Criptografia de Curvas Elípticas: Definição e Recomendações

Gustavo Pasqua de Oliveira Celani[1]

[1] Universidade Estadual de Campinas (UNICAMP)
Campinas - SP, Brasil
2019

gustavo_celani@hotmail.com

Abstract. *Elliptic Curve Cryptography (ECC), proposed by Koblitz and Miller in 1985, has been studied and applied bringing significant results to the area [Koblitz 2000]. However, not all curves are considered safe to the point of use in cryptographic systems. Their parameters are standardized by institutions to their properties become computationally inviolable.*

Resumo. *A criptografia de curvas elípticas (ECC), proposta por Koblitz e Miller em 1985, vêm sendo estudada e aplicada trazendo resultados um tanto quanto significativos para a área [Koblitz 2000]. Porém nem todas as curvas são consideradas seguras a ponto de serem utilizadas em sistemas criptográficos. Elas são padronizadas por instituições que recomendam parâmetros para que suas propriedades se tornem computacionalmente invioláveis.*

1. Introdução

A criptografia é uma das ferramentas mais importantes da área de segurança da informação. Ela garante a confidencialidade dos dados, limitando o acesso à informação tão somente para aqueles que possuem a devida autorização e conhecimento das convenções estabelecidas. As chaves criptográficas são utilizadas para controlar suas operações.

Existem inúmeros algoritmos criptográficos, podendo ser eles simétricos ou assimétricos. Na criptografia simétrica, existe uma única chave compartilhada entre as partes envolvidas, onde, a mesma chave é utilizada para cifrar e descifrar os dados. Já a criptografia assimétrica, empregada na ECC, utiliza-se de um par de chaves pública-privada. Seu funcionamento é explicado mais detalhadamente na sessão 2.1.

2. Revisão da Teoria

2.1. Criptografia Assimétrica

A criptografia assimétrica, também conhecida como "criptografia de chave pública", utiliza um par de chaves matematicamente relacionadas entre si para suas operações: uma pública e uma privada. É de fundamental importância que a chave privada permaneça em sigilo. Já a chave pública pode ser divulgada sem que ocorra nenhum comprometimento na segurança do sistema criptográfico.

Na operação de encriptação (cifragem) é utilizada a chave pública. Uma vez cifrada a mensagem só pode ser descriptografada (descifrada) pela chave privada. Estas operações garantem somente o sigilo da mensagem. Para se obter o não-repúdio é necessário utilizar o procedimento de assinatura digital. Neste processo a mensagem é assinada digitalmente utilizando a chave privada e todos que possuem a chave pública são capazes de garantir a procedência da mensagem.

2.2. Curvas Elipticas em Campos Finitos

Um campo finito pode ser definido, simplificadamente, como um conjunto numérico composto por um número finito de elementos. Exemplificando com um arquétipo de campo finito comumente utilizado neste contexto tem-se $\mathbb{F}_p^*$, onde p é um número primo. Este conjunto é definido por todos os inteiros entre 1 até $p-1$ ($\mathbb{F}_p^* = \{1, ..., p-1\}$) nos quais suas manipulações são feitas por meio da aritmética modular.

Curvas elípticas podem ser representadas sobre um campo finito. Uma curva elíptica $E(\mathbb{F}_p^*)$ é representada por um conjunto de pontos (x, y) de tal forma que satisfaça sua equação geral descrita pela Equação 1:

$$E(\mathbb{F}_p^*) \equiv y^2 \pmod{p} \equiv x^3 + ax + b \pmod{p} \tag{1}$$

Além de satisfazer sua equação característica, uni-se um ponto O chamado de "ponto no infinito"ao conjunto de saída. Também é necessário que os parâmetros constantes da equação pertençam ao campo finito descriminado. Por fim, são excluídas as curvas elípticas singulares. A Equação 2 expõe as condições citadas:

$$E(\mathbb{F}_p^*) \begin{cases} \{x, y, a, b, c, d, e\} \in \mathbb{F}_p^* \\ 4a^3 + 27b^2 \pmod{p} \neq 0 \\ \cup\{O\} \end{cases} \tag{2}$$

Para o contexto de criptografia de curvas elípticas é interessante citar as seguintes propriedades encontradas neste tipo específico de curvas [Laska 2012] [Portnoi 2005]:

I As curvas apresentam simetria com relação ao eixo x. Logo: $(x, -y) \in E(\mathbb{F}_p^*)$.

