**Thales Evandro Simas Júnior**

# Análise Sobre a Segurança do Algoritmo de Criptografia Posicional

Monografia de Graduação Apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como Parte das Exigências da Disciplina Projeto OrientadoII para Obtenção do Titulo de Bacharel em Ciência da Computação.

Orientador

Professor MSc. Bruno de Oliveira Schneider

Co-Orientador

Professor Mário Luiz Rodrigues Oliveira

Lavras

Minas Gerais – Brasil

2002

**Thales Evandro Simas Júnior**

**Análise Sobre a Segurança do Algoritmo de Criptografia Posicional**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 13 de dezembro de 2002

_____

Professor Msc. Bruno de Oliveira Schneider
(Orientador)

_____

Professor Mário Luiz Rodrigues Oliveira
(Co-Orientador)

Lavras

Minas Gerais – Brasil

2002

A
minha família,
que todo o tempo esteve comigo
me apoiando e dando força para continuar.
*In memorian* de meu avô Milton Simas

# Agradecimentos

Agradeço a todos os meus familiares por todo apoio que recebi durante todos estes anos.

Agradeço ao meu irmão Glaucus, pois sempre que pode me deu forças em minha cruzada.

Agradeço aos meus colegas de sala por todos estes anos de convivência.

Agradeço aos meus companheiros de república por tudo aquilo que vivemos.

Agradeço aos meus grandes amigos de Patos de Minas que me acompanharam nas festas e nas dificuldades.

Em especial venho a agradecer meus pais, Thales e Vera, por toda a formação que recebi e pela pessoa que sou hoje.

# Resumo

Neste projeto foi analisada a segurança do algoritmo de criptografia posicional sob o ataque de quebra criptográfica pelo método de análise de freqüência de símbolos.
Para isto verificou –se a influência da operação matemática *mod* na construção das chaves de codificação. Então, foi desenvolvido um *software* em C++ capaz de extrair a freqüência dos caracteres de uma linguagem, de criptografar os dados de um texto através do método posicional, de obter a quebra da criptografia pelo método de análise de freqüência de símbolos. Foi analisado também a importância do grau da função de criptografia para que as chaves de criptografia sejam encontradas.

# Abstract

In this project the security of the “algoritmo de criptografia posicional” was analyzed under the attack of cryptography by the method of analysis of frequency of symbols. For this it verified if the influence of the operation mathematical *mod* in the construction of the code keys. Then, a software was developed in C++ capable to extract the frequency of the characters of a language, of cryptography the data of a text through the method posicional”, of obtaining the break of the cryptography for the method of analysis of frequency of symbols. It was also analyzed the importance of the degree of the cryptography function so that the cryptography keys are found.

# Sumário

## Capítulo 1



## Capítulo 2

## Capítulo 3



## Capítulo 4



## Capitulo 5

## Capitulo 6



## Capitulo 7





## Apêndices

# Lista de Tabelas



# Lista de Figuras

# Capítulo 1

## Introdução

A necessidade humana em comunicar-se não é novidade. As figuras desenhadas nas paredes das cavernas denunciam a tentativa primitiva de estabelecer um tipo qualquer de contato, também a fala, veio como nosso majestoso instrumento de transmissão de conhecimento. A importância em transmitir informação trouxe consigo uma corrida desenfreada em busca de novos meios mais rápidos e práticos de comunicação. Desde a antigüidade, a comunicação ganhava novos meios de transpor barreiras. Comunicar-se cada vez mais longe e mais rápido, era este o desafio. Com uma tecnologia mais avançada, vieram os telégrafos com suas mensagens em código. Com o radio podia-se transmitir mensagem de voz. E finalmente a comunicação veio através de computadores. Mas acompanhando toda essa evolução veio também a necessidade de esconder as informações que estavam sendo transmitidas. A comunicação é preciosa, nem sempre a informação é de domínio publico. O crescimento contínuo da Internet e das redes privadas de computadores trouxe o desafio de se garantir a segurança das informações que trafegam por essas redes. Métodos para assegurar a privacidade da informação são pesquisados há muito tempo, varias soluções foram propostas, mas nem sempre com êxito absoluto. Assim a necessidade de prover a segurança dessas informações torna-se cada vez mais importante à medida que aumenta a informatização da sociedade atual e cresce o valor agregado às informações disponíveis nas redes de computadores.

A criptografia é um destes métodos que foram desenvolvidos para garantir a segurança da informação. Este método se baseia em utilizar técnicas matemáticas para conseguir esconder dados que não devem ser conhecidos por pessoas não gratas. Esta técnica então se baseia no fato de ser necessário encriptar os dados para domínio público, mas deve ser necessário permitir que um grupo restrito possa conseguir descriptografar estes dados para entendê-los.

Para que a criptografia atenda as suas expectativas é necessário que o algoritmo funcione de forma eficiente e segura. A eficiência é a velocidade com a qual pode-se criptografar e posteriormente descriptografar os dados sem muita perda de tempo. Isto pode ser testado considerando-se a complexidade e a velocidade de execução do mesmo. A segurança é constatada após uma bateria de testes que iram tentar quebrar a criptografia deste algoritmo. Estes testes começam em provas matemáticas e vão até testes feitos na prática. Na prática se pode tentar quebrar essa criptografia de duas maneiras, força bruta e análise de freqüência.

Este trabalho enfocou o algoritmo de criptografia posicional, que é um método criptográfico que leva em consideração a posição em que cada caractere se encontra no texto a ser criptografado. O objetivo a ser alcançado nesse projeto de fazer uma análise sobre a segurança deste método. Para esse propósito verificou-se a influência da operação matemática *mod* na construção das chaves, pois esta operação reduz o universo de chaves a serem usadas e isso a princípio tornaria possível decifrar o código em um tempo satisfatório.

O projeto apresentado pelo aluno Mário Luiz Rodrigues Oliveira intitulado "Uma análise da segurança e da eficiência do algoritmo de criptografia posicional" mostrou existir dificuldades em conseguir fazer a quebra da criptografia pelo método de força bruta. Segundo seu estudo o número de chaves a serem testadas no ataque por força bruta é $256^{n+1}$ onde n é o grau da função de criptografia. Isto inviabiliza este processo, pois este número de chaves que devem ser testados é extremamente grande.

O algoritmo criptográfico RSA, um dos mais conhecidos e utilizados mundialmente, necessita de chaves de 2048 bits para ser considerado seguro. Isto obedecendo às normas estabelecidas para utilização em meio militar e industrial. Com uma chave desta seria necessários ser testada $2^{2048}$ chaves, no pior caso, para que se possa quebrar a criptografia. Este número de chaves pode ser obtido no algoritmo de criptografia posicional se utilizar em sua função grau 255.

Como a quebra de segurança do algoritmo por força bruta não é viável, neste projeto utilizaremos a análise de freqüência como ferramenta para conseguir descobrir as chaves utilizadas para fazer a encriptação dos dados pelo algoritmo posicional.

Em principio isto não era considerado possível, pois a função codifica caracteres iguais em caracteres distintos. Entretanto voltamos à operação matemática *mod*256 associada à ordem seqüencial na qual os caracteres são criptografados. Esta operação nos retorna um conjunto de blocos de 256 caracteres onde a primeira posição do primeiro bloco foi encriptado da mesma forma da primeira posição do segundo bloco, e a primeira do terceiro e assim por diante. Isto nos permite verificar que se tivermos em mão um texto com um tamanho razoável onde possamos restringir um número considerável de caracteres encriptados da mesma forma, podemos utilizar a análise de freqüência e descobrir as chaves utilizadas no processo.

Verificou-se também que o segurança de tal método está fortemente relacionado com o grau da função de criptografia, no caso de tentar a quebra pelo método de força bruta, e quanto maior seu grau, maior sua segurança. Portanto, para decifrar a criptografia do algoritmo posicional conhecendo a priori o grau da função de criptografia será necessário aplicar o método da força bruta, e para descobrir as chaves de criptografia será suficiente resolver um sistema linear.

# Capítulo 2

# Referencial Teórico

## 2.1 Introdução

O que será apresentado a seguir tem por objetivo dar um embasamento teórico ao leitor para que este possa entender o que realmente está sendo realizado neste projeto.

Um breve histórico e conceitos sobre a criptografia serão mostrados para que se possa entender de onde veio este processo e o porque ele hoje tem tão relevada importância.

Alguns métodos criptográficos tradicionais e computacionais mostrarão a importância da criptografia na sociedade moderna. Apresentam-se também os conceitos matemáticos que constituem os pilares dos algoritmos de criptografia dando ênfase aos conceitos utilizados no algoritmo de criptografia posicional. Deixando claro que a abordagem aqui adotada não será suficiente à compreensão de todos os algoritmos criptográficos, sendo indicada uma bibliografia complementar aos interessados no assunto.

As formas de ataque aos algoritmos criptográficos, além dos critérios utilizados para avaliar a segurança dos algoritmos criptográficos serão apresentados para que o leitor possa ter contato não somente com a forma pela qual o projeto em questão irá demonstrar, mas também com outras técnicas existentes.

## 2.2 Histórico

A idéia inicial de criptografia pode ser percebida nos *hieróglifos* egípcios e também entre generais gregos e romanos, os quais usavam a criptografia para enviar mensagens em códigos aos comandantes de campo códigos estes simples comparados com os atuais. Sabe-se que uma técnica utilizada para encobrir mensagens era enrolar o papel em uma espada e escrever a mensagem assim, depois que o papel era desenrolado o resultado era incompreensível, apenas as pessoas que sabiam que isto havia sido feito enrolavam o papel novamente em outra espada para compreender a mensagem. Vê-se, portanto que as raízes da criptografia são de longa data, o que permitiu a essa arte um grande desenvolvimento.

Desenvolvimento este que, segundo Tanenbaum[10], deve-se historicamente há quatro categorias, militares, diplomatas, pessoas que gostam de guardar memórias, amantes.

