GEOVANDRO CARLOS C. F. PEREIRA

PARAMETRIZAÇÃO E OTIMIZAÇÃO DE CRIPTOGRAFIA DE CURVAS ELÍPTICAS AMIGÁVEIS A EMPARELHAMENTOS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2011

GEOVANDRO CARLOS C. F. PEREIRA

PARAMETRIZAÇÃO E OTIMIZAÇÃO DE CRIPTOGRAFIA DE CURVAS ELÍPTICAS AMIGÁVEIS A EMPARELHAMENTOS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração:

Sistemas Digitais

Orientador:

Prof. Dr. Paulo Sérgio L. M. Barreto

São Paulo
2011

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 26 de maio de 2011.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Pereira, Geovandro Carlos Crepaldi Firmino

Parametrização e Otimização de Criptografia de Curvas Elípticas Amigáveis a Emparelhamentos/ G. C. C. F. Pereira. – ed. rev. – São Paulo, 2011.

98 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais (PCS).

1. Criptologia. 2. Algoritmos. 3. Curvas algébricas. 4. Corpos finitos. 5. Segurança de redes. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais (PCS). II. t.

AGRADECIMENTOS

Aproveito a data de escrita desta seção, a última deste documento, para agradecer de início a duas grandes figuras do mundo do esporte que me motivaram nos quesitos dedicação e excelência. Faço um tributo ao goleiro Rogério Ceni do São Paulo Futebol Clube por ter marcado hoje seu centésimo gol, sendo o maior goleiro artilheiro da história do futebol e dando tanto orgulho a sua torcida. Homenageio também o tenista Roger Federer que, como Ceni em sua posição, é considerado o melhor de todos os tempos, com 16 vitórias em *Grand Slam*, o título mais almejado do tênis. Ambos Rogério Ceni e Roger Federer fazem lances incríveis parecerem fáceis para quem vê, como se não houvesse esforço. Ceni quase não se distancia da bola para cobrar uma falta e Federer executa uma *Grand Willy* no contrapé de seu adversário brincando. O fato é que só enxergamos o resultado ou o efeito de seus treinos.

Vejo a pesquisa científica de forma similar. Apenas o doce, como dizia Aristóteles, permanece visível. Contudo, para obtê-lo, é preciso trabalho árduo. Quando aqui utilizei o resultado do pequeno teorema de Fermat, foi porque foi despendido imenso esforço para prová-lo, não porque era mágico. É por isso que aproveito para agradecer aos pesquisadores que, ao custo de seus esforços, expandiram a fronteira do conhecimento fornecendo resultados tão valiosos para o desenvolvimento deste trabalho.

Também agradeço aos meus amigos do LARC, Laboratório de Arquitetura e Redes de Computadores, Bruno, Mateus e Pedro pelos momentos de descontração bem como pelas discussões empolgantes no quadro branco.

Ao meu amigo e também orientador Paulo Barreto que é craque em fazer pesquisa de ponta e vem mostrando que o Brasil é capaz sim de produzir e de não ser um mero utilizador de tecnologia. Ensinou-me a encarar o conhecimento sob um olhar próprio e de inconformismo, que é essencial no processo de pesquisa e inovação.

Deixo um agradecimento aos meus pais, Antônio e Ivone, que dedicaram parte de suas vidas para me fornecer os recursos de que precisei, mas sobretudo pela educação e exemplos que me passaram. Também ao meu irmão Adriano, pelas boas lembranças de nossa infância, o que determinou nosso forte laço de amizade e por saber que sempre posso contar com ele.

Por fim, deixo um brinde à bebida que foi largamente produzida no interior paulista, especialmente em Jaú, minha cidade natal, e que "estimulou" a conclusão deste trabalho. Uma notável descrição de sua importância para o interior paulista pode ser vista na tese de Rogério Naques Faleiros: **Fronteiras do Café: Fazendeiros e "Colonos" no Interior Paulista**.

RESUMO

A tendência para o futuro da tecnologia é a produção de dispositivos eletrônicos e de computação cada vez menores. Em curto e médio prazos, ainda há poucos recursos de memória e processamento neste ambiente. A longo prazo, conforme a Física, a Química e a Microeletrônica se desenvolvem, constata-se significativo aumento na capacidade de tais dispositivos . No intervalo de curto e médio prazos, entre 20 e 50 anos, até que a tecnologia tenha avanços, soluções leves de *software* se vêem necessárias.

No Brasil, o protocolo de assinatura digital RSA é o mais amplamente adotado, sendo obsolescente como padrão. O problema é que os avanços tecnológicos impõem um aumento considerável no tamanho das chaves criptográficas para que se mantenha um nível de segurança adequado, resultando efeitos indesejáveis em tempo de processamento, largura de banda e armazenamento. Como solução imediata, temos a criptografia de curvas elípticas sendo mais adequada para utilização por órgãos públicos e empresas.

Dentro do estudo de curvas elípticas, este trabalho contribui especificamente com a introdução de uma nova subfamília das curvas amigáveis a emparelhamento Barreto-Naehrig (BN). A subfamília proposta tem uma descrição computacionalmente simples, tornando-a capaz de oferecer oportunidades de implementação eficiente. A escolha das curvas BN também se baseia no fato de possibilitarem uma larga faixa de níveis práticos de segurança.

A partir da subfamília introduzida foram feitas algumas implementações práticas começando com algoritmos mais básicos de operações em corpos de extensão, passando por algoritmos de aritmética elíptica e concluindo com o cálculo da função de emparelhamento. A combinação da nova subfamília BN com a adoção de técnicas de otimização, cuidadosamente escolhidas, permitiu a mais eficiente implementação do emparelhamento Ate ótimo, operação bastante útil em aplicações criptográficas práticas.

Palavras-chave: Criptologia, Algoritmos, Curvas algébricas, Corpos finitos, Segurança de redes, Emparelhamentos bilineares.

ABSTRACT

The trend for the future consists of steadfast shrinking of electrical and computing devices. In the short to medium term, one will still find constrained storage and processing resources in that environment. In the long run, as Physics, Chemistry and Microelectronics progress, the capabilities of such devices are likely to increase. In 20 to 50 years from now, until technology has firm advances, lightweight software solutions will be needed.

In Brazil, the most widely adopted signature protocol, the RSA scheme, is obsolescent as a standard. The problem is that technological advances impose a considerable increase in cryptographic key sizes in order to maintain a suitable security level, bringing about undesirable effects in processing time, bandwidth occupation and storage requirements. As an immediate solution, we have the Elliptic Curve Cryptography which is more suitable for utilization in public agencies and industry.

In the field of elliptic curves, this work contributes specifically with the introduction of a new subfamily of the pairing-friendly Barreto-Naehrig (BN) curves. The proposed subfamily has a computationally simple description, and makes it able to offer opportunities for efficient implementation. The choice of the BN curves is also based on the fact that they allow a range of practical security levels.

Furthermore, there were made practical implementations from the introduced subfamily, like the most basic extension fields algorithms, elliptic curve arithmetic and pairing computation. The adoption of the new BN subfamily with carefully chosen optimization techniques allowed the most efficient implementation of the optimal Ate pairing, which is a very useful operation in many practical cryptographic applications.

Keywords: Cryptology, Algorithms, Algebraic curves, Finite fields, Network security, Bilinear pairings.

SUMÁRIO

LISTA DE ABREVIATURAS

AES	*Advanced Encryption Standard*
BN	*Barreto-Naehrig*
BDHP	*Bilinear Diffie-Hellman problem*
CBI	*Criptografia Baseada em Identidade*
CA	*Certification Authority*
CCA	*Chosen-Ciphertext Attack*
CL-PKC	*Certificateless Public Key Cryptography*
CM	*Complex Multiplication*
DLP	*Discrete Logarithm problem*
ECC	*Elliptic Curve Cryptography*
ECDLP	*Elliptic Curve Discrete Logarithm problem*
GDHP	*Gap Diffie-Hellman problem*
GSM	*Global System for Mobile Communications*
IBE	*Identity Based Encryption*
JSS	*Journal of Systems and Software*
KGC	*Key Generation Center*
MAC	*Message Authentication Code*
MNT	*Miyaji, Nakabayashi and Takano*
NESSIE	*New European Schemes for Signatures, Integrity and Encryption*
NONCE	*Number used ONCE*
PBC	*Pairing-Based Cryptography*
PKG	*Public Key Generator*
RSA	*Rivest Shamir Adleman*
RSSF	Redes de Sensores Sem Fio (*Wireless Sensor Networks*)
SMS	*Short Message Service*

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Dado o crescente cenário de usuários de dispositivos cada vez menores como *Pocket PC*, aparelhos celulares, BlackBerry, iPod, sensores, etc, observam-se muitos trabalhos de pesquisa em segurança de redes para fornecer aplicações satisfatórias. Uma linha de pesquisa que trata de RSSF (Redes de Sensores Sem Fio) é um destes trabalhos. Desenvolver *software* para esse tipo de ambiente demanda extremos cuidados com relação aos poucos recursos existentes.

Alguns exemplos de aplicações em redes com recursos limitados são: SMS (*Short Message Service*) sobre a rede GSM (Global System for Mobile Communications) e monitoração de condições ambientais em RSSFs. São cenários em que existe troca de informações entre usuários ou entre sistemas, onde os elementos da rede se comunicam através de canais públicos. Em todas as aplicações mencionadas existe alguma forma de preocupação com confidencialidade/autenticidade/disponibilidade das informações trocadas, ou seja, com serviços básicos de segurança. Sem tais procedimentos, uma transação financeira via SMS poderia ser fraudada ou até mesmo a estratégia trocada entre sócios de uma empresa via *smart phones* poderia cair nas mãos de adversários mal-intencionados.

Para garantir que a segurança seja preservada, esquemas criptográficos que assegurem todos os serviços mencionados tornam-se vitais.

Desenvolver soluções para tal ambiente exige um importante *trade-off* entre os seguintes items: consumo de memória, nível de segurança, sobrecusto de processamento,

comunicação e consumo de energia. Uma boa solução deveria avaliar cada um desses items. Em particular, é necessário que os algoritmos propostos consumam pouca memória e ao mesmo tempo apresentem razoável desempenho em processadores de baixo *clock* relativo (e.g. 7MHz para o TelosB (CROSSBOW, 2008)). Uma alternativa, o protocolo RSA (RIVEST et al., 1978), cuja segurança é baseada no problema da fatoração inteira, envolve tamanhos maiores de chaves relativamente a curvas elípticas. Em contrapartida, protocolos baseados em curvas elípticas têm sua segurança baseada principalmente no ECDLP (Elliptic Curve Discrete Logarithm problem), permitindo a utilização de chaves menores para um mesmo nível de segurança. A segunda alternativa é o corpo de estudo deste trabalho.

Para ilustrar a comparação entre aplicabilidade, segurança e tamanhos de chave de técnicas distintas de criptografia foi adicionada a Tabela 1.[1]

Nível de Proteção	Simétrico	Assimétrico	ECC	Hash
Proteção a curto prazo para pequenas organizações. Não deveriam ser usadas para confidencialidade em novos sistemas	64	816	128	128
Proteção a curto prazo p/ organizações médias e a médio prazo para pequenas organizações	72	1008	144	144
Chaves do 3DES restritas a 2^{40} *plaintext*/ *ciphertexts*. Proteção até 2012	80	1248	160	160
Legado do nível padrão 2-key 3DES restrito a 10^6 *plaintext*/*ciphertexts*. Proteção até 2020	96	1776	192	192
Proteção a médio prazo do 3-key 3DES, até 2030	112	2432	224	224
Proteção a longo prazo. Recomendação genérica independente da aplicação. Proteção até 2040	128	3248	256	256

Tabela 1: Tamanhos de Chave

Cada uma das linhas da tabela corresponde a um tamanho de chave simétrica e tamanhos de chaves assimétricas construídas de maneira similar como as usadas no projeto NESSIE [2].

Vale ressaltar que muito da pesquisa recente em criptografia de chave pública está voltado para curvas elípticas e emparelhamentos bilineares. Essa combinação vem se

[1] *http://www.keylength.com/en/3

[2] *http://www.cosic.esat.kuleuven.be/nessie

mostrando bastante versátil em aplicações práticas.

Inicialmente, os emparelhamentos bilineares foram usados para atacar o problema do logaritmo discreto em grupos formados por curvas elípticas através do ataque MOV (FREY et al., 1999; MENEZES et al., 1991). Mas foi em meados do ano 2000 que eles tiveram aplicação em criptografia para prover segurança. A primeira solução foi projetada por Joux (JOUX, 2000). Joux avistou uma generalização do protocolo de Diffie-Hellman (1976) a partir da nova ferramenta criptográfica. O protocolo original permite a criação de um segredo comum entre dois participantes e é usado como um dos blocos na construção de muitos outros protocolos. A versão proposta por Joux permitiu, através dos emparelhamentos, o acordo de um valor secreto entre três participantes.

O acordo de um segredo entre três participantes já existia por meio de outras técnicas conhecidas, contudo eram necessárias duas rodadas de comunicação para executá-lo. A técnica de Joux exigia apenas uma rodada. Em alguns protocolos, usar duas rodadas pode ser incômodo. Por exemplo, a troca de *e-mail* para combinar o segredo exige que os dois participantes estejam conectados, a fim de calcularem o segredo num determinado momento, o que é indesejável.

Naquela época conheciam-se dois tipos de emparelhamento – Weil e Tate – até ali usados apenas como ferramentas de criptoanálise para reduzir a complexidade do ECDLP em algumas curvas elípticas consideradas “fracas”.

Pouco tempo depois, no ano de 2003, Boneh e Franklin aplicaram curvas elípticas para propor uma nova instância do esquema IBE (Identity Based Encryption) (BONEH; FRANKLIN, 2001) de Shamir (SHAMIR, 1984). O esquema foi definido sobre grupos de curvas elípticas e emparelhamentos bilineares – primeiramente o emparelhamento de Weil – e continha a prova da segurança contra o ataque CCA (ataque de cifra escolhida). Uma alternativa para IBE foi proposta por Clifford Cocks em 2001. O esquema de Cocks é baseado em pressupostos bastante estudados (o pressuposto da residuosi-

dade quadrática) mas encripta apenas um bit da mensagem por vez, com um alto grau de expansão do texto criptografado. Portanto, é altamente ineficiente e não prático por enviar somente as mensagens mais curtas, tais como uma chave de sessão para uso com uma cifra simétrica.

Em CBI (Criptografia Baseada em Identidade), a chave pública de um usuário pode ser uma *string* escolhida de forma arbitrária, por exemplo, um valor conhecido como sua conta de *e-mail*, número de celular, etc. A partir desta escolha sutil, percebe-se que não é necessário obter o certificado do usuário para o qual se deseja enviar uma mensagem encriptada.

No caso em que a diretora de uma empresa, digamos Alice, deseja enviar informações estratégicas sobre um possível novo cliente para Bob, ela simplesmente encripta sua mensagem através de um protocolo assimétrico com a chave "bob@dominio.com.br". Note que Bob nem mesmo precisa ter obtido sua chave privada previamente. Neste caso Bob a obtém por meio de um centro gerador de chaves privadas, o PKG (Public Key Generator), da mesma forma como era feito com uma CA (Certification Authority). O único inconveniente desse tipo de esquema é a custódia de chaves, já que o PKG pode personificar os usuários, obrigando ser alguém de confiança incondicional.

Boneh e Franklin também apresentam uma aplicação de revogação de chaves. Imagine uma empresa que decidiu que as chaves públicas dos funcionários fossem seu *e-mail* concatenado com o ano atual (e.g. "bob@dominiodacompania.com.br || ano atual"). Uma vez por ano seria gerada uma chave privada para o funcionário Bob. Caso Bob fosse demitido, não mais seriam geradas chaves privadas para ele, não sendo ele capaz de decriptar seus novos *e-mails*.

A aplicabilidade dessas ferramentas não para por aí. Os dois criptógrafos discorrem também sobre a possibilidade de delegação de chaves de decriptação. Suponha que Bob represente o PKG. Ele primeiramente executaria um algoritmo de *setup* a fim

de gerar os parâmetros do sistema – *params* – junto com sua própria chave mestra. Pode-se visualizar *params* como sendo sua chave pública. Bob, então, obtém um certificado através de uma CA relativo à sua chave pública identificada como *params*. Note, também, que Bob é o único a conhecer a chave mestra, portanto não se observa o problema da custódia de chave. Algumas aplicações provenientes do paradigma CBI são listas a seguir:

1. Delegação para um *laptop*. Suponha que Alice encripte um *e-mail* usando a data atual como chave de encriptação (ela usa *params* de Bob como os parâmetros do sistema IBE). Uma vez que Bob tem a chave mestra, ele pode extrair a chave privada correspondente a esta chave de encriptação IBE e, então, decriptar a mensagem. Agora, suponha que Bob se ausente por uma viagem de sete dias. Normalmente, Bob armazenaria sua chave privada em seu laptop. Se o *laptop* for roubado a chave privada é comprometida. Por outro lado, ao adotar o sistema IBE, Bob poderia simplesmente instalar no seu *laptop* sete chaves privadas correspondendo aos sete dias da viagem. Se o *laptop* for roubado, apenas as chaves privadas para aqueles sete dias são comprometidas. A chave mestra está fora de perigo (BONEH; FRANKLIN, 2001, Seção 1.1.2).

2. Delegação de tarefas. Suponha que Alice encripte um *e-mail* para Bob usando o tipo do assunto como a chave de encriptação IBE. Bob pode decriptar o *e-mail* usando sua chave mestra. Agora, suponha que Bob tenha várias assistentes, cada uma delas responsável por uma tarefa diferente (e.g. uma cuida das “compras”, outra cuida dos “recursos humanos”, etc.). Bob fornece uma chave privada correspondente à responsabilidade de cada uma delas. Cada assistente pode então decriptar mensagens cujo assunto recaia nas suas responsabilidades, mas não se pode decriptar mensagens destinadas às outras. Note que Alice apenas obtém uma única chave pública de Bob (*params*), e usa aquela chave pública para enviar *e-mail* contendo qualquer tipo de assunto de sua escolha. O *e-mail* pode

apenas ser lido pela assistente responsável por aquele assunto.

1.1 Justificativa

Pesquisas recentes focaram em certas curvas elípticas individuais para atingir ganhos excepcionais de desempenho (BEUCHAT et al., 2010; NAEHRIG et al., 2010). Isso é essencial, uma vez que emparelhamentos são normalmente a operação mais cara computacionalmente em qualquer esquema criptográfico que utiliza emparelhamento. Por outro lado, pode-se argumentar que visar somente emparelhamentos rápidos é insuficiente e pode levar a ineficiências inoportunas ou inaceitáveis em plataformas limitadas. De fato, por conta do alto custo intrínseco dos emparelhamentos, muitos protocolos já são desenvolvidos baseando-se neles apenas quando as partes correspondentes do protocolo são maiores recursos computacionais (e.g. servidor ou clusters) enquanto as partes limitadas precisam efetuar apenas operações mais baratas (BARRETO et al., 2005; BONEH et al., 2003; LIBERT; QUISQUATER, 2005; ZHANG et al., 2004). Em tais cenários, parâmetros levando a emparelhamentos mais rápidos, com o preço de deteriorar o desempenho em alguma outra parte, seria prejudicial em vez de útil.

Uma linha de pesquisa distinta trata de obter curvas parametrizadas com certas propriedades previstas a fim de se evitar futuros testes computacionalmente caros durante a geração da curva ou, mais importante, evitar o teste de parâmetro da curva, requerido para detectar alguns tipos de ataques (e.g. verificar se a suposta curva BN contida em um dado certificado digital de fato apresenta as propriedades esperadas antes de usar aquele certificado). Este procedimento é trivial para curvas não amigáveis a emparelhamento, mas a proposta especial da natureza das curvas BN agrava a quantidade de cálculos necessários. Adotando-se uma curva onde certas propriedades são satisfeitas, o sobrecusto de testar seria bastante reduzido, e poderia ser efetuado em plataformas muito mais simples; e.g. um servidor de certificados leve precisaria apenas de aritmética inteira pura para verificação de primalidade (e nenhum suporte a

aritmética de curva elíptica) para atestar a consistência correta das curvas.

Construir o *twist* correto da curva sobre o corpo base $\mathbb{F}_p$ sem recorrer à aritmética de curva elíptica tem sido feito com sucesso (COHEN et al., 2006, Seção 13.1.5), (RUBIN; SILVERBERG, 2010).

Em contraste, as tarefas relacionadas com a escolha de representações apropriadas para todos os corpos envolvidos (os quais são normalmente escolhidos de antemão, baseados em *features* das bibliotecas de suporte e esquecidos pela natureza peculiar das curvas BN) e selecionar o *twist* correto no corpo de extensão $E'(\mathbb{F}_{p^2})$ tem tido pouca atenção na literatura e parece ainda necessitar de testes de caráter quadrático/cúbico nos corpos de extensão e de aritmética de grupo completa naquele *twist*.

1.2 Objetivos

Para solucionar os problemas descritos, esta dissertação analisa escolhas de curvas elípticas apropriadas para beneficiar as operações de aritmética elíptica, a otimização da operação de emparelhamento em si e amenizar os testes de parâmetros de curva.

Em particular, buscam-se aspectos teóricos de parametrização para a já consolidada família de curvas elípticas BN, amigáveis a emparelhamento, e que resultem em operações mais eficientes comparativamente a outros trabalhos na literatura. Tal família de curvas é conhecida até o momento por ser efetiva no desenvolvimento de protocolos de chave pública bem como de acordo de chaves.

A dissertação é idealizada de forma genérica, i.e. independente do *hardware* da plataforma alvo ou do protocolo utilizado, mas de forma que, para a implantação em uma arquitetura contendo dispositivos com recursos limitados, pode-se oferecer melhor desempenho tanto na parte pesada (servidor executando emparelhamentos) quanto na parte que executa as operações “mais leves”.

1.3 Metodologia

Uma vez que o papel central deste trabalho é fornecer uma parametrização especificada para implementação eficiente para as curvas da família BN, o método de avaliação do trabalho é em sua maior parte qualitativo, mas não deixando de lado o ponto de vista quantitativo.

Este é avaliado partindo da abrangência de oportunidades de "otimização" na aritmética elíptica e de corpos finitos dos grupos $\mathbb{G}_1$ e $\mathbb{G}_2$, que são gerados pela curva E e por seu *twist* E', respectivamente. As "otimizações" em questão se referem à viabilidade de se aplicar o melhor algoritmo conhecido para cada operação. Por exemplo, se a parametrização fornecida permite o uso do algoritmo de Cipolla-Lehmer [(CIPOLLA, 1903), (LEHMER, 1969),(RIESEL, 1985, pp. 287-288)] para extração de raiz quadrada modular, que é um dos mais eficientes conhecidos até o momento. Tais oportunidades dependem das estruturas geradas a partir da escolha dos parâmetros da curva.

No caso da implementação do emparelhamento Ate ótimo, o mais eficiente conhecido quando se utiliza uma curva BN, visa-se a análise quantitativa considerando como métrica de comparação com outros trabalhos como (ARANHA et al., 2011; BEUCHAT et al., 2010; NAEHRIG et al., 2010) o número equivalente de multiplicações, o número de reduções modulares e o número de adições no corpo base $\mathbb{F}_p$. Esse conjunto de métricas torna a comparação entre diferentes implementações independente da plataforma para a qual foram desenvolvidas.

As linguagens de programação usadas na análise prática são Magma, para se obter o parâmetro BN u da curva e Java para se implementar as estruturas de corpos finitos e suas extensões, aritmética elíptica e as funções de emparelhamentos bilineares.

