# TEORIA DOS NÚMEROS E CRIPTOGRAFIA

**Palavras-Chave: CRIPTOGRAFIA, TEORIA DOS NÚMEROS, CHAVE PÚBLICA**

**Autores:**
**ISABEL SOUZA GAIO, IMECC - UNICAMP**
**Prof. Dr. PIETRO SPEZIALI (orientador), IMECC - UNICAMP**

## 1 INTRODUÇÃO

O projeto aqui apresentado tem como foco o estudo das bases da criptografia: fundamentos da Teoria dos Números e bases de criação de criptossistemas. Assim foram vistos tipos de criptografia simétrica e assimétrica, e através dos princípios estudados em teoria dos números, foi possível entender o funcionamento de sistemas de Chave Pública, passando por modelos como o RSA, logaritmo discreto e problema da mochila. Como objetivo se teve, ao final, fornecer ao estudante os fundamentos teóricos e analíticos para entender o básico a respeito da teoria computacional dos números, com ênfase em sua utilização na criptografia.

## 2 METODOLOGIA

Foram apresentados ao longo do ano relatórios semanais a respeito do progresso da IC, com retirada de dúvidas sobre o que era estudado na semana. Foram desenvolvidos no primeiro semestre os capítulos I e II do livro de referência [1], tendo em mente um curso de teoria dos números para estabelecer uma boa base para o entendimento dos sistemas de chave pública, que se apoiam fortemente nesses conceitos para funcionarem.

Após essa primeira fase, era previsto para o segundo semestre o estudo dos capítulos III, IV e V. O capítulo III apresenta como seria o funcionamento de sistemas de criptografia sob uma perspectiva geral. Por exemplo, foram levantadas questões como: quando o sistema é simétrico ou não, quanto tempo de processamento seria necessário para gerar uma chave? O sistema é seguro levando em conta o quão computacionalmente demorado é quebrar sua encriptação? Apresentando também uma forma de assinar a mensagem para que seu interlocutor tenha certeza de que a mensagem não foi manipulada por um interceptador. No capítulo IV foram dados exemplos de criptossistemas assimétricos de chave pública, e o capítulo V ainda está sendo estudado. Para o desenvolvimento adequado em cada capítulo foram usados os livros de apoio referenciados ao final do resumo.

## 3 DISCUSSÃO E CONCLUSÕES:

### 3.1 Teoria dos números:

Sobre a primeira parte da IC temos os seguintes teoremas, importantíssimos para o desenvolvimento dos sistemas de chave pública que vimos no segundo semestre. Assim temos a respeito de teoria dos números:

**Teorema 1 (Teorema Fundamental da Aritmética):** todo inteiro maior que $1$ pode ser representado de maneira única como o produto de fatores primos.

**Prova:** se $n$ for primo, está provado. Supondo que $n$ é composto. E seja $p_1 > 1$, o menor divisor positivo de $n$. Posso afirmar que $p_1$ é primo, se não, existiria $p\ t.q.$

$1 < p < p_1$, o que contradiz a escolha de $p_1$. Então, $n = p_1 n_1$. Se $n_1$ for primo, essa é a prova. Se não, tomamos $p_2$ como o menor divisor de $n_1$, e pela mesma lógica, $p_2$ será primo e temos que $n = p_1 p_2 n_2$. Repetindo esse processo para qualquer número, a seguinte fórmula aparecerá:

$$n = {p_1}^{\alpha_1} {p_2}^{\alpha_2} \dots {p_k}^{\alpha_k}$$

Logo, está provado que existem tais fatores. Para a unicidade usamos a indução em $n$. Para $n = 2$ a afirmação é verdadeira. Assim, se assume que isso é verdade para para todo inteiro maior que 1 e menor que $n$ . Para a unicidade ser verdadeira em todo $n$ temos que se $n$ é primo, não temos nada a provar. Supondo então que $n$ seja composto e que tenha duas fatorações possíveis $t.q.$ :