II A condição de não-singularidade ($4a^3 + 27b^2 \pmod{p} \neq 0$) garante que as duas derivadas parciais sejam diferentes de zero para todo $(x, -y) \in E(\mathbb{F}_p^*)$.

III Uma curva elíptica intersecta o eixo x ou uma única vez ou três vezes. Onde as intersecções ocorrem na inflexão da condição de não-singularidade, ou seja: $4a^3 + 27b^2 \pmod{p} > 0$ para a primeira ocorrência, e $4a^3 + 27b^2 \pmod{p} < 0$ para a segunda e terceira, se houver.

IV Para dois pontos distintos pertencentes à curva $E(\mathbb{F}_p^*)$, tem-se que a soma entre esses pontos resulta em um terceiro ponto também pertencente à curva. Matematicamente: $M(x_m, y_m) + N(x_n, y_n) = R(x_r, y_r)$, onde $\{M, N, R\} \in E(\mathbb{F}_p^*)$.

2.3. Problema do Logaritmo Discreto

O problema do logaritmo discreto (*DLP - Discrete Logarithm Problem*) expõe uma debilidade computacional no sentido de não ser capaz de resolvê-lo em tempo polinomial.

O DLP descreve que: dados um grupo $\mathbb{G}$ e $\{y, \alpha\} \in \mathbb{G}$ no qual y é potência de α. O logaritmo discreto de y na base α é o menor inteiro não negativo x tal que $\alpha^x = y$, denotado por $\log_\alpha y = x$. Onde não é possível calcular x em tempo hábil [Flose 2011] [Hankerson 2004].

Simplificadamente, o problema descreve uma função considerada de sentido único computacionamente. Conhecidos P e Q, é computacionamente impraticável encontrar o valor de k onde $Q = k \times P$. Enquanto o cálculo de Q é executado em tempo polinomial por computadores considerados comuns.

3. Criptografia de Curvas Elípticas

A criptografia de curvas elípticas (*ECC - Elliptical Curves Cryptography*) é um sistema criptográfico assimétrico (seção 2.1) que faz uso de curvas elípticas sobre um campo finito (seção 2.2) e está assegurado pelo problema do logaritmo discreto (seção 2.3).

3.1. Geração de Chaves

A geração da chave privada (d) é a eleição de um número inteiro positivo aleatório diferente de zero e pertencente ao campo finito de alta ordem n. Logo, tem-se que $\{d\} \in \mathbb{F}_n^*$. Este número deve ser mantido em sigilo absoluto para que não haja nenhuma transgressão ao algoritmo [López 2000].

Já para a chave pública (D), seleciona-se uma curva elíptica e um ponto base (G) pertencente à curva selecionada. Tanto a curva, quanto o ponto base escolhidos podem ser publicados sem que haja qualquer comprometimento do algoritmo. Então é realizada a multiplcação da chave privada d pelo ponto base G. Matematicamente: $D = d \times G$. O DLP garante que a chave privada d não será calculada mesmo D e G sendo elementos públicos que não necessitam de sigilo [López 2000].

Ao final desse processo, foram definidos a curva elíptica e o ponto base na qual o algoritmo se baseará para realizar suas operações. E um par de chaves pública-privada foi gerado a partir destes parâmetros.

4. Diffie-Hellman Baseado em Curvas Elípticas

A troca de chaves utilizando Diffie-Hellman pode utilizar-se de propriedades das curvas elípticas em sua metodologia (*ECDH - Elliptic Curve Diffie-Hellman*). O protocolo de acordo de chaves possibilita que um segredo seja compartilhado por meio de um canal inseguro de tal forma que somente as duas partes envolvidas tenham acesso à ele e que nenhuma terceira consiga calculá-lo ou até mesmo deduzí-lo.

O ECDH (Figura 1) funciona da seguinte maneira: parte-se da premissa que duas partes (A e B) necessitem, por algum motivo, de compartilhar um segredo. Para isto, A e B acordam parâmetros públicos da curva, nos quais um adversário (E) pode ter acesso. Estes parâmetros públicos são:

- p: Campo no qual a curva é definida.
- a, b: Valores que definiem a equação característica da curva (Equação 1).
- G_x: Coordenada X do ponto base.
- G_y: Coordenada Y do ponto base.
- n: Número primo referente à ordem do ponto base G.
- h: Cofator.