Como aplicação de motivação, é implementado o protocolo BDCPS de cifrassinatura (cifração + assinatura), que é baseado em emparelhamento. Para efeito de comparação, o protocolo é implementado sobre ambos uma curva da subfamília BN proposta

e outra pertencente à família de curvas MNT.

1.4 Contribuições Originais

Este trabalho traz contribuições através da definição de uma subclasse de curvas elípticas da família BN (bastante grande) que é particularmente apropriada para construção/validação bem como implementação eficientes.

A parametrização em análise fornece automaticamente a curva *twist* correta $E'(\mathbb{F}_{p^2})$, o que é demonstrado através de teorema. Esse resultado dispensa a execução de testes de *twist*, anteriormente necessários (HESS et al., 2006, Seção 4.1), e dá às representações de corpo uma unidade geral que fornece melhores oportunidades de otimização. A família proposta tem intersecções com outras famílias interessantes que ocorrem na literatura (e.g. (NOGAMI et al., 2008; SHIRASE, 2010)), oferecendo benefícios adicionais naqueles casos.

O artigo resultante deste trabalho, intitulado "**A Family of Implementation-Friendly BN Elliptic Curves**" (PEREIRA et al., 2011), foi submetido e aceito no JSS (*Journal of Systems and Software*), um *journal* internacional que, na data de publicação, alcançava o mais alto extrato Qualis A1 na área Engenharias IV de acordo com a CAPES.

Salienta-se também que as técnicas propostas podem ser úteis para a obtenção de configurações otimizadas de outras classes de famílias amigáveis a emparelhamento, como a escolha das representações dos corpos de extensão.

Um exemplo de curva proposta foi implementado e está descrito na Seção 4.4, onde foi dada atenção especial à otimização da operação mais cara consumida pelos protocolos, o emparelhamento. A implementação na linguagem Java está disponível *online*[3]. Neste caso, o conjunto de parâmetros utilizado permitiu a quebra do recorde

[3] http://code.google.com/p/bnpairings/

mundial deste cálculo para curvas da família BN, cujo posto anterior pertenceu a Beuchat *et al*. (BEUCHAT et al., 2010) destacando que sua implementação não se utiliza da subfamília amigável à implementação indicada neste trabalho.

Também é apresentada neste trabalho uma errata do protocolo de cifrassinatura digital BDCPS, que foi publicado pelo autor e colaboradores no simpósio SBSeg'08.

1.5 Organização

O restante deste documento organiza-se da seguinte forma. O Capítulo 2 elenca uma série de conceitos e notações matemáticas necessários para a construção do arcabouço das curvas elípticas propostas, bem como para o emparelhamento. As técnicas mais importantes do ponto de vista prático são descritas e outras mais teóricas têm sua referência indicada.

Em seguida, no Capítulo 3 é então apresentada a contribuição deste trabalho, descrita sua escolha e as vantagens por ela fornecidas. Pode-se ver no Capítulo 4 a análise das decisões tomadas através de implementação prática e comparação com outros trabalhos. O trabalho é concluído no Capítulo 5.

O Apêndice A diferencia a contribuição aqui apresentada de um trabalho recente de Shirase *et al.* (SHIRASE, 2010) que tem intersecção, mas é mais restrito do que a família de curvas aqui proposta. O subsequente Apêndice B apresenta a correção do protocolo BDCPS anteriormente mencionado.

Pode-se observar que, no Apêndice C, são elencados os principais algoritmos usados na otimização do cálculo do emparelhamento Ate ótimo. Por fim, os artigos produzidos pelo autor até o momento são enumerados no Apêndice D.

2 FUNDAMENTOS MATEMÁTICOS

Para se implementar a função de emparelhamento de forma eficiente é necessário analisar e tomar decisões relacionadas às suas estruturas mais básicas constituintes – os grupos abelianos e os corpos finitos – bem como às estruturas de mais alto nível – os corpos de extensão e a aritmética de pontos sobre curvas elípticas. A escolha dos corpos e sua parametrização influenciam decisivamente nas oportunidades de otimização. Logo, estas estruturas devem ser detalhadamente analisadas.

Este capítulo tem por objetivo introduzir os conceitos matemáticos utilizados na implementação completa do emparelhamento bilinear Ate ótimo, sendo que algumas das decisões também podem ser utilizadas na implementação de outros tipos de emparelhamento. A Seção 2.1 introduz o conceito genérico de grupo abeliano, que instanciado sobre pontos de uma curva elíptica consistirá o domínio da função do emparelhamento. Logo depois, são apresentados nas Seções 2.2 e 2.3 os conceitos de corpo finito e de extensão, também conhecidos como corpos de *Galois*, cujos elementos são as coordenadas (x, y) dos pontos de uma curva elíptica bem como os valores do conjunto imagem da função de emparelhamento.

2.1 Grupos abelianos

Definição 1. *(SHOUP, 2004, Definição 6.1)(Grupo Abeliano) Um grupo abeliano é um conjunto G com uma operação binária $*$ associada, tal que*

1. *($*$ é associativa) para todo $a, b, c \in G$, $a * (b * c) = (a * b) * c$;*

2. *($*$ tem identidade) existe $e \in G$ tal que para todo $a \in G, a * e = a = e * a$;*

3. *($*$ tem inverso) para todo $a \in G$ existe $a' \in G$ tal que $a * a' = e = a' * a$;*

4. *($*$ é comutativa) para todo $a, b \in G, a * b = b * a$;*

Os grupos podem ser *cíclicos*, onde são definidos uma *ordem n*, que é o número de elementos do grupo, e um *elemento gerador*. Para gerar todo o grupo, basta somente o seu gerador. Qualquer elemento g do grupo pode gerar um novo grupo que é subgrupo do grupo original.

Definição 2. *(LIDL; NIEDERREITER, 1983, Definição 1.3)(Grupo Cíclico) Um grupo multiplicativo G é denominado cíclico se existe um elemento $a \in G$ tal que para qualquer elemento $b \in G$ há algum inteiro j com $b = j * a$. Tal elemento a é denominado gerador do grupo cíclico, e escreve-se $G = \langle a \rangle$.*

Por exemplo, se $G = (g^0, g^1, g^2, g^3, g^4, g^5)$ é um grupo, então $g^6 = g^0$, e G é cíclico. De fato, G é essencialmente isomorfo (o mesmo que) ao conjunto $\mathbb{Z}_6$, ou seja, $(0, 1, 2, 3, 4, 5)$ com adição módulo 6. Por exemplo, $1 + 2 = 3 \pmod 6$ corresponde a $g^1 \cdot g^2 = g^3$, e $2 + 5 = 1 \pmod 6$ corresponde a $g^2 \cdot g^5 = g^7 = g^1$, e assim por diante. Esse paralelo entre o grupo G acima e $\mathbb{Z}_6$ pode ser feito através do isomorfismo f definido por $f(g^i) = i$.

Definição 3. *(Subgrupo de Torção) Na teoria de grupos abelianos, o subgrupo de* torção $\mathbb{G}_T$ *de um grupo abeliano G é o subgrupo de G que consiste de todos os elementos que têm ordem finita, i.e.*

$$\mathbb{G}_T = \{g \in G \mid n \cdot g = 0\}$$

onde n é a ordem do elemento g. Um grupo abeliano G é chamado de um grupo de torção ou periódico se cada elemento de G tem ordem finita e é chamado livre de torção se cada elemento de G exceto a identidade tem ordem infinita.

Definição 4. *(Subgrupo de Torção p-potente) Para qualquer grupo abeliano G e qualquer número primo p, o conjunto $\mathbb{G}_{T_p}$ dos elementos de G que têm como ordem uma potência de p é um subgrupo denominado subgrupo de torção p-potente ou, abreviadamente, o subgrupo de p-torção*

$$\mathbb{G}_{T_p} = \{g \in G \mid \exists n \in \mathbb{N},\ p^n \cdot g = 0\}.$$

A partir do conceito de grupo, aliado a algumas propriedades adicionais, é possível especificar um corpo finito.

2.2 Corpos finitos ou de *Galois*

Para se construir uma curva elíptica é preciso previamente escolher o corpo finito sobre o qual ela é definida. Um corpo finito é denotado por $\mathbb{F}_q$ ou $GF(q)$ onde q é uma potência de um número primo p, $q = p^k$. Neste trabalho, sempre que $k = 1$, o corpo é denotado simplesmente como $\mathbb{F}_p$.

Os corpos são abstrações de sistemas numéricos familiares (tais como os números racionais $\mathbb{Q}$, os números reais $\mathbb{R}$ e os números complexos $\mathbb{C}$) e suas propriedades essenciais. Sua definição formal é dada a seguir:

Definição 5. *(HANKERSON et al., 2003, Seção 2.1)(Corpo) Um corpo consiste de um conjunto $\mathbb{F}$ juntamente com duas operações, adição (denotada por +) e multiplicação (denotada por ·), que satisfazem as seguintes propriedades aritméticas usuais:*

i. O conjunto $(\mathbb{F}, +)$ é um grupo abeliano com identidade (aditiva) denotada por 0.

ii. O conjunto $(\mathbb{F}\backslash\{0\}, \cdot)$ é um grupo abeliano com identidade (multiplicativa) denotada por 1.

iii. A lei distributiva vale: $(a + b) \cdot c = a \cdot c + b \cdot c$ para todo $a, b, c \in F$.

Se o conjunto $\mathbb{F}$ é finito, então o corpo é denominado *finito*.

O primo q é denominado a característica de $\mathbb{F}_q$ e tem a seguinte definição:

Definição 6. *(LIDL; NIEDERREITER, 1983, Definição 1.43)(Característica de um corpo finito) Se $\mathbb{F}_q$ é um corpo arbitrário e existe um inteiro positivo n tal que $n \cdot r = 0$ para todo $r \in \mathbb{F}_q$, então o menor inteiro n é denominado a característica de $\mathbb{F}_q$, $char(\mathbb{F}_q)$. Se tal n não existe, diz-se que $\mathbb{F}_q$ tem característica 0 (q não seria um primo e $\mathbb{F}_q$ não seria um corpo finito ou de* Galois *de ordem prima).*

Em particular, corpos de característica $p = 2$ são denominados *corpos binários*; para $p = 3$, têm-se os *corpos ternários* e corpos com valores maiores de p chamamos de *corpos primos*.

Destaca-se que uma boa escolha do primo p é extremamente importante, uma vez que pode tornar algumas das operações mais eficientes. Um exemplo é escolhermos $p \equiv 4 \pmod 9$ para acelerar a operação de extração de raiz cúbica (BARRETO; NAEHRIG, 2006, Seção 3.1)

Por fim, é descrito um importante teorema com finalidades teóricas e práticas

Teorema 1. *(Teorema da Raiz Primitiva) Seja q um número primo. Então existe um elemento $g \in \mathbb{F}_q^*$ cujas potências fornecem cada elemento de $\mathbb{F}_q^*$, i.e.,*

$$\mathbb{F}_q = \{1, g, g^2, g^3, \ldots, g^{q-2}\}. \tag{2.1}$$

Os elementos com esta propriedade são denominados raízes primitivas de $\mathbb{F}_q$ ou geradores de $\mathbb{F}_q^*$. Eles são elementos de $\mathbb{F}_q^*$ cuja ordem é $q - 1$.

2.3 Corpos de extensão

A partir do corpo finito $\mathbb{F}_p$ de p elementos, é possível construir um corpo de p^k elementos para qualquer $k > 1$. Esse corpo, denotado por $\mathbb{F}_{p^k}$ é chamado corpo de

extensão (ou corpo estendido) de grau k do corpo $\mathbb{F}_p$, e é único a menos de isomorfismo (AVANZI; MIHAILESCU, 2004).

Corpos de extensão podem ser construídos a partir de um polinômio irredutível $f(X)$ com grau k. Em analogia, o grau do polinômio seria o grau de extensão do corpo k e seus coeficientes seriam elementos no corpo base. O polinômio $f(X)$ com grau de extensão k pode ser então representado na forma

$$f(X) = a_k X^k + a_{k-1} X^{k-1} \ldots + a_1 X + a_0 \; com \; a_i \in \mathbb{F}_p, i \in \mathbb{N}$$

e, portanto, $f(X) \in \mathbb{F}_p[X]$. Se $f(X)$ for um polinômio irredutível de grau k, então $\mathbb{F}_{p^k} \cong \mathbb{F}_p[X]/f(X)$.

As operações usuais de soma e multiplicação são agora tomadas módulo o polinômio irredutível, $x \oplus y \mapsto x + y \pmod{f(X)}$.

Definem-se os *conjugados* de $a \in \mathbb{F}_{p^k}$ como os elementos a^{p^i}, $0 \leqslant i < k$. A norma de $a \in \mathbb{F}_{p^k}$ é o produto de todos os seus conjugados, $|a| := \prod_i a^{p^i}$.

Sempre que $p \equiv 3 \pmod 4$ o corpo finito $\mathbb{F}_{p^2}$ pode ser representado por $\mathbb{F}_p[i]/(i^2 + 1)$, como nos números complexos. Em analogia, o conjugado não trivial do elemento do corpo $f = \alpha + i\beta \in \mathbb{F}_{p^2}$ é $\bar{f} = f^p = \alpha - i\beta$.

Também é de interesse o conceito das *raízes n-ésimas da unidade*, ou números de *De Moivre*, que são todos os números complexos $\alpha \in \mathbb{C}$ que resultam em 1 quando são elevados a uma potência dada n, $\alpha^n = 1$, ou seja, $\alpha = 1^{1/n}$.

Para se aplicar algoritmos eficientes de exponenciação em um corpo de extensão com uma forma específica, precisamos do conceito de subgrupo ciclotômico. Este, por sua vez, utiliza a definição de polinômio ciclotômico e são dados a seguir.

Definição 7. *(Subgrupo ciclotômico) Um subgrupo ciclotômico do corpo de extensão amigável à* torre $\mathbb{F}_{p^k}$ *com* $k = 2^a 3^b$, $a, b \geq 1$ *é um subgrupo de ordem* $\Phi_k(p)$. *O polinômio* Φ_k *é o k-ésimo polinômio ciclotômico, o qual, quando* $6 \mid k$, *para k com a*

forma descrita, é sempre da forma

$$\Phi_{2^a 3^b}(x) = x^{2\cdot 2^{a-1}3^{b-1}} - x^{2^{a-1}3^{b-1}} + 1. \tag{2.2}$$

Denota-se o subgrupo ciclotômico por $G_{\Phi_k(p)}$, sendo o conjunto formado da seguinte forma

$$G_{\Phi_k(p)} = \{\alpha \in \mathbb{F}_{p^k} \mid \alpha^{\Phi_k(p)} = 1\} \tag{2.3}$$

2.4 Curvas elípticas e emparelhamentos bilineares

Na década de 80, Neal Koblitz e Victor Miller (KOBLITZ, 1987; MILLER, 1986) perceberam independentemente que criptossistemas baseados no DLP (*Discrete Logarithm Problem*) podem fornecer segurança maior quando definidos no grupo formado pelos pontos em uma curva elíptica em vez do grupo multiplicativo convencional de um corpo finito. Pode-se interpretar a partir desse fato que curvas elípticas poderiam possibilitar o uso de chaves mais compactas, e ao mesmo tempo fornecer um nível de segurança similar. Desde então, muitos esforços em pesquisas em ECC (*Elliptic Curve Cryptography*) foram feitos e grande quantidade de criptossistemas foram propostos. Alguns deles, contudo, provados menos seguros do que originalmente se supunha, uma vez que a estrutura das curvas propostas forneciam meios de ataque ao sistema.

Ao criarem a criptografia de curvas elípticas, nem Koblitz nem Miller se preocuparam em registrar uma patente acerca do assunto. Assim, a ideia básica de ECC se tornou disponível e grátis para uso geral. Às vistas dessa oportunidade, os criptógrafos Scott Vanstone e Ron Mullin fundaram a empresa de soluções de segurança baseadas no uso de ECC, a Certicom. Tanto a Certicom[1], com cerca de 350 patentes, quanto outras empresas, registraram patentes ao adicionar melhorias à ideia básica de ECC.

[1] http://www.certicom.com

2.4.1 Curvas elípticas

Um grande número de protocolos criptográficos são definidos genericamente sobre grupos cíclicos, independentemente do conjunto sobre o qual está definido o grupo.

Os grupos cíclicos também podem ser definidos sobre curvas elípticas cujas operações e elementos estão associados àquele domínio. Uma operação de soma é completamente diferente de uma operação de soma usual módulo um número primo. Contudo, ela satisfaz os axiomas de um grupo abeliano de comutatividade, elemento neutro, etc. Após a definição formal das curvas elípticas relevantes a este trabalho, é definido o grupo abeliano construído a partir dessas curvas.

Definição 8. *Uma curva elíptica E sobre um corpo $\mathbb{F}_q$ é definida através da equação:*

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.4}$$

onde $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q$ e $\Delta \neq 0$. O discriminante Δ serve para indicar se a equação da curva $E/(\mathbb{F}_q)$ corresponde ou não a uma equação cúbica não singular. Ser não singular indica que a equação cúbica $y^2 = f(x)$ não possui raízes repetidas e $\Delta \neq 0$ é suficiente e necessário para garantir tal requisito (HUSEMÖLLER, 2004, Remarks 2.1 e 3.5). O discriminante Δ é definido como se segue:

$$\begin{cases} \Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \\ d_2 = a_1^2 + 4a_2 \\ d_4 = 2a_4 + a_1a_3 \\ d_6 = a_3^2 + 4a_6 \\ d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2. \end{cases} \tag{2.5}$$

Pode-se também definir uma curva elíptica E sobre um corpo finito $\mathbb{F}_q$ ($q \geq 3$) como o conjunto das soluções em $\mathbb{F}_q \times \mathbb{F}_q$ da equação de Weierstrass

$$y^2 = x^3 + ax + b \tag{2.6}$$

onde $a, b \in \mathbb{F}_q$ e $\Delta = -16(4a^3 + 27b^2) \neq 0$, e às soluções (x, y) da equação, com $x, y \in \mathbb{F}_q$, adiciona-se o ponto no infinito O.

O conjunto

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 \mid y^2 = x^3 + ax + b\} \cup \{O\} \tag{2.7}$$

é chamado o conjunto dos pontos racionais $\mathbb{F}_q$ sobre a curva E. Denomina-se coordenada afim quando um ponto é representado por apenas duas coordenadas (x, y)[2]. O conjunto $E(\mathbb{F}_q)$ forma um grupo abeliano. Escreve-se + para a lei de grupo, ou seja, os pontos em uma curva elíptica formam um grupo aditivo. O elemento neutro é o ponto no infinito O.

O número de pontos de uma curva $E(\mathbb{F}_q)$, denotado por $\#E(\mathbb{F}_q)$, denomina-se a *ordem* da curva e é representada pela letra n. Note que a ordem da curva é o número de soluções da sua equação no corpo subjacente. Também se denota a ordem de um ponto P, pertencente à curva, como o menor inteiro positivo r tal que somando-se o ponto P, r vezes, segundo a lei de soma de uma curva elíptica, resulta o ponto no infinito, ou seja, $[r]P = O$. A partir desse fato, é possível definir o subgrupo de torção (veja Definição 4) de uma curva elíptica, que consiste no conjunto de pontos de ordem finita e múltipla de r, denotado por $E(\mathbb{F}_q)[r] = \{P \in E(\mathbb{F}_q) \mid [r]P = O\}$.

A quantidade de pontos em uma curva elíptica E sobre um corpo $\mathbb{F}_q$ é um conjunto finito próximo do número de elementos q do corpo, segundo o seguinte teorema (Hasse, 1930):

Teorema 2. *(Hasse) Seja E uma curva elíptica definida sobre o corpo $\mathbb{F}_q$ de q elementos, então*

$$\#E(\mathbb{F}_q) = q + 1 - t_q \quad com\ |t_q| \leq 2\sqrt{q} \tag{2.8}$$

[2]Mais adiante, para fins de eficência de implementação, é introduzida uma nova coordenada z

Definição 9. *A quantidade t_q escrita como*

$$t_q = q + 1 - \#E(\mathbb{F}_q) \tag{2.9}$$

é denominada o traço de Frobenius de $E(\mathbb{F}_q)$.

Reescrevendo-se a equação 2.8 de outra forma, temos

$$q + 1 - 2\sqrt{q} \leq n \leq q + 1 + 2\sqrt{q} \tag{2.10}$$

verifica-se que a ordem da curva, n, é da magnitude do tamanho do corpo, q.

2.4.2 Soma de pontos em uma curva elíptica

A adição de dois pontos P e Q sobre uma curva elíptica E na forma reduzida de Weierstrass 2.6 obedece a seguinte regra: o ponto $R = P + Q$ é obtido traçando-se uma reta r que passa por P e Q. A reta r intersecta o gráfico da curva E em um terceiro ponto denotado por $-R$. O sinal negativo significa que o ponto $-R$ é o reflexo do ponto desejado R em relação ao eixo Ox. Note que o ponto $-R$ está a mesma distância que R em relação ao eixo Ox, pois o gráfico da curva E é simétrico em relação a tal eixo. Então, se $-R$ tem coordenadas (x_R, y_R), o ponto R terá coordenadas $(x_R, -y_R)$. Observe que no caso em que $P = Q$, a reta r tangencia E nesse ponto e encontra E em apenas dois pontos: $P = Q$ e $-R$. O ponto R é obtido da mesma forma através da reflexão de $-R$ no eixo Ox.

Usando a regra da soma descrita, em que resulta a adição de dois pontos P e $-P$ pertencentes a uma mesma reta r vertical? A resposta é o ponto no infinito O, apresentado na Equação 2.7. Diz-se que se traçarmos a reta ligando P e $-P$, tal reta não encontra a curva E novamente, ou seja, a reta r encontra E no infinito. Então, temos que $R = P + (-P) = O$.

O cálculo da soma de pontos em uma curva elíptica é descrito pelo Algoritmo 1

Algoritmo 1 Algoritmo de soma de pontos de uma curva elíptica

Entrada: P_1 e P_2 pertencentes à curva E

1: Se $P_1 = O$, então $P_1 + P_2 = P_2$

2: Senão, se $P_2 = O$, então $P_1 + P_2 = P_1$

3: Senão, escreva $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$

4: Se $x_1 = x_2$ e $y_1 = -y_2$, então $P_1 + P_2 = O$

5: Senão, faça

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & se \quad P_1 \neq P_2, \\ \dfrac{3x_1^2 + a}{2y_1} & se \quad P_1 = P_2 \end{cases}$$

6: Compute $x_3 = \lambda^2 - x_1 - x_2$

7: Compute $y_3 = \lambda(x_1 - x_3) - y_1$

Saída: $P_1 + P_2 = (x_3, y_3)$

As fórmulas simplificadas para as coordenadas resultantes x_3 e y_3 no caso da duplicação de ponto ($P_1 = P_2$) são listadas abaixo.

$$\begin{aligned} x_3 &= \left(\frac{3x_1^2 + a}{2y_1}\right) - 2x_1, \\ y_3 &= \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 \end{aligned} \tag{2.11}$$

Note que o cálculo de x_3 e y_3 envolve uma divisão que, na prática, é vista como uma inversão no corpo $\mathbb{F}_q$. Para tamanhos grandes da característica q do corpo, como por exemplo $\lceil \lg p \rceil = 160$ bits, a inversão se torna uma operação cara. Visando implementações eficientes, evita-se tal inversão através da introdução de uma nova coordenada z a cada ponto de coordenadas (x, y).

2.4.3 Representação de ponto em coordenadas projetivas

Para se evitar inversões em corpos finitos nas operações de soma e duplicação de pontos em uma curva elíptica foi introduzido o conceito de coordenadas projetivas.