$$n = p_1 p_2 \dots p_s = q_1 q_2 \dots q_r$$

Como $p_1$ divide o produto $q_1 q_2 ... q_r$ , ele divide pelo menos um dos fatores de $q_j$. Sem perda de generalidade podemos supor que $p_1 | q_1$ . Como são ambos primos, isto implica $p_1 = q_1$ . Logo $\frac{n}{p_1} = p_2 ... p_s = q_2 ... q_r$ . Como $1 < \frac{n}{p_1} < n$, temos que as duas fatorações são iguais e $s = r$, a menos da ordem. Assim temos que qualquer número $n$ pode ser escrito como a multiplicação de fatores primos, sendo essa fatoração a menos da ordem.

□

**Teorema 2:** dados dois inteiros $a$ e $b$, com $b > 0$, existe um único par de inteiros $q$ e $r$ $t.q.$ :

$$a = qb + r, \text{ com } 0 \leq r < b \ (r = 0 \Leftrightarrow b|a)$$

**Prova:** como $b > 0$, $\exists\ q$ satisfazendo:

$$qb \leq a < (q + 1)b$$

temos graças a essas desigualdades que $0 \leq a - qb$ e $a - qb < b$. Desta forma, se definirmos $r = a - qb$ teremos que $\exists\ q$ e $r$. Para mostrar que é único, suponhamos que $\exists\ q_1$ e $r_1$ $t.q.$:

$$a = q_1 b + r_1 \text{ com } 0 \leq r_1 < b.$$

Supondo um $a = q_1 b + r_1 = qb + r$, temos: $(qb + r) - (q_1 b + r_1) = 0 \leq b(q - q_1) = r_1 - r$,assim $b|(r_1 - r)$. Mas, $r_1 < b$ e $r < b$, temos $|r_1 - r| < b$ e, portanto, como $b|(r_1 - r)$ devemos ter que $r_1 - r = 0$, o que implica $r_1 = r$. Logo $q_1 b = qb \Rightarrow q_1 = q$, uma vez que $b \neq 0$.

□

Com esses teoremas podemos apresentar o funcionamento do algoritmo de divisão Euclidiana:

Tomando 2 números quaisquer, queremos achar sua fatoração e $MDC$. Sendo eles $a = r_0$ e $b = r_1$ pelo teorema anterior, obtemos $r_0 = q_1 r_1 + r_2$, dividimos $r_1$ por $r_2$, obtendo $r_1 = q_2 r_2 + r_3$. Repetindo esse processo, lembrando que estamos tratando de números inteiros positivos, e $r_i + 1 < r_i i = 0, ..., n$, temos que, depois de certo número de operações $(n)$, $r_i$ será igual a 0 ou $< q_{i-1} r_{i-1}$. Logo, o resto $r_{n-1}$ será o $MDC(a, b)$.

**Definição:** Seja n um inteiro positivo, a função $(n)$ é definida para ser o número de inteiros positivos $b$ menores que $n$, primos de $n$:

$$(n) = |0 \leq b < n | MDC(b, n) = 1|$$

**Congruências:**

**Definição:** Dados três elementos $a$, $b$ e $m$, dizemos que $a$ é congruente a $b$ modulo $m$ se a diferença $a - b$ é divisível por $m$, ou seja, se dividirmos $a$ e $b$ por $m$, teremos o mesmo resto em cada uma das divisões. Escrevemos essa operação da seguinte forma: $a \equiv b (mod\ m)$.

