UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS FÍSICAS E MATEMÁTICA

Luiza Dilli Cadore

O Colapso Matemático da Criptografia: RSA e ECC Diante da Ameaça dos Computadores Quânticos

Florianópolis
2026

Luiza Dilli Cadore

O Colapso Matemático da Criptografia: RSA e ECC Diante da Ameaça dos Computadores Quânticos

Projeto apresentado como exigência parcial para a conclusão da disciplina "História da Matemática" do curso de Matemática, da Universidade Federal de Santa Catarina - Campus Reitor João David Ferreira Lima.

Florianópolis
2026

Lista de Figuras

Sumário

1 Introdução

Antes do desenvolvimento da fala, a humanidade se comunicava através de expressões faciais e gestos. Com o passar do tempo, grunhidos foram incorporados a essas expressões como uma forma de alertar indivíduos mais distantes sobre perigos próximos[1]. Em seguida, surgiu a protolinguagem, notada até hoje em bebês, que consiste na junção de gestos e vocalizações marcadas [4]. À medida que o cérebro humano evoluiu, esses sons ganharam complexidade, culminando no surgimento gradual da linguagem falada. Mais tarde, por volta de 3.000 a.C., as primeiras formas de linguagem escrita surgiram, civilizações como a suméria (com a escrita cuneiforme) e a egípcia (com os hieróglifos).

Atualmente, com o avanço da internet em todo o mundo, a sociedade se tornou amplamente conectada. Essa comunicação se dá via celulares, computadores, relógios inteligentes, entre outros. Através de aplicativos voltados tanto para o uso pessoal quanto profissional, as pessoas interagem constantemente, o que resulta no compartilhamento diário de um volume gigantesco de dados.

A partir disso, cria-se a necessidade de se estudar uma forma de manter a segurança desses dados compartilhados, ou seja, aqui se fala sobre a segurança da informação. Como principal ferramenta, desenvolveu-se a criptografia: uma técnica que converte dados em códigos confidenciais para impedir o acesso e a alteração por indivíduos não autorizados. Todo esse processo, no entanto, só se torna possível graças aos fundamentos da matemática.

Portanto, este trabalho dedica-se a analisar como os conceitos algébricos das teorias de grupos e anéis servem de base para o funcionamento do algoritmo RSA e do sistema criptográfico baseado em curvas elípticas, estendendo a discussão para o impacto e o papel da criptografia quântica diante dos desafios futuros de segurança.

2 Criptografia

A análise morfológica do termo “criptografia” mostra que essa é composta por dois radicais gregos: “kryptós”(oculto, secreto) e “gráphein”(escrever), ou seja, carrega consigo o significado de escrever algo de forma oculta.

A criptografia tem início muito antes do invento da internet. Suas origens são fortemente atreladas a períodos de conflitos militares. Um exemplo clássico disso está registrado na obra A Vida dos Césares (De Vita Caesarum), do historiador romano Suetônio. Nela, há uma menção direta a um método simples adotado por Júlio César, que consistia em uma bijeção onde cada letra do alfabeto romano era deslocada três posições adiante de forma cíclica. Na prática, a letra “A”era substituída por ”D”, a letra “B”transformava-se em ”E”, e o padrão se repetia sucessivamente para as 26 letras do alfabeto.

Segundo Oliveira [2], existem dois modelos principais de criptografia: o simétrico (ou de chave privada) e o assimétrico. Para compreender o funcionamento de ambos, é necessário definir o conceito de chave. William Stallings [3] a define da seguinte forma: “Chave é um valor (ou conjunto de valores) que, combinado a um algoritmo, permite cifrar e/ou decifrar uma mensagem”. A partir dessa definição, derivam-se os dois processos essenciais: a *encriptação* (transformar o texto claro em um texto cifrado) e a *desencriptação* (reverter a mensagem ao seu estado original).

Dessa forma, fica evidente que a compreensão da mensagem original é estritamente dependente do acesso à chave, sem ela, o conteúdo decifrado permanece inacessível e protegido.

Neste momento, vamos definir o que é a Criptografia Simétrica e Assimétrica.

2.1 Criptografia Simétrica

Esse, é o modelo mais antigo de criptografia que conhecemos. Oliveira, define da seguinte forma: a chave, que é utilizada para se ter acesso a mensagem criptografada é a mesma para o emissor e o receptor, isto é, a chave é simétrica e deve ser mantida em segredo, ou seja, é privada. Podemos entender esse processo com o seguinte exemplo:

Um emissor “A” codifica a mensagem utilizando um algoritmo de ciframento e uma chave. O receptor “B”recebe a mensagem cifrada e, para ler o conteúdo de forma clara, ele terá que utilizar o algoritmo de deciframento correspondente e a mesma chave utilizada pelo emissor. Se um intruso “C” interceptar a mensagem, mesmo que esse conheça o algoritmo

utilizado por “A” para codificar a mensagem, sem o acesso à chave (privada) ele não é capaz de transcrever o texto cifrado.

Como exemplo de criptografia simétrica iremos retomar a criptografia utilizada por Júlio César, também conhecida como a cifra de César. Dentro desse exemplo, a chave é representada pelo número 3 (o valor de deslocamento da letra) e o algoritmo é a operação de desolcar a letra 3 espaços.

2.2 Criptografia Assimétrica (ou chave pública)

Esse método criptográfico foi desenvolvido inicialmente na década de 1970 pelo matemático britânico Clifford Cocks. O sistema fundamenta-se no uso de um par de chaves distintas, assimétricas e complementares: a chave privada, que é de acesso exclusivo do proprietário, e a chave pública, disponibilizada abertamente a qualquer usuário. Essa estrutura foi projetada especificamente para garantir a segurança no canal de comunicação. Para ilustrar seu funcionamento, vamos retomar o cenário onde “A”atua como o emissor da mensagem e “B”como o receptor.

O receptor “B”gera um par de chaves composto por uma chave de ciframento (pública) e uma de deciframento (privada). Para que “A”envie uma mensagem segura a “B”, ele deve primeiro obter a chave pública de “B”e utilizá-la para codificar a informação. Uma vez cifrada, a mensagem é transmitida ao receptor, que emprega sua chave privada exclusiva para decodificá-la. Caso um terceiro não autorizado “C”intercepte o conteúdo no trajeto, ele será incapaz de lê-lo, dado que a chave necessária para a decodificação difere daquela utilizada para a cifragem.