Como podemos notar todas as categorias acima se remetem a pessoas que desejam que sua informação seja mantida em segredo. Vamos citar em especial os militares. Esta categoria que realmente despendeu esforços para que a criptografia atingisse hoje a proporção tomada. Os avanços tecnológicos em tempos de guerra e a necessidade crescente em comunicar-se em segredo e depois descobrir o significado da comunicação do inimigo fizeram um grande e notório avanço no campo.

O advento do computador foi o que permitiu a criptografia sair da limitação em que se encontrava para as grandes criptografias de hoje, por isto as eras são separada em criptografia tradicional e criptografia moderna ou computacional.

## 2.3 Conceitos

A palavra criptografia é definida pelo dicionário Aurélio como “a arte de escrever em cifra ou em código”, significado este que remete ao grego *cryptos* que significa secreto, oculto. E também da palavra grafia que nos remete a escrita. Então a palavra criptografia quer dizer escrita oculta.

O processo de criptografia é bem interessante. Ele consiste em pegar um determinado texto que está escrito de forma natural e utilizar alguma regra para esconder, mascarar a mensagem transmitida por ele. Deixando assim a leitura do mesmo impossível, pelo menos para se possa obter informações. O processo não é tão simples, não é necessário apenas esconder o que está descrito no texto. È necessário também que se possa recuperar as informações que foram camufladas. Para isto são utilizadas chaves. O detentor destas chaves deve ser capaz de criptografar e descriptografar este texto.

O processo de criptografia então se baseia em pegar um texto e através de um processo transformá-lo em um texto codificado. Extraindo-se deste processo chaves capazes de retroceder o processo ou executá-lo novamente.

Definições encontradas em livros retratam bem o que é criptografia

- Em Coutinho[2]

  “a criptografia estuda os métodos para codificar uma mensagem de modo que só seu destinatário legítimo consiga interpretá-la, é a arte dos códigos secretos"

- Em Goldreich[9]

  "a criptografia concerne a construção de esquemas que devem ser capazes de resistir a qualquer tipo de abuso, tais esquemas são construídos de modo a manter uma funcionalidade desejada, mesmo sob tentativas malicio-

sas com a intenção de fazê-los desviar de sua funcionalidade recomendada".

A criptografia é um método utilizado para proteger informações, pessoas que não terão acesso a suas informações estarão tentando entender estas informações a força. Isto deixa claro que se a mensagem que está sendo enviada não possui um conteúdo secreto ou restrito, não é necessário que se utilize técnicas criptográficas.

- Withfield Diffie[7], inventor da criptografia de chave pública.

  "se seus dados não estão sujeitos a este tipo de ataque, não é preciso criptografá-los".

A criptoanálise é um estudo onde os profissionais da área buscam decifrar os códigos apresentados pelos algoritmos criptográficos. Deve-se salientar que nem sempre o criptoanalista está interessado em quebrar a criptografia para fins de corromper alguma mensagem. A criptografia deve ser testada, e com o trabalho destes criptoanalistas pode-se comprovar se um algoritmo é ou não realmente seguro.

Deve-se separar bem a idéia de decifrar e decodificar. Decifrar é o processo que o criptoanalista tenta executar. Ele pega um texto codificado e através de técnicas visa obter informações dele. Decodificar é o processo feito quando um texto criptografado é decodificado através de sua chave de decodificação. É um processo natural de codificação e decodificação.

As chaves são peças importantes no processo de criptografia. Elas podem ser utilizadas de duas maneiras, podem codificar um arquivo ou decodificá-lo, depende de qual processo estiver executando.

Os algoritmos de criptografia são classificados segundo suas chaves e eles podem ser simétricos ou assimétricos.

Os algoritmos simétricos são aqueles onde a chave de codificação é a mesma da decodificação. Estes algoritmos são considerados mais fracos, mais fáceis de serem decifrados, apesar de nem sempre isto ocorrer. O algoritmo posicional estudado neste projeto é pertencente a esta classe.

Os algoritmos assimétricos são aqueles que possuem duas chaves distintas, uma para codificação e outra para decodificação. Estes algoritmos têm sua criptografia mais difícil de ser quebrada. Isto pode ser verificado pelo simples fato de que se você tiver posse de uma das duas chaves você terá poder de codificar ou decodificar seu texto e não os dois. Isto é interessante de diversas maneiras. Pense em um texto onde você quer que todos leiam, mas não podem alterá-lo, como por exemplo, a cotação enviada com seus preços a um determinado comprador, você quer que ele leia seus preços, mas não altere. Você distribui seu texto encriptado e permite que seja publico a chave de descriptografação, todos iram ter acesso às informações, mas apenas você pode gerar o código.

Outra maneira com a qual uma pessoa pode se defender contra uma alteração em seu texto é a chamada assinatura digital. A assinatura digital é um processo que como a criptografia visa à segurança da informação. O processo se baseia em uma varredura em todos os bits da mensagem, após essa varredura e identificação dos bits uma função é estabelecida e uma chave é gerada. Está chave só poderá ser repetida caso a mesma função execute novamente no mesmo conjunto de bits. Se apenas um for alterado a chave resultante não será a mesma. A isto se chama assinatura digital. Se alguma alteração em seu texto for executada o portador da chave de assinatura saberá. É importante frisar que a assinatura não visa impedir que alguém modifique seu texto, ela apenas indica se isto foi feito.

## 2.4 A Criptografia Tradicional

A criptografia era precária, baseava-se na substituição de um caractere por outro ou na troca de posição dos caracteres no texto, sendo por isso denominada criptografia orientada a caractere. Para aumentar a segurança, alguns métodos utilizavam tanto a substituição quanto a troca de posição dos caracteres e de preferência várias vezes. É interessante notar que tais métodos são todos simétricos (a mesma chave que codifica também decodifica).

### 2.4.1 Cifras de Substituição

Os métodos de substituição trocam cada caractere no texto simples por outro caractere no texto codificado. Tal substituição é realizada para tornar o texto codificado mais obscuro e incompreensível. Para decodificar o texto faz-se o processo inverso, de forma a restaurar o texto simples. Segundo Weber são considerados somente os 26 caracteres que são letras. Segue abaixo alguns tipos de cifras de substituição:

- **Substituição monoalfabética:** neste tipo de criptografia cada caracter é substituído por outro de acordo com uma tabela ou regra simples. A cifra de substituição mais conhecida é a *Cifra de Césa*r, na qual cada caractere é substituído por três caracteres adiante do alfabeto; assim A é substituído por D, B é substituído por E e assim sucessivamente até Z ser substituído por C. Para decodificar tal método é suficiente retroceder cada caractere do texto codificado em três posições.

Uma generalização deste método pode ser obtida deslocando-se o alfabeto de uma constante K posições, sendo K a chave de criptografia a ser utilizada.

Outra alternativa para a substituição monoalfabética é utilizar um mapeamento individual de cada caractere em substituição a um deslocamento constante, porém neste caso necessita-se de uma tabela para indicar as substituições que estão sendo feitas.

- **Substituição monofônica:** trabalha de maneira à substituição monoalfabética, no entanto cada caractere do texto simples pode ser mapeado para um ou vários caracteres no texto codificado. Assim, A pode ser substituído por 10, 2, 5 e B pode ser substituído por 7, 16, 41.

- **Substituição polialfabética:** este método usa uma combinação de várias substituições monoalfabéticas, usadas em rotação de acordo com algum critério ou chave. Como exemplo, considere uma substituição polialfabética em que são utilizadas quatro tabelas, usadas em alternância a cada quatro caracteres: a primeira tabela seria usada para substituir os caracteres nas posições 1, 5,9 e assim por diante; a segunda para substituir os caracteres nas posições 2, 6,10 e assim por diante; a terceira substituiria os caracteres das 6. Posições 3, 7,11 e assim por diante e a quarta tabela substituiria os caracteres nas posições 4, 8,12 e assim por diante. Um exemplo clássico deste tipo de cifra é a *Cifra de Vigenère* a qual é constituídos por 26 Cifras de César, cada uma com um deslocamento diferente.

- **Substituição por polígramos:** neste tipo de substituição são utilizados grupos de caracteres em vez de trabalhar com caracteres individuais. Então se poderia ter o grupo ABC substituído por RTQ ou ABB substituído por KSC.

Numa primeira análise tais métodos podem parecer seguros, no entanto tais cifras podem ser facilmente quebradas fazendo-se a análise de freqüência de cada caractere no texto codificado e comparando-se estas freqüências com aquelas que normalmente

encontram-se na língua na qual o texto simples foi escrito. Weber citando Tanenbaum afirma que a maioria das linguagens possui uma redundância tão alta que é necessária uma quantidade bem pequena de texto codificado para que se possa realizar a criptoanálise baseada em análise de freqüência.

### 2.4.2 Cifras de Transposição

Nos métodos que utilizam as Cifras de Transposição, cada caractere permanece inalterado, no entanto sua posição no texto codificado é alterada de acordo com uma regra ou função. Um exemplo de criptografia usando-se tal método é considerar o texto simples como uma matriz de caracteres e tal texto codificado é a matriz transposta da matriz de caracteres original.

Para diferenciar entre um texto codificado por cifras de substituição de outro codificado por cifras de transposição utiliza-se da análise de freqüência dos caracteres, se a freqüência for à mesma da língua, tem-se uma transposição, caso contrário uma substituição.

## 2.5 A Criptografia Moderna ou Computacional

Atualmente com o grande avanço da tecnologia computacional procedeu-se a substituição dos métodos criptográficos tradicionais por métodos criptográficos computacionais, onde as operações são implementadas por um computador ou por um circuito integrado especial tendo como conseqüência a aceleração dos processos de codificação e decodificação. Tal aumento nas velocidades de codificação/decodificação fez-se com que diversos passos extras de substituição e transposição fossem usados, a fim de dificultar a criptoanálise.