Acordados estes parâmetros publicamente, cada parte gera uma chave privada ($d_{\{A,B\}}$) e calcula seu par público ($D_{\{A,B\}}$) com base nos parâmetros previamente acordados (metodologia explicada na subseção 3.1). As chaves públicas são compartilhadas entre as A e B enquanto suas respectivas chaves privadas são mantidas em sigilo. O segredo compartilhado (K) será o resultado da multiplicação entre a chave privada com a chave pública da parte oposta ($K = d_{\{A,B\}} \times D_{\{B,A\}}$).

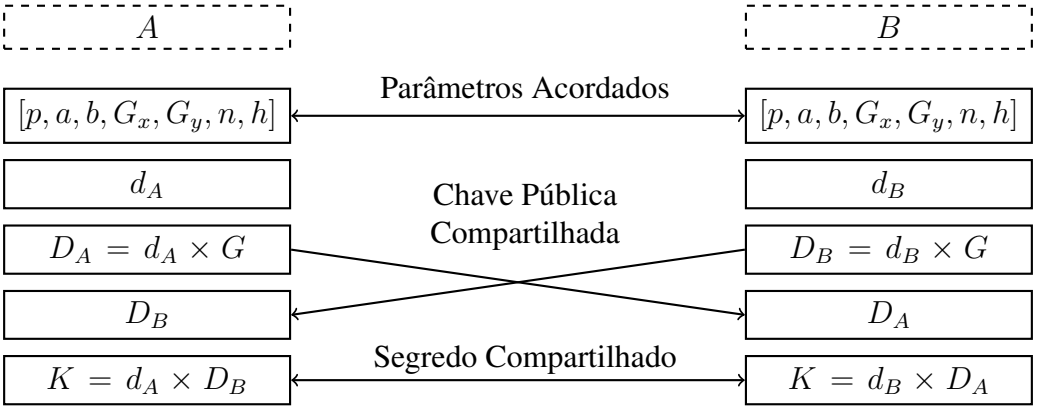

Figura 1. Diffie-Hellman baseado em Curvas Elípticas.

A equidade de K é matematicamente comprovado pelas Equações 3 à 6.

$$K_A = K_B \tag{3}$$

$$d_A \times D_B = d_B \times D_A \tag{4}$$

$$d_A \times (d_B \times G) = d_B \times (d_A \times G) \tag{5}$$

$$d_A \times d_B \times G = d_B \times d_A \times G \tag{6}$$

As chaves privadas envolvidas no protocolo ($d_{\{A,B\}}$) estão asseguradas pelo DLP (sessão 2.3). Logo, mesmo A possuindo K, d_A e G, ele não consegue calcular d_B. O mesmo ocorre com a parte B, que possui K, d_B e G e é incapaz de calcular d_A.

5. ElGamal Baseado em Curvas Elípticas

A criptografia de ElGamal pode ser aplicada sobre curvas elípticas uma vez que ela trabalha com um grupo. A seguir, são descritos os processos de encriptação/decriptação partindo da premissa que A deseja enviar uma mensagem sigilosa (m) para B por meio de um canal de comunicação inseguro [Lara 2010].

O primeiro passo é definir os parâmetros públicos da curva (vide sessão 4). Após acordar os parâmetros comuns entre as partes, cada uma delas gera seu par de chaves pública-privada (vide sessão 3.1). Feito isso, as chaves públicas são trocadas entre A e B. Todos estes compartilhamentos podem ser feitos em claro por um meio de canal inseguro de comunicação sem que haja comprometimento do protocolo. Neste ponto, ambas as partes conhecem os parâmetros públicos da curva e a chave pública de seu oposto.

O emissor (A) da mensagem computa o criptograma[1] (C) da seguinte maneira: $C = m + (d_A \times D_B)$. Uma vez computado, o criptograma é transmitido ao receptor (B), que por sua vez inicia o processo de decriptação para extrair a mensagem m do criptograma enviado pelo emissor. O processo de decriptação realizado por B se consiste em calcular m da seguinte forma: $m = C - (d_B \times D_A)$.