Conforme apontado por Oliveira, a grande vantagem desse modelo é permitir que qualquer indivíduo envie uma mensagem oculta utilizando a chave pública, enquanto a confidencialidade do conteúdo é rigidamente assegurada pela chave privada. Por outro lado, a alta complexidade matemática desse sistema exige um esforço computacional significativamente maior, o que torna o processo mais lento em comparação aos métodos simétricos. Como principais exemplos dessa categoria, destacam-se o Algoritmo RSA e a Criptografia de Curvas Elípticas (ECC), ambos detalhados nas seções seguintes deste estudo.

3 Algoritimo RSA

O Algoritmo RSA (Rivest-Shamir-Adleman) é um dos mais proeminentes algoritmos de chave assimétrica, tendo sido desenvolvido pelos pesquisadores Ronald Rivest, Adi Shamir e Leonard Adleman. Na atualidade, ele serve como o principal pilar para a maior parte das aplicações de segurança digital que dependem da criptografia de chaves públicas.

O funcionamento desse algoritmo é composto basicamente de três etapas: a criação das chaves de encriptação, encriptação da mensagem e desencriptação. Para a escolha da chave, precisamos escolher um número n em que $n = pq$, tais que p e q são números primos distintos.

Agora, calculamos a função Totiente de Euler em n, cuja a definição se encontra a seguir:

$$\varphi : \mathbb{N}^* \to \mathbb{N}^*$$
$$\varphi(n) = \#\{k \mid \text{mdc}(k,n) = 1; 1 \leqslant k \leqslant n\}$$

A maneira mais eficaz de coompreender o funcionamento da criptografia RSA é por meio de um exemplo. Então, suponha que queremos comunicar m, tal que $1 \leqslant m \leqslant n$. Para isso, escolha e de forma que $1 \leq e \leq \varphi(n)$ e $\text{mdc}(e, \varphi(n)) = 1$ assim, temos que, a mensagem cifrada x se dá na forma $m^e \equiv x \pmod{n}$.

Para a desencriptação, existe um único d número tal que $de \equiv 1 \pmod{\varphi(n)}$. Logo, para descriptografar x, sabemos que $m \equiv x^d \pmod{n}$ e assim, é sabido do Teorema de Euler, se $mdc(m,n) = 1$ então $m^{\varphi(n)} \equiv 1 \pmod{n}$.

Perceba que, como $de \equiv 1 \pmod{\varphi(n)}$, temos que $ed = k \cdot \varphi(n) + 1$ para algum $k \in \mathbb{N}$. Assim,

$$x^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$$

Substituindo $ed = k \cdot \varphi(n) + 1$, segue que

$$m^{k \cdot \varphi(n)+1} \equiv m^{k \cdot \varphi(n)} \cdot m \equiv m \pmod{n}$$

Isso, mostra que a chave pública é (e,n) e a privada é (d,n).

Após compreender o funcionamento deste algoritmo, podemos nos questionar o porquê ele funciona e o que de fato garante sua segurança. Ambas as questões serão exploradas ao longo deste capítulo.

3.1 A Álgebra por trás do RSA

A palavra álgebra se deriva do termo árabe al-jabr (الگبر) cujo significado é "reunião de partes quebradas"ou "restauração". A construção da álgebra, como conhecemos hoje, passou por muitas mudanças no decorrer da história. Antes do século XVI, a matemática não contava com os símbolos e notações atuais. Naquela época, os cálculos eram registrados por extenso, por exemplo: "Numerus aequalis qd & rebus"o que equivale, em notações atuais, à equação $n = x^2 + px$.

Em 1591, François Viète em sua obra "In Artem Analyticem Isagoge", introduz notações a fim que essas representassem as grandezas que eram trabalhadas. Durante os Séculos XIII e XVI, a matemática passou por um período de grandes avanços, a resolução das equações de terceiro e quarto graus por matemáticos como Cardano e Tartaglia, culminou no surgimento dos números complexos e isso, preparou o cenário para o surgimento da Geometria Análitica com Descartes.

Atualmente, a álgebra vai além da simples manipulação de equações com incógnitas. Ela se divide em diversas vertentes, como a Álgebra Abstrata (ou Moderna), Álgebra Linear, Geometria Algébrica, entre outras. Essas áreas, por mais que sejam distintas, comunicam-se de maneira excepcional. Para este trabalho, iremos explorar adiante como a Teoria de Grupos fundamenta a segurança do RSA.

Perceba que, para iniciar o processo de encriptação da mensagem, a primeira etapa consiste em definir n como o produto de dois números primos distintos, p e q. É exatamente nessa premissa que está o cerne da segurança deste algoritimo: o Problema da Fatoração de Inteiros.

3.2 Problema da Fatoração de Inteiros

Na presente seção vamos abordar o que é o problema da fatoração de inteiros e como isso garante a segurança do RSA.

Primeiramente, fatorar um número significa encontrar os fatores primos que compõe esse número, ou seja, no exemplo anterior os fatores de n são p e q. Agora, esse problema pode parecer fácil quando trabalhamos com números pequenos, por exemplo, a fatoração de 15 é 3×5. Mas, conforme os valores de n aumentam é quase impossível encontrar seus fatores em um tempo computacionalmente viável.

Essa dificuldade garante grande parte da segurança uma vez que

Em termos de algoritmo, suponha que o atacante intercepta o texto cifrado $x \equiv m^e \pmod{n}$, como a chave pública é (e,n), para descriptografar a mensagem é necessário calcular o valor do expoente privado d, ou seja, resolver a congruência linear $ed \equiv 1 \pmod{\phi(n)}$ e, para fazer, ele precisa obrigatoriamente conhecer o valor de $\phi(n)$. Caso o atacante consiga fatorar n e determinar os primos originais p e q, o cálculo de $\phi(n) = (p-1)(q-1)$ torna-se trivial, permitindo encontrar d.