Dessa forma deseja-se a obtenção de um método de codificação ideal e para tal o método deve garantir que a probabilidade de ocorrência de qualquer símbolo no texto

codificado seja exatamente igual às probabilidades de todos os demais símbolos, ou seja, a freqüência dos símbolos é homogênea. Dessa forma garante-se que a alteração de um único caractere no texto simples tenha a probabilidade de alterar metade dos símbolos do texto codificado e vice-versa e com isso impede-se qualquer análise por freqüência dos símbolos.

### 2.5.1 Exemplos de Métodos Criptográficos

Como foi dito anteriormente os algoritmos dividem-se em algoritmos criptográficos simétricos e algoritmos assimétricos.

**Algoritmos simétricos:**

- **DES:** O *DE*S (Data Encryption Standard) é o exemplo mais difundido de algoritmo criptográfico de chave única. Ele foi desenvolvido pela IBM e adotado como padrão nos Estados Unidos em 1977. O DES é um algoritmo de bloco e trabalha dividindo o texto simples em blocos de 8 caracteres, cifrando cada bloco com uma chave de 56 bits (mais 8 bits de paridade, totalizando uma chave de 64bits). Este método é passível de ser quebrado usando o método da força bruta, bastando para tal testar as $2^{56}$ chaves possíveis, no entanto, já foi demonstrado que é possível quebrá-lo utilizando-se o ataque do texto plano escolhido e adaptativo em $2^{47}$ tentativas ou até mesmo em $2^{43}$.
- **Triple-DES:** Este algoritmo é um método para tornar o DES mais seguro e para atingir tal objetivo aplica-se o algoritmo do DES três vezes com duas chaves diferentes: inicialmente cifra-se o texto com a chave C1, depois com a chave C2 e finalmente com a C1 novamente. Para quebrar tal método com

força bruta requer-se $2^{112}$ tentativas. Atualmente é usado por instituições financeiras como alternativa ao DES.

- **IDEA:** O algoritmo de criptografia de dados internacional - IDEA foi desenvolvido na Suíça por James Massey e Xuenjia Lai e públicado em 1990. Ele é um algoritmo de blocos com tamanho de 64 bits e utiliza chaves de 128 bits. Acredita-se que ele seja um algoritmo bastante poderoso, visto que ainda não existe nenhum método de ataque efetivo contra o mesmo e também por ele tem resistido bem contra os métodos aplicados com êxito sobre outros algoritmos. O uso generalizado de tal algoritmo foi bloqueado por uma série de patentes, atualmente em mãos da Ascom-Tech AG.

- **Blowfish:** Tal algoritmo de criptografia em bloco foi inventado por Bruce Schneier e permite a utilização de chaves de até 448 bits, sendo otimizado para serem executados em máquinas de 32 ou 64 bits. Além desses algoritmos acima descritos podem ser citados: RC2, RC4, RC5, MMB, Lucifer, NewDES entre outros. Aos interessados podem consultar, artigo no qual são citados os algoritmos de criptografia mais comumentes encontrados na literatura com uma pequena descrição sobre os mesmo e com referências para os que buscam detalhes aprofundados sobre tais algoritmos.

Descritos alguns algoritmos simétricos, descreve-se alguns métodos de criptografia de chave pública.

**Algoritmos assimétricos:**

- **Diffie-Hellman:** primeiro método de chave pública proposto tendo por objetivo resolver o problema de distribuição das chaves e não podendo ser utilizado para cifrar mensagens. Tal algoritmo baseia-se no problema do logaritmo discreto, o qual é NP completo.

- **Rabin:** este método baseia-se na dificuldade de se extrair a raiz quadrada em aritmética modular de um número composto. Para aplicá-lo escolhe-se dois números primos, *p* e q, sendo que ambos devem ser congruentes a 3 módulo 4 e tais primos são a chave privada e a chave pública é o número *n* calculado como o produto de *p* e *q*. Em [6] podem ser encontrados mais detalhes sobre tal algoritmo.

- **ElGamal:** criado por Taher ElGamal é um sistema criptográfico de chave pública que pode ser usado tanto para cifrar mensagens quanto para assinatura digital. Tem sua segurança baseada na dificuldade de calcular logaritmos discretos e aritmética modular.

Além dos métodos acima citados, podem ser encontrados outros na literatura, tais como: DSA, LUC, DSS, Método da Mochila entre outros. Aos interessados recomenda-se pesquisar em Weber o qual indica uma série de algoritmos de criptografia de chave pública com uma sucinta descrição sobre os mesmo, além de indicar fontes onde podem ser encontrados tais algoritmos com riqueza de detalhes.

## 2.6 Fundamentos Matemáticos

### 2.6.1 Relações de Equivalência

A aritmética modular é o ramo da matemática dedicado ao estudo dos fenômenos cíclicos. Por exemplo, pode-se somar 15 + 13 e obter como resultado 4 e este resultado está correto desde que se considere os números 15, 13 e 4 como horas, logo se são 15 horas e passam 13 horas, agora são 4 horas da manhã. Essa associação com o relógio é interessante e ajudará a compreender as propriedades desta aritmética.

Em Coutinho, tem-se a introdução deste assunto utilizando a noção de relações de equivalência, método este que também será usado neste trabalho. Vê-se que dado um conjunto X, o qual pode ser finito ou infinito, uma relação nesse conjunto **X** é definida dizendo-se como comparar dois elementos deste conjunto. Para tornar mais clara a idéia de relação considere os seguintes exemplos:

- No conjunto dos números inteiros, podem-se considerar duas relações naturais, a relação de igualdade e a relação de desigualdade.

- Num conjunto de bolas coloridas, pode-se considerar a relação de bolas de uma mesma cor.

Em ambos os exemplos define-se uma maneira de comparar os elementos de um conjunto, logo se define uma relação neste conjunto. E tendo clara a noção de relação, podem-se conceituar *relações de equivalênci*a. Dados um conjunto **X** e uma relação que será denotada por ~ definida neste conjunto e considerando-se ainda os elementos x, *y* e *z* pertencentes ao conjunto X, define-se esta relação como uma *relação de equivalência* se as seguintes propriedades forem satisfeitas para quaisquer x, *y* e z:

• $x \sim x$ ( reflexiva )

• se $x \sim y$ então $y \sim x$ ( simétrica )

• se $x \sim y$ e $y \sim z$ então $x \sim z$ (transitiva)

A primeira propriedade é denominada *reflexiva* e informa que se uma relação é de equivalência então um elemento pode ser comparado a si mesmo.

A segunda propriedade chamada *simétrica* nos informa que se um elemento *x* pode ser comparado com um elemento *y* então o elemento *y* também pode ser comparado com o elemento *x* e finalmente a terceira propriedade chamada *transitiva* nos informa que se um elemento pode ser comparado a um segundo elemento e este por sua vez pode ser comparado a um terceiro elemento, então o primeiro elemento pode ser comparado com o terceiro elemento.

Após esses conceitos iniciais, surge o questionamento sobre qual a utilidade de aplicar o conceito de *relação de equivalência* num conjunto e a resposta é simples, *relações de equivalência* são usadas para classificar os elementos de um conjunto em subconjuntos com propriedades semelhantes, sendo cada subconjunto produzido por essa classificação chamado de *classe de equivalência*.

Por tratar-se de um tema matemático, é conveniente uma maior formalidade, portanto segue-se a notação usada em [1]: seja **X** um conjunto e ~ uma relação definida em **X,** se *x* 2 **X** então a *classe de equivalência* de *x* é o conjunto dos elementos de **X** que são equivalentes a *x* por ~ e matematicamente denotada por: $\bar{x} = \{ y \in X: y \sim x\}$.

Como dito acima uma *relação de equivalência* divide um conjunto em *classes de equivalência* e tal divisão possui a seguinte propriedade: *qualquer elemento de uma classe de equivalência é um representante de toda a class*e, isto é, se um elemento da classe é conhecido, conhece-se toda a classe. Tal propriedade é matematicamente simbolizada por: se $x \in \mathbf{X}$ e $y \in \bar{x}$ então $= \bar{x} = \bar{y}$.

E para finalizar esta pequena introdução ao estudo de *relações de equivalência* enunciam-se as seguintes propriedades do conjunto **X** com a *relação de equivalência* ~ :

• a união de todas as *classes de equivalência* é o conjunto X;

• a intersecção de duas *classes de equivalência* é disjunta.

### 2.6.2 Inteiros Módulo *n*

No conjunto dos números inteiros pode-se definir a seguinte relação de congruência:

Dado um número *n* qualquer, dois inteiros *x* e *y* são equivalentes se a diferença entre eles é um múltiplo de n. Formalmente, diz-se que *x* e *y* são *congruentes módulo n* se x - y é um múltiplo de *n* e escreve-se simbolicamente como:

$$x \equiv y \ (mod \ \mathrm{n})$$

A prova que essa relação de congruência constitui-se uma relação de equivalência. Em nosso estudo, o conjunto que de fato interessa é o conjunto quociente de $\mathbb{Z}$ pela relação de congruência módulo n, conjunto este denominado *conjunto dos inteiros módulo n* e representado por $z$n. Dado que essa relação de congruência divide o *conjunto dos inteiros módulo n* em classes de equivalência, faz-se necessário identificar os elementos dessas classes e para tal considere a notação: seja $a \in \mathbb{Z}$, a classe *a* é formada pelos $b \in \mathbb{Z}$ que satisfazem *b* - *a* é múltiplo de n, isto é, *b* - *a* = *K* n, para algum *K* $\in \mathbb{Z}$. Simbolicamente pode-se representar a classe *à* por: $\bar{a} = \{ a + K\mathrm{n} : K \in \mathbb{Z} \}$.

Dessa forma pode-se notar que dado um inteiro *n* e a relação de congruência acima denotada, particiona-se o conjunto dos inteiros em *n* classes de equivalência, em outras palavras, o conjunto quociente $z_n$ é formado pelas classes $\bar{0}, \bar{1}, \cdots, \overline{n-1}$.