Existe mais de uma classe dessas coordenadas, contudo, neste trabalho serão usadas as coordenadas projetivas Jacobianas e homogêneas padrão, que possibilitam na prática uma operação de duplicação de ponto mais eficiente quando comparada a outros tipos de coordenadas. Ao ponto com coordenadas usuais x, y é adicionada uma nova coordenada z da seguinte forma: faz-se a mudança de variáveis, $x \mapsto \frac{x_{\text{proj}}}{z^2}$ e $y \mapsto \frac{y_{\text{proj}}}{z^3}$ com $z \neq 0$ na Equação 2.6 obtendo-se a seguinte equação projetiva

$$y_{\text{proj}}^2 = x_{\text{proj}}^3 + a x_{\text{proj}} z^4 + b z^6 \tag{2.12}$$

e temos o novo ponto (x, y, z) em coordenadas projetivas. Note que quando $z = 1$, as coordenadas x_{proj} e y_{proj} correspondem a seus valores afins.

Exemplo 1: Seja a curva elíptica $E : y^2 = x^3 + 7x + 5$ sobre o corpo $\mathbb{F}_{17}$ e o ponto $P = (3, 6)$ representado em coordenadas afins. A equação projetiva de E é dada por $E_{\text{proj}} : y_{\text{proj}}^2 = x_{\text{proj}}^3 + 7x_{\text{proj}} z^4 + 5z^6$. Pode-se observar diretamente que o ponto $Q = (3, 6, 1)$ é um ponto projetivo, pois pertence a E_{proj}. Para o caso em que $z \neq 0$ e $z \neq 1$, por exemplo, $z = 2$, obtemos a seguinte curva $E_{\text{proj}} : y_{\text{proj}}^2 = x_{\text{proj}}^3 + 10x_{\text{proj}} + 14$. Um ponto dessa curva poderia ser encontrado através da mudança de variáveis descrita, mas agora, no sentido inverso, $x_{\text{proj}} \mapsto x \cdot z^2$ e $y_{\text{proj}} \mapsto y \cdot z^3$. Logo, um ponto, convertido para coordenadas projetivas, seria $R = (x \cdot z^2, y \cdot z^3, z) = (3 \cdot 2^2, 6 \cdot 2^3, 2) = (12, 14, 2)$. □

Com o conceito de coordenadas projetivas, é possível reescrever as fórmulas da soma (e duplicação) de pontos evitando inversões. Fazendo-se a mesma mudança de variáveis $x \mapsto \frac{x_{\text{proj}}}{z^2}$ e $y \mapsto \frac{y_{\text{proj}}}{z^3}$ nas Equações 2.11 obtemos as novas coordenadas $(x_3, y_3, z_3)_{\text{proj}}$, isentas de inversão.

$$\begin{aligned} x_3 &= (3x_1^2 + az_1^4)^2 - 8x_1 y_1^2 \\ y_3 &= (3x_1^2 + az_1^4)^2 (4x_1 y_1^2 - x_3) - 8y_1^4 \\ z_3 &= 4y_1^2 z_1^2 \end{aligned} \tag{2.13}$$

No cálculo de emparelhamento, por exemplo, é possível poupar um grande número

de inversões nos corpos finitos, através da manutenção dos pontos representados na sua forma projetiva na maior parte do tempo.

Existe uma operação de normalização de ponto que faz a conversão de um ponto projetivo para um ponto afim. Para o ponto R do Exemplo 1, representado em coordenadas projetivas Jacobianas, sua normalização é feita através de multiplicação de cada uma de suas coordenadas pelos inversos quadrado e cúbico de z, respectivamente, como segue $R_{norm} = (x \cdot z^{-2}, y \cdot z^{-3}, 1) = (6 \cdot 2^{-2}, 12 \cdot 2^{-3}, 1) = (3, 6, 1)$. Note que a coordenada z não precisa ser multiplicada pois sempre resultará no elemento identidade do corpo, 1.

Mais algumas definições importantes sobre pontos de curvas elípticas são listadas abaixo:

Definição 10. *(HUSEMOLLER, 1987, Definição 1.1)(Mapa ou endomorfismo de Frobenius) Seja uma curva elíptica E definida sobre um corpo finito $\mathbb{F}_q$. O endomorfismo de Frobenius $\phi_E : E \to E$ é dado por $\phi_A(x, y) = (x^q, y^q)$.*

Definição 11. *(Invariante j) Seja a curva elíptica E definida sobre o corpo $\mathbb{F}_q$ com equação $y^2 = x^3 + ax + b,\ a, b \in \mathbb{F}_q$ e $char(\mathbb{F}_q) \neq 2, 3$.*

Admita também a seguinte mudança de variáveis

$$x_1 = \mu^2 x,\ y_1 = \mu^3 y,\ \mu \in \mathbb{F}_p^*. \tag{2.14}$$

Ao aplicarmos a mudança 2.14 à E, obtém-se a curva $E' : y_1^2 = x_1^3 + a_1 x_1 + b_1$, com $a_1 = \mu^4 a$, $b_1 = \mu^6 b$. O invariante j de E é definido através da expressão

$$j = j(E) = 1728 \frac{4a^3}{4a^3 + 27b^2}$$

Logo, aplicando-se a mudança de variáveis fornecida pela Equação 2.14 em E, e calculando o novo invariante $j(E')$, temos

$$j(E') = 1728 \frac{4(\mu^4 a)^3}{4(\mu^4 a)^3 + 27(\mu^6)^2} = j(E)$$

Observa-se, então, que o invariante j não se altera para o isomorfismo proposto.

2.4.4 Curvas Barreto-Naehrig ou BN

As curvas BN (BARRETO; NAEHRIG, 2006) são uma subfamília de curvas elípticas ordinárias da família cuja equação é obtida fazendo-se $a = 0$ na Equação de Weierstrass 2.6 e parametrizando-se os primos p e n como é descrito a seguir

$$E_b : y^2 = x^3 + b, \tag{2.15}$$

e elimina-se o b subscrito de E_b, escrevendo-se simplesmente E quando o coeficiente específico da equação b é irrelevante para discussão ou está claro no contexto.

A ordem de E deve ser prima e calculada em função de um inteiro u através da expressão $n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1$.

A característica p do corpo finito relacionado $\mathbb{F}_p$ é definida em função do mesmo parâmetro u, segundo a expressão $p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ também prima para algum $u \in \mathbb{Z}$.

O traço de Frobenius da curva pode ser obtido através da aplicação da Equação 2.9. Podemos reescrevê-la, apenas modificando as notações, como $t(u) = p(u) + 1 - n(u)$, obtendo $t(u) = 6u^2 + 1$.

Uma vez que curvas BN têm seu invariante j igual a 0, é relativamente fácil encontrá-las quando comparadas a outras famílias de curvas amigáveis a emparelhamento (veja (FREEMAN et al., 2010) para uma consulta extensiva). Em particular, para se encontrar curvas do tipo BN, não há necessidade de recorrer ao método CM (Complex Multiplication) explicitamente como é feito para as curvas também ordinárias e amigáveis a emparelhamento da família MNT, que é apresentada na Seção 2.4.5.

Para encontrar curvas BN, escolhem-se inteiros de tamanho apropriado para u até

que ambos os polinômios $n(u)$ e $p(u)$ resultem em números inteiros primos. Em seguida, testam-se valores para o coeficiente b até que a curva tenha a ordem correta dentre as ordens possíveis. Note que este último passo era necessário antes deste trabalho.

Para os passos descritos acima, testes de primalidade são necessários, possivelmente testes de caráter quadrático e cálculo de raízes cúbicas em $\mathbb{F}_p$ para se obter um ponto em $E(\mathbb{F}_p)$, e finalmente, uma multiplicação escalar para verificar a ordem n.

O corpo finito $\mathbb{F}_p$ contém uma raiz cúbica primitiva da unidade $\zeta(u) = 18u^3+18u^2+9u+1$ como se pode verificar por inspeção direta. Dizendo de outra forma, $\zeta(u)^3 \equiv 1 \pmod{p(u)}$. A raiz primitiva é utilizada em operações como conjugação, cálculo do endomorfismo de Frobenius entre outras, que são operações necessárias para otimizar os cálculos nos corpos de extensão.

Curvas BN têm grau de mergulho $k = 12$ e admitem um *twist* sêxtico ($d = 6$). Isso significa que se pode obter um subgrupo de n-torção $\mathbb{G}_2$ definido sobre o *twist* da curva E', isto é, $\mathbb{G}_2 = E'(\mathbb{F}_{p^{k/d}})[n] = E'(\mathbb{F}_{p^2})[n]$. A seguinte condição $p \equiv 3 \pmod 4$ se mantém, se e somente se, u é ímpar. Tal condição permite representar o corpo de extensão $\mathbb{F}_{p^2}$ como um polinômio $f(X) = aX + b; a, b \in \mathbb{F}_p$ com $X^2 = -1$, imitando os números complexos. Tal representação facilita operações tais como quadrado, onde $(aX + b)^2 = a^2 + b^2$.

O *twist* correspondente $E'(\mathbb{F}_{p^2})$ da curva $E(\mathbb{F}_p)$ é normalmente selecionado encontrando-se um não-quadrado e não-cubo $\xi \in \mathbb{F}_{p^2}$ e verificando então via multiplicação escalar se a curva $E' : y^2 = x^3 + b'$ dada por $b' = b/\xi$ ou por $b' = b/\xi^5$ tem ordem divisível por n. O elemento ξ pode ser usado para representar as extensões do corpo $\mathbb{F}_{p^2}$ contidas no corpo $\mathbb{F}_{p^{12}}$ uma vez que o polinômio $z^r - \xi$ é irredutível sobre $\mathbb{F}_{p^{2h}}$ para $r \in \{2, 3, 6\}$ e $h \in \{1, 2, 3\}$ sempre que $\gcd(h, r) = 1$ (NAEHRIG, 2009, Lema 2.14).

Exemplo 2: Seja $p^e \equiv 1 \pmod 6$. Para cada $\xi \in \mathbb{F}_{p^e}$ que não é nem quadrado nem um

cubo, pode-se representar $\mathbb{F}_{p^{6e}}$ como uma torre de extensão de $\mathbb{F}_{p^e}$ das três seguintes maneiras:

- $\mathbb{F}_{p^{6e}} = \mathbb{F}_{p^e}[u]/(u^6 - \xi)$;
- $\mathbb{F}_{p^{6e}} = \mathbb{F}_{p^{2e}}[v]/(v^3 - \xi)$ com $\mathbb{F}_{p^{2e}} = \mathbb{F}_{p^e}[s]/(s^2 - \xi)$;
- $\mathbb{F}_{p^{6e}} = \mathbb{F}_{p^{3e}}[w]/(w^2 - \xi)$ com $\mathbb{F}_{p^{3e}} = \mathbb{F}_{p^e}[t]/(t^3 - \xi)$.

Os componentes de um elemento de $\mathbb{F}_{p^{6e}}$ em qualquer um desses exemplos podem ser extraídos diretamente, sem a necessidade de efetuar computações caras. Portanto: $a_0 + a_1u + a_2u^2 + a_3u^3 + a_4u^4 + a_5u^5 \leftrightarrow (a_0 + a_3s) + (a_1 + a_4s)v + (a_2 + a_5s)v^2 \leftrightarrow (a_0 + a_2t + a_4t^2) + (a_1 + a_3t + a_5t^2)w$, para $a_i \in \mathbb{F}_{p^e}$. Isso mostra que a configuração sugerida automaticamente fornece os denominados "compostos" ou corpos amigáveis a torre (GRANGER; SCOTT, 2010), com ganhos de eficiência associados.

As curvas BN constituem uma das classes mais versáteis de curvas elípticas amigáveis a emparelhamentos. Entre outras coisas, elas são conhecidas por (BARRETO et al., 2007):

- facilitar o desenvolvimento de emparelhamentos bilineares para um nível de segurança de 128 bits (DEVEGILI et al., 2007);
- possibilitar a construção de todos os esquemas e protocolos criptográficos baseados em emparelhamentos inclusive fornecendo assinaturas curtas (GALBRAITH et al., 2008);
- serem facilmente encontradas para vários tamanhos em bits (NAEHRIG, 2009, Seção 2.1.1);
- admitirem um *twist* sêxtico (HESS et al., 2006), logo os parâmetros do emparelhamento podem ser definidos sobre os corpos relativamente pequenos $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$, respectivamente;

- serem propícias para compressão dupla ou tripla (NAEHRIG et al., 2008);
- atingir alta eficiência para todos os algoritmos de cálculo de emparelhamento conhecidos, incluindo os emparelhamento de Tate (SCOTT, 2005), ate (HESS et al., 2006), Eil (HESS, 2008), R-ate (LEE et al., 2009), Xate (NOGAMI et al., 2008) e Ótimo (VERCAUTEREN, 2010);
- admitir otimizações baseadas em endomorfismos e homomorfismos para todos os grupos envolvidos [(GALBRAITH et al., 2009), (GALBRAITH; SCOTT, 2008)], permitindo assim, operações rápidas não apenas de emparelhamento;
- serem apropriadas para implementações em software e hardware em uma vasta gama de plataformas [(FAN et al., 2009), (GOUVÊA; LÓPEZ, 2009)].

2.4.5 Curvas MNT

Até o ano de 2001, curvas supersingulares eram as únicas curvas conhecidas por terem grau de mergulho pequeno suficiente para se aplicar os emparelhamentos de Tate ou de Weil. No mesmo ano, contudo, foi descrito um método de construção de curvas ordinárias com graus de mergulho $k = 3, 4, 6$ (MIYAJI et al., 2001). Refere-se a estas curvas como curvas MNT. Sua descrição é a seguinte. Seja $E(\mathbb{F}_q)$ uma curva elíptica de ordem prima $n = \#E(\mathbb{F}_q) = q + 1 - t$, onde t é o traço de E. Fixe um k e verifique quais restrições nos parâmetros n, q e t são impostas pela condição $m \mid (q^k - 1)$ e $m \nmid (q^\ell - 1)$ para $0 < \ell < k$.

Teorema 3. *(MIYAJI et al., 2001, Teoremas 2–4) Seja $E(\mathbb{F}_q)$ uma curva elíptica com traço t.*

1. *Se (q, t) podem ser representados por $q = 12\ell^2 - 1$ e $t = -1 \pm 6\ell$ para algum $\ell \in \mathbb{Z}$, então o grau de mergulho de E é $k = 3$ obtendo-se a curva MNT3.*
2. *Se (q, t) podem ser representados por $q = \ell^2 + \ell + 1$ e $t = -\ell, \ell + 1$ para algum $\ell \in \mathbb{Z}$, então o grau de mergulho de E é $k = 4$ obtendo-se a curva MNT4.*

3. *Se (q, t) podem ser representados por* $q = 4\ell^2 + 1$ *e* $t = 1 \pm 2\ell$ *para algum* $\ell \in \mathbb{Z}$*, então o grau de mergulho de E é* $k = 6$ *obtendo-se a curva MNT6.*

O próximo passo é construir curvas que recaem em uma das três classes descritas no Teorema 3. Isso é feito usando o método de multiplicação complexa (CM) (para mais detalhes veja (ATKIN; MORAIN, 1993)), que é o único conhecido para construir uma curva para determinados t e q. A dificuldade é que para se aplicar o método CM, precisa-se conhecer a solução de uma certa equação, denominada equação CM. Miyaji *et al.* (MIYAJI et al., 2001) mostraram que para $k \in (3, 4, 6)$ e q primo, a equação CM pode ser reduzida para uma equação de forma particular, a equação de Pell, para a qual a solução é conhecida.

Para um k qualquer, a equação CM não pode ser reduzida à equação de Pell e, portanto, o método proposto por (MIYAJI et al., 2001) não funciona. Os estudos independentes de Barreto *et al.* (BARRETO et al., 2003) e Dupont *et al.* (DUPONT et al., 2002) generalizam o critério MNT para k arbitrário, i.e. não restrito a $k \in (3, 4, 6)$. Eles contornam o problema escolhendo-se, primeiramente, um valor para servir como solução da equação CM. Só depois, buscam-se parâmetros que forneçam uma equação CM que tenha aquela solução particular. Eles também fornecem exemplos de curvas com grau de mergulho 7, 11 e 12.

O nível de segurança relacionado a uma curva MNTk, onde k é seu grau de mergulho, pode ser obtido pela expressão $N = \lceil \lg q \rceil \cdot k$, onde $\lceil \lg q \rceil$ é o tamanho do corpo sobre o qual a curva está definida. Por exemplo, uma curva MNT4 definida sobre $\mathbb{F}_q$ com q de 256 bits, temos que o nível é dado por $N = 256 * 4 = 1024$, ou seja, tem-se um nível equivalente ao do RSA-1024.

2.5 Divisores e emparelhamentos

Nesta seção são expostos os aspectos mais essenciais da teoria de divisores da geometria algébrica e sua relação com o conceito de emparelhamentos bilineares para aplicações criptográficas. A abordagem será sucinta, seguindo de perto a exposição em (BARRETO, 2003). O leitor interessado em detalhes do assunto pode consultar, por exemplo, o trabalho de Silverman (SILVERMAN, 1986).

2.5.1 Visão geral do emparelhamento

Exemplos de emparelhamentos bilineares podem ser encontrados no estudo de Álgebra Linear. Por exemplo, a operação de produto escalar é um emparelhamento bilinear no espaço vetorial $\mathbb{R}^n$,

$$\beta(\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \upsilon_1\omega_1 + \upsilon_2\omega_2 + \ldots + \upsilon_n\omega_n.$$

A operação β representa um emparelhamento no sentido de que ela toma um par de vetores e retorna um único número. Em outras palavras, podemos dizer grosseiramente que ela emparelhou seus dois vetores de entrada retornando o resultado desse emparelhamento. Tal operação também é dita bilinear, uma vez que produz uma transformação linear em ambos os argumentos. Ou seja, para quaisquer vetores v_1, v_2, w_1, w_2 e quaisquer números reais a_1, a_2, b_1, b_2 temos

$$\beta(a_1v_1 + a_2v_2, \mathbf{w}) = a_1\beta(v_1, \mathbf{w}) + a_2\beta(v_2, \mathbf{w}),$$
$$\beta(\mathbf{v}, b_1w_1 + b_2w_2) = b_1\beta(\mathbf{v}, w_1) + b_2\beta(\mathbf{v}, w_2)$$

A operação de emparelhamento bilinear a ser definida é similar à β, uma vez que ela recebe como entrada dois pontos em uma curva elíptica e retorna como saída um número contido em um corpo de extensão. Contudo, a condição de bilinearidade é ligeiramente diferente, porque o valor da saída deve ser um elemento não nulo do corpo.

Desta forma, o sinal de soma do lado direito das Equações em 2.5.1 é substituído por um produto para garantir que tal elemento nulo não seja produzido.

Definição 12. *(Emparelhamento Bilinear) Sejam dois grupos $\mathbb{G}_1$, $\mathbb{G}_2$ e também o subgrupo multiplicativo $\mathbb{G}_T$, todos de ordem prima n. Um emparelhamento bilinear, $\hat{e}$, é um mapa bilinear não degenerado e eficientemente computável, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Normalmente, $\mathbb{G}_1$ e $\mathbb{G}_2$ são escritos aditivamente, enquanto $\mathbb{G}_T$ é escrito multiplicativamente. As propriedades a serem satisfeitas pelo emparelhamento são*

1. *Bilinearidade: $\forall (P, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\forall a, b \in \mathbb{Z}_n$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.*
2. *Não-degenerado: $\forall P \in \mathbb{G}_1$, $\hat{e}(P, Q) = 1$ para todo $Q \in \mathbb{G}_2$ se e somente se $P = O_{\mathbb{G}_1}$, a identidade de $\mathbb{G}_1$.*
3. *Computabilidade: $\forall (P, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$, $e(P, Q)$ é eficientemente computável.*

A Definição 12 é mais genérica. Ela também pode ser instanciada sobre os grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ de curvas elípticas, e estas por sua vez podem ser de naturezas distintas, ou seja, curvas supersingulares ou ordinárias. Para cada uma delas a definição do emparelhamento é obtida de forma singular.

Para fins criptográficos são usados outros tipos de emparelhamento, diferentes do produto escalar β, que são capazes de fornecer propriedades essenciais para se apoiar a segurança dos protocolos criptográficos. Em outras palavras, criptosistemas baseados em emparelhamentos confiam sua segurança na intratabilidade de problemas matemáticos relacionados a emparelhamentos. Há muitos problemas computacionais conjecturados intratáveis no contexto de emparelhamentos. Por exemplo, o GDHP (*Gap Diffie-Hellman Problem*, ou problema Diffie-Hellman lacunar) e o BDHP (*Bilinear Diffie-Hellman Problem*, ou problema Diffie-Hellman bilinear) são comuns no estabelecimento de provas formais de segurança de protocolos baseados em emparelhamentos.

Definição 13. *(Gap Diffie-Helman problem) O problema GDHP é definido da seguinte forma: dados $P, \alpha P \in \mathbb{G}_1$ e $Q, \beta Q \in \mathbb{G}_2$, calcule $\alpha\beta P$ e/ou $\alpha\beta Q$ com a ajuda do emparelhamento nestes grupos.*

Definição 14. *(Bilinear Diffie-Hellman problem) O problema BDHP é definido da seguinte forma: dados $\alpha P, P \in \mathbb{G}_1$, $\beta Q, Q \in \mathbb{G}_2$ e $\gamma T, T \in \{\mathbb{G}_1, \mathbb{G}_2\}$ calcule $\hat{e}(P, Q)^{\alpha\beta\gamma}$.*

Os tipos de emparelhamento mais comumente usados na prática são de Weil (MILLER, 2004), de 1940, e de Tate (SCOTT, 2005), cujos nomes fazem menção aos seus idealizadores, André Weil e John Tate. Ambos os emparelhamentos utilizam o algoritmo de Miller em sua implementação, contudo, o emparelhamento de Weil é menos eficiente na prática pois necessita de duas aplicações de Miller, enquanto o de Tate apenas uma com a adição de uma exponenciação final mais leve que o algoritmo Miller.

Algoritmos típicos de emparelhamento são baseados no algoritmo de Miller (MILLER, 2004) com um número de otimizações (BARRETO et al., 2002; HESS et al., 2006; LEE et al., 2009; NOGAMI et al., 2008; VERCAUTEREN, 2010), os mais notáveis emparelhamentos ótimos (VERCAUTEREN, 2010) os quais têm uma ordem de laço de tamanho $\lceil \lg n \rceil / \varphi(k)$ em geral (onde φ é a função tociente de Euler), comparando bem com o emparelhamento original de Tate o qual tem laço de ordem de $\lceil \lg n \rceil$.

2.5.2 Teoria de divisores

Seja P um ponto em E de ordem prima r onde $r^2 \nmid n$. Diz-se que o subgrupo $\langle P \rangle$ tem *grau de mergulho k* para algum inteiro $k > 0$ se $r \mid q^k - 1$ e $r \nmid q^s - 1$ para todo $0 < s < k$. Aplicações criptográficas requerem curvas cujo grau de mergulho seja grande o suficiente para manter um nível elevado de segurança, mas pequeno o bastante para permitir que operações aritméticas em $\mathbb{F}_{q^k}$ sejam efetuadas eficientemente. O grau de mergulho de curvas elípticas ordinárias é, via de regra, enorme (BALASUBRAMANIAN; KOBLITZ, 1998). Entretanto, se E é supersingular, o valor de k é limitado por $k \leqslant$

6 (MENEZES et al., 1991). Este limite é atingido por curvas definidas sobre corpos de característica 3 mas não sobre corpos de característica 2, onde o valor máximo atingível é $k = 4$ (HANKERSON et al., 2003, seção 5.2.2). Métodos para construir curvas ordinárias de ordem prima só são conhecidos para alguns valores de k, a saber, $k \in \{3, 4, 6\}$ (MIYAJI et al., 2001), $k = 10$ (FREEMAN, 2006), e $k = 12$ (BARRETO; NAEHRIG, 2006).