[1]Mensagem cifrada, codificada, criptografada

O protocolo é matematicamente comprovado pelas Equações 7 à 12.

$$\begin{cases} C = m_A + (d_A \times D_B) \\ m_B = C - (d_B \times D_A) \end{cases} \tag{7}$$

$$\begin{cases} C = m_A + (d_A \times D_B) \\ C = m_B + (d_B \times D_A) \end{cases} \tag{8}$$

$$m_A + (d_A \times D_B) = m_B + (d_B \times D_A) \tag{9}$$

$$m_A + [d_A \times (d_B \times G)] = m_B + [d_B \times (d_A \times G)] \tag{10}$$

$$m_A + \cancel{[d_A \times (d_B \times G)]} = m_B + \cancel{[d_B \times (d_A \times G)]} \tag{11}$$

$$m_A = m_B \tag{12}$$

6. Equivalências

Quando comparado às outras técnicas criptográficas conhecidas, como criptografia simétrica com DES, ou criptografia assimétrica utilizando RSA, o ECC se mostra mais eficiente do que o AES porém inferior ao DES, no que se refere ao tamanho da chave necessária. Na Tabela 1 e no gráfico da Figura 2, pode-se observar a equivência entre estes criptosistemas (comparação levando em consideração ataques conhecidos) [López 2000]:

Tabela 1. Equivalência de chaves entre diferentes algoritmos criptográficos.

Criptografia Simétrica		Criptografia Assimétrica		Curvas Elípticas
Algoritmo Exemplo	Tamanho da Chave [bits]	Algoritmo Exemplo	Tamanho da Chave [bits]	Tamanho da Chave [bits]
SKIPJACK	80	DSA/RSA	1024	160
Triple-DES	112	DSA/RSA	2048	224
128-bit DES	128	DSA/RSA	3072	256
192-bit DES	192	DSA/RSA	7680	384
256-bit DES	256	DSA/RSA	15360	512

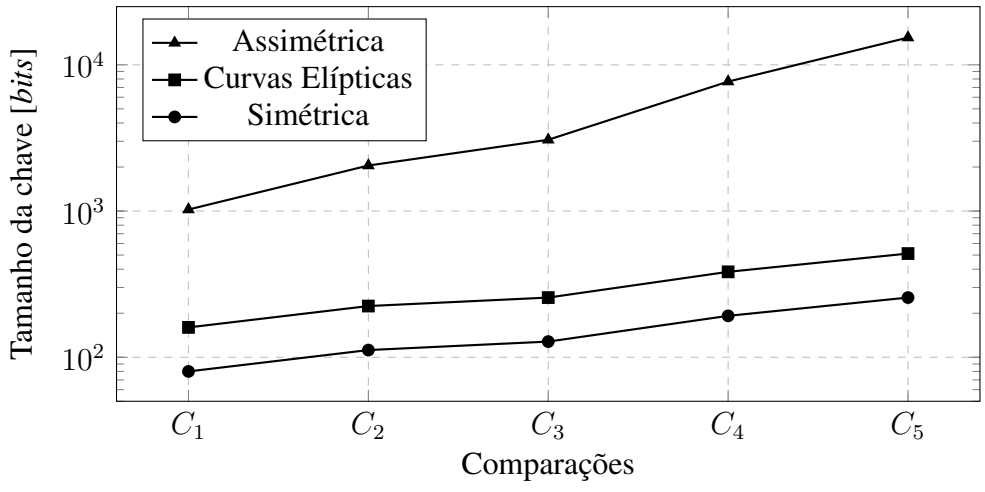

Figura 2. Equivalência de chaves entre diferentes algoritmos criptográficos.

7. Recomendações de Uso

Nem todas curvas elípticas são consideradas suficientemente seguras para serem utilizadas em sistemas criptográficos. Para obter uma curva elíptica segura para uso é necessário calcular seus parâmetros de tal forma que nenhuma propriedade seja violada.

Para evitar que todo criptógrafo que almeje fazer uso da ECC calcule esses parâmetros, existem entidades que os definem. Um exemplo é o NIST (*National Institute of Standards and Technology*), que publica um documento chamado *FIPS PUB 186-X Digital Signature Standard (DSS)* contendo parâmetros para serem utilizados em curvas para que as mesmas sejam consideradas seguras. A Tabela 2 apresentada alguns parâmetros recomendados para a curva P-256 publicados no *FIPS PUB 186-4* em 2013 [NIST 2013]:

Tabela 2. FIPS PUB 186-4 [NIST 2013]: Recomendação para ECC sobre P-256.