**Algumas propriedade de congruência:**

1. $a \equiv a(mod\, m)$;

2. Se $a \equiv b(mod\, m)$ e $b \equiv c(mod\, m)$, então $a \equiv c(mod\, m)$;

3. Se $a \equiv b(mod\, m)$ e $c \equiv d(mod\, m)$, então $a \pm c \equiv b \pm d(mod\, m)$ e $ac \equiv bd(mod\, m)$;

4. Se $a \equiv b(mod\, m)$ então $a \equiv b(mod\, d)$ para qualquer divisor de $m$ tal que $d|m$;

5. Se $a \equiv b(mod\, m)$ e $a \equiv b(mod\, n)$ e $m$ e $n$ são relativamente primos, então $a \equiv b(mod\, mn)$.

**Teorema 3 (pequeno teorema de Fermat):** seja $p$ um primo, se $p \nmid a$, então $a^{p-1} \equiv 1(mod\, p)$.

**Prova:** suponha que $p \nmid a$. Primeiro dizemos que $0a, 1a, 2a, 3a, \ldots, (p-1)a$ (I) é um conjunto completo dos resíduos de módulo $p$, já que nenhum dado $ia$ e $ja$ está na mesma classe, ou seja, nenhum satisfaz $ai \equiv aj(mod\, p)$, já que isso significaria que $p|(i-j)a$, e sendo $p \nmid a$, teríamos $p|(i-j)$ e por $i$ e $j$ serem menores que $p$, a única possibilidade seria se $i = j$. Logo concluímos que $a, 2a, \ldots, (p-1)$ (II) é simplesmente um rearranjo de $1, 2, \ldots, p-1$ se considerado o módulo $p$. Logo, o produto dos termos de (I) é congruente ao produto dos termos de (II), ou seja, $a^{p-1}(p-1)! \equiv (p-1)!(mod\, p)$. Logo $p|((p-1)!(a^{p-1}-1))$. Já que $(p-1)!$ não é divisível por $p$, é necessário que $p|(a^{p-1}-1)$. Por fim, se multiplicarmos ambos os lados da congruência $a^{p-1} \equiv 1(mod\, p)$ por a temos que sendo a divisível ou não por $p$, conseguimos a congruência $a^p \equiv a(mod\, p)$.

□

**Teorema 4 (Teorema do resto Chinês):** Supondo que queremos resolver u sistema de congruências de diferentes módulos:

$$\begin{aligned} x &\equiv a_1\ (mod\, m_1) \\ x &\equiv a_2\ (mod\, m_2) \\ &\cdots \\ x &\equiv a_r\ (mod\, m_r) \end{aligned}$$

Supondo que cada par de módulos são relativamente primos: $MDC(m_i, m_j) = 1$ para $i \neq f$. Então existe simultaneamente solução $x$ para todas as congruências,e quaisquer duas são congruentes a um outro módulo $M = m_1 m_2 \ldots m_r$.

**Prova:** primeiro provamos a unicidade do módulo $M$. Supondo que $x'$ e $x$" são duas soluções. Seja $x = x' - x$". Então $x$ precisa ser congruente a 0 em cada módulo $m_i$ e consequentemente módulo $M$. Agora, para construir $x$:

Definindo $M_i = M/m_1$ para ser o produto de todos os módulos com exceção do $i$-ésimo. Então $MDC(m_i, M_i) = 1$, então existe um inteiro $N_1$ (que pode ser achado pelo algorítimo euclidiano), tal que $M_i N_i \equiv 1(mod\, m_i)$. Então,defina $x = \sum_i a_i M_i N_i$. Então para cada $i$ nós vemos que os termos no somatório além do $i$-ésimo termo são divisíveis por $m_i$, pois $m_i|M_j$ sempre que $i \neq j$. Logo, para cada $i$ temos: $x \equiv a_i M_i N_i \equiv a_i\ (mod\, m_i)$, como buscávamos.

□

**Corolário:** a função $\varphi$ de Euler apresentada anteriormente é "multiplicativa", significando que $\varphi(mn) = \varphi(m)\varphi(n)$ sempre que $MDC(m, n) = 1$.