O grupo $E(\mathbb{F}_q)$ é isomorfo a um subgrupo de $E(\mathbb{F}_{q^k})$ (justificando o nome grau de mergulho). Seja $P \in E(\mathbb{F}_q)$ um ponto de ordem r tal que $\langle P \rangle$ tenha grau de mergulho k. Então $E(\mathbb{F}_{q^k})$ contém um ponto Q da mesma ordem r mas linearmente independente de P, no sentido de que a única combinação linear da forma $\alpha P + \beta Q = O$ é aquela onde $\alpha = \beta = 0$.

Seja $E(\mathbb{F}_q)$ uma curva elíptica contendo um subgrupo de ordem prima r com grau de mergulho k. Um *divisor*[3] sobre E é uma soma formal $\mathcal{D} = \sum_{P \in E(\mathbb{F}_{q^k})} a_P(P)$ onde $a_P \in \mathbb{Z}$. Em outras palavras, um divisor é uma associação de um coeficiente inteiro a_P a cada ponto $P \in E(\mathbb{F}_{q^k})$, isto é, uma função $\mathcal{D} : E(\mathbb{F}_{q^k}) \to \mathbb{Z}$, $P_i \mapsto a_i$, para todos os pontos $P_i \in E(\mathbb{F}_{q^k})$, $i = 1, \ldots, n$, onde $\#E(\mathbb{F}_{q^k}) = n$. Os parênteses ao redor dos pontos P_i só são usados para lembrar que não se refere aqui ao valor da soma, isto é, $[a_1]P_1 + \cdots + [a_n]P_n$, mas aos coeficientes dos pontos. Em particular, a notação (P) é uma abreviação para o divisor em que os coeficientes de todos os pontos são nulos, exceto o coeficiente do ponto P, que é $a_P = 1$. Assim, $(P) \equiv 0(P_1) + 0(P_2) + \cdots + 1(P) + \cdots + 0(P_n)$.

O conjunto dos pontos $P \in E(\mathbb{F}_{q^k})$ tais que $n_P \neq 0$ chama-se suporte de $\mathcal{D}$. O *grau* de $\mathcal{D}$ é o valor da soma dos coeficientes $\deg(\mathcal{D}) = \sum_P a_P$. O divisor nulo, denotado 0, tem todos os coeficientes nulos, $n_P = 0$. A soma de dois divisores $\mathcal{D} = \sum_P n_P(P)$ e $\mathcal{D}' = \sum_P n'_P(P)$ é o divisor $\mathcal{D} + \mathcal{D}' = \sum_P (n_P + n'_P)(P)$. Induz-se assim uma estrutura de grupo abeliano sobre o conjunto de divisores; em particular, $r\mathcal{D} = \sum_P (ra_P)(P)$.

[3] Divisores são geralmente definidos sobre o fecho algébrico $\overline{\mathbb{F}_q}$ de $\mathbb{F}_q$ sujeito à condição adicional de que apenas um número finito de coeficientes a_P são não nulos. Contudo, restringiremos nossa atenção a divisores definidos sobre $\mathbb{F}_{q^k}$, de modo que o número total de coeficientes a_P seja finito.

Uma *função racional* $f : \mathbb{F}_{q^k} \times \mathbb{F}_{q^k} \to \mathbb{F}_{q^k}$ é uma função da forma $f(x, y) = N(x, y)/D(x, y)$, onde $N, D \in \mathbb{F}_{q^k}[x, y]$. Um zero de f é qualquer ponto $(x, y) \in \mathbb{F}_{q^k} \times \mathbb{F}_{q^k}$ tal que $N(x, y) = 0$ e $D(x, y) \neq 0$, e um polo de f é qualquer ponto $(x, y) \in \mathbb{F}_{q^k} \times \mathbb{F}_{q^k}$ tal que $N(x, y) \neq 0$ e $D(x, y) = 0$ (note-se que f não está propriamente definida em seus polos). A multiplicidade de f em P, escrita $\mathsf{Ord}_P(f)$, é definida como $\deg N$ se P for um zero de f, como $\deg D$ se P for um polo de f, e como 0 de P for um ponto ordinário de f (i.e. nem um zero nem um polo de f). Por extensão, define-se $f : E(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}$ como $f(P) = f(x, y)$ para $P = (x, y)$.

Dada uma função racional $f : E(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}$, define-se o *divisor de* f como sendo o divisor $(f) = \sum_P \mathsf{Ord}_P(f)(P)$. Segue desta definição que $(fg) = (f)+(g)$ e $(f/g) = (f)-(g)$ para quaisquer duas funções f e g definidas sobre $E(\mathbb{F}_{q^k})$; além disso, $(f) = 0$ se, e somente se, f for uma constante não nula. Em consequência, duas funções distintas por um fator constante $c \neq 0$ têm o mesmo divisor, isto é, $(cf) = (f)$ para todo $c \in \mathbb{F}_{q^k}^*$.

Um divisor $\mathcal{D}$ é dito *principal* se $\mathcal{D} = (f)$ para alguma função (f). Sabe-se (HANKERSON et al., 2003, teorema 2.25) que um divisor $\mathcal{D} = \sum_P a_P(P)$ é principal se, e somente se, o grau de $\mathcal{D}$ é zero e $\sum_P a_P P = O$. Dado um ponto $P \in E[r]$, um exemplo importante de divisor principal é $r(P) - r(O)$.

Diz-se que dois divisores $\mathcal{D}$ e $\mathcal{D}'$ são equivalentes, $\mathcal{D}' \sim \mathcal{D}$, se existe uma função g tal que $\mathcal{D}' = \mathcal{D} + (g)$, isto é, se a diferença entre eles for um divisor principal. Em particular, $(P + R) - (R) \sim (P) - (O)$ para quaisquer pontos P e R.

Para qualquer função f e qualquer divisor $\mathcal{D} = \sum_P a_P(P)$ de grau zero, define-se $f(\mathcal{D}) = \prod_P f(P)^{a_P}$. Se f for constante quando calculada sobre os pontos da curva, seu valor será 1 quando calculada sobre um divisor de grau zero, isto é, $f(\mathcal{D}) = 1$ independentemente do valor da constante $f(P)$.

2.5.3 Formalização de emparelhamentos bilineares

Definição 15. *Seja $E(\mathbb{F}_q)$ uma curva elíptica contendo um subgrupo de ordem prima r e grau de mergulho k, e seja ℓ um múltiplo de r que divide $q^k - 1$. Sejam $P \in E(\mathbb{F}_q)[\ell]$, $Q \in E(\mathbb{F}_{q^k})$, f_ℓ uma função racional cujo divisor satisfaça $(f_\ell) = \ell(P) - \ell(O)$ e $\mathcal{D} \sim (Q) - (O)$ um divisor com suporte disjunto do suporte de f_ℓ. O* emparelhamento de Tate *(por vezes chamado emparelhamento de Tate-Lichtenbaum)) de ordem ℓ é a função racional $e_\ell : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^*$ definida por $e_\ell(P, Q) \equiv f_\ell(\mathcal{D})^{(q^k-1)/\ell}$.*

Note-se que, como $P \in E(\mathbb{F}_q)$, f_ℓ é uma função racional com coeficientes em $\mathbb{F}_q$.

O emparelhamento de Tate satisfaz as seguintes propriedades (FREY et al., 1999):

- *(Bilinearidade)* $e_\ell(P_1 + P_2, Q) = e_\ell(P_1, Q) \cdot e_\ell(P_2, Q)$ e $e_\ell(P, Q_1 + Q_2) = e_\ell(P, Q_1) \cdot e_\ell(P, Q_2)$ para todo $P, P_1, P_2 \in E(\mathbb{F}_q)[\ell]$ e todo $Q, Q_1, Q_2 \in E(\mathbb{F}_{q^k})$. Segue daí que $e_\ell([a]P, Q) = e_\ell(P, [a]Q) = e_\ell(P, Q)^a$ para todo $a \in \mathbb{Z}$.

- *(Não-degeneração)* Se $e_\ell(P, Q) = 1$ para todo $Q \in E(\mathbb{F}_{q^k})$, então $P = O$. Alternativamente, para todo $P \neq O$ existe $Q \in E(\mathbb{F}_{q^k})$ tal que $e_\ell(P, Q) \neq 1$.

- *(Compatibilidade)* Se $\ell = h\ell'$, $P \in E(\mathbb{F}_q)[\ell]$, e $Q \in E(\mathbb{F}_{q^k})$, então $e_\ell(P, Q) = e_{\ell'}(hP, Q) = e_{\ell'}(P, Q)^h$, isto é, $f_\ell(\mathcal{D})^{(q^k-1)/\ell} = f_{\ell'}(\mathcal{D})^{(q^k-1)/\ell'}$.

A propriedade de compatibilidade permite escrever simplesmente $e(P, Q)$, subentendendo-se $f_\ell(\mathcal{D})^{(q^k-1)/\ell}$ para qualquer múltiplo ℓ de r que divida $q^k - 1$.

De modo geral, é inviável representar $e(P, Q)$ diretamente como a razão de dois polinômios. Contudo, calculando o valor do emparelhamento sob demanda mantém a complexidade computacional do emparelhamento de Tate (ou de Weil) igual à de uma multiplicação por escalar sobre a extensão da curva E para o corpo finito $\mathbb{F}_{q^k}$. Esta é a ideia básica do algoritmo de Miller.

2.5.4 Algoritmo de Miller

Sejam $P \in E(\mathbb{F}_q)[r]$ e $Q \in E(\mathbb{F}_{q^k})$ pontos linearmente independentes, com $k > 1$. Seja f_r uma função racional sobre $\mathbb{F}_{q^k}$ com divisor $(f_r) = r(P) - r(O)$. A definição 15 do emparelhamento de Tate não sugere um método efetivo de cálculo, pois a função f_r não está explicitamente definida. Uma ideia de cálculo mostrada em (MILLER, 2004) é decompor f_r em funções de grau 1, a saber, as linhas retas definidas pela construção geométrica ("secantes e tangentes") da lei de grupo encontradas durante o cálculo de $[r]P$.

Seja $g_{U,V}$ a (equação da) reta que passa pelos pontos $U, V \in \langle P \rangle$, e seja $g_{U,V}(Q)$ o valor da equação dessa reta no ponto $Q \in E(\mathbb{F}_{q^k})$. A abreviação g_V denota $g_{V,-V}$. As funções $g_{U,V}$ são denominadas *funções de linha*. Em coordenadas afins, se a curva for definida pela equação $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$, para $U = (x_U, y_U)$, $V = (x_V, y_V)$ e $Q = (x, y)$ temos:

$$\begin{aligned} g_{U,V}(Q) &= 1,\ Q \in \langle P \rangle. \\ g_{U,U}(Q) &= \lambda_1(x - x_U) + y_U - y,\ Q \notin \langle P \rangle. \\ g_{U,V}(Q) &= \lambda_2(x - x_U) + y_U - y,\ Q \notin \langle P \rangle,\ U \neq V. \\ g_U(Q) &= x - x_U,\ Q \notin \langle P \rangle. \end{aligned}$$

onde

$$\lambda_1 = \frac{3x_U^2 + a}{2y_U},\quad \lambda_2 = \frac{y_V - y_U}{x_V - x_U}.$$

Teorema 4 (Fórmula de Miller)**.** *Seja P um ponto de $E(\mathbb{F}_q)$ e f_c uma função com divisor $(f_c) = c(P) - ([c]P) - (c-1)(O)$, $c \in \mathbb{Z}$. Para todo $a, b \in \mathbb{Z}$, $f_{a+b}(\mathcal{D}) = f_a(\mathcal{D}) \cdot f_b(\mathcal{D}) \cdot g_{[a]P,[b]P}(\mathcal{D}) / g_{[a+b]P}(\mathcal{D})$ salvo um fator não nulo constante.*

Demonstração. Os divisores das funções de linha satisfazem:

$$\begin{aligned}(g_{[a]P,[b]P}) &= ([a]P) + ([b]P) + (-[a+b]P) - 3(O),\\ (g_{[a+b]P}) &= ([a+b]P) + (-[a+b]P) - 2(O).\end{aligned}$$

Logo, $(g_{[a]P,[b]P}) - (g_{[a+b]P}) = ([a]P) + ([b]P) - ([a+b]P) - (O)$. Da definição de f_c vemos que:

$$\begin{aligned}(f_{a+b}) &= (a+b)(P) - ([a+b]P) - (a+b-1)(O)\\ &= a(P) - ([a]P) - (a-1)(O)\\ &+ b(P) - ([b]P) - (b-1)(O)\\ &+ ([a]P) + ([b]P) - ([a+b]P) - (O)\\ &= (f_a) + (f_b) + (g_{[a]P,[b]P}) - (g_{[a+b]P}).\end{aligned}$$

Portanto, $f_{a+b}(\mathcal{D}) = f_a(\mathcal{D}) \cdot f_b(\mathcal{D}) \cdot g_{[a]P,[b]P}(\mathcal{D}) \,/\, g_{[a+b]P}(\mathcal{D})$. □

Pela definição de f_c, vemos que $(f_0) = (f_1) = 0$, significando que f_0 e f_1 são constantes sobre os pontos da curva, e portanto $f_0(\mathcal{D}) = f_1(\mathcal{D}) = 1$ para qualquer divisor $\mathcal{D}$ de grau zero (cfr. seção 2.5.2). Assim, $f_{a+1}(\mathcal{D}) = f_a(\mathcal{D}) \cdot g_{[a]P,P}(\mathcal{D}) / g_{[a+1]P}(\mathcal{D})$ e $f_{2a}(\mathcal{D}) = f_a(\mathcal{D})^2 \cdot g_{[a]P,[a]P}(\mathcal{D}) / g_{[2a]P}(\mathcal{D})$ para $a > 0$. Essas observações permitem escrever um algoritmo iterativo que calcula $f_r(\mathcal{D})$ combinando as fórmulas acima com o método da decomposição binária para calcular $[r]P$. Infelizmente, não é possível usar diretamente $\mathcal{D} = (Q) - (O)$ para calcular $g_{U,V}(\mathcal{D})$, pois $g_{U,V}$ tem um polo (de ordem 2 ou 3) em O. Podemos, porém, usar um divisor equivalente $\mathcal{D} = (Q+R) - (R)$, onde o ponto arbitrário $R \in E(\mathbb{F}_{q^k})$ não é um polo nem um zero de $g_{U,V}$.

Seja então $(r_t, r_{t-1}, \ldots, r_1, r_0)$ a representação binária da ordem $r \geqslant 0$ de P, onde $r_i \in \{0,1\}$ e $r_t \neq 0$. Assume-se $r \mid q^k - 1$. O cálculo do emparelhamento de Tate de ordem r, $e_r(P,Q)$, procede da seguinte maneira:

Algoritmo de Miller:

$R \leftarrow$ **aleat**$(E(\mathbb{F}_{q^k}))$, $\mathcal{D} \leftarrow (Q + R) - (R)$

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

para $i \leftarrow t-1, t-2, \ldots, 1, 0$ **faça** {

$f \leftarrow f^2 \cdot g_{V,V}(\mathcal{D})/g_{[2]V}(\mathcal{D})$, $V \leftarrow [2]V$

se $r_i = 1$ **então** $f \leftarrow f \cdot g_{V,P}(\mathcal{D})/g_{V+P}(\mathcal{D})$, $V \leftarrow V + P$

}

devolva $e_r(P, Q) \leftarrow f^{(q^k-1)/r}$

Este algoritmo falha para a escolha feita do ponto R se esse ponto for um polo ou um zero de alguma das ocorrências de $g_{U,V}$. Nesse caso, um novo ponto R teria que ser escolhido e o processo reiniciado, mas a probabilidade de uma tal ocorrência na prática é negligível.

O algoritmo de Miller, sendo concomitante ao cálculo de uma multiplicação por escalar, beneficia-se das mesmas técnicas de otimização disponíveis para este cálculo (MENEZES et al., 1997, seção 14.6).

Os algoritmos mais eficientes conhecidos para o cálculo de emparelhamentos, de uma forma ou de outra, são baseados no algoritmo de Miller acoplado a um grande número de técnicas de otimização, na maioria das vezes dependente da escolha das curvas elípticas subjacentes.

Sobre as curvas da forma descrita na Seção 2.4.4, podemos definir o emparelhamento bilinear, $\hat{e}$. Esta função recebe como argumentos dois pontos P e Q na curva elíptica[4], geradores de $\mathbb{G}_1$ e $\mathbb{G}_2$, e os mapeia para o conjunto $\mathbb{G}_T \in \mathbb{F}_{p^k}$, sendo k o grau de mergulho da curva. Para implementações mais eficientes e seguras é desejável que o grau de mergulho, k, seja pequeno, enquanto o primo p seja um número grande. O grau de mergulho deve ser escolhido de forma que seja o menor inteiro tal

[4]Para curvas ordinárias como as curvas BN os pontos não estão necessariamente na mesma curva, mas em curvas isomorfas.

que $n \mid (p^k - 1)$, n é uma restrição para que a curva definida sobre $\mathbb{F}_{p^k}$ contenha todos os pontos de n-torção (i.e. $E[n] \subseteq E(\mathbb{F}_{p^k})$).

Na prática, os grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ são mais comumente determinados pelo auto-espaço vetorial do endomorfismo de Frobenius ϕ_p em alguma curva elíptica $E(\mathbb{F}_p)$ de grau de mergulho $k > 1$. Especificamente, $\mathbb{G}_1$ é escolhido para ser o auto-espaço-1, $E[n] \cap \ker(\phi_p - [1]) = E(\mathbb{F}_p)[n]$, e $\mathbb{G}_2$ é escolhido para ser a imagem inversa de $E'(\mathbb{F}_{p^e})[n]$ do auto-espaço-p $E[n] \cap \ker(\phi_p - [p]) \subseteq E(\mathbb{F}_{p^k})[n]$ sob um isomorfismo de *twist* $\psi : E' \to E$, $(x, y) \mapsto (\mu^2 x, \mu^3 y)$ para algum $\mu \in \mathbb{F}_{p^k}$, onde E' é definido sobre $\mathbb{F}_{p^e}$ e $e \mid k$ é o menor possível (ou, equivalentemente, onde o *twist* E' tem o maior grau $d = k/e$).

Quando um emparelhamento é definido sobre curvas amigáveis a emparelhamento, são usadas na prática curvas supersingulares ou ordinárias. As primeiras são as curvas mais simples existentes, definidas sobre corpos de característica binária, ternária ou prima com grau de mergulho pequeno ($k \leq 6$)[5].

Há uma alternativa para curvas supersingulares (com escolha limitada do grau de mergulho k). São as famílias especiais de curvas ordinárias ou também chamadas de não-supersingulares, representadas por $E(\mathbb{F}_p)$. Neste caso, o emparelhamento (e.g. Tate) recai sobre o corpo estendido $\mathbb{F}_{p^k}$, onde k é o grau de mergulho da curva.

Emparelhamentos sobre curvas supersingulares são ditos do tipo 1. Eles possuem a propriedade simétrica $e(P, Q) = e(Q, P)$. Emparelhamentos sobre curvas ordinárias são ditos do tipo 3, e diferentemente do tipo 1, não são simétricos. Outra propriedade dos emparelhamentos do tipo 3 é que um dos seus parâmetros deve ser um ponto em $E(\mathbb{F}_p)$ e o melhor que pode ser feito para o outro parâmetro é que ele seja um ponto em $E'(\mathbb{F}_{p^d})$, onde $d \mid k$ e E' é algum *twist* da curva original.

No caso do emparelhamento de Tate, se $q = 2^m$, definido em uma curva do tipo amigável a emparelhamento (existem curvas em que há ataques eficientes com cálculo

[5]Note que também existem curvas supersingulares com $k > 6$ de gênero $g > 1$, mas estas são denominadas curvas hiperelípticas e não fazem parte do escopo deste trabalho

de índices que não são recomendadas), a função mapeia os elementos de seu domínio para um corpo $\mathbb{F}_{q^k}$, $\hat{e}(P, Q) \mapsto \mathbb{F}_{q^k}$. Os pontos P e Q são pontos de ordem r na curva, e para curvas supersingulares, r é um divisor primo grande de $2^m \pm 2^{(m+1)/2} + 1$, em que $2^m \pm 2^{(m+1)/2} + 1 = w \cdot r$ e 2^m é a ordem de q, e r a ordem de $w \cdot r$. A ordem de t é $2^{(m+1)/2}$, o traço de Frobenius.

A seguir temos um exemplo de *hash* em curvas supersingulares:

Exemplo 3: Um exemplo prático foi proposto no trabalho de Szczechowiak (SZCZECHOWIAK et al., 2009). Consideram-se curvas supersingulares com $q = 2^m$ e o mapeamento das identidades ID_A e ID_B, $A = H(\mathsf{ID}_A)$, $B = H(\mathsf{ID}_B)$.

O custo de mapear identidades para pontos de curva deve ser considerado. Um método simples é calcular o hash da identidade para a coordenada x e então resolver a equação quadrática para encontrar y. Caso a equação não tenha solução, incrementa-se x e tenta-se novamente. Caso contrário, tome uma das duas soluções de y e multiplique o ponto (x, y) pelo cofator $w = 487805$, para obter um ponto de ordem r. O tempo para tal mapeamento é pequeno comparado ao do cálculo do emparelhamento.

Seja a curva: $y^2 + y = x^3 + x$ sobre $\mathbb{F}_{2^{271}}$ com número de pontos na curva: $2^{271} + 2^{136} + 1 = 487805 \cdot r$. O valor de r deve ser grande o suficiente para tornar qualquer ataque do tipo Pohlig-Hellman em ECDLP intratável. O emparelhamento mais rápido conhecido nessas curvas é o η_T (BARRETO et al., 2007).

Neste caso, o problema do ECDLP em $E(\mathbb{F}_q)$ pode ser reduzido ao DLP em $\mathbb{F}_{q^k}$. Para que o emparelhamento seja seguro, ambos os problemas devem ser difíceis a níveis aceitáveis de segurança. Para $q = 2^{271}$ o nível de segurança em $\mathbb{F}_{2^{271\times4}}$ é aproximadamente $L = 4 \cdot 271 = 1084$ quando comparado ao RSA-1024. □

2.6 Sinopse

Este capítulo elencou uma série de notações matemáticas e conceitos necessários à construção do arcabouço para as curvas elípticas propostas mais adiante, particularmente os fundamentos de álgebra de grupos e corpos finitos e o conceito de emparelhamento bilinear. Descreveram-se técnicas, algoritmos e parâmetros mais relevantes do ponto de vista prático, com ênfase para curvas amigáveis a emparelhamentos, tais como as curvas BN.