Portanto, se em algum momento fossemos capazes de encontrar a fatoração de um número grande teriamos inviabilizado a utilização do RSA.

4 Curvas Elípticas

As curvas elípticas conectam três grandes áreas: a teoria dos números, a geometria algébrica e a ciência da computação. Neste capítulo, vamos entender como essas estruturas funcionam como um grupo abeliano, criando a base necessária para compreender o seu papel crucial na criptografia.

A fundamentação teórica apresentada a seguir baseia-se nos textos [6], [5].

4.1 Teoria Básica

A equação de uma Curva Elíptica mais geralmente é dada por:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Para aplicações no ramo da criptografia são consideradas cuvas com características diferentes de 2 e 3. Assim, a equação da curva E toma a forma:

$$y^2 = x^3 + Ax + B$$

Além disso, ara garantir que a curva seja não singular (sem autointerseções ou bicos), exigimos que seu discriminante seja distinto de zero, ou seja, $\Delta = 4A^3 + 27B^2 \neq 0$.

Somado a isso, para ser possível construir um grupo abeliano utilizando curvas elípticas é necessário construir o ponto no infinito, que aqui denotamos por $\mathcal{O}$ ou ∞. Para nós, esse ponto nada mais é que o lugar geométrico em que duas retas paralelas se encontram no plano projetivo.

Com isso, temos a seguinte definição:

Definição 4.1. *Seja K um corpo tal que a característica de K seja diferente de 2 e 3, e seja* $x^2 = Ax + B$ *com* $A, B \in K$ *sem raízes múltiplas, ou seja, o discriminante* $\Delta = 4A^3 + 27B^2 \neq 0$. *A equação de uma curva elíptica é dada por :*

$$E : y^2 = x^3 + Ax + B$$

e também definimos "∞*" o ponto no infinito.*

Uma equação da forma $y^2 = x^3 + Ax + B$ é chamada de equação de Weierstrass.

4.1.1 Estrutura de Grupo

A estrutura de grupo de uma curva elíptica E é definida pela própria curva e por uma operação de soma de pontos, denotada por $(+)$. É importante ressaltar que a soma de pontos em uma curva elíptica não segue a aritmética vetorial trivial encontrada em $\mathbb{R}^n$.

Para somar dois pontos $P_1 = (x_1, y_1)$ e $P(x_2, y_2)$ em E seguimos o seguinte processo para encontrar $P_3 = P_1 + P_2$:

1) Traçamos a reta l que passa por P_1, P_2;

2) encontramos o terceiro ponto T de interseção dessa reta com a curva;

3) O ponto P_3 é então a reflexão do ponto T em relação ao eixo x.

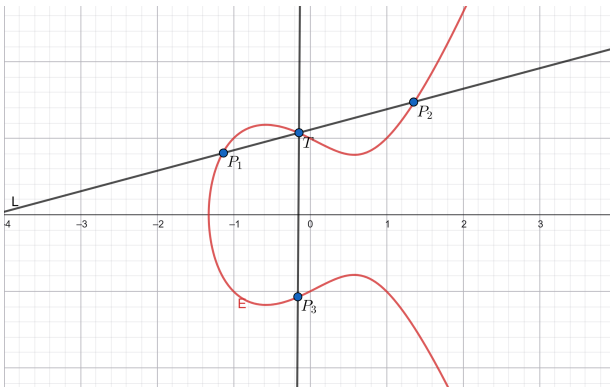

Figura 4.1.1: soma de dois pontos sobre a curva $y^2 = x^3 - x + 1$

Fonte: Elaborado pela autora (2026) utilizando o Geogebra.

Essa descrição geométrica da soma pode ser estendida para uma definição algébrica, tal qual é válida para quaisquer pontos pertencentes à curva. Cosidere então $E : y^2 = x^3 + Ax + B$ uma curva elíptica. Sejam $P_1 = (x_1, y_1)$ e $P(x_2, y_2)$ pontos em E tais que $P_1, P_2 \neq \infty$. Definimos $P_1 + P_2 = P_3 = (x_3, y_3)$ da seguinte forma:

1) Se $x_1 \neq x_2$ então:

$$x_3 = m^2 - x_1 - x_2;\ y_3 = m(x_1 - x_3) - y_1 \text{ em que } m = \frac{y_2 - y_1}{x_2 - x_1}$$

2) Se $x_1 = x_2$ mas $y_1 \neq y_2$ então, $P_1 + P_2 = \infty$

3) Se $P_1 = P_1$ e $y_1 \neq 0$ então:

$$x_3 = m^2 - 2x_1;\ y_3 = m(x_1 - x_3) - y_1 \text{ em que } m = \frac{3x_1^2 + A}{2y_1}$$

4) Se $P_1 = P_2$ e $y_1 = 0$ então, $P_1 + P_2 = \infty$

5) Definimos também que $\forall P \in E\ P + \infty = P$

As demonstrações dos fatos expostos acima serão omitidas por brevidade, as demonstrações completas podem ser encontradas na referência [5].

Para demonstrar que o conjunto de pontos de uma curva elíptica possui uma estrutura de grupo, ainda mais, abeliano, é necessário demonstrar que:

Teorema 4.1. *A adição de pontos de uma curva elíptica satisfaz a seguintes propriedades:*

1) $P_1 + P_2 = P_2 + P_1,\ \forall P_1, P_2 \in E$

2) $P + \infty = P,\ \ \forall P \in E$

3) Dado $P \in E$*, existe* $(-P) \in E$ *tal que* $P + (-P) = \infty$

4) $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3),\ \forall P_1, P_2, P_3 \in E$

Demonstração:

A demonstração dos itens (1),(2) e (3) são relativamente simples. Veja que

1) Como a reta que passa por P_1 e P_2 é a mesma que passa por P_2 e P_1 a comutatividade é óbvia.

2) Pela própria definição do ponto ∞ temos a validade do item (2).