É interessante notar também que a única maneira de dois números entre *0* e *n – 1* serem congruentes módulo *n* é se forem iguais e assim pode-se representar $z_n$ por:

$z_n = \{ \bar{0}, \bar{1}, \cdots, \overline{n-1} \}$.

Finalmente cumpre ressaltar a possibilidade de representar cada classe de $z_n$ na forma reduzida seguindo a notação, assim quando uma classe estiver representada na forma $\bar{a}$ com $0 \leq a \leq n - 1$, diz-se que está na *forma reduzid*a.

### 2.6.3 Aritmética Modular

Como visto na seção 2.6.2 pode-se particionar o conjunto dos inteiros em *n* classes disjuntas através da *relação de congruência módulo n* e fez-se tal partição considerando-se o resto da divisão de um inteiro por n. Passados todos esses conceitos iniciais, pode-se pensar numa aritmética aplicável ao conjunto $z_n$ e este é o propósito desta seção. Salienta-se que serão apenas apresentadas as operações definidas em $z_n$ e suas respectivas propriedades, sem, no entanto provar tais propriedades.

Dessa forma apresentam-se as seguintes operações em *z*n:

- adição,
- subtração,
- multiplicação,
- divisão.

A soma em $z_n$ é definida usando a operação soma de inteiros e é representada da seguinte forma: $\bar{a} + \bar{b} = \overline{a+b}$. Para verificar tal igualdade considere a soma $\bar{5} + \bar{5}$ no conjunto *z*8, logo pela definição de soma tem-se que $\bar{5} + \bar{5} = \overline{5+5}$ e, portanto $\bar{5} + \bar{5} \equiv 2 \ (mod\ 8)$, pois $\overline{10} \equiv 2 \ (mod\ 8)$.

A subtração é definida de maneira análoga não oferecendo maiores dificuldades. Apenas para exemplificar considere a seguinte subtração em $z8$: $\overline{13}$ - $\overline{5}$ que terá como resultado $\overline{0}$, pois $\overline{13}$ - $\overline{5}$ = $\overline{13-5}$ e $\overline{8} \equiv 0 \ (mod\ 8)$.

A multiplicação é definida da seguinte maneira: $\overline{a} \times \overline{b} = \overline{ab}$ e por ser semelhante à adição apresentada julga-se não necessária à apresentação de um exemplo.

Para definir a divisão em $z_n$ considere $x$ e $y$ dois números quaisquer e com $y \neq 0$ e interpreta-se a divisão de $x$ por $y$ como a seguinte multiplicação $x \times \frac{1}{y}$ sendo o número $\frac{1}{y}$ conhecido como o *inverso* de y. Transpondo tal definição para $z_n$ e supondo que se deseja dividir $\overline{2}$ por $\overline{3}$ em $z8$, pode-se transformar tal divisão numa multiplicação como indicado acima e posteriormente resolver tal multiplicação.

### 2.6.4 Sub-módulos

Seja M um módulo-A, e N um subconjunto de M, então N se diz um sub-módulo de M se for subgrupo e se, para todo o a $\in$ A e x $\varepsilon$ N, a.x $\in$ N. Notamos que N < M.

É simples de verificar que um sub-módulo N $\leq$ M é módulo-A, aliás, um sub-módulo é apenas um subconjunto que é módulo-A para as restrições de todas as operações.

Claramente {0} e M são sub-módulos de M; ao primeiro chamaremos sub-módulo zero, e nota-lo-emos apenas por 0.

Sendo X C M, definimos combinação linear de X (coeficientes em A) como qualquer elemento M da forma:

$$a_1x_1 + ... + a_nx_n = \sum_{i=1}^{n} a_i x_i \ ,$$

com $x_i \in X$ e $a_i \in A$, para todo i.
Note que não exigimos que o conjunto X seja finito.

## 2.7 Formas de Ataques a Métodos Criptográficos

Criptoanalistas, harckers, estudantes entre outros, são pessoas que atacaram um sistema criptográfico com o intuito de quebrar sua segurança.

Estes atacantes são divididos em duas categorias: atacantes ativos e passivos. Os atacantes ativos podem interceptar uma mensagem e alterá-la ou trocá-la por outra mensagem, ao passo que os atacantes passivos somente conseguem ter acesso a mensagens sem, no entanto modificá-las.

Os ataques construtivos que visam à melhoria dos métodos criptográficos são passivos, enquanto que os nocivos de onde surgiu à necessidade de combater são os que têm o real intuito de corromper uma mensagem.

Para atacar um sistema de criptografia, o criptoanalista pode agir das seguintes formas segundo Schneier:

- **ataque do texto cifrado:** neste tipo de ataque o criptoanalista tem a sua disposição uma grande quantidade de textos codificados, mas desconhece os textos simples e as chaves utilizadas e sua função é deduzir tais chaves.

- **ataque do texto conhecido:** aqui o criptoanalista tem a sua disposição uma grande quantidade de textos codificados, conhece também os textos simples correspondentes e sua tarefa é deduzir as chaves utilizadas no processo de codificação. Na prática existem vários tipos de mensagens padrões como cabeçalhos de e-mail e arquivos, nomes de pessoas e mensagens de login, das quais pode-se extrair informações não codificadas.

- **ataque adaptativo do texto escolhido:** este tipo de ataque diferencia-se do anterior por permitir uma realimentação entre um texto escolhido para codificação e o próximo texto. No método anterior o criptoanalista somente era capaz de fornecer textos uma única vez. Neste ele pode alimentar o sistema com um pequeno texto, analisar os resultados e realimentar o sistema com outro pequeno texto. Dessa forma, este ataque mais efetivo do que o anterior. O objetivo aqui também é descobrir as chaves utilizadas.

- **ataque do texto cifrado escolhido:** neste tipo de ataque o criptoanalista não só conhece uma grande quantidade de textos simples e seus equivalentes codificados, mas também pode produzir um texto codificado específico para ser decodificado e obter o resultado produzido. Aqui também se objetiva encontrar as chaves envolvidas no processo de codificação. Em todos os tipos de ataques descritos acima se presume que o criptoanalista tenha total conhecimento dos métodos de codificação e decodificação.

Além dos tipos de ataques enumerados pode-se citar o ataque da força bruta, na qual testa-se todas as chaves possíveis e dessa forma obterá sucesso com uma delas. Não é possível defender-se deste tipo de ataque, pois não se pode evitar que um atacante tente decifrar uma mensagem testando todas as chaves possíveis, no entanto que este tipo de ataque é ineficiente visto a quantidade de chaves a serem testadas.

Estas são formas de atacar um sistema criptográfico e não a maneira usada para inferir a chave de criptografia. Como maneira para descobrir a chave pode-se citar a análise de freqüência, que constitui em analisar as freqüências do texto codificado; feito isso se supõe que o caractere de maior freqüência no texto codificado corresponde ao código do caractere de maior freqüência na língua na qual foi escrito o texto simples

e a partir dessa informação o criptoanalista pode tentar descobrir a chave usada na codificação.

## 2.8 Segurança dos Métodos Criptográficos

Segurança é um atributo muito difícil e complexo de ser implementado num sistema e assim um sistema pode ser considerado seguro se ainda não foi possível determinar uma maneira de torná-lo inseguro. Sendo esta opinião também compartilhada por Goldreich que afirma ser muito complicado avaliar a segurança de um método criptográfico.

Em Goldreich encontram-se duas abordagens sobre segurança em métodos criptográficos, a saber:

- **clássica:** nessa abordagem um método é considerado seguro apenas se a chave em uso é pelo menos tão longa quanto o comprimento do texto simples, tal abordagem é baseada na teoria da informação.

- **moderna:** abordagem baseada na complexidade computacional e considera seguro um sistema onde não é possível extrair informações eficientemente de um texto codificado, mesmo que o texto codificado contenha dados sobre o texto simples.

Para que um sistema criptográfico seja considerado robusto deve se encontrar os seguintes requisitos:

- o criptoanalista tem acesso à descrição do algoritmo.
- o criptoanalista tem acesso a uma grande quantidade de textos codificados e os seus correspondentes textos simples.

- o criptoanalista é capaz de escolher quais mensagens serão cifradas e receber as mensagens cifradas correspondentes.

Considera-se inseguro o método que não atende às premissas acima citadas. Também se percebe que atualmente não se julga segurança de um método criptográfico baseado na obscuridade do mesmo.

Finalmente considere a premissa de Kerckhoffs sobre a segurança dos métodos criptográficos: o mecanismo deve ser tão seguro que nem mesmo seu autor deve ser capaz de decifrar uma mensagem sem dispor da chave utilizada, ou ainda "a chave de criptografia pode cair nas mãos do inimigo sem inconvenientes" disse Kerckhoffs.

## 2.9 O Algoritmo de Criptografia Posicional

O Algoritmo de Criptografia Posicional desenvolvido em 2001 por Moreno e Chiaramonte propõe que a posição do caractere no texto simples interfira na chave de criptografia utilizada. Para isto introduziu-se uma função polinomial f(x) sobre o algoritmo da cifra de César que foi visto na seção 2.4.1. Assim ao codificar um símbolo considera-se o valor numérico da tabela ASCII para este símbolo, a esse valor soma-se a imagem da função polinomial f(x), onde x é a posição do símbolo no texto simples e finalmente aplica-se a operação (*mod* 256) sobre resultado da soma. Matematicamente a função de criptografia é a seguinte:

$$\text{ValorCodificado} = (\text{ASCII} + f(x)) \ (mod\ 256)$$

sendo que o uso da operação *mod* garante que o tamanho do texto codificado seja o mesmo do texto simples.

A função polinomial f(x) pode ser de qualquer grau e com coeficientes inteiros, os quais constituem a chave de codificação. Note que quanto maior o grau da função

polinomial, maior a robustez do sistema criptográfico. Para uma melhor compreensão do processo de codificação do algoritmo posicional considere a palavra JUNIOR usando como chave os números *1, 2 e 3* que são os coeficientes da função polinomial.