3 UMA SUBFAMÍLIA DE CURVAS BN AMIGÁVEIS À IMPLEMENTAÇÃO

Este trabalho propõe uma subfamília de curvas BN livre de testes de caráter quadrático e/ou cúbico comumente necessários ao se representar as extensões de corpos finitos que ocorrem em uma implementação típica de protocolos baseados em emparelhamentos. O seguinte lema apanha uma importante propriedade da classe de curvas BN que diz que, se a ordem da curva for prima, então o coeficiente b não é quadrado nem cubo em $\mathbb{F}_p$:

Lema 1. (NAEHRIG, 2009, Lema 2.7) *Seja $E : y^2 = x^3 + b$ uma curva elíptica sobre $\mathbb{F}_p$ de ordem $n = \#E(\mathbb{F}_p)$ tal que $2 \nmid n$ e $3 \nmid n$. Então o coeficiente b não é um quadrado nem um cubo em $\mathbb{F}_p$.*

Demonstração. Para qualquer $\gamma, \delta \in \mathbb{F}_p$, o ponto $(0, \gamma) \in E : y^2 = x^3 + \gamma^2$ tem ordem 3 e, portanto, $3 \mid n$, enquanto o ponto $(-\delta, 0) \in E : y^2 = x^3 + \delta^3$ tem ordem 2 e, portanto, $2 \mid n$, ambos os caminhos contradizendo o pressuposto de que $2 \nmid n$ e $3 \nmid n$. □

Como consequência, pode-se definir o coeficiente b como sendo a norma de um elemento ξ pertencente a alguma extensão $\mathbb{F}_{p^e}$ e, dado que b não é quadrado nem cubo em $\mathbb{F}_p$, então o elemento ξ também não o será em $\mathbb{F}_{p^e}$. Tal fato permite a liberdade de parametrização do coeficiente b em função da livre escolha do elemento ξ. O lema a seguir nos dá suporte a tal propriedade:

Lema 2. *Seja $\xi \in \mathbb{F}_{p^e}^*$ e seja $b = |\xi| \in \mathbb{F}_p$. Se $E : y^2 = x^3 + b$ sobre $\mathbb{F}_p$ tem ordem $n = \#E(\mathbb{F}_p)$ com $2 \nmid n$ e $3 \nmid n$, então ξ não é nem um quadrado nem um cubo em $\mathbb{F}_{p^e}$.*

Demonstração. Assuma que ξ seja um quadrado ou um cubo em $\mathbb{F}_{p^e}$, i.e. $\xi = \gamma^r$ para algum $\gamma \in \mathbb{F}_{p^e}$ e $r \in \{2, 3\}$, então $b = |\xi| = |\gamma^r| = (|\gamma|)^r$, i.e. b é um quadrado ou um cubo em $\mathbb{F}_p$, contradizendo o Lema 1. □

Isso significa que testar se b é de caráter quadrático ou cúbico não é necessário em $\mathbb{F}_p$ nem em $\mathbb{F}_{p^e}$. Em particular, o elemento ξ especificado no Lema 2 pode ser usado para definir todas as extensões de $\mathbb{F}_{p^e}$ que são de interesse na implementação de emparelhamento e para facilitar trocas entre representações nas torres de corpos, como mostrado no Exemplo 2.

O próximo resultado se refere a como evitar o cálculo de ordem e a aritmética elíptica em $E'(\mathbb{F}_{p^2})$ para *twists* sêxticos, revelando imediatamente qual deles tem a ordem correta. Antes de apresentá-lo, precisamos da seguinte propriedade adicional:

Lema 3. *Seja* $p \equiv 1 \pmod 3$ *um primo. Para qualquer* $\xi \in \mathbb{F}_{p^2}$*, seja* $b = |\xi| = \xi\bar{\xi}$*. Então* b/ξ^5 *é um cubo.*

Demonstração. Primeiro nota-se que $b = \xi\bar{\xi} = \xi\xi^p = \xi^{p+1}$ e portanto $b/\xi^5 = \xi^{p-4}$. Uma vez que $p - 4$ é divisível por 3, constata-se que b/ξ^5 é um cubo, como afirmado. □

Neste ponto, estamos finalmente em posição de enunciar o seguinte teorema, que permite a captura da curva *twist* correta:

Teorema 5. *Dada uma curva BN da forma* $E : y^2 = x^3 + b$ *com* $b = |\xi|$ *para algum* $\xi \in \mathbb{F}_{p^2}$*, o* twist *sêxtico particular* $E' : y'^2 = x'^3 + \bar{\xi}$ *satisfaz* $\#E(\mathbb{F}_p) \mid \#E'(\mathbb{F}_{p^2})$.

Demonstração. Uma vez que E é assumida ser uma curva BN, o parâmetro b não pode ser a norma de um elemento em $\mathbb{F}_p$, porque tal norma é um quadrado em $\mathbb{F}_p$, contradizendo o Lema 1. Portanto, as considerações implicam $\xi \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$. O Remark 2.13 em (NAEHRIG, 2009) mostra que a ordem do *twist* desejado E' sobre $\mathbb{F}_{p^2}$ é

$n' = \#E'(\mathbb{F}_{p^2}) = n(2p - n)$. Uma vez que n é ímpar, n' também o é. Se $\xi \in \mathbb{F}_{p^2}$ não é nem um quadrado nem um cubo, o *twist* correto é dado por $y^2 = x^3 + b/\xi$ ou por $y^2 = x^3 + b/\xi^5$. Uma vez que $p \equiv 1 \pmod 3$ e $b = |\xi|$, o valor b/ξ^5 é um cubo pelo Lema 3. Isso significa que a curva dada por $y^2 = x^3 + b/\xi^5$ tem um ponto de ordem 2, portanto a ordem do *twist* particular é par. Logo, $E' : y^2 = x^3 + b/\xi$ é o *twist* procurado. Note que $b/\xi = \bar{\xi}$. □

Propõe-se usar curvas BN da forma $E_{b,\ell} : y^2 = x^3 + b$ onde

$$b = c^4 + d^6 \text{ ou } b = c^6 + 4d^4 \tag{3.1}$$

para $c, d \in \mathbb{N} \setminus \{0\}$. O primo BN p satisfaz $p \equiv 3 \pmod 4$ (e possivelmente também $p \equiv 4 \pmod 9$). Os pesos de Hamming das representações binárias (com sinal) da ordem do laço ω do emparelhamento Ate ótimo ou do parâmetro u (ou ambos) são mínimos para cada tamanho em bits $\ell := \lceil \lg p \rceil$, e b é tão pequeno quanto possível (de preferência com baixo peso de Hamming).

Com um leve abuso de notação identificamos o inteiro b como um elemento do corpo $\mathbb{F}_p$. O correspondente $\xi \in \mathbb{F}_{p^2}$ tal que $b = |\xi|$ é então

$$\xi = c^2 + d^3 i \text{ ou } \xi = c^3 + 2d^2 i, \tag{3.2}$$

respectivamente. Note que a escolha de b é consistente com ambos o Teorema 2 e o Teorema 5, e é compatível também com (RUBIN; SILVERBERG, 2010, Algoritmo 3.5).

Como principal contribuição deste trabalho, propõe-se o uso de curvas com a forma determinada pelas Equações em 3.1, pertencendo a uma subfamília das curvas BN definida como segue:

Definição 16. *Uma curva BN $E_b : y^2 = x^3 + b$ sobre $\mathbb{F}_p$ é denominada* amigável *se $p \equiv 3 \pmod 4$ e se existem $c, d \in \mathbb{F}_p^*$ tais que $b = c^4 + d^6$ ou $b = c^6 + 4d^4$.*

Uma curva BN amigável $E = E_b$ com os parâmetros correspondentes c e d de-

finidos acima tem as seguinte propriedades. Note que, uma vez que $p \equiv 3 \pmod 4$, representa-se $\mathbb{F}_{p^2}$ por $\mathbb{F}_p[i]/(i^2+1)$.

- Os parâmetros c e d automaticamente fornecem $\xi \in \mathbb{F}_{p^2}$ com $b = |\xi|$ de acordo com o Lema 2 para representar as extensões requeridas pelo corpo $\mathbb{F}_{p^2}$. O elemento ξ é $\xi = c^2 + d^3 i$ ou $\xi = c^3 + 2d^2 i$, respectivamente.

- Uma vez que $b = |\xi|$, os parâmetros c e d determinam o *twist* sêxtico de ordem correta de acordo com o Teorema 5, e é dado pela equação $y^2 = x^3 + \bar{\xi}$.

- Os geradores de $E(\mathbb{F}_p)$ são dados por soluções óbvias para a equação da curva como $G = (-d^2, c^2)$ ou $G = (-c^2, 2d^2)$, respectivamente.

- Os geradores para $E'(\mathbb{F}_{p^2})[n]$ podem ser encontrados como hG', onde $h = 2p - n$ e $G' = (-di, c)$ ou $G' = (-c, d(1-i))$, respectivamente.

A justificativa para a parametrização descrita é resumida como segue.

3.1 Eficiência do emparelhamento

Primeiramente, o cálculo do emparelhamento deve ser tão eficiente quanto possível, uma vez que é a operação mais cara em qualquer protocolo baseado em emparelhamento. A ordem ω de baixo peso minimiza o custo do laço de Miller no emparelhamentos Ate ótimo, enquanto o baixo peso de u minimiza o custo da exponenciação final (SCOTT et al., 2009b).

A possibilidade de usar pequenos valores de b acelera o cálculo do emparelhamento (COSTELLO et al., 2010), especialmente quando b tem baixo peso de Hamming, o que é claramente possível com a forma prescrita sugerida (e.g. se c e d são pequenas potências de 2). Uma das melhores situações surge quando $b = 2$ e $\xi = 1 + i$. Uma vez que multiplicações por tais valores de b são eficientes e todas as plataformas (não

apenas naquelas onde uma multiplicação dedicada por uma constante pequena está prontamente disponível, mas também aquelas onde multiplicação tem que ser emulada com operações como *shifts* ou adições) e o cálculo de conjugados, que envolve multiplicações por ξ, causa mínimo sobrecusto.

3.2 Eficiência geral

Todas as operações envolvidas em protocolos baseados em emparelhamentos devem ser as mais eficientes possíveis. Trabalhos como (BEUCHAT et al., 2010) apenas consideram o cálculo rápido do emparelhamento como métrica, desconsiderando operações como gerar pontos aleatórios ou calcular *hash* para os grupos de emparelhamento $\mathbb{G}_1$ e $\mathbb{G}_2$ que é essencial para a maior parte dos esquemas criptográficos baseados em emparelhamentos.

Para curvas BN, isso significa que deve ser um método muito eficiente de se calcular raízes quadradas em $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$. Isto é menos caro quando $p \equiv 3 \pmod 4$ e $p^2 \equiv 9 \pmod{16}$, uma vez que o método de Cipolla-Lehmer simplifica para um teste de caráter quadrático e uma exponenciação para raízes quadradas em $\mathbb{F}_p$, isto é, $\sqrt{a} = a^{(p+1)/4}$, e o método de KCYL (KONG et al., 2006) se aplica ao cálculo de raízes quadradas em $\mathbb{F}_{p^2}$, levando um teste de caráter quadrático e 1.5 exponenciações. O caso $p^2 \equiv 17 \pmod{32}$ é quase tão eficiente, levando um teste de caráter quadrático e 2 exponenciações para extrair raízes em $\mathbb{F}_{p^2}$ com o método de (MÜLLER, 2004).

Em certos cenários (e.g. quando é desejada compressão tripla do emparelhamento) pode-se querer impor $p \equiv 4 \pmod 9$ também, uma vez que isso facilita o cálculo de raízes cúbicas com métodos similares àqueles para o cálculo de raízes quadradas (BARRETO; NAEHRIG, 2006, Seção 3.1).

3.3 Aritmética uniforme em corpos finitos

Aritmética envolvida deve ser eficiente. Operações em $\mathbb{G}_1$ e $\mathbb{G}_2$ também precisam de aritmética eficiente em $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$, e processamento adicional (e.g. exponenciação implícita ou explícita) dos valores de emparelhamento necessitam de algoritmos eficientes para o próprio $\mathbb{F}_{p^{12}}$, ou em alguns casos para o subcorpo $\mathbb{F}_{p^6}$ ou $\mathbb{F}_{p^4}$, se as técnicas de compressão de emparelhamentos são adotadas (por fatores de 2 e 3, respectivamente). Também, potencial suporte para conversões eficientes entre diferentes representações têm que ser planejadas para propiciar a interoperabilidade.

3.4 Simplicidade de gerador

Geradores óbvios que não envolvem nenhum processamento ou armazenamento adicional são claramente desejados. Uma equação de curva da forma $E : y^2 = x^3 + (c^4 + d^6)$ admite a solução óbvia $G = (-d^2, c^2)$, enquanto uma da forma $E : y^2 = x^3 + (c^6 + 4d^4)$ admite a solução $G = (-c^2, 2d^2)$.

Além disso, pelo Teorema 5 o *twist* sêxtico da forma respectivamente $E' : y'^2 = x'^3 + (c^2 - d^3 i)$ ou $E' : y'^2 = x'^3 + (c^3 - 2d^2 i)$ sempre contém um subgrupo de mesma ordem n que E, e a equação da curva para E' admite a solução óbvia $G' = (-di, c)$ ou $G' = (-c, d(1-i))$, respectivamente, para que o ponto $h \cdot G'$, onde $h = p - 1 + t$, apenas falha em ser um gerador de $E'(\mathbb{F}_{p^e})[n]$ com probabilidade desprezível $O(1/h)$.

Uma escolha particulamente adequada é fazer $d = 1$ e procurar o menor c tal que E tenha ordem n. A multiplicação por cofator pode ser feita de forma muito eficiente (SCOTT et al., 2009a, Seção 6).

3.5 Tamanhos apropriados de corpo

Um gargalo óbvio é a aritmética em $\mathbb{F}_{p^2}$, uma vez que ela está sob todas as operações em $\mathbb{G}_2$, $\mathbb{G}_T$, e o cálculo do emparelhamento. Escolher p ligeiramente menor do que um múltiplo da palavra da plataforma (digamos, dois bits a menos) é interessante porque isto permite não apenas adiar reduções modulares em operações críticas como multiplicação ou quadrado no corpo $\mathbb{F}_{p^2}$, mas também simplificar a redução real quando ela é finalmente aplicada, como destacado em (BEUCHAT et al., 2010, Seção 5.2).

3.6 Sinopse

Neste capítulo apresentou-se a contribuição central deste trabalho, a saber, uma família de curvas BN especialmente favorável a implementações eficientes, denominadas curvas BN amigáveis. Detalharam-se as razões que nortearam a escolha dessa família de curvas, e expuseram-se as vantagens por ela fornecidas.

4 IMPLEMENTAÇÃO E RESULTADOS

Este capítulo trata de inspecionar as propriedades apresentadas pela subfamília de curvas através de exemplos práticos e da sua implementação através da linguagem Java. Foi implementado o emparelhamento Ate ótimo sobre uma curva específica desta família definida em um corpo finito de tamanho 254 bits.

4.1 Exemplos de curvas para a família proposta

Algumas das curvas apropriadas para este trabalho são listadas como exemplos a seguir.

Na Tabela 2, são apresentadas curvas práticas da família proposta para corpos de tamanho $\ell := 32m - 2$ em bits, onde $5 \leqslant m \leqslant 20$. Portanto, tais corpos apresentam tamanhos de 158 a 638 bits, o que fornece níveis de segurança (equivalente simétrico) variando de 80 até 192 bits.

A fim de se encontrar curvas com o menor parâmetro b possível, sendo ele da forma $b = c^4 + d^6$, fixou-se a variável de maior grau, d^6, igual a um, obtendo-se $b = c^4 + 1$. A partir dessa parametrização, foram procuradas as curvas para cada tamanho de corpo.

Todas as curvas da Tabela 2 possuem a forma $E_{c^4+1,\ell} : y^2 = x^3 + (c^4 + 1)$ sobre $\mathbb{F}_{p(u)}$, ordem prima $n(u)$, e admitem um *twist* de ordem correta dado por $E' : y'^2 = x'^3 + (c^2 - i)$ sobre $\mathbb{F}_{p^2}$. O parâmetro c é sempre uma potência de 2. Estes parâmetros foram obtidos

a partir da criação de um *script* na linguagem Magma (BOSMA et al., 1997) que procura números primos com propriedades pré-determinadas.

Também é importante notar que os valores para u e c determinam unicamente todos os parâmetros necessários, i.e. os primos p e n, as equações da curva para E e seu *twist* bem como os pontos geradores. As extensões de corpos $\mathbb{F}_{p^{2r}}$ podem ser representadas, se desejado, diretamente como $\mathbb{F}_{p^2}[z]/(z^r - c^2 - i)$ por $r = 2, 3, 6$, ou via torres como indicado no Exemplo 2.

Os grupos de emparelhamento são $\mathbb{G}_1 = \langle G \rangle$ para $G = (-1, c^2)$, e $\mathbb{G}_2 = \langle G' \rangle$ para $G' = h \cdot (-i, c)$ com $h = p - 1 + t$, respectivamente. O baixo peso de u permite multiplicação muito eficiente pelo cofator h (SCOTT et al., 2009a, Seção 6).

A escolha peculiar $\ell := 32m - 2$ merece atenção, uma vez que é menor (embora não muito) do que um múltiplo de tamanhos típicos de palavra (mais precisamente, um múltiplo de 8 bits) e, portanto, leva a níveis de segurança que são ligeiramente menores que o usual. Isto foi feito para que, adotando aritmética de Montgomery no corpo base, todos os valores listados aqui permitam que todas as reduções modulares envolvidas em uma multiplicação ou quadrado em $\mathbb{F}_{p^2}$ sejam postergadas e efetuadas apenas uma única vez no final da operação, de maneira muito simples e eficiente como sugerido por (BEUCHAT et al., 2010, Seção 5.2).

O valor $\lfloor 2^{32m}/p \rfloor$ indica quantas reduções modulares podem ser adiadas uma vez que elementos em $\mathbb{F}_p$ são mantidos em variáveis de $32m$ bits. Com a escolha sugerida de $\ell = 32m - 2$, $\lfloor 2^{32m}/p \rfloor = 7$ para todos os exemplos na Tabela 2 exceto para a entrada em $\ell = 254$, onde são 6 (multiplicações ou quadrados em $\mathbb{F}_{p^2}$ não é necessário que este valor seja maior que 5).

Raízes quadradas em $\mathbb{F}_{p^2}$ podem ser eficientemente calculadas com o método sugerido, ou KCYL (KONG et al., 2006) ou com Müller (MÜLLER, 2004).

Exemplo 4: Os parâmetros para a curva de 254-bit definida por $u = -(2^{62} + 2^{55} + 1)$

m	ℓ	u	$\mathsf{wt}(6u+2)$	c	b	$\sqrt{\mathbb{F}_{p^2}}$
5	158	$-(2^{38}+2^{28}+1)$	5	2	17	KCYL
6	190	$-(2^{46}+2^{23}+2^{22}+1)$	5	8	4097	KCYL
7	222	$2^{54}-2^{44}+1$	5	4	257	Müller
8	254	$-(2^{62}+2^{55}+1)$	5	1	2	KCYL
9	286	$-(2^{70}+2^{58}+2^{38}+1)$	7	1	2	KCYL
10	318	$2^{78}+2^{62}+2^{1}+1$	6	1	2	KCYL
11	350	$-(2^{86}-2^{69}+2^{28}+1)$	7	1	2	KCYL
12	382	$-(2^{94}+2^{76}+2^{72}+1)$	7	1	2	KCYL
13	414	$-(2^{102}+2^{84}-2^{55}+1)$	7	1	2	KCYL
14	446	$2^{110}+2^{36}+1$	5	4	257	Müller
15	478	$-(2^{118}-2^{55}-2^{19}+1)$	7	1	2	KCYL
16	510	$-(2^{126}+2^{53}-2^{50}+1)$	6	4	257	KCYL
17	542	$-(2^{134}+2^{114}+2^{30}+1)$	7	1	2	KCYL
18	574	$-(2^{142}+2^{120}-2^{99}+1)$	7	1	2	KCYL
19	606	$-(2^{150}-2^{95}+2^{8}+1)$	7	1	2	KCYL
20	638	$2^{158}-2^{128}-2^{68}+1$	7	4	257	Müller

Tabela 2: Curvas de exemplo $E_{b,\ell}$

são $E_{2,254} : y^2 = x^3 + 2$, $G = (-1, 1)$, $E' : y'^2 = x'^3 + (1 - i)$, $G' = h \cdot (-i, 1)$. □

Exemplo 5: Todos os exemplos na Tabela 2 satisfazem a primeira forma de curvas amigáveis à implementação sugeridas. Como um exemplo da segunda forma, em um cenário onde o cálculo eficiente de raízes cúbicas é desejado pode-se adotar a curva de 254 bits (não listados na Tabela 2) definida por $u = -(2^{62} - 2^{49} - 2^2 + 1)$ são $E_{5,254} : y^2 = x^3 + 5$, $G = (-1, 2)$, $E' : y'^2 = x'^3 + (1 - 2i)$, $G' = h \cdot (-1, 1 - i)$. Pode-se verificar por inspeção direta que $p \equiv 4 \pmod 9$ para esta curva. □

A curva particular do Exemplo 4 foi aparentemente sugerida por (NOGAMI et al., 2008, Seção 4.2), e curvas com $c = 1$ (e portanto $b = 2$), que compõem a maioria da Tabela 2, foram ressaltadas em (SHIRASE, 2010), embora sem o benefício de uma visão unificada da equação da curva, seu *twist* correto, e os corpos finitos envolvidos como mostrado na Seção 2.4.4.

4.2 Tratamento do produto de conjugados para inversão no corpo de extensão $\mathbb{F}_{p^{12}}$

Para o tratamento da inversão em $\mathbb{F}_{p^{12}}$, são utilizados a notação do Exemplo 2 e a Seção 3.1.

Pode-se inverter $\gamma \in \mathbb{F}_{q^6} \setminus \{0\}$ calculando (veja e.g. (LAUTER et al., 2010, Seção 3.1))

$$\gamma^{-1} = \gamma^{v-1} \cdot \gamma^{-v},$$

onde $v := 1 + q + q^2 + q^3 + q^4 + q^5$. Definindo as quantidades $\lambda := \gamma^{1+q^3} \in \mathbb{F}_{q^3}$, $\mu := \lambda^q \cdot \lambda^{q^2} \in \mathbb{F}_{q^3}$, $\varepsilon := \gamma^v \in \mathbb{F}_q$, e $\eta := \mu \cdot \varepsilon^{-1}$, pode-se escrever $\varepsilon = \gamma^{1+q+q^2+q^3+q^4+q^5} = \lambda \cdot \mu$ e $\gamma^{v-1} = \gamma^{q+q^2+q^3+q^4+q^5} = \gamma^{q^3} \cdot \gamma^{q(1+q^3)} \cdot \gamma^{q^2(1+q^3)} = \gamma^{q^3} \cdot \lambda^q \cdot \lambda^{q^2} = \gamma^{q^3} \cdot \mu$, onde $\gamma^{-1} = \gamma^{q^3} \cdot (\mu \cdot \varepsilon^{-1}) = \gamma^{q^3} \cdot \eta$.

Escrevendo $\gamma = \alpha + \beta w$ para $\alpha, \beta \in \mathbb{F}_{q^3}$, percebe-se que $\lambda = (\alpha + \beta w)(\alpha - \beta w) = \alpha^2 - \beta^2 \xi$, onde os quadrados em $\mathbb{F}_{q^3}$ podem ser efetuados via o método de Chung-Hasan SQR_3 para $\mathbb{F}_{q^3}$ sobre $\mathbb{F}_q$, resultando um custo total de $2\tilde{m} + 8\tilde{s}$, enquanto $\mu = (\lambda \cdot \lambda^q)^q$ pode ser calculado pelo custo de $3\tilde{m} + 3\tilde{s}$, além da conjugação. Calcular o produto $\varepsilon = \lambda \cdot \mu$ requer apenas $3\tilde{m}$, uma vez sabe-se que esse valor recai no corpo $\mathbb{F}_q$. Computar $\eta = \mu \cdot \varepsilon^{-1}$ então acarreta uma inversão em $\mathbb{F}_q$ e uma multiplicação entre um elemento de $\mathbb{F}_{q^3}$ e outro de $\mathbb{F}_q$ (o que custa $3\tilde{m}$). Finalmente nos deparamos com a multiplicação $\gamma^{-1} = \gamma^{q^3} \cdot \eta$ entre um elemento de $\mathbb{F}_{q^6}$ e outro de $\mathbb{F}_{q^3}$, a qual custa $2 \cdot 6\tilde{m}$, além da conjugação.

