Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 12, n. 1, 2026.

Trabalho apresentado no XLIV CNMAC, Fundação Getulio Vargas - Rio de Janeiro - RJ, 2025.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Um Estudo Sobre Algoritmos de Ataque ao ECC

Hudsson D. A. de Andrade[1]
ICEI PUC MG, Belo Horizonte, MG
Neila M. G. de Oliveira[2]
DME PUC MG, Belo Horizonte, MG
Divane A. de M. Dantas[3]
DM CEFET MG, Belo Horizonte, MG

Com o avanço da tecnologia, muitos métodos criptográficos antigos, como a Cifra de César, tornaram-se vulneráveis. Já a criptografia com curvas elípticas (ECC) continua segura dentro de limites computacionais viáveis. Devido ao seu baixo custo computacional, a ECC é eficiente para dispositivos com recursos limitados, sendo usada em smartcards, IoT e criptomoedas.

A segurança da ECC baseia-se na dificuldade de resolver o problema do logaritmo discreto (PLD) em curvas elípticas. Esse problema surge na estrutura matemática dessas curvas, que são definidas pela equação de Weierstrass $y^2 = x^3 + ax + b$ sobre um corpo finito $\mathbf{F}_q$. Os pontos da curva, com a operação de soma, formam um grupo em que o PLD é formulado. Para curvas bem escolhidas, resolver o PLD é considerado computacionalmente inviável.

A complexidade computacional dos algoritmos utilizados para atacar o PLD é denotada por $O(f(t))$, em que f é uma função que descreve o crescimento do tempo de execução do algoritmo em função do parâmetro t. Nesse trabalho analisamos o custo computacional de alguns desses algoritmos. Em nossa análise, calculamos a complexidade baseada na quantidade de multiplicações de pontos, pois essa operação é fundamental para os cálculos em criptografia de curvas elípticas e influencia diretamente o custo computacional dos ataques. Usamos [3] como base para esses cálculos.

Implementamos operações fundamentais em curvas elípticas utilizando a linguagem de programação Python. Além disso, desenvolvemos e analisamos a implementação dos algoritmos de ataque ao PLD: o método de Pohlig-Hellman, o Big-Step Giant-Step e a abordagem de força bruta.

A abordagem de força bruta consiste em testar exaustivamente todas as possíveis chaves privadas até encontrar a correta. Sua complexidade é $O(n)$, em que n é a ordem do corpo $\mathbf{F}_q$. Isso significa que seu custo cresce linearmente com a ordem da curva e a chave privada. Embora ineficiente para curvas grandes, serve como referência para avaliar métodos mais avançados.

O método Big-Step Giant-Step otimiza a busca ao dividir o espaço de soluções em duas fases: uma pré-computação de valores (passos pequenos) armazenados em uma tabela e uma busca com saltos maiores (passos grandes). Sua complexidade é proporcional à raiz quadrada da ordem do grupo dos pontos da curva. Apesar de reduzir significativamente as operações em comparação à força bruta, exige mais memória para armazenar os valores pré-calculados.

Já o algoritmo de Pohlig-Hellman utiliza a fatoração da ordem do grupo da curva para dividir o problema em subproblemas menores, resolvendo cada um separadamente e combinando as soluções com o Teorema Chinês do Resto. Sua complexidade depende do somatório dos fatores primos da ordem do grupo, sendo especialmente eficiente quando esses fatores são pequenos. O desempenho varia pouco com a chave, pois depende principalmente da estrutura do grupo.

Para avaliar a eficácia dessas estratégias, realizamos uma análise da complexidade computacional da nossa implementação, considerando o número de operações de multiplicação de pontos

[1]hdaandrade@sga.pucminas.br
[2]neilaoliveira@pucminas.br
[3]divane@cefetmg.br

necessárias para encontrar a chave privada em um sistema baseado em ECC. Executamos os algoritmos em diferentes cenários e realizamos a contagem exata das multiplicações de pontos efetuadas em cada caso.

No exemplo analisado, utilizamos o corpo $\mathbf{F}_{7919}$, a curva elíptica $y^2 = x^3 + 1001x + 75$ e o ponto gerador $P(4023, 6036)$. A figura 1 representa os resultados obtidos, e nos mostra algumas características dos algoritmos de ataque citados previamente.

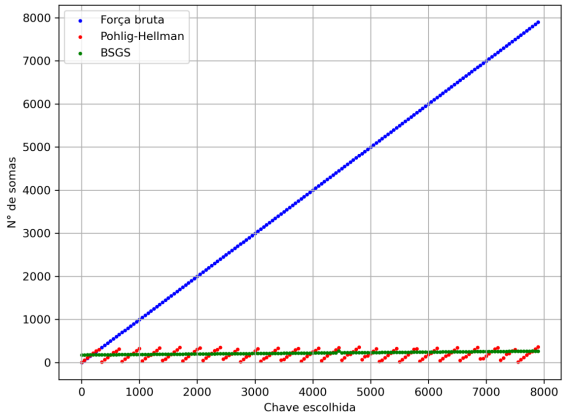

Figura 1: Gráfico de complexidade. Fonte: Próprio autor.

O gráfico da figura 1 foi obtido a partir da variação da chave privada escolhida. Todos os demais parâmetros — o corpo finito, a curva elíptica e o ponto gerador — permaneceram constantes. No gráfico, nota-se que o tempo de execução do ataque por força bruta cresce linearmente com o valor da chave privada, o que é esperado, já que chaves maiores exigem mais iterações para serem encontradas. Por outro lado, os algoritmos de Pohlig-Hellman e Baby-Step Giant-Step (BSGS) apresentam variações mínimas no número de operações, que podem ser consideradas constantes. Isso ocorre porque esses algoritmos dependem da ordem do grupo de pontos, que não se altera uma vez que os parâmetros da curva e do corpo finito permanecem fixos.

Agradecimentos

Os autores agradecem ao CEFET-MG, à PUC-MINAS e à FAPEMIG (PCE-00114-25) pelo apoio financeiro

Referências

[1] D. R. Hankerson. **Guide to Elliptic Curve Cryptography**. Ed. por Alfred J. Menezes e Scott A. Vanstone. 1st ed. 2004. Springer Professional Computing. Includes bibliographical references (p. [277]-304) and index. New York, NY: Springer New York, 2004. 1312 pp. ISBN: 9780387218465.

[2] N. Koblitz. "Elliptic curve cryptosystems". Em: **Mathematics of Computation** 48.177 (1987), pp. 203–209. ISSN: 1088-6842. DOI: 10.1090/s0025-5718-1987-0866109-5.

[3] A. J. Menezes, P. C. V. Oorschot e P. C. Vanstone. **Handbook of Applied Cryptography**. CRC Press, dez. de 2018. ISBN: 9780429466335. DOI: 10.1201/9780429466335.

[4] R. A. Miranda. "Criptossistemas Baseados em Curvas Elípticas". Dissertação de mestrado. IC, Unicamp, 2002.