SBC — Soc. Bras. de Computação

um múltiplo de d não divisível por n. No cálculo desta parte difícil da exponenciação final estão envolvidas as operações: multiplicações em $\mathbb{F}_{p^{12}}$; quadrados no subgrupo ciclotômico $\mathbb{G}_{\phi_6}$ [Aranha et al. 2013]; p, p^2 e p^3-potências de Frobenius sobre $\mathbb{F}_{p^{12}}$; exponenciações por u; conjungações em $\mathbb{F}_{p^{12}}$; inversões em $\mathbb{F}_{p^{12}}$. Portanto, a forma adotada para o cálculo da exponenciação final [Fuentes-Castañeda et al. 2011] tem custo de:

$$\begin{aligned} C_{\text{exp. final}} &= 389\tilde{m_u} + 1170\tilde{s_u} + 7929\tilde{a} + 931\tilde{r} + 4\tilde{i} + 20m_u + 12a \\ &= 2365m_u + 2348s_u + 23782a + 4988r + 4i. \end{aligned}$$

O número de operações necessárias para calcular a exponenciação final foi estimado de forma semelhante às demais equações de custo.

6. Custo computacional dos emparelhamentos Ate, R-Ate e Optimal Ate em número de operações sobre $\mathbb{F}_p$

A Tabela 2 apresenta o custo de execução de cada emparelhamento para coordenadas afins e projetivas. Os valores apresentados foram obtidos utilizando as equações de custo computacional das operações presentes no algoritmo de Miller e na Exponenciação final (Seções 4 e 5).

Observando a Tabela 2 percebe-se que, tanto em coordenadas afins como projetivas, o emparelhamento R-Ate apresentou um custo total ligeiramente inferior ao custo do emparelhamento Optimal Ate, principalmente nas operações de maior impacto, que são multiplicações, quadrados, reduções e inversões. Em coordenadas projetivas homogêneas, o emparelhamento Optimal Ate apresentou custo inferior ao do emparelhamento R-Ate apenas no número de multiplicações sobre $\mathbb{F}_p$. Porém, em relação às demais operações o Optimal Ate demonstrou custo superior ao do R-Ate confirmando resultados experimentais existentes na literatura. Porém, não é de nosso conhecimento a existência de outras análises teóricas similares a esta que validem tais resultados. O emparelhamento Ate não foi considerado nas comparações anteriores, devido ao seu custo superior em ambos os tipos de coordenadas.

7. Análise do custo computacional dos emparelhamentos Ate, R-ate e Optimal Ate em um processador de referência

Estimado o custo de cálculo dos emparelhamentos Ate, R-Ate e Optimal Ate em número de operações sobre $\mathbb{F}_p$ (Tabela 2), esta seção apresenta os resultados da avaliação deste custo em processadores de referência hipotéticos. Para as mais diversas situações foram considerados processadores genéricos com diferentes características, como palavras de 8, 16, 32, 64, 128 e 256-bits e diferentes opções de multiplicadores. Buscando melhor precisão na comparação foi considerado que estes processadores possuem um mesmo conjunto de instruções com operações aritméticas básicas (adição, subtração, multiplicação), operações lógicas (OR, AND, XOR, SHIFT, entre outras), condicionais e de leitura e escrita em memória.

A fim de estimar o custo de cálculo dos emparelhamentos em função de operações básicas do processador, o custo de cada operação que compõe as equações de custo apresentadas na Tabela 2 foi definido em função do número de instruções de adição (a') e multiplicação (m') nativas do processador. Por fim, para colocar este custo em função da