3) Para encontrar o inverso de um ponto P conhecido basta considerar o ponto $-P$ a reflexão do ponto P pelo eixo x, assim a reta que passa por $P, (-P)$ é a reta vertical que por definição também passa pelo ponto ∞. Portanto, $P + (-P) = \infty$

□

A demonstração do item (4) é mais delicada, pois existem vários casos a serem considerados. Como o objetivo deste trabalho não é o estudo aprofundado da estrutura de grupo a demonstração do item (4) não será apresentada em detalhes, no entanto, exibiremos a sua ideia central.

A maneira mais elegante de mostrar a associatividade da soma é utilizando o plano projetivo.

A prova da associatividade da soma se dá seguindo a seguinte ideia: Começamos com uma curva elíptica E e três pontos $P,Q,R \in E$, no final das contas, queremos mostrar que $-((P+Q)+R) = -(P+(Q+R))$. Para computar o ponto $-((P+Q)+R)$ são necessarias as retas $l_1 = \overline{PQ}$, $m_2 = \overline{\infty, P+Q}$, $l_3 = \overline{R, P+Q}$ e perceber em que ponto na curva E essas retas se interceptam.

E, para computar o ponto $-(P+(Q+R))$, precisamos das retas $m_1 = \overline{QR}$, $\overline{\infty, QR}$ e $m_3 = \overline{P, Q+R}$ e perceber em que ponto da curva E essas retas interceptam E.

Vamos perceber agora que, todos os pontos da forma $P_{ij} = l_i \cap m_j$ estão em E, salvo o ponto P_{33}. Note que,

- $P_{11} = l_1 \cap m_1 = Q$
- $P_{12} = l_1 \cap m_2 = -(P+Q)$
- $P_{13} = l_1 \cap m_3 = P$
- $P_{21} = l_2 \cap m_1 = -(Q+R)$
- $P_{22} = l_2 \cap m_2 = \infty$
- $P_{23} = l_2 \cap m_3 = (Q+R)$
- $P_{31} = l_3 \cap m_1 = R$
- $P_{32} = l_3 \cap m_2 = P+Q$

Queremos mostrar que, se todos os oito pontos $P_{ij} \neq P_{33}$ estão em E então obrigatóriamente o ponto P_{33} está em E, e assim conseguimos mostrar que $P_{33} = \overline{R, P+Q} \cap \overline{P, Q+R} = -((P+Q)+R) = -(P+(Q+R))$ e, portanto, a soma é associativa.

A demonstração completa pode ser consultada na referência [5].

5 Criptografia das Curvas Elípticas

Diferente do sistema RSA, o qual a confiabilidade repousa na complexidade computacional da fatoração de inteiros, a Criptografia das Curvas Elípticas (ECC) extrai sua robustez das propriedades algébricas de grupos abelianos, constituídos por pontos de uma curva elíptica definidos sobre um corpo finito.

O principal diferencial da ECC reside na sua eficiência. À medida que o poder computacional aumenta, o RSA exige chaves cada vez maiores para manter o mesmo nível de segurança, o que impacta o desempenho em dispositivos com recursos limitados. Em contraste, as curvas elípticas permitem alcançar o mesmo nível de proteção com chaves significativamente menores.

Para que seja possível estudarmos a Criptografia das Curvas Elípticas é necessário coompreender o O Problema do Logaritmo Discreto, uma vez que ele é a base para a segurança da troca de chaves.

5.1 Problema do Logaritimo Discreto

O Problema do Logaritmo Discreto é considerado um "problema de mão única"uma vez que dados g, x e p é simples calcular:

$$h = g^x \pmod{p}$$

Contudo, dados g, h e p pode ser extremamente difícil encontrar x que resolva a equação.

Dentro do contexto de Curvas Elípticas, no lugar de números inteiros, utilizaremos pontos em uma curva. Neste caso, a operação não é a multiplicação comum, mas a adição de pontos, ou seja, ao somar um ponto P a si mesmo k vezes obtemos um ponto Q em que :

$$Q = kP$$

Assim, o desafio do intruso é descobrir qual é o valor de k conhecendo somente os valores de P e Q.

Para uma breve implementação utilizaremos a curva "secp256k1"[7], definida pela equação $y^2 = x^3 + 7$, visto que esta é amplamente difundida na criptografia por fundamentar a

segurança da Criptomoeda Bitcoin.

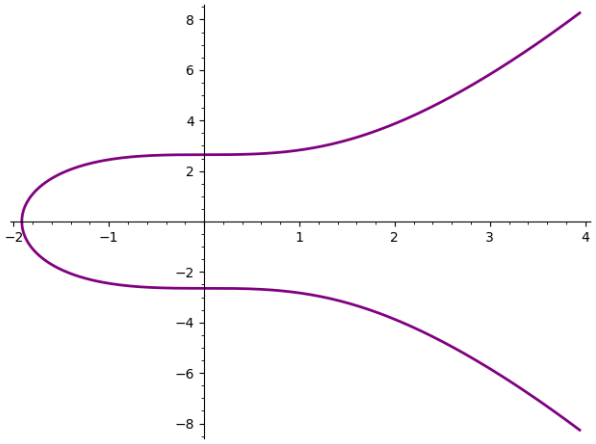

Figura 5.1.1: curva $y^2 = x^3 + 7$ sobre os reais

Fonte: Elaborado pela autora (2026) utilizando o software SageMath.

Para ilustrar a aplicação prática da criptografia, apresentamos a seguir o algoritmo Diffie-Hellman.

Algoritmo 5.1 (Diffie-Hellman)**.** *O Algoritmo Diffie-Hellman tem como finalidade estabelecer, de maneira segura e utilizando chaves assimétricas, um meio para que duas partes possam definir e trocar a chave de um algoritmo simétrico. Esta chave será utilizada posteriormente para garantir a segurança da troca de informações em um canal inseguro.*

Suponha que Alice e Bob queiram se comunicar de forma que um terceiro não consiga compreender a mensagem caso a intercepte.