$f(x) = x^2 + 2x + 3$.

| Palavra | J | U | N | I | O | R |
|---|---|---|---|---|---|---|
| ASCII | 106 | 117 | 110 | 105 | 111 | 114 |
| Posição | 1 | 2 | 3 | 4 | 5 | 6 |
| F(x) | 6 | 11 | 18 | 27 | 38 | 51 |
| F(x) + ASCII | 112 | 128 | 128 | 132 | 149 | 165 |
| Valor criptografado | 112 | 128 | 128 | 132 | 149 | 165 |

*Tabela 2.1: Exemplo de encriptação utilizando algoritmo posicional*

Para decodificar usa-se a seguinte expressão matemática:

$$(c - f(x)) \ (mod\ 256) \quad 2.2$$

onde: c é o valor codificado do símbolo na posição x e $f(x)$ é a imagem da chave de criptografia também para o símbolo da posição x. Portanto usa-se a mesma chave para decodificar os dados e assim tem-se um método de criptografia simétrico.

| Valor criptografado | 112 | 128 | 128 | 132 | 149 | 165 |
|---|---|---|---|---|---|---|
| Posição | 1 | 2 | 3 | 4 | 5 | 6 |
| F(x) | 6 | 11 | 18 | 27 | 38 | 51 |
| Valor criptografado | 112 | 128 | 128 | 132 | 149 | 165 |
| Valor criptografado – F(x) | 106 | 117 | 110 | 105 | 111 | 114 |
| Palavra | J | U | N | I | O | R |

*Tabela 2.2: Exemplo de descriptografação utilizando algoritmo posicional*

## 2.10 Analisando a Segurança do Algoritmo Posicional

Ao analisar-se a segurança do algoritmo posicional verificou-se a influência da operação matemática *mod* na construção das chaves de codificação e a possibilidade de quebrar o algoritmo usando-se os métodos da força bruta e de análise de freqüência.

**Força Bruta**: A operação matemática *mod* induz uma partição no conjunto dos números inteiros. A operação matemática *mod*256 na chave de codificação, a qual é formada por números inteiros. Portanto esta operação divide o universo de chaves possíveis em *256* classes de equivalência e assim sendo cada coeficiente da função f(x) usada na codificação pode variar de *0* a *25*5.

Como visto acima temos 256 possibilidades diferentes para cada coeficiente inteiro da função f(x) usada na codificação e isto reduz o universo de chaves possíveis, visto que se não utilizasse a operação matemática *mod*256 na chave de criptografia ter-se-ia infinitas possibilidades para cada coeficiente. A infinidade de chaves possíveis ao excluir-se a operação *mod*256 deve-se ao fato de cada coeficiente da função f(x) ser inteiro e o conjunto dos números inteiros possui cardinalidade infinita.

Para calcular o número de chaves possíveis em função do grau de f(x) usou-se a seguinte fórmula: $256^{n+1}$, onde *n* é o grau de f(x). Essa fórmula é deduzida considerando todas as combinações possíveis entre os coeficientes de f(x), como para cada coeficiente existem 256 possibilidades diferentes e o número de coeficientes numa função f(x) de grau *n* é *n* + 1, tem-se então que o número de chaves possíveis é $256^{n+1}$.

Percebe-se que mesmo a operação *mod*256 reduzindo o universo de chave possíveis, tem-se que a quantidade de chaves possíveis para um teste usando a força bruta aumenta exponencialmente em função do grau da função de codificação.

Por isto o projeto apresentado pelo bacharel Mário encontrou problemas com a quebra do algoritmo por força bruta.

**Análise de freqüência**: consiste em analisar a freqüência dos símbolos num determinado idioma e fazer-se o mesmo no texto codificado. A partir dessa análise supor que o caráter de maior freqüência num idioma coincide com o caráter de maior freqüência no texto codificado e com essa informação tentar inferir a chave de codificação usada no processo.

## 2.11 Analisando a influência da operação matemática *mod* no algoritmo de criptografia posicional

Ao analisar o método criptográfico, verificou-se que na montagem da função de criptografia aplica-se a operação matemática *mod*256 na função f(x), isto indicou uma falha na segurança do algoritmo pois, foi constatado que está operação faz com que a cada posição múltipla de 256 no texto ocorra uma codificação igual, variando apenas o código ASCII e portanto quando caracteres iguais aparecerem nestas posições serão codificados da mesma maneira.

Ao fazer-se a análise de freqüência no texto criptografado basta considerar os caracteres nas posições múltiplas de 256 (assim começando da posição 1 o próximo símbolo a ser verificado é o da posição 257, começando da posição 2 o próximo símbolo a ser analisado é o da posição 258 e assim sucessivamente).

Hipótese: $\overline{f(x)} = \overline{f(x+256*k)}$

Como: $\overline{f(x)} = f(\bar{x})$

Prova: $\overline{f(x+256*k)} = f(\overline{x+256*k}) = f(\bar{x} + \overline{256*k}) =$

$$f(\bar{x}) = \overline{f(x)}$$

Pode-se constatar então que apenas caracteres nas posições múltiplas de *256* são codificados da mesma forma e pode-se demonstrar praticamente tal fato da seguinte maneira:

Cada caractere é criptografado com a expressão:

$(f(x) + \text{ASCII}) \ (mod\ 256)$

Onde *x* é a posição do caractere no texto simples; dessa forma o valor de f(x) repete-se a cada intervalo de *256* posições e sendo o valor de f(x) o mesmo para dois caracteres iguais, esse caracteres serão codificados com os mesmos valores.
Considere $f(x) = x^2 + x + 1$ e ASCII = 65, então:

$(f(1) + 65) \ (mod\ 256) = (3 + 65) \ (mod\ 256) = 68$
$(f(2) + 65) \ (mod\ 256) = (7 + 65) \ (mod\ 256) = 72$

e assim para cada valor diferente de *x* tem-se um f(x) diferente, a não ser quando *x* é múltiplo de 256, como mostrado abaixo:

$(f(257) + 65) \ (mod\ 256) = (66307 + 65) \ (mod\ 256) = 68$
$(f(258) + 65) \ (mod\ 256) = (66823 + 65) \ (mod\ 256) = 72$

assim f(1) = f(257) e f(2) = f(258) devido à operação *mod*256.

Um texto de um considerável tamanho pode nos devolver um conjunto de caracteres criptografados da mesma maneira para ser analisado, e como foi dito anterior-

mente, Weber[7] citando Tanenbaum[10] afirma que a maioria das linguagens possui uma redundância tão alta que é necessária uma quantidade bem pequena de texto codificado para que se possa realizar a criptoanálise baseada em análise de freqüência.

# Capítulo 3

# Metodologia

Como foi constatado pelo estudo feito sobre o algoritmo de criptografia posicional, é bem provável que o método de análise de freqüência de caracteres seja eficiente na quebra da segurança criptográfica do algoritmo.

Neste projeto foi desenvolvido um software em linguagem C++ de análise de freqüência de caracteres em um texto. Este programa recebe um arquivo texto como entrada e devolver quantos caracteres ele possui e com qual freqüência. Os textos a serem analisados deveram se encontrar na língua portuguesa, pois a análise foi feita sobre esta língua. O programa faz esta análise em diversos textos e armazena em um arquivo os resultados. A intenção é que quanto maior o número de textos processados pelo programa mais próximo os resultados estarão da real freqüência da língua escolhida.

Após este software ser desenvolvido e seus resultados extraídos, um novo software foi implementado também em linguagem C++ que irá codificar um texto pelo método de criptografia posicional. Alguns textos foram encriptados para executar testes neles.

Os textos criptografados foram processados pelo programa de análise de freqüência que devolve a freqüência dos caracteres encriptados.

Conjuntos de caracteres que se encontrem em posições múltiplas de 256 são agrupados, pois a função *mod* detectada no método garante que estes caracteres foram

encriptados da mesma forma. A freqüência destes caracteres foi constatada e o resultado permite inferir a chave de criptografia.

Uma análise dos resultados obtidos foi feita para que realmente seja constatada a veracidade dos mesmos. Está análise despende testes.

# Capítulo 4

# Ataque criptográfico pelo método de análise de freqüência

## 4.1 Introdução

O método de ataque criptográfico por análise de freqüência será utilizado neste projeto como prosseguimento ao projeto realizado por Mario Luiz Rodrigues Oliveira[1] que visa fazer um estudo detalhado da segurança do algoritmo de criptografia posicional.

Como foi constatada, a quebra por força bruta foi considerada é inviável, pois mesmo a operação *mod*256 reduzindo o universo de chaves possíveis este universo cresce exponencialmente em relação ao grau da função de codificação. Por isto este projeto estuda como o algoritmo reage a uma quebra por análise de freqüência.

Neste capítulo será analisado como o método de quebra criptográfica por análise de freqüência será utilizado como ferramenta para que possa ser testada a segurança do algoritmo de criptografia posicional.

## 4.2 A utilização do método de análise de freqüência

A análise de freqüência é um método de quebra criptográfica muito eficiente quando se trata de uma cifra monoalfabética. Como neste tipo de cifra, independentemente do

método utilizado para cifrar o texto, iremos conseguir para cada caracter no texto decodificado um único código cifrado. Podemos deduzir que quando analisarmos a freqüência deste texto cifrado e compararmos com a freqüência da língua utilizada poderá facilmente extrair o texto original.

Moreno e Chiaramonte[3] para evitar a quebra por analise de freqüência propuseram que a posição do byte no texto interferisse sobre a chave de criptografia com isso acreditavam que o algoritmo de criptografia posicional não fosse uma cifra monoalfabética e sim polialfabética, o que tornaria o método de análise de freqüência inútil.

Mas ao ser analisada a influência da operação matemática *mod*256 podemos constatar que a cada múltiplo de 256 obtemos uma cifra feita da mesma forma. Como foi demonstrado em 2.11.