 ©2015 SBC — Soc. Bras. de Computação

Tabela 2. Custo computacional dos emparelhamentos Ate, R-Ate e Optimal Ate

Coordenadas Afins	
Ate	
Loop de Miller	$11514m_u + 2278s_u + 56048a + 13170r + 136i$
Exp. Final	$2365m_u + 2348s_u + 23782a + 4988r + 4i$
Custo total	$13879m_u + 4626s_u + 79830a + 18158r + 140i$
R-Ate	
Loop de Miller	$5887m_u + 1146s_u + 28644a + 6708r + 70i$
Exp. Final	$2365m_u + 2348s_u + 23782a + 4988r + 4i$
Custo total	$8252m_u + 3494s_u + 52426a + 11696r + 74i$
Optimal Ate	
Loop de Miller	$5891m_u + 1164s_u + 28721a + 6736r + 71i$
Exp. Final	$2365m_u + 2348s_u + 23782a + 4988r + 4i$
Custo total	$8256m_u + 3512s_u + 52503a + 11714r + 75i$
Coordenadas Projetivas	
Ate	
Loop de Miller	$14404m_u + 3028s_u + 67584a + 16578r$
Exp. Final	$2365m_u + 2348s_u + 23782a + 4988r + 4i$
Custo total	$16769m_u + 5376s_u + 91366a + 21566r + 4i$
R-Ate	
Loop de Miller	$7374m_u + 1524s_u + 34550a + 8460r$
Exp. Final	$2365m_u + 2348s_u + 23782a + 4988r + 4i$
Custo total	$9739m_u + 3872s_u + 58332a + 13448r + 4i$
Optimal Ate	
Loop de Miller	$7362m_u + 1548s_u + 34646a + 8488r$
Exp. Final	$2365m_u + 2348s_u + 23782a + 4988r + 4i$
Custo total	$9727m_u + 3896s_u + 58428a + 13476r + 4i$

operação mais simples do processador (a adição a') foram supostos três casos diferentes: os casos onde uma multiplicação (m') tem custo equivalente a dez e quatro adições (a'), e o caso ótimo, onde o custo da multiplicação e da adição são equivalentes.

Para avaliar o custo das operações m_u, s_u, a, r e i nos processadores considerados foram desenvolvidos pseudocódigos utilizando o conjunto de instruções destes processadores hipotéticos. O algoritmo utilizado para realizar a multiplicação sobre $\mathbb{F}_p$ (m_u) foi o método da multiplicação em blocos [Comba 1990]. O mesmo método foi utilizado para realizar a operação de quadrado sobre $\mathbb{F}_p$ (s_u). A adição sobre $\mathbb{F}_p$ (a) foi feita também em blocos. A redução modular (r) foi feita combinando o método de multiplicação e redução de Montgomery [Menezes et al. 1996]. A operação de inverso multiplicativo sobre $\mathbb{F}_p$ (i) utilizou o algoritmo de Euclides Estendido, mas, como este não é um algoritmo de tempo constante, foi considerado o pior caso de execução. Por exemplo, para avaliar o custo da operação de adição sobre $\mathbb{F}_p$ (a) nos diferentes ambientes considerados, primeiramente se estimou quantas palavras do processador são necessárias para armazenar cada um dos elementos envolvidos na operação de adição. Supondo que sejam necessárias W palavras, onde $W = \lceil log_2(p)/w \rceil$ e w é o tamanho da palavra do processador, então o custo da

 ©2015 SBC — Soc. Bras. de Computação

adição simples (sem redução) sobre $\mathbb{F}_p$ é dado por $4wa' + 4a'$, considerando o custo de cada operação de leitura e escrita equivalente a uma a'.

As Tabelas 3 e 4 apresentam o custo de cálculo dos emparelhamentos Ate, R-Ate e Optimal Ate em número de operações de adição nativa do processador. Observando estas tabelas é possível efetuar quatro análises diferentes variando a razão m'/a', o tipo de coordenada adotada, o tipo de emparelhamento e o tamanho da palavra do processador. Primeiramente pode-se observar que um multiplicador mais eficiente (menor razão m'/a') traz benefícios para o cálculo dos emparelhamentos.

Tabela 3. Custo computacional dos emparelhamentos em um processador com palavra de 8, 16 e 32 bits (unidade: número de adições nativas a')