## 3.2 Criptografia:

A partir desse ponto passamos a falar sobre sistemas de criptografia. Que em poucas palavras, é o método que se usa para codificar uma mensagem. De forma que pessoas não desejadas não consigam entender a informação, enquanto quem faz parte da conversa troca o conhecimento com tranquilidade. A princípio podemos pensar em dois grandes tipos de criptografia, a Simétrica e a Assimétrica.

Na criptografia simétrica os interlocutores compartilham uma mesma chave que codifica suas mensagens. Por um lado, isso facilita a encriptação, uma vez que o mesmo processo é aplicado repetidamente a cada mensagem, pois usa sempre as mesmas correspondências entre mensagem e código. Mas, por não ser trocada, aumenta o tempo disponível para tentativas externas de descobrir a chave, facilitando também a desencriptação. Hoje, sistemas como o DES (Data Encryption Standard), que usa a criptografia simétrica, já são considerados inseguros, apesar de poderem 1 gerar trilhões de chaves diferentes. Isso porque alguns computadores já possuem capacidade de processamento que, usando da força bruta, conseguem por tentativa e erro descobrir qual chave está sendo aplicada, ou seja, conseguem testar todas as chaves até chegar na em questão. Já a criptografia assimétrica tem como exemplo sistemas de chave pública, nos quais um dos principais algoritmos usados é o RSA. Neste algoritmo cada parte do diálogo tem sua própria chave, e a quantidade de informação usada para criar cada uma é maior, fazendo com que assim a força bruta não funcione mais em tentativas de descobrir a encriptação feita. Isso, pois, não existe computador que consiga analisar todas as possibilidades possíveis de combinações em tempo viável.

Dos sistemas de chave pública que foram estudados ao longo da IC temos o RSA, o logaritmo discreto e o problema da mochila. Passemos a falar sobre cada: O RSA é o criptossistema mais usado e conceitualmente é bem simples, porém gera grande dificuldade na tentativa de desencriptação indesejada. Para indicar o processo de como funciona, primeiro vamos definir dois números primos extremamente grandes, chamando-os de $p$ e $q$, e colocando $n = pq$. Uma vez que se tem a fatoração de n, basta usar a função totiente de Euler definida como: $\varphi(n) = (p-1)(q-1) = n+1-p-q$. Em seguida se escolhe um inteiro $i$ entre 1 e $\varphi(n)$ que seja coprimo deste último e, por fim, se calcula outro inteiro tal que: $io \equiv mod\ \varphi(n)$. Assim a chave que é tornada pública é $(n, o)$ e a chave que apenas o destinatário saberá é $(p, q, i)$. Sistemas assim possuem o que chamamos de direção única, pois, criar uma chave se mostra uma tarefa razoavelmente tranquila, no caso, é fácil achar um $n$ sabendo quem são $p$ e $q$. Mas, achar $p$ e $q$ sabendo quem é $n$ se torna uma operação muito trabalhosa, já que existe um número enorme de possibilidades que satisfaçam a igualdade.

No Logaritmo Discreto temos outro sistema que usa a ideia de direção única, nesse caso, pensando na expressão $a \equiv b^x (mod\ y)$, podemos afirmar que é consideradamente fácil encontrar a para grandes valores de $x$, sendo conhecidos $b$ e $y$. Porém, fazer o caminho inverso e tentar achar $x$ sabendo o valor de $a$, pode se tornar uma tarefa basicamente impraticável para um grande a levando em consideração a capacidade computacional que temos hoje. Esse é um criptossistema que, em comparação com o RSA, precisa armazenar menos dados e oferece o mesmo nível de segurança, sendo assim, é uma alternativa muito boa. Por fim temos o problema da mochila (ou knapsack problem), para sua explicação é feita uma analogia com mochilas, razão de seu nome. Pensando em uma mochila de certo volume $V$ , e um conjunto com $k$ itens de volume $v_i$ cada, com $1 \leq i \leq k$, se deve achar uma combinação de $v_i$ tal que a soma de todos eles seja igual ao volume $V$ . Ou seja, $\sum_{i \in I} v_i = V$.