Isto permite deduzir que se for analisado o texto de forma a identificar apenas caracteres de 256 em 256 posições consecutivas conseguimos extrair um conjunto formado por cifra monoalfabética, assim tornaremos um texto de cifra polialfabética em até 256 conjuntos de caracteres compostos por cifra monoalfabética. Cada conjunto deste então formaria a quantidade bem pequena de texto codificado para que se possa realizar criptoanálise baseado em análise de freqüência, citado por Weber.

Será feito então a análise de freqüência em cada um destes 256 subtextos e após isto ser feito, as partes serão novamente compostas com as cifras substituídas por seus respectivos caracteres identificados por suas freqüências para conseguir a quebra da criptografia.

## 4.3 A operação matemática *mod e o software*

O algoritmo de criptografia posicional possui um código simples para se fazer a encriptação do texto.

O pseudocódigo seria o seguinte:

*Inicio*

*1: num = número de caracteres do arquivo*

*2: para posição = 1, enquanto posição < num faça*

*3: leia caracter*

*4: (caracter lido + f(posição)) mod*256

5: posição é incrementada

*fim*

O problema deste código está na função, a posição do caracter no texto atua como a sua variável. Em um texto de tamanho considerável como é o que será analisado, se pode atingir posições muito grandes.

Um texto que alcance a posição 99865 e irá utilizar 256 chaves para atingir um grau 255. Ao se encriptar o último caracter terá uma função onde será determinado o valor de:

$$A(99865)^{255} + B(99865)^{254} + C(99865)^{253} + ... + D(99865)^{1} + E = f(x)$$

Como se pode analisar mesmo para um computador não é uma conta fácil de se realizar. Por isto neste projeto será alterado o pseudocódigo para facilitar as contas sem com isso é claro alterar o resultado do algoritmo.

Como foi demonstrado em 2.11, quando for encriptado um caracter na posição 1 e na posição 257 será obtido o mesmo resultado. O mesmo ocorre quando para a 2 posição e a posição 258, assim por diante.

Por isso, ao alterar o pseudocódigo inserindo outra operação matemática *mod*256, mas desta vez na posição e não no resultado da função. O pseudocódigo então assumiria está forma:

*Inicio*

*1: num = número de caracteres do arquivo*

*2: para posição = 1, enquanto posição < num faça*

*3: leia caracter*

*4: pos = posição mod*256

*5: (caracter lido + f(posição)) mod*256

6: posição é incrementada

*fim*

Observa-se que quando a posição 257 for encriptada a função receberá 1 (um) como parâmetro, o que não vai alterar o resultado, pois o algoritmo irá devolver o mesmo resultado para a posição 257 e 1, caso tivesse sendo encriptado o mesmo caracter.

Nossa função exemplo agora ficaria na forma:

99865 *mod*256 = 25

$A(25)^{255} + B(25)^{254} + C(25)^{253} + ... + D(25)^{1} + E = f(x)$

O resultado obtido já é bem mais fácil de ser executado. E a integridade do algoritmo foi mantida já que o resultado final não foi alterado.

## 4.4 A freqüência

Como foi descrito acima, para que a quebra criptográfica por análise de freqüência seja realizada, é necessário saber se a freqüência com a qual a linguagem que queremos decifrar foi descrita. Como este projeto está sendo desenvolvido no Brasil e o algoritmo criptográfico alvo é brasileiro, utilizaremos a língua brasileira como padrão para nossas pesquisas. Como foi constatado será adotado como caracter de maior freqüência o "espaço" depois o caracter "a" depois o caracter "e" e assim por diante. A tabela contendo todos os caracteres e as freqüências obtidas está anexada a este projeto.

## 4.5 O *software*

Para que testes fossem realizados e que se pudesse constatar todas as teorias que pesquisamos, foi desenvolvido um software. Este foi capaz de fornecer um embasamento pratico para dissertar neste projeto.

O primeiro objetivo do *software* é o de extrair a freqüência dos caracteres da linguagem escolhida para os testes. Para isto ele possui uma função que faz a leitura de textos e insere a freqüência dos caracteres lidos em um arquivo especial que é incrementado à medida que mais textos forem sendo lidos. Quanto maior o número de textos lidos pelo programa maior a possibilidade de se descobrir a real freqüência da língua escolhida.

O segundo objetivo é o de obter textos encriptados segundo o algoritmo de criptografia posicional para, através destes, fazer testes para testar toda a teoria. Para isto foi implementada uma função que criptografa qualquer texto utilizando o pseudo-código modificado descrito em 3.2.

O terceiro objetivo é realmente fazer uso do método de análise de freqüência para com que o código encriptado seja decriptado. Para isto o *software* utiliza-se do

arquivo que contém a freqüência da linguagem escolhida e através deste arquivo realiza a quebra.

Para melhor entender o funcionamento ele será descritos em fases:

1ºfase: É gerada uma tabela com os caracteres e sua freqüência a partir do arquivo que armazena a freqüência da linguagem escolhida.

2ºfase: O texto encriptado é lido e armazenado em um vetor.

3ºfase: Os caracteres são lidos a partir do vetor em posições múltiplas de 256.

4ºfase: Os caracteres vão sendo analisados e suas freqüências inseridas em uma tabela.

5ºfase: Quando todos os caracteres múltiplos da primeira posição são lidos, a tabela gerada é comparada com a tabela de freqüência e é gerada uma nova tabela, contendo o caracter encriptado, o seu referente decriptado e a posição onde isto é verdade.

6ºfase: Todo o processo é repetido até que todos os múltiplos da posição 256 sejam analisados.

7ºfase: O vetor onde o texto encriptado foi armazenado é relido e os caracteres vão sendo alterados e escritos em um arquivo.

Esse *software*, como é previsível, não conseguirá retornar um texto completamente descriptografado. Para isto, deve ser feito um refinamento de sua saída, com uma nova análise de freqüência, mas não de símbolos e sim de palavras da língua esco-

lhida, obtendo assim um banco de dados de palavras que existem na língua. A partir dele realizar uma consulta ao texto no qual o programa executou a quebra e então verificar quantas palavras do arquivo de saída são legíveis, e quantas ainda estão ilegíveis. O programa deve ser novamente executado, mas pequenas alterações na freqüência da língua devem ser feitas para que o arquivo de saída se encontre diferente, com algumas alterações. O novo arquivo de saída deve ser novamente analisado e comparado ao banco de palavras. Se for obtido um maior número de palavras pertencentes à língua as alterações devem ser mantidas. Se o inverso ocorrer, essas alterações devem ser desfeitas. Isto deve ser refeito com uma distorção cada vez maior da freqüência até que o arquivo obtido se encontre satisfatório.

# Capítulo 5

# Avaliação do algoritmo de criptografia posicional sobre o ataque do método de análise de freqüência

## 5.1 Introdução

Neste Capítulo será apresentado como o algoritmo de criptografia posicional se mostrou perante o ataque pelo método de análise de freqüência. Será avaliada sua segurança a este tipo de ataque e os dados atingidos.

## 5.2 A segurança do algoritmo de criptografia posicional

### 5.2.1 Introdução

Os autores[3] do algoritmo de criptografia posicional acreditavam que sua segurança estaria fortemente ligada ao fato de que “foi introduzida uma função sobre o algoritmo *Cifra de César* para que cada posição seja criptografada de forma diferente eliminando o problema de freqüência de letras”. Além disto também acreditavam que “nesse tipo de criptografia, quanto maior o grau da expressão posicional, maior será a complexida-

de do algoritmo e por sua vez a segurança dos dados encriptados usando-se desse algoritmo".

Neste projeto será demonstrado que a introdução da função não torna a *Cifra de Cezar* uma cifra polialfabética e sim uma combinação de varias substituições monoalfabéticas, usadas em rotação. Será demonstrado também que o grau da função de criptografia nada interfere na quebra por analise de freqüência de símbolos e sim na quebra por força bruta e ao tentar descobrir as chaves de criptografia.

### 5.2.2 Importância de posição

O critério utilizado pelo algoritmo é a posição em que o caracter está posicionado, isto realmente impede que caracteres iguais sejam encriptados da mesma forma o que impossibilita a quebra por análise de freqüência, mas na fórmula de criptografia é inferida a operação *mod*256 o que após estudos mostrou que a cada posição múltipla de 256 temos uma encriptação feita da mesma maneira. Quando isto foi detectado pode-se perceber que os subconjuntos de caracteres múltiplos de 256 formados no texto criptografado formavam cifras monoalfabéticas.

Desenvolveu-se então um *software* capaz de executar a análise em um texto criptografado pelo algoritmo posicional. Ele irá analisar o conjunto de caracteres cujo as posições sejam múltiplas de 256 e assim obtendo a análise de freqüência.

### 5.2.3 Resultados práticos do S*oftware*

Foram realizados testes em um computador AMD K6-II 500MHZ com 128MB de memória RAM e com sistema operacional Windows 98 sobre diversos textos foram observados alguns resultados práticos.

Pode-se observar que o tempo de quebra cresce linearmente com o tamanho do texto, este resultado era esperado, pois o tempo de quebra de cada caracter é o mesmo então o aumento do tempo é devido ao aumento do número de caracteres. O gráfico de tempo X tamanha ficaria então desta maneira.

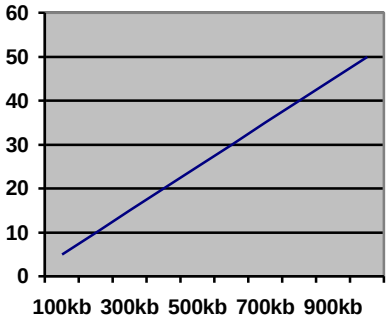


**Figura 5.1:** Tempo(min) X tamanho de quebra do algoritmo posicional

Realizou-se também testes para saber o quanto do texto foi realmente descriptografado. Para isso comparou-se caracter por caracter do texto no formato normal e depois de quebrar sua criptografia. Os resultados obtidos foram:

| Tamanho | Porcentagem de quebra |
|---|---|
| 163kb | 46% |
| 288kb | 55% |
| 989kb | 74% |

**Tabela 5.1:** Tamanho do texto pela porcentagem de quebra obtida

### 5.2.4 Análise dos resultados do *Software*

Como foi possível identificar o grau da função de criptografia não influência de forma alguma na quebra da cifra por análise de freqüência, mas é aconselhável utilizar um alto grau, pois o grau da função é de grande importância se a tentativa de quebra for por força bruta e também para identificar as chaves de criptografia na análise de freqüência.