Seguimos então com os seguintes passos:

1) *Alice e Bob escolhem uma Curva Elíptica E sobre o corpo finito $\mathbb{F}_q$ em que o Problema do Logaritimo Discreto seja difícil de se resolver em $E(\mathbb{F}_q)$.*

   Também, é escolhido um ponto $P \in E(\mathbb{F}_q)$ tal que o subgrupo formado por P é grande o suficiente;

2) *Alice escolhe um $a \in \mathbb{Z}$ secreto, computa $P_a = a \cdot P$ e compartilha com Bob;*

3) *Bob escolhe um $b \in \mathbb{Z}$ secreto, computa $P_b = b \cdot P$ e compartilha com Alice;*

4) *Alice calcula $a \cdot P_b = abP$;*

5) *Bob calcula $b \cdot P_a = baP$;*

6) Alice e Bob concordam em um método para extrair a chave de abP.

Uma vez definida a estrutura da curva, $y^2 = x^3 + 7$, introduzimos um número primo p para restringir os cálculos ao corpo finito $\mathbb{F}_p$, permitindo assim a construção das chaves criptográficas.

Para este exemploo corpo escolhido foi $\mathbb{F}_{97}$. Assim, a curva elíptica $y^2 = x^3 + 7$ sobre os o corpo $\mathbb{F}_{97}$ tem a forma:

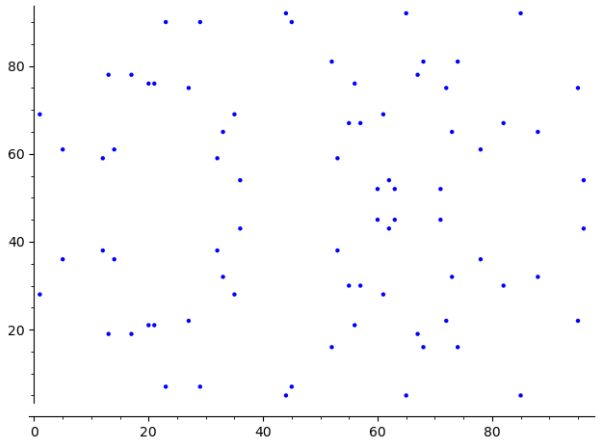

Figura 5.1.2: curva $y^2 = x^3 + 7$ sobre os o corpo $\mathbb{F}_{97}$

Fonte: Elaborado pela autora (2026) utilizando o software SageMath.

5.2 Uma Breve Implementação

A implementação do algoritmo proposto neste trabalho foi desenvolvida no software SageMath, fundamentando-se no protocolo Diffie-Hellman e estruturando-se nas seguintes etapas:

1) Calcular a ordem da curva e seus pontos:

Com o corpo $\mathbb{F}_{97}$ e a curva $y^2 = x^3 + 7$ obtivemos como resposta:

- Ordem da curva: 79;
- Lista de pontos:

(0 : 1 : 0)	(1 : 28 : 1)	(1 : 69 : 1)	(5 : 36 : 1)
(5 : 61 : 1)	(12 : 38 : 1)	(12 : 59 : 1)	(13 : 19 : 1)
(13 : 78 : 1)	(14 : 36 : 1)	(14 : 61 : 1)	(17 : 19 : 1)
(17 : 78 : 1)	(20 : 21 : 1)	(20 : 76 : 1)	(21 : 21 : 1)
(21 : 76 : 1)	(23 : 7 : 1)	(23 : 90 : 1)	(27 : 22 : 1)
(27 : 75 : 1)	(29 : 7 : 1)	(29 : 90 : 1)	(32 : 38 : 1)
(32 : 59 : 1)	(33 : 32 : 1)	(33 : 65 : 1)	(35 : 28 : 1)
(35 : 69 : 1)	(36 : 43 : 1)	(36 : 54 : 1)	(44 : 5 : 1)
(44 : 92 : 1)	(45 : 7 : 1)	(45 : 90 : 1)	(52 : 16 : 1)
(52 : 81 : 1)	(53 : 38 : 1	(53 : 59 : 1)	(55 : 30 : 1)
(55 : 67 : 1)	(56 : 21 : 1)	(56 : 76 : 1)	(57 : 30 : 1)
(57 : 67 : 1)	(60 : 45 : 1)	(60 : 52 : 1)	(61 : 28 : 1)
(61 : 69 : 1)	(62 : 43 : 1)	(62 : 54 : 1)	(63 : 45 : 1)
(65 : 5 : 1)	(65 : 92 : 1)	(67 : 19 : 1)	(67 : 78 : 1)
(68 : 16 : 1)	(68 : 81 : 1)	(71 : 45 : 1)	(71 : 52 : 1)
(72 : 22 : 1)	(72 : 75 : 1)	(73 : 32 : 1)	(73 : 65 : 1)
(74 : 16 : 1)	(74 : 81 : 1)	(78 : 36 : 1)	(78 : 61 : 1)
(82 : 30 : 1)	(82 : 67 : 1)	(85 : 5 : 1)	(85 : 92 : 1)
(88 : 32 : 1)	(88 : 65 : 1)	(95 : 22 : 1)	(95 : 75 : 1)
(96 : 43 : 1)	(96 : 54 : 1)		

2) Geração do ponto base P:

   No contexto deste trabalho, os parâmetros obtidos para o ponto base durante a fase de testes foram:

   - Ponto Base P: (78 : 36 : 1)
   - Ordem de P: 79

   Note que, como a ordem de P é igual a ordem da curva P é um gerador do grupo. Somado a isso, como a ordem de P é um número primo, pelo Teorema de Lagrange, a ordem de

qualquer subgrupo deve dividir a ordem do grupo, e como os únicos divisores de 79 são 1 e 79 isso significa que não existem subgrupos menores dentro do grupo gerado por P.

O fato de não existirem subgrupos menores é importante pois, se a ordem de P fosse, composta um atacante poderia usar o Teorema Chinês dos Restos para descobrir partes da chave privada de Alice resolvendo o problema do logaritmo discreto nesses subgrupos menores.