Parâmetro	Valor Recomendado
p	115792089210356248762697446949407573530 086143415290314195533631308867097853951
n	115792089210356248762697446949407573529 996955224135760342422259061068512044369
$Semente$	c49d3608 86e70493 6a6678e1 139d26b7 819f7e90
a	-3
b	5ac635d8 aa3a93e7 b3ebbd55 769886bc 651d06b0 cc53b0f6 3bce3c3e 27d2604b
c	7efba166 2985be94 03cb055c 75d4f7e0 ce8d84a9 c5114abc af317768 0104fa0d
G_x	6b17d1f2 e12c4247 f8bce6e5 63a440f2 77037d81 2deb33a0 f4a13945 d898c296
G_y	4fe342e2 fe1a7f9b 8ee7eb4a 7c0f9e16 2bce3357 6b315ece cbb64068 37bf51f5
h	1

8. Conclusão

A criptografia de curvas elípticas está ganhando espaço no que se refere à ser elegida para uso dentre todos algoritmos criptorgáficos disponíveis devido à sua eficiência. Foi visto que para garantir o mesmo nível de segurança de um AES utilizando chave de, por exemplo, 3072 bits, o ECC necessita de uma chave de apenas 256 bits. Isto representa que, para este caso, o ECC possui uma eficiência 12 vezes maior do que o AES. O desempenho do ECC é uma grande vantagem quando comparado com os demais algoritmos assimétricos.

Porém, para se atingir a segurança oferecida pelo ECC, é necessário eleger os parâmetros corretos para sua utilização, uma vez que não são todas as curvas elípticas que são seguras quando usadas em sistemas criptográficos. Este fator é o que, normalmente, torna as implementações de criptografia de curvas elípticas vulneráveis.

9. Trabalhos Futuros

No decorrer do estudo, foram identificados pontos que condescendem possíveis extensões na pesquisa. Um dos principais pontos a serem estudados seria desenvolver um aprofundamento em outros protocolos que utilizam ECC em seu funcionamento. Além do ECDH abordado neste trabalho, pode-se destacar:

- Elliptic Curve Digital Signature Algorithm (ECDSA)
- Elliptic Curve AES (ECAES)

Referências

Flose, V. B. S. (2011). Criptografia e Curvas Elípticas. *Universidade Estadual Paulista Júlio de Mesquita Filho - Instituto de Geociência e Ciências Exatas*. Disponível em <https://repositorio.unesp.br/bitstream/handle/11449/94347/flose_vbs_me_rcla.pdf>. Acessado em dez 2019.

Hankerson, Menezes, V. (2004). *Guide to Elliptic Curve Cryptography*. Springer.

Koblitz, Menezes, V. (2000). The State of Elliptic Curve Cryptography. *Kluwer Academic Publishers*. Disponível em <https://www.cse.iitk.ac.in/users/nitin/courses/WS2010-ref2.pdf>. Acessado em dez 2019.

Lara, O. (2010). Curvas Elípticas: Aplicação em Criptografia Assimétrica. *Laboratório Nacional de Computação Científica*. Disponível em <https://www.lncc.br/~borges/doc/Curvas\%20Elipticas\%20-\%20Aplicacao\%20em\%20Criptografia\%20Assimetrica.pdf>. Acessado em dez 2019.

Laska, T. (2012). Criptografia com curvas elípticas – uma pequena introdução. *Institut für Mathematik, Freie Universität Berlin*. Disponível em <http://random-ti.bplaced.net/ecc/curvas-elipticas.pdf>. Acessado em dez 2019.

López, D. (2000). An Overview of Elliptic Curve Cryptography. *State University of Campinas - Institute of Computing*. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.2771&rep=rep1&type=pdf>. Acessado em dez 2019.

NIST (2013). FIPS PUB 186-4 - Digital Signature Standard (DSS). *NIST - National Institute of Standard and Technology*. Disponível em <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>. Acessado em dez 2019.

Portnoi (2005). Criptografia com Curvas Elípticas. *Universidade Salvador - Núcleo de Pesquisa Interdepartamental em Redes de Computadores*. Disponível em <https://www.eecis.udel.edu/~portnoi/publications/criptografia_com_curvas_elipticas.pdf>. Acessado em dez 2019.