$w = 8$ bits $(W = 32)$				
Emparelhamento	Coordenadas	$m'/a' = 10$	$m'/a' = 4$	$m'/a' = 1$
Ate	Afins	2.077.683.363	1.632.608.355	1.410.070.851
	Projetivas	2.422.615.259	1.892.628.827	1.627.635.611
R-Ate	Afins	1.326.643.282	1.042.556.434	900.513.010
	Projetivas	1.503.838.595	1.176.116.099	1.012.254.851
Optimal Ate	Afins	1.328.963.946	1.044.453.354	902.198.058
	Projetivas	1.506.493.403	1.178.236.379	1.014.107.867
$w = 16$ bits $(W = 16)$				
Emparelhamento	Coordenadas	$m'/a' = 10$	$m'/a' = 4$	$m'/a' = 1$
Ate	Afins	546.868.755	432.874.227	375.876.963
	Projetivas	625.728.299	489.997.163	422.131.595
R-Ate	Afins	348.183.762	275.393.970	238.999.074
	Projetivas	388.683.395	304.723.331	262.743.299
Optimal Ate	Afins	348.857.186	275.958.434	239.509.058
	Projetivas	389.370.971	305.272.667	263.223.515
$w = 32$ bits $(W = 8)$				
Emparelhamento	Coordenadas	$m'/a' = 10$	$m'/a' = 4$	$m'/a' = 1$
Ate	Afins	150.623.659	120.762.139	105.831.379
	Projetivas	166.588.915	131.038.867	113.263.843
R-Ate	Afins	95.416.226	76.334.738	66.793.994
	Projetivas	103.609.123	81.604.387	70.602.019
Optimal Ate	Afins	95.631.894	76.521.654	66.966.534
	Projetivas	103.793.083	81.751.483	70.730.683

A relação entre o custo de cálculo com coordenadas afins e projetivas está ligada diretamente à escolha das instruções do processador utilizadas para implementar as operações aritméticas sobre $\mathbb{F}_p$. As coordenadas projetivas se mostraram mais vantajosas que as afins em processadores de 64 bits ($m'/a' = 1$) e de 128 e 256 bits (qualquer m'/a'). Em todos os casos analisados foi possível perceber uma leve vantagem (inferior a $0,5\%$) do emparelhamento R-Ate com relação ao Optimal Ate, resultado coerente com a avaliação teórica aqui apresentada e com alguns resultados experimentais da literatura [Gouvêa and López 2009].

O impacto mais significativo no cálculo dos emparelhamentos nos casos avalia-

 ©2015 SBC — Soc. Bras. de Computação

Tabela 4. Custo computacional dos emparelhamentos em um processador com palavra de 64, 128 e 256 bits (unidade: número de adições nativas a')

$w = 64$ bits $(W = 4)$				
Emparelhamento	Coordenadas	$m'/a' = 10$	$m'/a' = 4$	$m'/a' = 1$
Ate	Afins	44.797.823	36.650.999	32.577.587
	Projetivas	46.845.423	37.149.279	32.301.207
R-Ate	Afins	28.151.270	22.938.878	20.332.682
	Projetivas	29.197.995	23.189.451	20.185.179
Optimal Ate	Afins	28.229.570	23.009.234	20.399.066
	Projetivas	29.250.147	23.231.235	20.221.779
$w = 128$ bits $(W = 2)$				
Emparelhamento	Coordenadas	$m'/a' = 10$	$m'/a' = 4$	$m'/a' = 1$
Ate	Afins	14.961.393	12.583.965	11.409.129
	Projetivas	14.430.293	11.601.941	10.203.893
R-Ate	Afins	9.299.716	7.775.608	7.024.036
	Projetivas	9.023.999	7.268.183	6.401.891
Optimal Ate	Afins	9.332.226	7.805.754	7.053.054
	Projetivas	9.040.247	7.281.263	6.413.459
$w = 256$ bits $(W = 1)$				
Emparelhamento	Coordenadas	$m'/a' = 10$	$m'/a' = 4$	$m'/a' = 1$
Ate	Afins	5.811.530	5.046.812	4.664.453
	Projetivas	5.086.860	4.177.614	3.722.991
R-Ate	Afins	3.568.763	3.077.231	2.831.465
	Projetivas	3.194.871	2.629.077	2.346.180
Optimal Ate	Afins	3.584.096	3.091.784	2.845.628
	Projetivas	3.200.667	2.633.793	2.350.356

dos foi trazido pela alteração da palavra do processador, pois diversas operações que compõem este cálculo possuem seu custo descrito por uma função quadrática em relação ao número de palavras necessárias para representar cada elemento do corpo finito $\mathbb{F}_p$. Portanto, espera-se que o cálculo de emparelhamentos seja cada vez mais eficiente em arquiteturas que permitam cálculos com palavras mais longas, o que deverá popularizar o uso dos mesmos à medida que novas e mais poderosas arquiteturas surgirem no mercado.