Portanto, além das conjugações e uma inversão em $\mathbb{F}_q$, o custo total é $23\tilde{m} + 11\tilde{s} \approx 91m$, que apresenta certa relevância ao se comparar com o custo de $288m$ necessárias ao se calcular ingenuamente γ^{-1} como o produto de conjugados de forma sequencial.

Operação em $\mathbb{F}_{p^2}$	Correspondência em $\mathbb{F}_p$
$\tilde{m}$	$3m_u + 2r + 8a$
$\tilde{s}$	$2m_u + 2r + 3a$
$\tilde{m}_u$	$3m_u$
$\tilde{s}_u$	$2s_u$
$\tilde{r}$	$2r$
$\tilde{a}$	$2a$

Tabela 3: Correspondência entre operações de $\mathbb{F}_{p^2}$ e $\mathbb{F}_p$

4.3 Otimizações no emparelhamento Ate Ótimo

É instrutivo comparar a eficiência relativa da família proposta com os resultados disponíveis na literatura. Curvas com mesmo nível de segurança que $E_{2,254}$ do Exemplo 4 apareceram em (BEUCHAT et al., 2010) e (NAEHRIG et al., 2010). Os resultados são resumidos na Tabela 4.4.

Seguindo (ARANHA et al., 2011; BEUCHAT et al., 2010), denota-se por $\tilde{m}_u$ o número de multiplicações em $\mathbb{F}_{p^2}$ efetuadas sem reduções modulares, por $\tilde{s}_u$ o número de quadrados em $\mathbb{F}_{p^2}$ efetuados sem reduções modulares, por $\tilde{r}$ o número de reduções modulares em $\mathbb{F}_{p^2}$ (contando metade quando apenas uma redução modular simples em $\mathbb{F}_p$ é necessária), e por $\tilde{a}$ o número de adições/subtrações em $\mathbb{F}_{p^2}$ envolvidas no cálculo de um emparelhamento Ate ótimo, com o laço de Miller (LM), a exponenciação final (EF) e o custo total (CT). Também é denotado por m_u o número de multiplicações em $\mathbb{F}_p$ efetuado sem reduções modulares, por r o número de reduções modulares em $\mathbb{F}_p$, e por a o número de $\mathbb{F}_p$ adições/subtrações correspondendo ao número de operações em $\mathbb{F}_{p^2}$, com cada multiplicação $\tilde{m}$ em $\mathbb{F}_{p^2}$ ocasionando $3m_u + 2r + 8a$ e cada quadrado $\tilde{s}$ em $\mathbb{F}_{p^2}$ ocasionando $2m_u + 2r + 3a$.

Em resumo, pode-se formular a Tabela 3 de mapeamento de correspondências entre operações nos corpos $\mathbb{F}_{p^2}$ e $\mathbb{F}_p$:

As contagens fornecidas referem-se apenas à aritmética *on-the-fly* sem técnicas de pré-cálculos.

Os resultados da Tabela 4.4 na seção 4.1 se referem às seguintes decisões de implementação:

- Cálculos conjuntos de ponto-e-linha dentro do laço de Miller como sugerido por Costello *et al.* (COSTELLO et al., 2010, Seção 5) (veja também (COSTELLO et al., 2009, Seção 4)).

- Multiplicação dedicada em $\mathbb{F}_{p^{12}}$ para acumular os valores da função de linha, os quais são conhecidos por serem bastante esparsos para graus de mergulho pares (COSTELLO et al., 2010, Seção 3).

- Cálculo melhorado da parte pesada da exponenciação final como proposto por Scott *et al.* (SCOTT et al., 2009b).

- Quadrado em $\mathbb{F}_{p^{12}}$ via algoritmo de Chung-Hasan SQR_3 (CHUNG; HASAN, 2007), o qual afeta positivamente ambos os cálculos conjuntos de ponto-e-linha e a parte leve da exponenciação final após o laço de Miller;

- Quadrado melhorado de Granger-Scott (GRANGER; SCOTT, 2010) e quadrado comprimido (KARABINA, 2010) no subgrupo ciclotômico $G_{\phi_6(q)}$, o qual afeta positivamente a parte pesada da exponenciação final depois do laço de Miller e também contribui para a eficiência pós-processamento (e.g. exponenciação adicional) dos valores de emparelhamento, necessário em vários protocolos.

- Cuidadoso escalonamento do produto de conjugados durante o cálculo do inverso em $\mathbb{F}_{p^{12}}$ (e.g. (LAUTER et al., 2010, Seção 3.1)), a fim de concentrar a maioria das operações nos subcorpos.

- (Otimização menor) Simplificação das funções de linha finais que aparecem no emparelhamento Ate ótimo (NAEHRIG et al., 2010, Seção 3.2). Isso inclui omitir a terceira função de linha e multiplicar os valores esparsos junto dos dois restantes.

- (Otimização menor) Quando o parâmetro BN u é negativo, substituição da inversão extra necessária por uma conjugação após a exponenciação final (ARANHA et al., 2011).

O algoritmo do emparelhamento Ate ótimo foi implementado conforme descrito acima para a curva $E_{2,254}$ mostrada na Tabela 2. A implementação foi feita puramente em Java e não utiliza qualquer otimização em *Assembly* ou específica de processador. Trata-se, portanto, de uma implementação mais flexível, bastante independente de restrições de plataformas.

4.4 Resultados

Na otimização do emparelhamento Ate ótimo sobre a curva $y^2 = x^3 + 2$ foi obtida uma melhoria de 38% em multiplicações no corpo base em relação à melhor implementação conhecida de Beuchat et al (que não se utiliza da família proposta), que é apresentada na Tabela 4.4. As métricas comparativas utilizadas são o número equivalente de multiplicações m_u, o número de reduções modulares r e o número adições a no corpo base.

Após apresentarem uma versão preliminar do artigo referente aos resultados deste trabalho, Aranha *et al.* destacaram que Karabina *et al.* oferecem oportunidades adicionais de otimização para exponenciar elementos pertencentes ao subgrupo ciclotômico com expoentes esparsos quando se utiliza a subclasse BN aqui apresentada (veja (ARANHA et al., 2011; KARABINA, 2010)). Contudo, devido a diferentes escolhas de implementação motivadas por outro foco de pesquisa, Aranha *et al.* não atingem o nível completo de otimização acima descrito com o conjunto de técnicas recomendado.

Não se pode afirmar que as otimizações aqui efetuadas são ótimas, contudo uma implementação em C para um processador de alto nível usando a biblioteca Miracl (SHAMUS SOFTWARE, 2010) atinge cerca de 9% de melhora na velocidade do

fonte	parte	$\tilde{m}_u$	$\tilde{s}_u$	$\tilde{r}$	$\tilde{a}$	m_u	r	a
Naehrig *et al.*	LM	2022	590	2612	7140	NA	NA	NA
(NAEHRIG et al., 2010)	EF	673	1719	2392	7921	NA	NA	NA
	CT	2695	2309	5004	15061	NA	NA	NA
Beuchat et al.	LM	1952	568	2520	6912	6992	5040	13824
(BEUCHAT et al., 2010)	EF	403	1719	2122	7021	4647	4244	14042
	CT	2355	2287	4642	13933	11639	9284	27866
Aranha et al.	LM	1857	392	1335	10047	6355	2670	20094
(ARANHA et al., 2011)	EF	430	1179	963	8435	3648	1926	16870
	CT	2287	1571	2298	18482	10003	4596	36964
Este trabalho	LM	1256	1209	1169	10515	6186	2338	21030
	EF	1162	66	902	6795	3618	1804	13590
	CT	2418	1275	2071	17310	9804	4142	34620

Tabela 4: Comparação experimental de desempenho do emparelhamento Ate ótimo

cálculo do emparelhamento, em termos de m_u, em relação aos melhores resultados anteriores (ARANHA et al., 2011) para este exemplo particular, ou cerca de 38% a mais de velocidade do que os melhores resultados que não usam a vantajosa família apresentada (BEUCHAT et al., 2010). Uma implementação em Java para a mesma plataforma atinge uma velocidade de calculo de emparelhamento que é 7% maior do que as técnicas em (ARANHA et al., 2011) e 70% maior que a proposta em (BEUCHAT et al., 2010). Estes resultados são complementados com o bônus de aritmética melhorada e mais simples (incluindo extração de raiz quadrada mais rápida) nos corpos finitos $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$, os grupos subjacentes $\mathbb{G}_1$ e $\mathbb{G}_2$ conforme é necessário para a maioria dos protocolos baseados em emparelhamento. Automaticamente, isso também se mantém para aritmética em $\mathbb{G}_T$ possibilitado pela natureza amigável a torres da representação de corpo finito sugerida.

A maioria das técnicas listadas são aplicáveis a diferentes escolhas de parâmetros bem como para outras famílias de curvas elípticas amigáveis a emparelhamento. Esta subfamília foi escolhida para favorecer os melhores casos de cada técnica e, portanto, maximizar seu potencial para processamento eficiente.

4.4.1 Comparação experimental

A partir dos resultados da implementação do cálculo do emparelhamento Ate ótimo na Tabela 4.4 para a curva $y^2 = x^3 + 2$ do Exemplo 4, pode-se fazer um paralelo dos trechos de otimização no emparelhamento e as técnicas relacionadas. Além disso, são comparados resultados anteriormente publicados. Nota-se também que os resultados obtidos por Aranha *et al.* também estão apresentados nas figuras, mas utilizam a mesma subfamília e por isso os efeitos não são tão grandes.

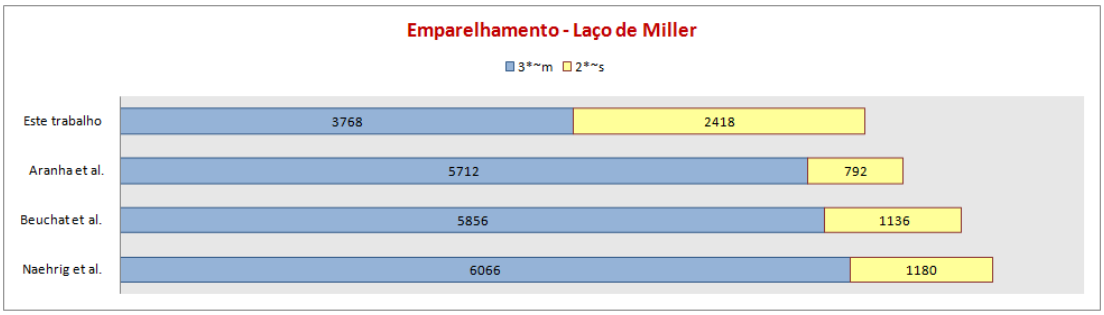

Figura 1: Comparação do número de $\tilde{m}_u$ e $\tilde{s}_u$ no laço de Miller

Para a implementação relacionada ao laço de Miller, a Figura 1 ilustra os custos do número de multiplicações, quadrados, reduções modulares e adições no corpo de extensão $\mathbb{F}_{p^2}$. As técnicas relacionadas a este trecho são listadas a seguir:

- Cálculos conjuntos de ponto-e-linha (adição e duplicação) dentro do laço de Miller como sugerido por (COSTELLO et al., 2010).
- Quadrado no corpo $\mathbb{F}_{p^2}$ melhorado com o algoritmo SQR_3 (CHUNG; HASAN, 2007).

A Figura 2 mostra que, com o conjunto de técnicas adotadas foi possível diminuir o número equivalente m_u mesmo com o aumento do número $\tilde{m}_u$. Isso de deve principalmente ao fato dos algoritmos adotados minimizarem drasticamente o número de quadrados $\tilde{s}$ efetivos. As técnicas relacionadas a este procedimento são

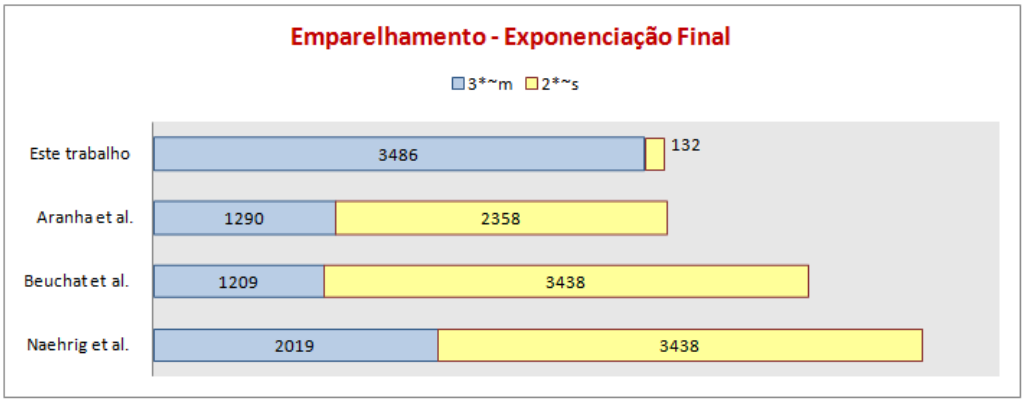

Figura 2: Comparação do número de $\tilde{m}_u$ e $\tilde{s}_u$ na exponenciação final

- A parte pesada da exponenciação final foi melhorada com
  - Técnica de exponenciação proposta em (SCOTT et al., 2009b) que pode ser usada em qualquer subgrupo de $\mathbb{F}_{p^{12}}$.
  - Quando o expoente é grande e esparso, no caso o parâmetro u de baixo peso de Hamming, utiliza-se a técnica de exponenciação associada a operação de quadrado comprimido apresentada em (KARABINA, 2010).
  - Após a parte leve da exponenciação final, o elemento pertence ao subgrupo ciclotômico e, a partir de então, utiliza-se a técnica de quadrado neste subgrupo proposta em (GRANGER; SCOTT, 2010).
  - Cuidadoso escalonamento do produto de conjugados no inverso em $\mathbb{F}_{p^{12}}$, concentrando as operações nos subcorpos 4.2.

O custo total, ilustrado na Figura 3, é o efeito da combinação obtida pela minimização do número $\tilde{m}_u$ no laço de Miller com a minimização do número $\tilde{s}_u$ na exponenciação final. O efeito causado teve maior influência no número $\tilde{s}_u$ porporcionando a dimunuição da métrica m_u. Alguns comentários sobre o custo total são dados a seguir

- Como o parâmetro BN u é negativo para a curva utilizada, foi possível aplicar um truque proposto em (ARANHA et al., 2011) que economiza uma inversão no

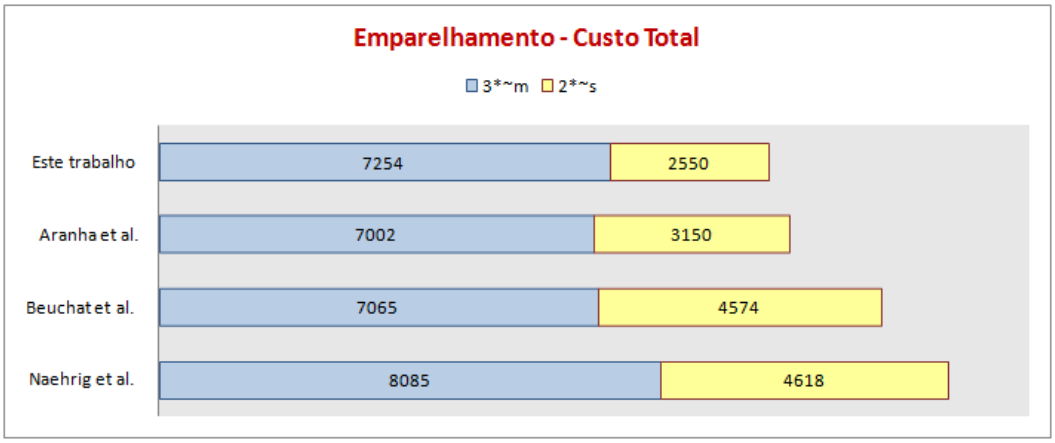

Figura 3: Comparação do número de $\tilde{m}_u$ e $\tilde{s}_u$ para o custo total

corpo $\mathbb{F}_{p^{12}}$ necessária após o laço de Miller, que custa $91m_u$ em troca de uma conjugação leve custando apenas $6a$.

- A eficiência pós-processamento (e.g. exponenciação adicional necessária em vários protocolos) dos valores de emparelhamento fica garantida, pois é efetuada no subgrupo ciclotômico que possibilita o uso de algoritmos mais leves.

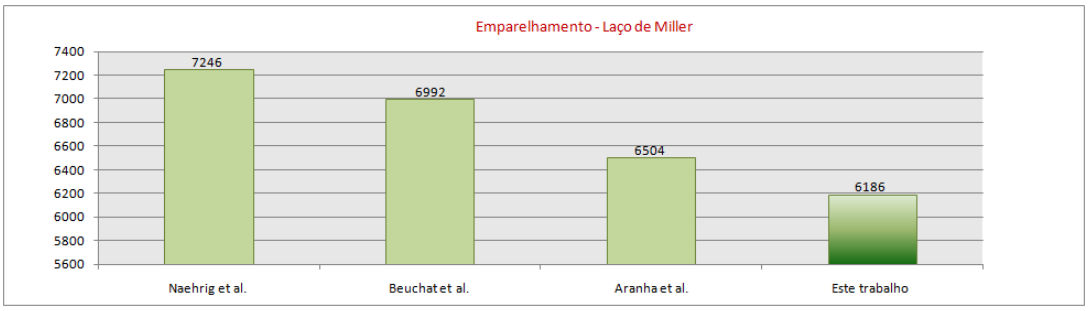

Figura 4: Comparação do número de m_u no laço de Miller

Analisando-se o efeito final das técnicas relacionadas apenas ao laço de Miller, obtemos a Figura 4. A seguir é feita a uma comparação da evolução entre os trabalhos.

- Otimização em % entre cada trabalho
  - $Naehrig \overset{3,5\%}{\rightarrow} Beuchat \overset{7\%}{\rightarrow} Aranha \overset{5\%}{\rightarrow}$Este trabalho.

Assim como para o laço de Miller, é avaliado também o efeito equivalente na exponenciação final, ilustrado na Figura 5.

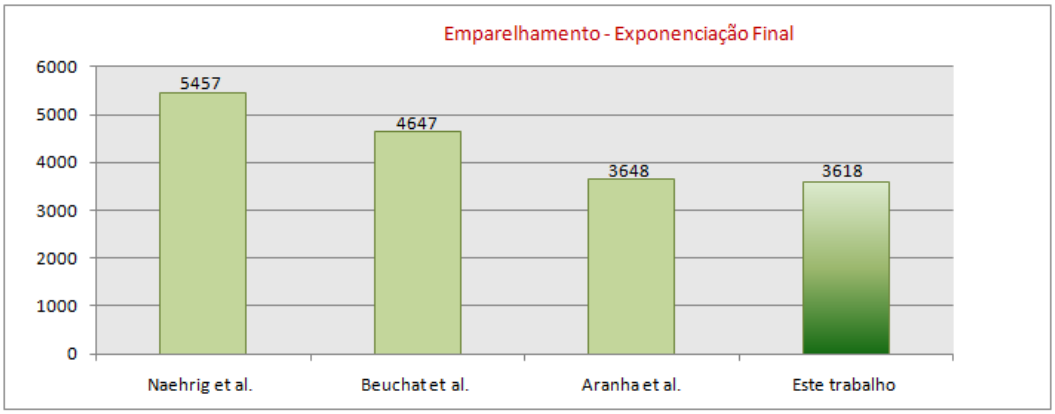

Figura 5: Comparação do número de m_u na exponenciação final

- Otimização em % entre cada trabalho
  - $Naehrig \overset{15\%}{\rightarrow} Beuchat \overset{21,5\%}{\rightarrow} Aranha \overset{1\%}{\rightarrow}$Este trabalho.

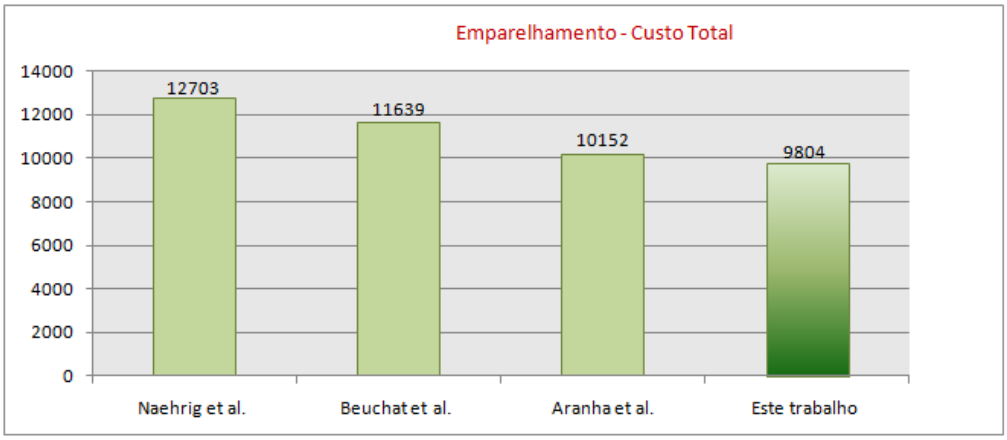

Figura 6: Comparação do número de m_u no custo total

Agora, do ponto de vista do emparelhamento completo, pode-se observar o efeito final representado na Figura 6.

- Otimização em % entre cada trabalho
  - $Naehrig \overset{8\%}{\rightarrow} Beuchat \overset{13\%}{\rightarrow} Aranha \overset{3,5\%}{\rightarrow}$Este trabalho.

4.5 Aplicação da subfamília BN ao protocolo de cifra-assinatura BDCPS

Atualmente, assinaturas digitais vêm substituindo as assinaturas de próprio punho e o antigo carimbo, evitando assim, a presença física do signatário no procedimento de autenticação em um sistema. Além disso, não mais se necessita de um operador para fazer a verificação da assinatura, pois um dispositivo com processamento é capaz de realizar até milhões de vezes esse procedimento por dia.

O desafio surge quando tal dispositivo apresenta baixas capacidades de processamento, armazenamento e largura banda, ou ainda, quando enorme demanda de processamento é requerida em um curto período de tempo. Um cartão SIM ou o próprio chip, encontrados nos aparelhos celulares, são bons exemplos. Um processador típico do cartão SIM, o ATMega128L, apresenta velocidade de *clock* de 7MHz e fornece 128KiB de memória RAM que são bastante reduzidos quando comparados aos 2GHz de *clock* e 3GB de RAM encontrados em um *desktop*.

Dado o cenário mais limitado, conforme descrito anteriormente, é necessário que as operações realizadas pelo protocolo de assinatura sejam rápidas a ponto de tratar um grande número de verificações quando executadas em um *desktop* ou de tratar uma única verificação rapidamente quando executada num dispositivo de capacidade limitada.

O protocolo de assinatura implementado, o BDCPS (BARRETO et al., 2008), é baseado em emparelhamento e foi desenvolvido e publicado pelo autor e colaboradores no simpósio SBSeg'08. Contudo, durante o desenvolvimento deste trabalho, foi produzida uma errada para o protocolo, sendo proposta no Apêndice B deste documento.