O grau da função não influência na quebra da cifra por análise de freqüência, pois a quebra será executada no resultado e não na função, independente do grau da função utilizada se dois caracteres iguais que estiverem em posições múltiplas de 256 receberam a mesmo número de codificação, e este será analisado pela freqüência que aparecerá no texto. Como o método atua no resultado, o grau da função não importa.

Também foi possível observar que o tamanho do texto é um parâmetro de grande importância na medida da segurança do algoritmo. Quanto maior o texto mais vulnerável à quebra.

Isto pode ser constatado pela própria idéia da análise de freqüência. Como será identificada a freqüência com que cada caracter aparece no texto, precisa-se de uma quantidade de caracteres que permita retornar um resultado expressivo sobre a freqüência com que eles aparecem.

Isto deixa claro que quanto maior o número de caracteres que um texto possui, maior o conjunto de caracteres capazes de denunciar sua freqüência. Um conjunto pequeno de caracteres pode denunciar um resultado falso sobre a freqüência da língua utilizada, o que levaria a uma quebra de criptografia de pouca clareza.

O resultado obtido é um texto decodificado, porém a qualidade do texto obtido deve ser aprimorada, para que a legibilidade dele seja boa. Como foi descrito em 3.4 após o texto descriptografado ser obtido, é necessário realizar um refinamento dos dados, isto pode ser feito, utilizando-se um dicionário e realizando uma varredura no texto identificando quantas palavras pertencentes a língua foram extraídas do texto.

Após isto ter sido feito um novo texto deve ser descriptografado, mas com pequenas alterações na freqüência para que um texto de maior qualidade seja obtido. O processo de ser realizado quantas vezes for necessário.

Todo este processo sendo realizado é possível obter a quebra da criptografia e obter um texto legível, isto atenderia as pretensões de atacantes ativos e passivos.

Os atacantes passivos desejam apenas ter acesso aos dados, e isto será possível já que o texto estará legível.

Os ativos que pretendem modificar os dados poderão fazê-lo. O atacante ativo terá acesso a todos os caracteres na forma criptografada e também descriptografada. Como se conhece a posição dos caracteres no texto é só construir uma tabela com o valor do caracter seu correspondente criptografado e a posição na qual isto é verdadeiro, com isto é só alterar a criptografia.

### 5.2.5 Chaves de criptografia

No processo de quebra criptográfica, porém, o interessante é obter as chaves de criptografia, com as chaves de criptografia descobertas toda a segurança do método é desfeita.

Ao aplicar-se o método de análise de freqüência de símbolos no algoritmo de criptografia posicional deparou-se com o problema de não se conhecer o grau da função de criptografia.

Para decodificar um símbolo codificado pelo algoritmo posicional usa-se a seguinte expressão ASCII = (c - f(x)) (*mod* 256), onde *c* é o valor do símbolo codificado e *ASCII* é o valor do símbolo decodificado. Ao utilizar o método se consegue obter o valor do caracter decodificado e também o valor do caracter codificado, com isso conseguimos obter o valor de f(x) ao utilizar a fórmula descrita acima.

Entretanto deseja-se descobrir a chave de codificação, a qual é composta pelos coeficientes de f(x) e para tal necessita-se do grau de f(x), pois o número de coeficientes de uma função é o número do seu grau mais um.

Ao aplicar-se o método da análise de freqüência no algoritmo de criptografia posicional, tem-se uma maneira de descobrir-se à imagem de f(x) e a partir deste fato encontra-se a chave de codificação. Para encontrar a chave de codificação monta-se um sistema linear com os valores conhecidos de f(x), obtendo como resultado desse sistema o valor da chave de codificação usada no processo. Ao montar tal sistema necessita-se do grau da função de criptografia, pois ele determina o número de equações que formaram o sistema. Caso contrário tem-se que supor um grau qualquer e resolver o sistema com este suposto grau. Se o resultado não for satisfatório repete-se a operação com um novo grau, uma espécie de força bruta, até que o resultado seja satisfatório.

Para montar-se tal sistema linear encontra-se a imagem de f(x) com método da análise de freqüência de símbolos e como se sabe que o valor de *x* é a posição do caracter no texto, a qual é conhecida, tem-se que os únicos valores desconhecidos são os coeficientes de f(x), os quais objetiva-se descobrir. Dessa forma deve-se ter um sistema com n equações, assim formadas: a primeira equação é formada tomando como valores de *ASCII* e *COD* os símbolos mais freqüentes no idioma no qual foi escrito o texto simples e no texto codificado, respectivamente e obtém-se a primeira imagem de f(x), e a segunda a terceira e as outras equações são formadas de maneira semelhantes, considerando como valores de *ASCII* e *COD* do segundo e assim por diante os outros símbolos mais freqüentes tanto no texto simples quanto no texto codificado. O valor de *x* é a posição do símbolo no texto, a qual é determinada pela primeira posição considerada ao iniciar a análise de freqüência no texto codificado. Dessa forma pode-se concluir que a segurança do algoritmo de criptografia posicional é dependente do grau da função de criptografia tanto para quebra por força bruta quanto para quebra por análise de freqüência, quando se trata do fato de encontrar a chave de criptografia. Uma maneira eficiente de decifrar tal código é saber qual o grau utilizado na função de criptografia.

# Capítulo 6

# Conclusão

Neste projeto foi analisada a segurança do algoritmo de criptografia posicional sob o ataque de quebra criptográfica pelo método de análise de freqüência de símbolos.

Foi desenvolvido um *software* capaz de extrair de textos a freqüência dos caracteres que formam sua linguagem. Isto foi necessário, pois para realizar a quebra por análise de freqüência de símbolos é necessário saber a priori qual a freqüência real da linguagem escolhida. No caso do projeto em questão a linguagem escolhida foi à língua portuguesa. Escolha esta feita em função de ser a língua pátria do algoritmo de criptografia e do autor deste projeto.

O *software* é capaz também de criptografar os dados de um texto através do método posicional. Esta função foi implementada com o intuito de obter textos encriptados para que testes fossem realizados sobre eles. É importante observar que uma pequena modificação foi realizada na idéia inicial do algoritmo de criptografia com o intuito de facilitar os cálculos que seriam realizados. Está modificação como foi provada não interfere no resultado da criptografia com isso não altera os dados do problema, tornando assim os testes válidos.

O processo de quebra foi realizado com base na influência de operação matemática *mod*256. Como foi atestado que esta operação faz com que caracteres iguais que estiverem em posições múltiplas de 256 sejam encriptadas da mesma maneira. O *software* consegue identificar e separar os caracteres múltiplos de 256 em todo o texto

e realizar sobre eles a análise de freqüência de símbolos. Alterando assim os caracteres de freqüências iguais.

Como foi está mostrado em anexo, o texto descriptografado não está legível, o resultado obtido necessita de refinamento. Foi proposto que este refinamento se realize utilizando um dicionário contendo palavras da língua escolhida e que se faça uma varredura no texto descriptografado, buscando encontrar quantas palavras no texto existem na língua. Após isto ser feito uma nova operação de descriptografia deve ser feita, mas com pequenas alterações na freqüência dos símbolos, um novo texto deve ser gerado e neste novamente deve ser feito uma varredura para identificar quantas palavras contidas no texto existem na língua. Se o número de palavras for maior que no texto anterior as modificações devem ser mantidas, caso contrário devem ser desfeitas. A operação deve ser repetida até que um texto legível possa ser obtido.

Como foi averiguado, para que o que a cifra seja quebrada o grau da função de encriptação não influência em nada, já que a quebra é feita utilizando-se os símbolos resultantes da operação. Mas como em toda a quebra de criptografia o interessante é que seja obtida a chave de criptografia.

Para que a chave de criptografia seja obtida é necessário que se construa um sistema formado por equações lineares, estas equações devem conter o código ASCII do caracter encriptado, o código ASCII do seu equivalente decriptado e então deduzir um grau à função para que possa ser testado. Se o resultado for negativo, será necessário realizar a operação novamente atribuindo a função um novo grau, até que se descubra qual o seu verdadeiro grau e com isso a chave de criptografia.

Pode se perceber então claramente que se tratando de encontrar as chaves de criptografia, o grau da função de encriptação influência na complexidade do problema.

O algoritmo posicional apresentou também uma fragilidade em sua segurança quando o texto a ser encriptado aumenta de tamanho. Como na quebra pelo método de análise de freqüência é necessário que se análise a freqüência dos caracteres que se encontram encriptados. Fica fácil verificar que quanto maior o texto, maior a quantida-

de de caracteres encriptados da mesma maneira. E com isso maior a probabilidade de encontrar a freqüência correta dos caracteres com isso uma quebra do código mais eficiente.

# Capítulo 7

# Trabalhos Futuros

Dentre os possíveis trabalhos futuros sugere-se:

- Desenvolver o software capaz de analisar o número de palavras existentes na língua e no texto descriptografado e após isso ser capaz de descriptografá-lo novamente com pequenas alterações na freqüência mantendo-as se o resultado for positivo e alterando novamente se for negativo. Isto é necessário para o refinamento do resultado.

- Analisar a segurança pelo método de analise de freqüência só que através de sistemas, tentando descobrir o grau da função por força bruta.

- Analisar a freqüência com um *lookahead* para que se possa identificar as freqüências que não estiverem de acordo com o texto. No caso de se encontrar por exemplo o caracter ‘d’ e com o *lookahead* verificar que o proximo caracter a ser inserido será o ‘c’, na língua portuguesa não existe ‘dc’ então isto deverá ser tratado.