Esse método é baseado na ideia de que a pessoa que o aplicará, trabalhará com o caso especial crescente da mochila, em que vi está ordenado crescentemente, mas que se alguém externo tentar fazer o processo de desencriptação precisaria lidar com o caso geral de uma classe de problemas chamados NP-completo, que são extremamente difíceis de se resolver. Entretanto existem falhas em algumas formas de construção (como no criptossistema de Merkle-Hellman), nas quais computadores competentes, através da criptoanálise conseguem descobrir as chaves de desencriptação. E em 1982 Adi Shamir achou um algoritmo que resolve esse tipo de criptosistema. Desde então, muitos perderam a confiança no método, apesar de ainda existirem formas de construção que não têm solução.

Todos esses sistemas oferecem segurança, porém temos a questão de que apesar disso, é necessário

muito tempo de processamento para que sejam calculados. Portanto é comum que se use a chave pública para compartilhar uma chave de criptografia simétrica com a qual é possível codificar com mais velocidade as mensagens. Assim a segurança é mantida, mas a eficiência também é assegurada.

Depois de entender um pouco do funcionamento de cada sistema, sabemos que existem métodos muito eficientes para criptografar uma mensagem. Mas, ainda é necessário um método que ajude quem recebe a mensagem a constatar se quem a enviou foi sua fonte segura, ou que se sua conversa foi invadida. Esse método é o que chamamos de ZKP - Zero Knowledge Protocol.

O ZKP pode ser explicado de forma simples. Imaginemos que a pessoa $A$ tenha os códigos $E_a$ e $D_a$ ($E$ - encriptação e $D$ - desencriptação) e a pessoa $B$ tenha os códigos $E_b$ e $D_b$. Ao em vez de a pessoa $A$ mandar uma mensagem $m$ para a pessoa $B$ através apenas da função $E_b(m)$, que é a chave pública de $B$, ela manda no formato $E_b(D_a(m))$. Assim a pessoa $B$ conseguirá decriptar $E_b$ através de sua chave $D_b$, e aplicando a chave $E_a$, que é a chave pública de $A$, terá sua mensagem sem problemas. Isso ocorre pois, se a pessoa que enviou a mensagem não for $A$, a chave pública $E_a$ não será capaz de fazer o caminho inverso de $D_a$, e decriptar a mensagem. Ou seja, a mensagem só se tornará clara se a pessoa que a enviou tiver a chave correspondente a $E_a$, que é $D_a$. Por definição a única pessoa que possui $D_a$ é $A$, portanto se for possível ler $m$, a única pessoa que pode te-la enviado é realmente $A$.

Assim, a veracidade do interlocutor é constatada. Pois como vimos, usamos outro método que não seja apenas aplicando $E_b$, que pode ser adquirida por qualquer pessoa uma vez que é pública. Esse protocolo portanto se torna uma ferramenta muito eficiente, por não ser necessário expor nenhuma informação individual de cada uma das partes que já não seja pública.

# Referências


[1] KOBLITZ, N. *A course in Number Theory and Cryptography*. 2. ed. New York: Springer-Verlag, 1994.

[2] MILIES, C.P.; COELHO, S.P. *Números:* Uma introdução à Matemática. 3. ed. São Paulo: Edusp, 2013.

[3] LIDL, R.; NIEDERREITER, H. *Finite Fields*. 2. ed. Cambridge: Cambridge University Press, 1997.

[4] MARTINEZ, F.B. Et al. *teoria dos números:* um passeio com primos e outros números familiares pelo mundo inteiro. 4. ed. Rio de Janeiro: IMPA, 2018.

[5] COUTINHO, S.C. *Números Inteiros e Criptografia RSA*. 2. ed. Rio de Janeiro: IMPA, 2011.

[6] SANTOS, José Plínio de Oliveira. *Introdução à Teoria dos Números*. 3. ed. Rio de Janeiro: IMPA, 2012.