Agora, é ilustrado como se aplicar a parametrização aqui desenvolvida para encontrar os parâmetros referentes a um nível de segurança especificado.

Para obter um nível de segurança de, aproximadamente, 80 bits de uma curva da

família proposta, pode-se usar o parâmetro de segurança $k = 158$ bits.

Com $k = 158$, utiliza-se um *script* que busca um parâmetro BN u que forneça parâmetros $p(u)$ e $n(u)$ primos. A preferência é escolher u com baixo peso de Hamming e, então, podemos usar os seguintes valores práticos encontrados:

$$
\begin{aligned}
&u = -0x4000800023, \ ou \\
&u = (1000000000000001000000000000000100011)_2, \\
&p(u) = 0x24012003af565be6394ac09e1e2d54a04c50525b, \\
&n(u) = 0x24012003af565be6394a609c9e2b6b9f7a5035a5.
\end{aligned}
$$

Escolhendo-se os parâmetros $c, d \neq 0$ de forma a obter o menor valor possível para $b = c^4 + d^6$, temos:

$$
\begin{aligned}
&c = d = 1, \\
&b = c^4 + d^6 = (1)^4 + (1)^6 = 2, \\
&\xi = c^2 + d^3 i = 1^2 + 1^3 i = 1 + i, \\
&\bar{\xi} = 1 - i.
\end{aligned}
$$

A curva definida sobre o corpo base $\mathbb{F}_{p(u)}$ e seu gerador G são obtidos por:

$$
\begin{cases}
E : y^2 = x^3 + 2, \\
G = (-d^2, c^2) = (-1, 1).
\end{cases}
$$

e seu *twist* definido sobre o corpo de extensão $\mathbb{F}_{p(u)^2}$ com o respectivo gerador são calculados da seguinte forma:

$$
\begin{cases}
b_{twist} = b/\xi = (\xi \cdot \bar{\xi})/\xi = \bar{\xi} = (e - fi) = 1 - i, \\
E_{twist} : y'^2 = x'^3 + (1 - i), \\
G_{twist} = h(-di, c) = 1(-i, 1).
\end{cases}
$$

Sobre as curvas definidas acima foi, então, implementado em Java o protocolo de cifrassinatura utilizando a subfamília BN assim como o emparelhamento Ate ótimo.

Para efeitos de ilustração, o protocolo também foi instanciado sobre uma curva da subfamília MNT4 e o emparelhamento Ate. O tamanho do corpo base para um

nível de segurança equivalente para estas curvas é $k = 256$ bits. A Tabela 5 ilustra os *benchmarks* executados em um processador *Intel Core 2 Duo* com 2GHz de *clock*. As operações de emparelhamento, multiplicação, exponenciação e adição são denotadas por e, m, exp e a, respectivamente.

Algoritmo	Tempo(MNT4)	Tempo(BN)	e	op. $\in \mathbb{G}_T$	op. $\in \mathbb{G}_1$	op. $\in \mathbb{G}_2$
Private-Key-Extract	31ms	3ms	-	-	m	-
Check-Private-Key	90ms	31ms	1	-	-	m+a
Set-Public-Value	5ms	22ms	-	exp	-	-
Set-Public-Key	28ms	30ms	-	exp	m	-
Public-Key-Validate	95ms	50ms	1	exp+m	-	m+a
Signcrypt	11ms	22ms	-	exp	-	-
Unsigncrypt	16ms	44ms	-	2exp+m	-	-

Tabela 5: *Benchmarks* do protocolo BDCPS em um *desktop*.

Note que os algoritmos da Tabela 5 que se baseiam principalmente na aritmética em $\mathbb{G}_T$ como Set-Public-Value, Signcrypt e Unsigncrypt, apresentam tempos menores para a curva MNT4 pelo fato do corpo usado pelo grupo $\mathbb{G}_T$ ser menor. Tal grupo se baseia no corpo de extensão $\mathbb{F}_{p^4}$ onde p tem 256 bits e, portanto, p^4 tem 1024 bits. Enquanto isso, a aritmética no grupo $\mathbb{G}_T$ para as curvas BN é feita sobre o corpo de extensão $\mathbb{F}_{p^{12}}$, com p possuindo 158 bits, porém, a extensão p^{12} contém 1896 bits sendo razoavelmente maior que os 1024 bits usados em MNT4.

Note que para o caso crucial, os algoritmos mais caros, ou seja, os que aplicam emparelhamento, como Check-Private-Key e Public-Key-Validate, a curva BN é a mais indicada, apresentando grande vantagem (tempo 3 vezes menor no algoritmo Check-Private-Key).

Percebe-se ainda que, quando a aritmética é realizada no grupo $\mathbb{G}_1$ (definido sobre $\mathbb{F}_p$), tem-se um enorme ganho com a curva BN (veja o algoritmo Private-Key-Extract) uma vez que o grupo $\mathbb{G}_1$ na curva BN implica aritmética no corpo $\mathbb{F}_p$ com p de 158 bits e, na curva MNT4, o corpo tem 256 bits.

Em vista dos *trade-offs* mencionados, podem-se aproveitar os ganhos da subfamília BN na aritmética em $\mathbb{G}_1$ e no emparelhamento, sendo possível fazer uma modifica-

ção no protocolo para transferir as operações dos algoritmos para tais domínios mais vantajosos. A Tabela 5 foi então reescrita segundo o novo mapeamento das operações. Além disso, para contextualizar os resultados em um cenário de um dispositivo com capacidade mais restrita, os novos *benchmarks* foram executados em um aparelho celular, modelo Nokia 6275, com cerca de 150MHz de *clock*. Os tempos são mostrados nas Tabelas 6 e 7. Note que o grupo menos relevante $\mathbb{G}_2$ foi omitido.

Algoritmo	Tempo(MNT4)	Tempo(BN)	e	op. $\in \mathbb{G}_T$	op. $\in \mathbb{G}_1$
Private-Key-Extract	7.9s	0.8s	-	-	m
Set-Public-Value	1.4s	6.3s	-	exp	-
Set-Public-Key	7.9s	0.8s	-	exp	m
Check-Private-Key	23.0s	7.8s	1	-	-
Public-Key-Validate	23.8s	12.5s	1	exp+m	-
Signcrypt	3.2s	6.4s	-	exp	-
Unsigncrypt	3.8s	10.3s	-	2exp+m	-

Tabela 6: *Benchmarks* do BDCPS no Nokia 6275. Segurança de 80 bits.

Algoritmo	Tempo(MNT4)	Tempo(BN)	e	op. $\in \mathbb{G}_T$	op. $\in \mathbb{G}_1$
Private-Key-Extract	7.9s	0.8s	-	-	m
Set-Public-Value	4.5s	0.2s	-	-	m
Set-Public-Key	7.9s	0.8s	-	-	2m
Check-Private-Key	23.0s	7.8s	1	-	-
Public-Key-Validate	81.8s	10.3s	2	m	m
Signcrypt	9.0s	0.2s	-	-	m
Unsigncrypt	13.6s	0.6s	-	m	2m

Tabela 7: *Benchmarks* da variante BDCPS no Nokia 6275. Segurança de 80 bits.

Após a modificação do protocolo e sua implementação, nota-se que todos os algoritmos passam a ser mais rápidos para a subfamília BN (Tabela 7). Outro resultado importante foi a obtenção de alguns tempos abaixo do limiar de percepção humano. Por outro lado, as operações que consomem tempo maior que um segundo são executadas apenas uma vez para validação de chave privada (Check-Private-Key) ou uma única vez para validar cada novo contato (Public-Key-Validate).

4.6 Sinopse

Este capítulo explorou aspectos práticos de implementação de curvas BN amigáveis, incluindo exemplos para níveis realísticos de segurança, otimizações e comparação com trabalhos relacionados. Descreveu-se também uma aplicação completa na forma de um protocolo de cifrassinatura desenvolvido originalmente pelo autor e colaboradores, mas também implementado sobre outra família de curvas elípticas.

5 CONCLUSÕES

Este trabalho apresentou uma subfamília de curvas Barreto-Naehrig que, genericamente, favorece a implementação eficiente através de uma descrição bastante simples. A análise aborda não só a velocidade do emparelhamento mas também a eficiência das operações aritméticas típicas necessárias para se instanciar protocolos baseados em emparelhamento. É dada ênfase no suporte a oportunidades de otimização num cenário com maior variedade de plataformas possível, em vez de se concentrar em uma plataforma específica.

Adotar uma curva BN com uma das formas descritas na Definição 16 (independente dos coeficientes c e d) é uma condição suficiente para

- evitar completamente testes de detecção de quadrado e cubo em $\mathbb{F}_p$ e $\mathbb{F}_{p^2}$ na fase de construção da curva;
- automaticamente sugerir representações de corpo finito apropriadas para aritmética eficiente;
- indicar o *twist* sêxtico correto diretamente;
- fornecer geradores simples para a curva e seu *twist*;
- possibilitar virtualmente todas as otimizações mais efetivas encontradas na literatura para todas as estruturas algébricas envolvidas.

5.1 Trabalhos futuros

Como indicação para pesquisa futura, destaca-se um problema em aberto considerando implementação eficiente de criptografia baseada em emparelhamento, que trata da operação de *hashing* de forma determinística e eficiente para os grupos $\mathbb{G}_1$ e $\mathbb{G}_2$. Embora este trabalho trate parcialmente tal problema, através do suporte dos algoritmos mais rápidos conhecidos para aritmética destes grupos (particularmente extração de raiz quadrada), técnicas mais avançadas de *hashing* como aquelas de Icart (ICART, 2009) não são atualmente aplicáveis para nenhuma curva BN. Encontrar um método seguro de *hashing* daquele tipo para esses grupos ou descrever uma subfamília de curvas BN onde tal método seja possível é de grande importância para muitos protocolos baseados em emparelhamento.

Salienta-se que a escolha aqui feita de representação para extensões de corpos finitos pode favorecer a implementação de outras famílias de curvas elípticas amigáveis a emparelhamento (e.g. KSS, veja (FREEMAN et al., 2010)). Prosseguir com essa possibilidade, contudo, não foi do escopo deste trabalho.

REFERÊNCIAS

ARANHA, D. F.; KARABINA, K.; LONGA, P.; GEBOTYS, C. H.; LÓPEZ, J. Faster explicit formulas for computing pairings over ordinary curves. In: *Advances in Cryptology – EUROCRYPT 2011*. Tallinn, Estonia: Springer, 2011. (Lecture Notes in Computer Science). To appear.

ATKIN, A.; MORAIN, F. Elliptic curves and primality proving. *Math. Comp*, v. 61, p. 29–68, 1993.

AVANZI, R. M.; MIHAILESCU, P. Generic efficient arithmetic algorithms for PAFFs (processor adequate finite fields) and related algebraic structures. In: *In Selected Areas in Cryptology – SAC 2003*. Melbourne, Florida EUA: Springer, 2004. p. 320–334.

BAEK, J.; SAFAVI-NAINI, R.; SUSILO, W. Certificateless public key encryption without pairing. In: *Information Security Conference – ISC 2005*. Singapore: Springer, 2005. (Lecture Notes in Computer Science, v. 3650), p. 134–148.

BALASUBRAMANIAN, R.; KOBLITZ, N. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *Journal of Cryptology*, Springer-Verlag, Berlin, Germany, v. 11, n. 2, p. 141–145, 1998.

BARRETO, P.; KIM, H.; LYNN, B.; SCOTT, M. Efficient algorithms for pairing-based cryptosystems. In: *Advances in Cryptology – Crypto 2002*. Santa Barbara, California (USA): Springer, 2002. (Lecture Notes in Computer Science, v. 2442), p. 354–368.

BARRETO, P. S. L. M. *Criptografia Robusta e Marcas d'Água Frágeis - Construção e Análise de Algoritmos para Localizar Alterações em Imagens Digitais*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, Brasil, 2003.

BARRETO, P. S. L. M.; DEUSAJUTE, A. M.; CRUZ, E. S.; PEREIRA, G. C. F.; SILVA, R. Toward efficient certificateless signcryption from (and without) bilinear pairings. In: *The Brazilian Symposium on Information and Computer System Security (SBSeg)*. Gramado, RS, Brasil: Brazilian Computer Society (SBC), 2008. http://sbseg2008.inf.ufrgs.br/proceedings/data/pdf/st03_03_artigo.pdf.

BARRETO, P. S. L. M.; GALBRAITH, S.; HÉIGEARTAIGH, C. .; SCOTT, M. Efficient pairing computation on supersingular abelian varieties. *Designs, Codes and Cryptography*, Springer, Boston/Norwell (USA), v. 42, n. 3, p. 239–271, 2007. ISSN 0925-1022.

BARRETO, P. S. L. M.; LIBERT, B.; MCCULLAGH, N.; QUISQUATER, J.-J. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: *Advanced in Cryptology – ASIACRYPT 2005*. Chennai, India: Springer, 2005. (Lecture Notes in Computer Science, v. 3788), p. 515–532.

BARRETO, P. S. L. M.; LYNN, B.; SCOTT, M. Constructing elliptic curves with prescribed embedding degrees. In: *Proceedings of the Third Workshop on Security in Communications Networks – SCN 2002)*. Amalfi, Italy: Springer-Verlag, 2003. (Lecture Notes on Computer Science, v. 2576), p. 257–267.

BARRETO, P. S. L. M.; NAEHRIG, M. Pairing-friendly curves of prime order. In: *Selected Areas in Cryptography – SAC'2005*. Santa Fe, New Mexico (USA): Springer, 2006. (Lecture Notes in Computer Science, v. 3897), p. 319–331.

BARRETO, P. S. L. M.; NAEHRIG, M.; SCOTT, M. *Pairing-Friendly Curves of Prime Order with Embedding Degree 12*. 2007. IEEE P1363.3 Standard Specifications For Public-Key Cryptography – Identity Based Public Key Cryptography using Pairings. Technique submitted to standardization body.

BEUCHAT, J.-L.; DíAZ, J. E. G.; MITSUNARI, S.; OKAMOTO, E.; RODRíGUEZ-HENRíQUEZ, F.; TERUYA, T. High-speed software implementation of the optimal ate pairing over barreto-naehrig curves. In: *Proceedings of the 4th international conference on Pairing-based cryptography*. Yamanaka Hot Spring, Japan: Springer-Verlag, 2010. (Pairing'10), p. 21–39. http://portal.acm.org/citation.cfm?id=1948966.1948969.

BONEH, D.; FRANKLIN, M. Identity-based encryption from the Weil pairing. In: *Advanced in Cryptology – Crypto 2001*. Santa Barbara, California (USA): Springer, 2001. (Lecture Notes in Computer Science, v. 2139), p. 213–229.

BONEH, D.; GENTRY, C.; LYNN, B.; SHACHAM, H. Aggregate and verifiably encrypted signatures from bilinear maps. In: *Advances in Cryptology – Eurocrypt 2003*. Warsaw, Poland: Springer, 2003. (Lecture Notes in Computer Science, v. 2656), p. 416–432.

BOSMA, W.; CANNON, J.; PLAYOUST, C. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, v. 24, p. 235–265, 1997. http://dx.doi.org/10.1006/jsco.1996.0125.

CHUNG, J.; HASAN, M. A. Asymmetric squaring formulae. In: *IEEE Symposium on Computer Arithmetic – ARITH 2007*. Montpellier, France: IEEE Press, 2007. (Proceedings), p. 113–122.

CIPOLLA, M. Un metodo per la risoluzione della congruenza di secondo grado. *Rendiconto dell'Accademia delle Scienze Fisiche e Matematiche Napoli*, v. 9, p. 154–163, 1903. http://cr.yp.to/bib/entries.html#1903/cipolla.

COHEN, H.; FREY, G.; AVANZI, R. *Handbook of elliptic and hyperelliptic curve cryptography*. Boca Raton: Chapman and Hall/CRC, 2006. http://nla.gov.au/nla.cat-vn3799525.

COSTELLO, C.; HISIL, H.; BOYD, C.; NIETO, J. G.; WONG, K. K.-H. Faster pairings on special Weierstrass curves. In: *Pairing-Based Cryptography – Pairing 2009*. Palo Alto, CA (USA): Springer, 2009. (Lecture Notes in Computer Science, v. 5671), p. 89–101.

COSTELLO, C.; LANGE, T.; NAEHRIG, M. Faster pairing computations on curves with high-degree twists. In: *Public Key Cryptography – PKC 2010*. Paris, France: Springer, 2010. (Lecture Notes in Computer Science, v. 6056), p. 224–242.

CROSSBOW. *TelosB Datasheet*. 2008. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

DEVEGILI, A. J.; SCOTT, M.; DAHAB, R. Implementing cryptographic pairings over Barreto-Naehrig curves. In: *Pairing-Based Cryptography – Pairing 2007*. Tokyo, Japan: Springer, 2007. (Lecture Notes in Computer Science, v. 4575), p. 197–207.

DUPONT, R.; ENGE, A.; MORAIN, F. Building curves with arbitrary small mov degree over finite prime fields. *Journal of Cryptology*, v. 18, p. 79–89, 2002.

FAN, J.; VERCAUTEREN, F.; VERBAUWHEDE, I. Faster arithmetic for cryptographic pairings on Barreto-Naehrig curves. In: *Cryptographic Hardware and Embedded Systems – CHES 2009*. Lausanne, Switzerland: Springer, 2009. (Lecture Notes in Computer Science, v. 5747), p. 240–253.

FREEMAN, D. Constructing pairing-friendly elliptic curves with embedding degree 10. In: *Algorithmic Number Theory Symposium – ANTS-VII*. Berlin, Germany: Springer, 2006. (Lecture Notes in Computer Science, v. 4076), p. 452–465.

FREEMAN, D.; SCOTT, M.; TESKE, E. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, v. 23, n. 2, p. 224–280, 2010.

FREY, G.; MÜLLER, M.; RÜCK, H. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, v. 45, n. 5, p. 1717–1719, 1999.

GALBRAITH, S. D.; LIN, X.; SCOTT, M. Endomorphisms for faster elliptic curve cryptography on general curves. In: *Advanced in Cryptology – Eurocrypt 2009*. Cologne, Germany: Springer, 2009. (Lecture Notes in Computer Science, v. 5479), p. 518–535.

GALBRAITH, S. D.; PATERSON, K. G.; SMART, N. P. Pairings for cryptographers. *Discrete Applied Mathematics*, v. 156, n. 16, p. 3113–3121, 2008.

GALBRAITH, S. D.; SCOTT, M. Exponentiation in pairing-friendly groups using homomorphisms. In: *Pairing-Based Cryptography – Pairing 2008*. London (UK): Springer, 2008. (Lecture Notes in Computer Science, v. 5209), p. 211–224.

GOUVÊA, C. P. L.; LÓPEZ, J. C. Software implementation of pairing-based cryptography on sensor networks using the msp430 microcontroller. In: *Progress in Cryptology – Indocrypt 2009*. New Delhi, India: Springer, 2009. (Lecture Notes in Computer Science, v. 5922), p. 248–262.

GRANGER, R.; SCOTT, M. Faster squaring in the cyclotomic subgroup of sixth degree extensions. In: *Public Key Cryptography – PKC 2010*. Paris, France: Springer, 2010. (Lecture Notes in Computer Science, v. 6056), p. 209–223.

HANKERSON, D.; MENEZES, A. J.; VANSTONE, S. *Guide to Elliptic Curve Cryptography*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

HESS, F. Pairing lattices. In: *Pairing-Based Cryptography – Pairing 2008*. London (UK): Springer, 2008. (Lecture Notes in Computer Science, v. 5209), p. 18–38.

HESS, F.; SMART, N. P.; VERCAUTEREN, F. The eta pairing revisited. *IEEE Transactions on Information Theory*, v. 52, n. 10, p. 4595–4602, 2006.

HUSEMÖLLER, D. Elementary properties of the chord-tangent group law on a cubic curve. In: *Elliptic Curves*. New York: Springer, 2004. v. 111, p. 23–43. http://dx.doi.org/10.1007/0-387-21577-8_2.

HUSEMOLLER, D. *Elliptic curves*. Berlin: Springer-Verlag, 1987.

ICART, T. How to hash into elliptic curves. In: *Advanced in Cryptology – Crypto 2009*. Santa Barbara, California (USA): Springer, 2009. (Lecture Notes in Computer Science, v. 5677), p. 303–316.

JOUX, A. A one round protocol for tripartite diffie-hellman. In: *ANTS IV*. Brussels, Belgium: Springer, 2000. p. 385–394.

KARABINA, K. *Squaring in cyclotomic subgroups*. 2010. Cryptology ePrint Archive, Report 2010/542. http://eprint.iacr.org/.

KOBLITZ, N. Elliptic Curve Cryptosystems. *Mathematics of Computation*, v. 48, n. 177, p. 203–209, 1987. http://www.jstor.org/stable/2007884.

KONG, F.; CAI, Z.; YU, J.; LI, D. Improved generalized Atkin algorithm for computing square roots in finite fields. *Information Processing Letters*, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands, v. 98, p. 1–5, 2006.

LAUTER, K.; MONTGOMERY, P. L.; NAEHRIG, M. An analysis of affine coordinates for pairing computation. In: *Pairing-Based Cryptography – Pairing 2010*. Yamanaka Hot Spring, Japan: Springer, 2010. (Lecture Notes in Computer Science). To appear.

LEE, E.; LEE, H. S.; PARK, C.-M. Efficient and generalized pairing computation on {Abelian} varieties. *IEEE Transactions on Information Theory*, IEEE Press, v. 55, n. 4, p. 1793–1803, 2009.

LEHMER, D. H. Computer technology applied to the theory of numbers. In: *Studies in Number Theory*. [S.l.]: Mathematical Association of America, 1969. p. 117–151. http://cr.yp.to/bib/entries.html#1969/lehmer.

LIBERT, B.; QUISQUATER, J. J. Improved signcryption from q-Diffie-Hellman problems. In: *Security in Communication Networks – SCN 2004*. Amalfi, Italy: Springer, 2005. (Lecture Notes in Computer Science, v. 3352), p. 220–234.

LIDL, R.; NIEDERREITER, H. *Finite Fields*. [S.l.]: Cambridge University Press, 1983.

MENEZES, A.; VANSTONE, S.; OKAMOTO, T. Reducing elliptic curve logarithms to logarithms in a finite field. In: *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. New Orleans, Louisiana (USA): ACM, 1991. (STOC '91), p. 80–89. http://doi.acm.org/10.1145/103418.103434.

MENEZES, A. J.; OORSCHOT, P. C. V.; VANSTONE, S. A.; RIVEST, R. L. *Handbook of Applied Cryptography*. 1997.

MILLER, V. S. Use of elliptic curves in cryptography. In: *Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO'85*. New York, NY, USA: Springer-Verlag New York, Inc., 1986. p. 417–426. ISBN 0-387-16463-4. http://portal.acm.org/citation.cfm?id=18262.25413.

____. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, v. 17, n. 4, p. 235–261, 2004. See also "Short programs for functions on curves" 1986 unpublished manuscript, http://crypto.stanford.edu/miller/miller.pdf.

MIYAJI, A.; NAKABAYASHI, M.; TAKANO, S. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A, n. 5, p. 1234–1243, 2001.

MÜLLER, S. On the computation of square roots in finite fields. *Designs, Codes and Cryptography*, Kluwer Academic Publishers, Norwell, MA, USA, v. 31, n. 3, p. 301–312, 2004. http://dx.doi.org/10.1023/B:DESI.0000015890.44831.e2.

NAEHRIG, M. *Constructive and Computational Aspects of Cryptographic Pairings*. 151 p. Tese (Doutorado) — Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2009.