3) Geração das chaves;

Selecionamos dois inteiros aleatórios a e b, pertencentes ao intervalo $[1,79]$, que atuarão como as chaves privadas. A partir deles, computamos os pontos $A = a \cdot P$ e $B = b \cdot P$, os quais representam as respectivas chaves públicas:

Obtivemos como resultado:

- O valor de a é 16
- O valor de b é 47
- Chave Publica da Alice (A): (32 : 59 : 1)
- Chave Publica do Bob (B): (45 : 90 : 1)

4) Calcular o segredo em comum de Alice e Bob:

Agora, cada parte multiplica sua chave privada pela chave pública recebida do outro. Matematicamente, o sucesso dessa operação reside no fato de que $a \cdot B$ e $b \cdot A$ resultam no mesmo ponto da curva elíptica e extrai sua coordenada x, que será utilizada como base para a criptografia simétrica:

Assim, obtemos como resultado:

- Ponto secreto da Alice: (67 : 78 : 1)
- Ponto secreto do Bob: (67 : 78 : 1)
- Sucesso! Ambos chegaram ao mesmo ponto secreto.
- Coordenada x do segredo: 67

6 Computação Quântica

A palavra “quântico” tem sua origem no latim “quantum”(quantidade/quantia). Para a física, “quantum”é a quantidade mínima de qualquer coisa física necessária para interagir com outra coisa [15], ou seja, os fenômenos que ocorrem em escalas extremamente pequenas, como as dimensões de partículas subatômicas [14]. Alguns conceitos fundamentais da física quântica são:

- Interferência;
- Superposição: princípio fundamental que afirma que um sistema físico pode existir em múltiplos estados ou locais simultaneamente.
- Emaranhamento quântico: fenômeno físico em que duas ou mais partículas ficam tão interligadas que a alteração no estado de uma afeta instantaneamente as demais, sem importar a distância entre elas [16].

A computação quântica surge da junção da ciência da computação com a mecânica quântica. Ela aproaproveita princípios da mecânica quântica para realizar cálculos de forma muito mais eficiente do que os supercomputadores da era moderna[13].

6.1 Computador Quântico

Nos computadores tradicionais, as informações são processadas e armazenadas na forma de bits, que utilizam o código binário (sequências de 0 e 1) como base. Assim, o sistema distribui esses dados conforme a demanda do momento: os arquivos e programas em execução imediata vão para a memória RAM, enquanto o armazenamento de longo prazo é direcionado ao HD ou SSD.

Já em um computador quântico as informações são armazenadas em q-bits. Por mais que ainda seja utilizado o sistema binário, o q-bit possui a capacidade de existir em múltiplos estados simultaneamente, ou seja, ele pode ser 0, pode ser 1, ou pode ser uma combinação de ambos ao mesmo tempo, esse fato explora o princípio da superposição quântica.

Além disso, alguns q-bits podem apresentar emaranhamento quântico. Isso será explorado adiante para entender o funcionamento do algoritmo de Shor.

6.2 RSA vs. Computação Quântica

Sabemos que a segurança do RSA se baseia na dificuldade de, uma vez que $n = pq$, é extremamente complexo encontrar a fatoração de n sem conhecer ao menos um dos fatores p ou q.

Para os tamanhos de chaves utilizados hoje, como o RSA de 2048 bits, um computador clássico levaria milhões de anos testando combinações para descobrir a resposta. Contudo, com a chegada dos computadores quânticos essa não é mais uma realidade. Peter Williston Shor, em 1994, desenvolveu um algoritmo para encontrar os fatores primos de um inteiro, conhecido como algoritmo de Shor.

Os computadores tradicionais tratam o problema de fatoração dos inteiros como um processo complexo de tentativa e erro, no qual ele procura dividir o número sucessivamente por todos os primos até a sua raiz quadrada. O algoritmo de Shor segue o seguinte procedimento:

Suponha que queremos encontrar a fatoração de n em que n é ímpar e não uma potência de primo, então

1) Escolha um número aleatório a tal que $1 < a < n$;

2) Agora iremos analisar a função:

$$f(x) = a^x \pmod{n}$$

Ao calcular essa função para variados valores de x em algum momento eles começarão a se repetir, uma vez que estamos analisando a função módulo n. Então, é de interesse encontrar o padrão de repetição, isso se traduz em encontrar a ordem do elemento a no grupo multiplicativo $\mathbb{Z}_n^*$. Assim, queremos resolver

$$a^r \equiv 1 \pmod{n}$$

Como essa equação é extremamente dificil de resolver aqui é inserido um computador quântico.

3) O computador então faz segue os seguintes passos:

- registra, em q-bits, uma superposição de todos os valores possíveis de x simultaneamente para $f(x)$;

- calcula a função $f(x) = a^x \pmod{n}$ para todos esses estados ao mesmo tempo;
- utiliza a Transformada Quântica de Fourier criando assim um padrão de interferência;
- mede os qubits que entregam um valor diretamente relacionado ao período r.

4) Se r for um número par computamos:

$$(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \pmod{N}$$

e calculamos:

$$p = \gcd(a^{r/2} - 1, N)$$

$$q = \gcd(a^{r/2} + 1, N)$$

5) Se r for ímpar, o número a escolhido no começo é descartado e o algoritmo reinicia.

De maneira teórica o algoritmo de Shor "quebra"completamente a segurança do RSA. Mas, em aplicações práticas para quebrar uma chave RSA de 2048 bits, os computadores quânticos atualmente possuem apenas algumas centenas de qubits que, somado a isso, são instáveis.

Portanto, por mais que o algoritimo realmente funcione, atualmente não existem computadores quânticos que consigam atacar o RSA que comumente utilizamos [18].

6.3 Criptografia Curvas Elípticas vs. Computação Quântica

A segurança da Criptografia Curvas Elípticas repousa na dificuldade da solubilidade do Problema do Logaritmo Discreto em Curvas Elípticas. A computação quântica coloca essa segurança em risco através do Algoritmo de Shor. Para o leitor interessado são sugeridas as leituras das referências [19], [20]

Dado um corpo finito $\mathbb{F}_q$, uma curva elíptica $E(\mathbb{F}_q)$, um ponto base P de ordem n e um ponto $Q \in \langle P \rangle$, a premissa da criptografia é que seja computacionalmente inviável encontrar o escalar $k \in \mathbb{Z}_n$ (a chave privada) tal que:

$$Q = kP$$

Para mostrar como o Algoritmo de Shor quebra a Criptografia de Curvas Elílipticas é necessário entender o que é o Problema do Subgrupo Oculto.