8. Conclusões

Este trabalho apresentou primeiramente uma comparação teórica do custo de cálculo dos emparelhamentos Ate, R-Ate e Optimal Ate fixados sobre uma curva BN definida sobre um corpo finito de 254 bits e comprovou resultados experimentais mostrando ligeira vantagem do emparelhamento R-Ate sobre o Optimal Ate. Em seguida apresentou uma análise do custo de cálculo destes emparelhamentos em diferentes tipos de processadores genéricos hipotéticos. Assim foi possível exercitar vários cenários não disponíveis para experimentação e confirmar os resultados da análise teórica, bem como constatar que o tamanho da palavra do processador é o fator de maior impacto no desempenho. Como trabalhos futuros pretende-se fazer análises semelhantes com implementações destes emparelhamentos em diferentes tipos de processador a fim de validar as conclusões obtidas

 ©2015 SBC — Soc. Bras. de Computação

nesse trabalho, bem como avaliar que tipo de arquitetura seria mais eficiente para estes cálculos.

Referências

A. J. Devegili, M. S. and Dahab, R. (2007). Implementing Cryptographic Pairings over Barreto-Naehrig Curves. Cryptology ePrint Archive, Report 2012/408. https://eprint.iacr.org/2007/390.pdf.

Aranha, D. F., Barreto, P. S. L. M., Longa, P., and Ricardini, J. E. (2013). The Realm of the Pairings. To appear.

Aranha, D. F. and López, J. (2009). Paralelização em Software do Algoritmo de Miller. In *IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG 2009)*, pages 27–40.

Barreto, P. and Naehrig, M. (2002). Pairing-Friendly Elliptic Curves of Prime Order. http://eprint.iacr.org/2005/133.pdf.

Barreto, P. S. L. M., Galbraith, S., Ó hÉigeartaigh, C., and Scott, M. (2004). Efficient Pairing Computation on Supersingular Abelian Varieties. Cryptology ePrint Archive, Report 2004/375. http://eprint.iacr.org/2004/375.pdf.

Comba, P. G. (1990). Exponentiation cryptosystems on the ibm pc. *IBM Syst. J.*, 29(4):526–538.

E. Lee, H. S. L. and Park, C. M. (2008). Efficient and Generalized Pairing Computation on Abelian Varieties. http://eprint.iacr.org/2008/040.pdf.

Fuentes-Castañeda, L., Knapp, E., and Rodríguez-Henríquez, F. (2011). Faster hashing to g2. Center for Applied Cryptographic Research. http://cacr.uwaterloo.ca/techreports/2011/cacr2011-26.pdf.

Galbraith, S. D. and Paterson, K. G., editors (2008). *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, volume 5209 of *Lecture Notes in Computer Science*. Springer.

Gouvêa, C. P. and López, J. (2009). Software implementation of pairing-based cryptography on sensor networks using the msp430 microcontroller. In *Proceedings of the 10th International Conference on Cryptology in India: Progress in Cryptology*, INDOCRYPT '09, pages 248–262, Berlin, Heidelberg. Springer-Verlag.

Hess, F., Smart, N., and Vercauteren, F. (2006). The Eta Pairing Revisited. Cryptology ePrint Archive, Report 2006/110. https://eprint.iacr.org/2006/110.pdf.

Hoffstein, J., Pipher, J., and Silverman, J. (2008). *An Introduction to Mathematical Cryptography*. Springer Publishing Company, Incorporated, 1 edition.

Menezes, A. J., Vanstone, S. A., and Oorschot, P. C. V. (1996). *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.

Miyaji, A., Nakabayashi, M., and TAKANO, S. (2001). New Explicit Conditions of Elliptic Curve Traces for FR-reduction. IEICE Transactions on Fundamentals, E84-A.

Vercauteren, F. (2010). Optimal Pairings. http://eprint.iacr.org/2008/096.pdf.

 ©2015 SBC — Soc. Bras. de Computação