NAEHRIG, M.; BARRETO, P. S. L. M.; SCHWABE, P. On compressible pairings and their computation. In: *Progress in Cryptology – Africacrypt 2008*. Casablanca, Morocco: Springer, 2008. (Lecture Notes in Computer Science, v. 5023), p. 371–388.

NAEHRIG, M.; NIEDERHAGEN, R.; SCHWABE, P. New software speed records for cryptographic pairings. In: *Progress in Cryptology – Latincrypt 2010*. Puebla, México: Springer, 2010. (Lecture Notes in Computer Science, v. 6212). 109–123.

NOGAMI, Y.; AKANE, M.; SAKEMI, Y.; KATO, H.; MORIKAWA, Y. Integer variable χ-based ate pairing. In: *Pairing-Based Cryptography – Pairing 2008*. London (UK): Springer, 2008. (Lecture Notes in Computer Science, v. 5209), p. 178–191.

PEREIRA, G. C. C. F.; JR, M. A. S.; NAEHRIG, M.; BARRETO, P. S. L. M. A family of implementation-friendly BN elliptic curves. *Journal of Systems and Software*, Elsevier, 2011.

RIESEL, H. *Prime numbers and computer methods for factorization*. Cambridge, MA (USA): Birkhauser Boston Inc., 1985.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, ACM, New York, NY (USA), v. 21, p. 120–126, 1978. http://doi.acm.org/10.1145/359340.359342.

RUBIN, K.; SILVERBERG, A. Choosing the correct elliptic curve in the CM method. *Mathematics of Computation*, v. 79, p. 545–561, 2010.

SAKAI, R.; KASAHARA, M. ID based cryptosystems with pairing on elliptic curve. In: *SCIS 2003*. Hamamatsu, Japan: IEICE, 2003.

SCHNORR, C. P. Efficient signature generation by smart cards. *Journal of Cryptology*, v. 4, n. 3, p. 161–174, 1991.

SCOTT, M. Computing the Tate pairing. In: *Topics in Cryptology – CT-RSA 2005*. San Francisco, CA (USA): Springer, 2005. (Lecture Notes in Computer Science, v. 3376), p. 293–304.

SCOTT, M.; BENGER, N.; CHARLEMAGNE, M.; PéREZ, L. J. D.; KACHISA, E. J. Fast hashing to $\mathbb{G}_2$ on pairing friendly curves. In: *Pairing-Based Cryptography – Pairing 2009*. Palo Alto, CA (USA): Springer, 2009. (Lecture Notes in Computer Science, v. 5671), p. 102–113.

____. On the final exponentiation for calculating pairings on ordinary elliptic curves. In: *Pairing-Based Cryptography – Pairing 2009*. Palo Alto, CA (USA): Springer, 2009. (Lecture Notes in Computer Science, v. 5671), p. 78–88.

SHAMIR, A. Identity based cryptosystems and signature schemes. In: *Advances in Cryptology – Crypto'84*. Santa Barbara, California (USA): Springer, 1984. (Lecture Notes in Computer Science, v. 0196), p. 47–53.

SHAMUS SOFTWARE. *Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL) v.5.4.4*. 2010. http://www.shamus.ie/.

SHIRASE, M. *Barreto-Naehrig Curve With Fixed Coefficient*. 2010. IACR ePrint Archive, report 2010/134. http://eprint.iacr.org/2010/134.

SHOUP, V. *A Computational Introduction to Number Theory and Algebra*. 2004.

SILVERMAN, J. H. *Silverman JH. The Arithmetic of Elliptic Curves.* [S.l.]: Springer-Verlag, 1986.

SZCZECHOWIAK, P.; KARGL, A.; SCOTT, M.; COLLIER, M. On the application of pairing based cryptography to wireless sensor networks. In: *Proceedings of the second ACM conference on Wireless network security – WiSec '09*. Zurich, Switzerland: ACM, 2009. p. 1–12.

VERCAUTEREN, F. Optimal pairings. *IEEE Transactions on Information Theory*, IEEE Press, Piscataway, NJ, USA, v. 56, p. 455–461, 2010. http://dx.doi.org/10.1109/TIT.2009.2034881.

ZHANG, F.; SAFAVI-NAINI, R.; SUSILO, W. An efficient signature scheme from bilinear pairings and its applications. In: *Public Key Cryptography – PKC 2004*. Singapore: Springer, 2004. (Lecture Notes in Computer Science, v. 2947), p. 277–290.

ZHENG, Y. Digital signcryption or how to achieve cost(signature & encryption) « cost(signature) + cost(encryption). In: *Advanced in Cryptology – Crypto'97*. Santa Barbara, California (USA): Springer, 1997. (Lecture Notes in Computer Science, v. 1294), p. 165–179.

APÊNDICE A – COMPARAÇÃO COM AS CURVAS APRESENTADAS POR SHIRASE

Uma vez que muitas curvas na Tabela 2 aparecem em (SHIRASE, 2010), isto é, curvas com $b = 2$, é instrutivo comparar aquela proposta com este trabalho.

Em certo sentido, este trabalho generaliza a ideia de (SHIRASE, 2010) de eliminar testes de *twist* para $b = 2$ para todo b satisfazendo a Definição 16 (na verdade (SHIRASE, 2010) descreve também como fazer o mesmo para $b = \pm 16$, mas esta escolha não produz uma curva BN apropriada, uma vez que b deve ser um não-quadrado pelo Lema 1).

Contudo, (SHIRASE, 2010) não fornece uma visão unificada da equação da curva, seu *twist* correto (isto é, argumentavelmente, tão importante quanto encontrar a curva em si), e os corpos finitos envolvidos, ele também não fornece automaticamente geradores apropriados para *ambas E* e seu *twist E′*, uma tarefa que por outro lado exigiria calcular raízes quadradas explicitamente.

As fórmulas melhoradas de Costello *et al.* (COSTELLO et al., 2010) também permanecem completamente úteis enquanto b for pequeno (digamos, cabendo em uma única palavra de processador) e tiver baixo peso de Hamming, a qual é certamente uma possibilidade conforme indicado pelos exemplos na Tabela 2 (também no Exemplo 5).

A habilidade de calcular eficientemente raízes cúbicas é importante quando compressão tripla no emparelhamento é desejada. Para curvas BN isto é viável (conforme

destacado em (BARRETO; NAEHRIG, 2006, Seção 3.1)) por requerer $p \equiv 4 \pmod 9$. Infelizmente, isto é impossível para curvas com $b = 2$, que precisa do parâmetro u que determina que $p = p(u)$ seja $2, 11 \pmod{12}$ como indicado em (RUBIN; SILVERBERG, 2010; SHIRASE, 2010) e, portanto, implica em $p \equiv 1 \pmod 9$. Por outro lado, há curvas neste trabalho que de fato provêem esse caso, como indicado no Exemplo 5, onde $b = 5$.

Finalmente, curvas com $b = 2$ sozinhas não podem ser suficientes para todos os tamanhos de corpos desejados; os exemplos na Tabela 2 onde $b \neq 2$ ilustra tal fato.

APÊNDICE B – PROTOCOLO BDCPS CORRIGIDO

Este apêndice é dedicado a apresentar uma correção ao protocolo de cifrassinatura BDCPS (BARRETO et al., 2008). O BDCPS integra técnicas 1) de criptografia baseada em identidade BLMQ (BARRETO et al., 2005) para efetuar a validação de chave privada; 2) de assinaturas Schnorr (SCHNORR, 1991) compondo a funcionalidade de assinatura; 3) de cifrassinatura Zheng (ZHENG, 1997) que combina encriptação com a assinatura, produzindo um esquema eficiente auto-certificado de cifrassinatura. Pode-se construir tal esquema usando o modelo de criptografia em (BAEK et al., 2005), em que os usuários escolhem seu par de chaves convencional, mas não certificado, independentemente de suas chaves baseadas em identidade. Estas chaves são validadas posteriormente através de um mecanismo baseado em identidade.

O protocolo consiste dos seguintes algoritmos:

- **Setup:** dado um parâmetro de segurança k, este algoritmo escolhe um número primo n de k bits, grupos de mapa bilinear $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ de ordem n com suporte a um emparelhamento eficientemente computável, não-degenerado $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, geradores $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ e funções de hash
  $h'_0 : \mathbb{G}_T \times \{0,1\}^* \times \mathbb{G}_T \to \mathbb{Z}_n^*$,
  $h_1 : \{0,1\}^* \to \mathbb{Z}_n^*$,
  $h_2 : \mathbb{G}_T \to \{0,1\}^*$,
  $h'_3 : (\mathbb{G}_T \times \{0,1\}^*)^3 \to \mathbb{Z}_n^*$.

Uma chave mestra $s \overset{R}{\leftarrow} \mathbb{Z}_n^*$ também é escolhida, para a qual a chave pública $Q_{pub} = sQ \in \mathbb{G}_2$ é associada. O gerador $g = \hat{e}(P, Q) \in \mathbb{G}_T$ é incluído também entre os parâmetros públicos, que são $\texttt{params} = (k, n, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P, Q, Q_{pub}, g, h'_0, h_1, h_2, h'_3)$.

•**Private-Key-Extract:** tem como entrada o identificador $\mathsf{ID}_A \in \{0,1\}^*$ da entidade A e extrai a chave privada baseada em identidade $P_A = (h_1(\mathsf{ID}_A) + s)^{-1}P \in \mathbb{G}_1$.

•**Check-Private-Key:** A entidade A pode verificar a consistência da chave P_A testando se $\hat{e}(P_A, h_1(\mathsf{ID}_A)Q + Q_{pub}) = g$. Essa configuração é denominada estilo de chave Sakai-Kasahara (SAKAI; KASAHARA, 2003).

A correção aqui se trata da função de *hash* h_1, que deve receber apenas a identidade ID_A como parâmetro de entrada, e não mais o valor público y_A conforme descrito na versão publicada no SBSeg'08. O modelo que contempla a assinatura de uma chave pública y_A é o public key cryptography convencional e não é o que se deseja neste protocolo. O paradigma alvo para o qual o protocolo BDCPS foi desenvolvido é o *self-certified cryptography* (BAEK et al., 2005) ou auto-certificado.

•**Set-Secret-Value:** dados $\texttt{params}$, este algoritmo obtém $x_A \overset{R}{\leftarrow} \mathbb{Z}_n^*$ como o valor secreto da entidade A.

•**Set-Private-Key:** dada a chave privada parcial $P_A \in \mathbb{G}_1$ da entidade A e o valor secreto $x_A \in \mathbb{Z}_n^*$, este algoritmo atribui o par $(x_A, P_A) \in \mathbb{Z}_n^* \times \mathbb{G}_1$ ao par completo da chave privada da entidade A.

•**Set-Public-Value:** dado o valor secreto $x_A \in \mathbb{Z}_n^*$ da entidade A, calcula-se $y_A \leftarrow g^{x_A} \in \mathbb{G}_T$, o valor público de A.

•**Set-Public-Key:** dada a chave privada parcial $P_A \in \mathbb{G}_1$ de A, o valor secreto

$x_A \in \mathbb{Z}_n^*$ e o valor público correspondente $y_A = g^{x_A}$. O signatário obtém $u_A \overset{R}{\leftarrow} \mathbb{Z}_n^*$ e calcula

1. $r_A \leftarrow g^{u_A}$
2. $h_A \leftarrow h_0'(r_A, \mathsf{ID}_A, y_A)$
3. $S_A \leftarrow (u_A - x_A h_A) P_A$

A chave pública completa da entidade A é a tripla $(y_A, h_A, S_A) \in \mathbb{G}_T \times \mathbb{Z}_n^* \times \mathbb{G}_1$. Essa configuração é uma combinação de uma assinatura Schnorr (sob a chave x_A) com uma assinatura BLMQ (sob a chave P_A) sobre a identidade ID_A e o valor público y_A.

- **Public-Key-Validate:** dada a chave pública completa (y_A, h_A, S_A) de A, este algoritmo verifica se y_A tem ordem n (i.e. que $y_A^n = 1$) e calcula

1. $r_A \leftarrow e(S_A, h_1(\mathsf{ID}_A)Q + Q_{pub}) y_A^{h_A}$
2. $v_A \leftarrow h_0'(r_A, \mathsf{ID}_A, y_A)$

O *verifier* aceita a mensagem assinada se e somente se $v_A = h_A$. O processo de validação combina a verificação de uma assinatura de Schnorr com a assinatura BLMQ.

- **Signcrypt:** para encriptar $m \in \{0,1\}^*$ sob a chave pública $y_B \in \mathbb{G}_T$ do receptor previamente validada para a identidade ID_B e P_{pub}, e a chave privada do emissor $x_A \in \mathbb{Z}_n^*$, a chave pública $y_A \in \mathbb{G}_T$ e a identidade ID_A, o emissor obtém $u \overset{R}{\leftarrow} \mathbb{Z}_n^*$ e calcula

1. $r \leftarrow y_B^u$
2. $c \leftarrow h_2(r) \oplus m$
3. $h \leftarrow h_3'(r, m, y_A, \mathsf{ID}_A, y_B, \mathsf{ID}_B)$
4. $z \leftarrow u - x_A h$

O criptograma de assinatura é a tripla $(c, h, z) \in \{0,1\}^* \times \mathbb{Z}_n^2$.

- **Unsigncrypt:** dada a chave pública do emissor $y_A \in \mathbb{G}_T$ previamente validada para a identidade ID_A e P_{pub}, e a chave privada do receptor $x_B \in \mathbb{Z}_n^*$, a chave pública $y_B \in \mathbb{G}_T$ e a identidade ID_B, sob a recepção da tripla (c, h, z) o receptor verifica se $h, z \in \mathbb{Z}_n^*$ e calcula

  1. $r \leftarrow y_A^{hx_B} y_B^z$

  2. $m \leftarrow h_2(r) \oplus c$

  3. $v \leftarrow h_3'(r, m, y_A, \mathsf{ID}_A, y_B, \mathsf{ID}_B)$

  O receptor aceita a mensagem se e somente se $v = h$. A Equação 1 de recuperação do *nonce* é ligeiramente mais simples que sua versão Zheng devido ao estilo Schnorr adotado para cifrassinatura.

A análise de segurança foi esboçada na versão publicada e considera os modelo de segurança de cada assinatura de forma independente. Contudo, como o esquema consiste da combinação de vários modelos, acredita-se que o modelo de segurança mais apropriado para tal protocolo ainda não existe. Este tópico pode ser visto como uma tarefa de pesquisa futura.

APÊNDICE C – ALGORITMOS

Neste apêndice, são listados os principais algoritmos adotados, que foram capazes de fornecer os resultados apresentados na Tabela 4.4. O principal deles é dado pelo Algoritmo 2, que descreve o emparelhamento Ate ótimo e está desenvolvido na forma otimizada para parâmetros u negativos de baixo peso de Hamming.

Também são listados os algoritmos sugeridos por Costello *et al.* (Algoritmos 3 e 4) para otimizar o laço de Miller, reaproveitando o cálculo de adição e duplicação de ponto em coordenadas projetivas para calcular a respectiva função de linha.

Algoritmo 2 Emparelhamento Ate ótimo sobre curvas BN.

Entrada: $P \in \mathbb{G}_1, Q \in \mathbb{G}_2,\ \omega = |6u+2| = (1, \omega_{s-1}, \ldots, \omega_0)_2$

1: $R \leftarrow Q,\ f \leftarrow 1$

2: **for** $i \leftarrow s;\ i \geq 0;\ i--$ **do**

3: $\quad f \leftarrow f^2 \cdot l_{R,R}(P)$

4: $\quad R \leftarrow [2]R$

5: $\quad$ **if** $\omega_i = 1$ **then**

6: $\quad\quad f \leftarrow f \cdot l_{R,Q}(P),\ R \leftarrow R + Q$

7: $\quad$ **end if**

8: **end for**

9: **if** $u < 0$ **then**

10: $\quad f \leftarrow f^{p^6}$

11: **end if**

12: $Q_1 = \phi_p(Q),\ Q_2 = \phi_{p^2}(Q)$

13: $f \leftarrow f \cdot l_{R,Q_1}(P),\ \ R \leftarrow R + Q_1$

14: $f \leftarrow f \cdot l_{R,-Q_2}(P),\ R \leftarrow R - Q_2$

15: $f \leftarrow f^{p^6-1}$

16: $f \leftarrow f^{p^2+1}$

17: $f \leftarrow f^{(p^4-p^2+1)/n}$

Saída: $f = a_{opt}(Q, P)$

Algoritmo 3 Duplicação de ponto integrada com o cálculo da função de linha (COSTELLO et al., 2010).

Entrada: $Q = (x_2, y_2, z_2),\ x_2, y_2, z_2 \in \mathbb{F}_{p^2}$.

1: $A \leftarrow x_2^2$,

2: $B \leftarrow y_2^2$,

3: $C \leftarrow z_2^2$,

4: $D \leftarrow 3bC$,

5: $E \leftarrow (x_2 + y_2)^2 - A - B$,

6: $F \leftarrow (y_2 + z_2)^2 - B - C$,

7: $G \leftarrow 3D$,

8: $X_3 \leftarrow E \cdot (B - G)$,

9: $Y_3 \leftarrow (B + G)^2 - 12D^2$,

10: $Z_3 \leftarrow 4B \cdot F$,

11: $L_{1,0} \leftarrow 3A$,

12: $L_{0,1} \leftarrow -F$,

13: $L_{0,0} \leftarrow D - B$,

Saída: $R = 2Q = (X_3, Y_3, Z_3);\ l_{R,R}(P) = L_{0,1} + L_{1,0}u + L_{0,0}u^3$.

Algoritmo 4 Adição de ponto integrada com o cálculo da função de linha (COSTELLO et al., 2010).

Entrada: $P = (x_0, y_0),\ x_0, y_0 \in \mathbb{F}_p$; $Q = (x_1, y_1, z_1);\ x_1, y_1, z_1 \in \mathbb{F}_{p^2}$; $T = (x, y, z), x, y, z \in \mathbb{F}_{p^2}$.

1: $A \leftarrow x - z \cdot x_1$,

2: $B \leftarrow y - z \cdot y_1$,

3: $L_{0,1} \leftarrow A \cdot y_0$,

4: $L_{1,0} \leftarrow B \cdot -x_0$,

5: $L_{0,0} \leftarrow B \cdot x_1 - A \cdot y_1$,

6: $C \leftarrow A^2$,

7: $x \leftarrow x \cdot C$,

8: $C \leftarrow C \cdot A$,

9: $D \leftarrow B^2 \cdot z + C - 2x$,

10: $y \leftarrow B \cdot (x - D) - y \cdot C$,

11: $x \leftarrow A \cdot D$,

12: $z \leftarrow C$

Saída: $R = (x, y, z);\ l_{R,Q}(P) = L_{0,1} + L_{1,0}u + L_{0,0}u^3$

Algoritmo 5 Quadrado em $\mathbb{F}_{p^{12}}$ SQR_3 (CHUNG; HASAN, 2007).

Entrada: $a(t) = a_0 + a_1 t + a_2 t^2 \in \mathbb{F}_{p^{12}}$.

1: $c_0 = S_0 = a_0^2$,

2: $S_1 = (a_2 + a_1 + a_0)^2$,

3: $S_2 = (a_2 - a_1 + a_0)^2$,

4: $c_3 = S_3 = 2a_1 a_2$,

5: $c_4 = S_4 = a_2^2$,

6: $T_1 = (S_1 + S_2)/2$,

7: $c_1 = S_1 - T_1 - S_3$,

8: $c_2 = T_1 - S_4 - S_0$

Saída: $c(t) = c_0 + c_1 t + c_2 t^2 \in \mathbb{F}_{p^{12}}$.

Algoritmo 6 Quadrado no subgrupo ciclotômico $G_{\phi_6(q)}$ (GRANGER; SCOTT, 2010).

Entrada: $a(t) = a_0 + a_1 t + a_2 t^2 \in \mathbb{F}_{p^{12}}$.

1: $c_0 \leftarrow 3a_0^2 - 2\bar{a}_0$,

2: $c_1 \leftarrow 3ia_2^2 - 2\bar{a}_1$,

3: $c_2 \leftarrow 3a_1^2 - 2\bar{a}_2$

Saída: $c(t) = c_0 + c_1 t + c_2 t^2 \in \mathbb{F}_{p^{12}}$.

Algoritmo 7 Exponenciação em $G_{\phi_6(q)}$ baseada no quadrado do Algoritmo 6

Entrada: $a \in \mathbb{F}_{p^{12}};\ k \in \mathbb{Z}$

1: $b \leftarrow a$;

2: **for** $i \leftarrow bits(k) - 2;\ i \geq 0;\ i--$ **do**

3: $\quad b \leftarrow b^2$ (Algoritmo 6)

4: $\quad$ **if** $testaBit(k, i)$ **then**

5: $\quad\quad b \leftarrow b \cdot a$

6: $\quad$ **end if**

7: **end for**

Saída: $b = a^k,\ b \in \mathbb{F}_{p^{12}}$

Algoritmo 8 Exponenciação em $\mathbb{F}_{p^{12}}$ com quadrado comprimido de Karabina

Entrada: $a \in \mathbb{F}_{p^{12}};\ k \in \mathbb{Z};\ k$ esparso

1: $b \leftarrow a;\ c \leftarrow a$

2: **for** $i \leftarrow bits(k) - 2;\ i \geq 0;\ i - -$ **do**

3: $\quad b \leftarrow QuadradoComprimido(b)$

4: $\quad$ **if** $testaBit(k, i)$ **then**

5: $\quad\quad c \leftarrow Descomprime(b)$

6: $\quad\quad b \leftarrow b \cdot c$

7: $\quad$ **end if**

8: **end for**

Saída: $c = a^k,\ c \in \mathbb{F}_{p^{12}}$

APÊNDICE D – PUBLICAÇÕES DO AUTOR

1.Pereira, G.C.C.F. ; Simplicio Jr, M.A. ; Santos, M.A.S. ; Margi, C.B. ; Oliveira, B.T. ; Barreto, P.S.L.M. ; Ruggiero, W.V. **Lightweight Cryptographic Framework for Secure Mobile Applications**. Submitted to Journal of Systems and Software.

2.Pereira, G.C.C.F. ; Simplício Jr, M.A. ; Naehrig, M. ; Barreto, P.S.L.M. **A Family of Implementation-Friendly BN Elliptic Curves**. To appear in Journal of Systems and Software.

3.Santos, M.A.S. ; Margi, C.B. ; Simplício JR, M.A. ; Pereira, G.C.C.F. ; Oliveira, B.T. **Implementation of Data Survival in Unattended Wireless Sensor Networks Using Cryptography**. In: Fifth IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2010), 2010, Denver. The 35th IEEE Conference on Local Computer Networks (LCN), 2010. p. 961–967.

4.Barreto, P.S.L.M. ; Deusajute, A.M. ; Cruz, E. ; Pereira, G. ; Silva, R.R. **Toward Efficient Certificateless Signcryption from (and without) Bilinear Pairings**. In: Brazilian Symposium on Information and Computer Systems Security – SBSeg'2008, 2008, Gramado (RS). Proceedings of the 8th Brazilian Symposium on Information and Computer Systems Security – SBSeg'2008. Porto Alegre (RS) : Brazilian Computer Society (SBC), 2008.

5.Cruz, E. ; Pereira, G. ; Silva, R.R. ; Barreto, P.S.L.M. **Construção de um Sistema de SMS Seguro**. In: Workshop de Trabalhos de Iniciação Científica e Graduação (WTICG'2008), 2008, Gramado (RS). Anais do Workshop de Trabalhos de Iniciação Científica e Graduação (WTICG'2008), 2008.