Definição 6.1. *Seja G um grupo finito, X um conjunto finito e $f : G \to X$ uma função e um subgrupo $H \leq G$. Dizemos que a função f oculta o subgrupo H se ela for constante nas classes laterais de H e assumir valores distintos em classes laterais diferentes, ou seja,*

$$f(x) = f(y) \iff xH = yH$$

O objetivo do Problema do Subgrupo Oculto é encontrar um conjunto gerador para o subgrupo H.

Dada essa definição, podemos aplicar o Problema do Subgrupo oculto para descobrir a chave privada k seguindo o seguintes passos:

1) Defina $G = \mathbb{Z}_n \times \mathbb{Z}_n$ e defina a função $f : \mathbb{Z}_n \times \mathbb{Z}_n \to E(\mathbb{F}_q)$ da seguinte maneira:

$$f(a,b) = aP + bQ$$

2) Como $Q = kP$, substituimos essa relação na função f:

$$f(a,b) = aP + b(kP) = (a + bk)P$$

3) Agora é avaliado quando dois pares diferentes de entradas (a,b) e (c,d) resultam no mesmo ponto na curva elíptica, ou seja,

$$f(a,b) = f(c,d) \Rightarrow (a + bk)P = (c + dk)P$$

Como P tem ordem n, os coeficientes devem ser congruentes módulo n, então

$$\begin{aligned} a + bk &\equiv c + dk \pmod{n} \\ \Rightarrow (a-c) + (b-d)k &\equiv 0 \pmod{n} \end{aligned}$$

Logo, $f(a,b)$ é constante exatamente nas classes laterais de um subgrupo específico $H \leq G$.

O subgrupo oculto H é formado por todos os pares (x,y) que satisfazem a equação $x+yk \equiv 0 \pmod{n}$.

Note que o subgrupo H que satisfaz $x+yk \equiv 0 \pmod{n}$ é gerado por gerado pelo elemento $(-k,1)$ pois, substituindo $x=-k$ e $y=1$ temos que

$$(-k)+(1)k = -k+k = 0 \equiv 0 \pmod{n}$$

e também, $(-k,1)$ gera todos os elementos de H, vamos mostrar esse fato. Seja (x,y) um par qualquer que pertença a H. Como ele está no subgrupo, sabemos por definição que:

$$x+yk \equiv 0 \pmod{n}$$

Logo,

$$x \equiv -yk \pmod{n}$$

Agora, subtituindo o valor de x no par original (x,y) segue que:

$$(x,y) = (-yk,y) = y(-k,1)$$

Portanto,

$$H = \langle(-k,1)\rangle$$

4) Agora, como $G = \mathbb{Z}_n \times \mathbb{Z}_n$ é abeliano utilizamos o Algoritmo de Shor da seguinte maneira:

   - O computador quântico avalia a função $f(a,b)$ em superposição para todos os pares possíveis;
   - Através da Transformada Quântica de Fourier e medição, ele isola a periodicidade da função;
   - Isso revela os geradores do subgrupo oculto H;
   - Ao descobrir que o gerador é o elemento $(-k,1)$, o atacante extrai diretamente o valor de k.

5) O algoritmo está quebrado!

6.4 Criptografia Quântica

É notável que com o rápido avanço da computação quântica, os algoritmos de criptografia utilizados atualmente correm o risco de se tornarem obsoletos. Diante dessa ameaça à segurança da informação, cientistas da computação têm se dedicado ao desenvolvimento de uma nova forma para manter as informações seguras: a Criptografia Quântica e a Criptografia Pós-Quântica.

Como a computação quântica, a criptografia quãntica é construida sobre os pilares da física, ou seja, em vez de se basear em anéis, corpos ou complexidade matemática, a criptografia quântica se constrói sobre as leis da mecânica quântica.

Já a criptografia pós-quântica não depende de hardware quântico ou de princípios da física. O objetivo aqui é desenvolver novos algoritmos matemáticos que possam ser executados em computadores clássicos e redes atuais, mas que sejam resistentes aos ataques teóricos de um computador quântico, alguns exemplos de criptografia pós-quântica são:

- Criptografia multivariada;
- Criptografia baseada em rede;
- Criptografia baseada em código;
- Criptografia baseada em hash.

7 Conclusão

Este trabalho buscou analisar os sistemas de segurança digital RSA e Criptografia de Curvas Elípticas (ECC) e como o avanço da compuatação quântica coloca em xeque a eficácia desses algoritmos.

A robustez de tais algoritmos provém diretamente de fundamentos matemáticos profundos contidos na Álgebra Abstrata. Por um lado, o RSA estabelece sua segurança na complexidade computacional do Problema da Fatoração de Inteiros. Por outro lado, a ECC baseia sua eficiência e segurança utilizando propriedades de grupos abelianos sobre corpos finitos e da intratabilidade do Problema do Logaritmo Discreto.

Contudo, a chegada da computação quântica introduz um risco de obsolescência para essas técnicas tradicionais. Pautada em princípios da física quântica, como superposição, interferência e emaranhamento, essa nova tecnologia opera com q-bits, os quais são capazes de processar múltiplos estados simultaneamente. A principal ameaça matemática concreta é representada pelo algoritmo de Shor que, de maneira teórica, reduz drasticamente o tempo necessário para solucionar tanto a fatoração de grandes inteiros quanto o Problema do Subgrupo Oculto em curvas elípticas, quebrando os dois algoritimos acima listados.

Atualmente , o colapso desses sistemas ainda não ocorreu devido às limitações físicas atuais dos computadores quânticos, que contam com poucas centenas de qubits e sofrem com instabilidades. No entanto, com o rápido avanço da tecnologia, aprimorar a maneira com a qual mantemos as informações seguras passa de apenas problemas teóricos para práticos.

Referências

[1] BORGES, Lucivanda Cavalcante; SALOMÃO, Nádia Maria Ribeiro.**Aquisição da linguagem: considerações da perspectiva da interação social**. Psicologia: Reflexão e Crítica, v. 16, n. 2, p. 327–336, 2003. Disponível em: https://doi.org/10.1590/S0102-79722003000200013. Acesso em: 11 jun. 2026.

[2] OLIVEIRA, Ronielton Rezende.**Criptografia simétrica e assimétrica: os principais algoritmos de cifragem.** [S. l.]: IFSUL, [2012?]. Disponível em: http://uab.ifsul.edu.br/tsiad/conteudo/modulo5/src/biblioteca/1_Leitura _Complementar_Algoritmos_de_Criptografia.pdf.pdf. Acesso em: 11 jun. 2026.

[3] STALLINGS, W. **Criptografia e segurança de redes**: princípios e práticas. 6. ed. São Paulo: Pearson, 2014.

[4] Villa, I. (1995). Aquisição da linguagem. Em C. Coll, J. Palácios & A. Marchesi (Orgs.), **Desenvolvimento psicológico e educação** (Vol. 1, pp. 69-80). Porto Alegre: Artes Médicas.

[5] WASHINGTON, Lawrence C. **Elliptic curves: number theory and cryptography**. 2. ed. Boca Raton: Chapman & Hall/CRC, 2008. (Discrete Mathematics and Its Applications).

[6] SILVERMAN, Joseph H. **The arithmetic of elliptic curves**. 2. ed. New York: Springer, 2009. (Graduate Texts in Mathematics, 106).

[7] ESTÚDIO BITCOIN. **Elliptic Curve in Bitcoin**. [S. l.], [s. d.]. Disponível em: https://estudiobitcoin.com/elliptic-curve-in-bitcoin/. Acesso em: 2 maio 2026.

[8] SILVA, Wagner Jorcuvich Nunes da. **Uma introdução à Computação Quântica**. 2018. Trabalho de Formatura (Bacharelado em Matemática Aplicada e Computacional) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2018. Disponível em: https://www.ime.usp.br/ mapweb/tcc/2018/WagnerJorcuvichV3.pdf. Acesso em: 14 jun. 2026.

[9] IBM. **O que é computação quântica?** IBM Think. Disponível em: https://www.ibm.com/br-pt/think/topics/quantum-computing. Acesso em: 14 jun. 2026.

[10] HAYASHI, Victor Takashi et al. **Introdução à Computação Quântica e Impactos em Criptografia**. In: Minicursos do SBSeg 2025. [S. l.]: SBC, 2025. cap. 3, p. 94-153. Disponível em: https://books-sol.sbc.org.br/index.php/sbc/catalog/download/178/788/1537?inline=1. Acesso em: 14 jun. 2026.

[11] CAMPOS, Breno Augusto Ribeiro; SILVA, Bruno de Jesus; SILVA, Kelly de Araújo. **Computação Quântica e Criptografia: Uma Revisão de Literatura sobre os Desafios e Impactos na Segurança Digital**. In: CONGRESSO ESTADUAL DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA DO IF GOIANO, 12., 2023, Rio Verde. Anais [...]. Rio Verde: IF Goiano, 2023. p. 1-4. Disponível em: https://repositorio.ifgoiano.edu.br/bitstream/prefix/4631/1/art_Breno%20Augusto%20Ribeiro%20Car Acesso em: 17 jun. 2026.

[12] DIAS, Beatryz Fernandes et al.**O futuro da criptografia RSA-2048 frente ao avanço da Computação Quântica e da Inteligência Artificial**. Observatório de la Economía Latinoamericana, [S. l.], v. 24, n. 1, p. e12953, 2026. DOI: https://doi.org/10.55905/oelv24n1-114. Disponível em: https://ojs.observatoriolatinoamericano.com/ojs/index.php/olel/article/view/12953/8105. Acesso em: 14 jun. 2026.

[13] RAPATÃO, Éric D'Andréa; FARIA, Renata Mirella.**O impacto da computação quântica na criptografia moderna**. Revista Científica RevistaFT, [S. l.], v. 29, ed. 152, 13 nov. 2025. ISSN 1678-0817. DOI: 10.69849/revistaft/dt10202511132318. Disponível em: https://revistaft.com.br/o-impacto-da-computacao-quantica-na-criptografia-moderna/. Acesso em: 14 jun. 2026.

[14] **FÍSICA Quântica**. In: Toda Matéria. [S. l.], [2026?]. Disponível em: https://www.todamateria.com.br/fisica-quantica/. Acesso em: 16 jun. 2026.

[15] NASA. **Google and NASA Achieve Quantum Supremacy**. 2019. Disponível em: https://www.nasa.gov/technology/computing/google-and-nasa-achieve-quantum-supremacy/. Acesso em: 16 jun. 2026.

[16] NASA. **What is the Spooky Science of Quantum Entanglement?**2024. Disponível em:

https://science.nasa.gov/what-is-the-spooky-science-of-quantum-entanglement/. Acesso em: 16 jun. 2026.

[17] MICROSOFT AZURE. **What is a qubit?** [s.d.]. Disponível em: https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-qubit. Acesso em: 16 jun. 2026.77

[18] TAKU, Dave. **Como esclarecer os fatos sobre a computação quântica e a criptografia RSA**. Blog da RSA, 28 out. 2024. Disponível em: https://www.rsa.com/pt_br/resources/blog/zero-trust/setting-the-record-straight-on-quantum-computing-and-rsa-encryption/. Acesso em: 16 jun. 2026.

[19] KAYE, Phillip; LAFLAMME, Raymond; MOSCA, Michele. **An Introduction to Quantum Computing**. Oxford: Oxford University Press, 2006.

[20] FERNANDES, Tharso Dominisini. **Problema do Subgrupo Escondido em grupos Nilpotentes de Classe 2**. 2008. Dissertação (Mestrado em Modelagem Computacional) – Laboratório Nacional de Computação Científica, Petrópolis, 2008. Disponível em: https://tede.lncc.br/bitstream/tede/83/1/Thesis_Tharso_Dominisini_Fernandes_2008.pdf. Acesso em: 17 jun. 2026.

[21] SCHNEIDER, Josh; SMALLEY, Ian.**O que é criptografia quântica?** IBM Think, [s. l.], [2024 a 2026]. Disponível em: https://www.ibm.com/br-pt/think/topics/quantum-cryptography. Acesso em: 18 jun. 2026.