# Bibliografia

[1]Oliveira, R. M. L.. *Uma Análise da Segurança e da Eficiência do Algoritmo de criptografia Posicional.* Monografia, UFLA, 2001.

[2]Coutinho, Severino Collier. *Números Primos e Criptografia RS*A. IMPA, 2000.

[3]Moreno. Edward David e Chiaramonte. Rodolfo Barros. Criptografia posicional: Uma solução para segurança de dados, 2001. Artigo públicado na Revista de Iniciação Científica da SBC.

[4]Prazeres, Cássio Vinícius Serafim e Mata Júnior. Gilberto Bittencourt e Reis Júnior. Paulino Batista. Fundamentos teóricos da criptografia, 2000. Mono-grafia de Graduação.

[5]Hemmessy, John L. e Patterson. David A. *Computer Architecture: A Quanti-tative Approac*h. Morgan Kaufmann, 1996.

[6]Garfinkel, Simson e Spafford. Gene. *Comércio e Segurança na We*b. Market Books, 1999.

[7]Weber, Raul Fernando. Criptografia contemporânea, 1995. Artigo publicado no sítio www.módulo.com.br.

[8]Ziviani, Nívio. *Projeto de Algoritmos - Com Implementações em Pascal e* C. Editora Pioneira, 1999.

[9]Goldreich, Oded. *Foundations of Cryptograpy - Basic Tool*s. Cambridge Univ Press, 2001.

[10]Tanenbaum, Andrew S. *Rede de Computadore*s. Editora Campus, 1997.

# Apêndices

## Tabela de freqüência

Foram analisados 1.53281e+06 caracteres e foi encontrando 134 caracteres diferentes e a freqüência encontrada está ordenada na tabela:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **01** | Espaço | **35** | A | **69** | ? | **103** | & |
| **02** | a | **36** | j | **70** | 8 | **104** | [ |
| **03** | e | **37** | E | **71** | w | **105** | = |
| **04** | o | **38** | C | **72** | ‘ | **106** | ] |
| **05** | s | **39** | ; | **73** | L | **107** | Á |
| **06** | r | **40** | ó | **74** | 5 | **108** | \ |
| **07** | i | **41** | O | **75** | 3 | **109** | % |
| **08** | d | **42** | ê | **76** | ° | **110** | ö |
| **09** | n | **43** | õ | **77** | W | **111** | – |
| **10** | t | **44** | S | **78** | G | **112** | $ |
| **11** | m | **45** | à | **79** | É | **113** | Í |
| **12** | u | **46** | M | **80** | H | **114** | Y |
| **13** | c | **47** | ú | **81** | ü | **115** | ´ |
| **14** | l | **48** | N | **82** | 6 | **116** | è |
| **15** | p | **49** | D | **83** | 4 | **117** | ¢ |
| **16** | , | **50** | P | **84** | 7 | **118** | _ |
| **17** | v | **51** | : | **85** | ( | **119** | \| |
| **18** | g | **52** | I | **86** | ) | **121** | Ó |
| **19** | . | **53** | — | **87** | ª | **122** | … |
| **20** | f | **54** | B | **88** | y | **123** | ñ |
| **21** | b | **55** | T | **89** | k | **124** | Ê |
| **22** | h | **56** | F | **90** | À | **125** | < |
| **23** | q | **57** | R | **91** | X | **126** | ½ |
| **24** | ã | **58** | V | **92** | º | **127** | ` |
| **25** | ç | **59** | â | **93** | K | **128** | Ú |
| **26** | - | **60** | 1 | **94** | Z | **129** | È |
| **27** | Tabulação | **61** | ô | **95** | / | **130** | * |
| **28** | Pulo | **62** | 2 | **96** | Ã | **131** | ¯ |
| **29** | à | **63** | Q | **97** | + | **132** | Ë |
| **30** | z | **64** | 9 | **98** | * | **133** | |
| **31** | é | **65** | 0 | **99** | “ | **134** | > |
| **32** | í | **66** | J | **100** | ” | | |
| **33** | x | **67** | ! | **101** | • | | |
| **34** | ¨ | **68** | U | **102** | Ç | | |

# Texto descriptografado

## Pedaço do texto de 989kb

Este uivrop seuamatne adrrars A pufldcinade, sem amparo ne huapbmer dancro
a, ,ara que os ,rstestos cottra as balsiiades hce auaoo encerrasse se eéercitasoem perheidamente desahogados, cot-cisnoc - franua e espsmtRiea h eávrossa ,elo secs teufores Nrg
osv a gratne sitlania tohilimanora na miiha terral qce d-o socncdten e ãce me nesvadeleb js Nnruso tesuiées aponnados leca crátila sço, leca pr;pria tesvaldap gastante eco-3edmes to deuanarem a segciadãa ias rtxias e proposd-jes avedtaiasf
éR s -ce demodotra esma resedfa iq,itaNéíú ç . . g Merleizriss inconsciednes
g Ostranfou-se a eí,ressãob Pas nevo madtNçla6 matnethoãa.áá:
o trve s intuddo de de.edner oo serteneAssp por.ue este livro dío E cm ,nvrs no debeoaV x, idfeciímette, de ataquefá
Ama-ce .ratls e, devo dijEçlo, itvouctt
rie. óesse idvesnir, apareddemetie desafiador, com es sidbccar
osimos udviudéados qce tss sert:es, niante de somdhár.aros, estadeaiam táo uasmit
veis selvatigceáasv o.enecn as rigor rduserlávec na veriadeb Cidgu
m e devarzbíáõ oe não nemesse envardar-te em paralelo que ião mere-ol .ravarda na lrimaira pzgina a frase ns.retedte siicera de ôulxdinesp ao esurover a fistória da gcerra ts Oeuopoieoo - ,or-ce eu tam.ém emhora sem a mesma visço ahuipiia, escrevi
ásem iar ur
nino és ,rimeiras nestomcdhas hce edpontrerl nem ;s miihas próprdas im,ressàes, mas narradne apedas oo alodmeuimedtos de hue bui es,eltanor ou sobre os buais tive rdformaçCes se.cras.çãIB - .g. nesabriganas ne tods anne e auideA lorrosiva ios agcaceiros metpestuoses... h
4iuqse desma qrase uma rnejadinão e um dos imaginosos draqos do meu apenre;ano nefeli.atisms liein
fico fíá Bevdsda do Cedtie de 0etras e ármes, ne Camlnias, if0 T, F4 ne Aadeiro de ôFC1g
çzra, eslasseantoçme o mempo lara amar aunores, limino-ne a aponnar a l
gnda S8U da Ueolovra re àonteçead oofre a eros-o tao roubasDíémes altiots l-Goihces et uhimd-ues ,ar leo eauó pluviales ,ccs ou monds l.argzes d4a,nde car.stiãue - lritcipalemetn scr les rocqes ues pclo attahuabces ací auites, comme ces laucaires enl.

áêara o uaso es,epdal do ;rasil, eicodtra-se ainna N pégnna :1P to li,re de émh 1iais, so.re a iossa lodgorma-zo geocOgiua, a uarapternía
zo ns henúmemo -ua se motmre ed mrGr granne
cfeule, sano nelte ; uause de ia brxhcetpe et ie uQ acidinõ des ppuies d0orabe-zíRe ettadno o liétdlo cepioiaM Nem as pfmvas lauoat erosEes por lonmerem acvumas mol luuas a mais de titro ou ne ameiãauo p sen
o peua rrzeéa ia lameda foriíodtau slperior em reuaãço às camadas moceo inqerdores enugz
éénraoriin-ria beolo.ia, esmaf.gq-OBó ç .b. as qavelas tem, das folfao, de esnCmatos ezlaididos et vipesidates b . g b
íúpressohme em uorri.ir evddednássrmo envano, dratannoqse te noçço n-o oitpcesb
ã1eia-seú nas houfasp te cáuuuas eA,atdidas em viuosddanesbAxRP q J bue a mor.ocsvia da terra vioca as ueis geiais nos puimas .á
Pctro tiéer macsiiado. Gmlugdaho resleim
geu uiedtdsmaGzz1etso fue se a natureza lembame es tesertos, apenas o -a,ies geogr hico tonihica as coddnã;es eítiínseuas ns meiob é se violéiuia importa msdi.ica

o, giocar é deso.etouer ao creesmaqecelidob éssim, no b viocaççs contre as leis gerais doo cuimasp eis o -ue no patece tu,ina g íí éorreie na Mam-á ne 1 te fevereiro de TRUTb zzPteãplilvep cotmraditav esta, hce investe uot todas as loducusjos da meteo-rocsvia modernaB Easma saber
se -ue setdo as peis gerads te lm lcima as bme se terivat tas relaçjes aotrodVticas ç as ,rj,rias odtcla-àes dss isodernos , rdiiscdlcnnadatenne reuurvos , mas qle segcnriam os paraueloo oe respeitassem afuelas leds, s
s cm amesdato da viopaããoféé Nem lrepis
vanoo ejempcifdcar o lreiomAdio ,ermanente nas cacsas partilucares ol oeludnãrias da codsmdncdãão luim
tila ne huacquer paisg De 1adtos, ccNo ccima e-latoriac z uma adoma,na em uatntcte superior A io trNlicsp E 1roocVndia co-orta de .elos grodteiia às laragets fetivnas ta Rsrue.al enlodnrarjatos espcóddinos ezemluosb
Oidda reledtemedne, no belo ud,ro sohro a lsilelovna dos itbcesesp EsctmG aositala o hano ne ter a Rd.uaterrav no ,aralecs te I7gL, nem,eramcra iguau a 1ú.ü no lad.l dos Estados Vnidosg
ã4cem quor que acompatfe nut ma,a o isoderuo de S.4, partirE na brigimássima 7scúimiap aganãar
,ara o suu, numa uurga capril.osa, para a Rnvlamerra, çle dão tolarõA norcerA nelors para o eítrems norte ta 5oruegap e vouvorz de dego as sul e se a,roximaiqp nos meses hiiosl ne Bario e de 5ieda - que assim se ,igam malgrato canitldes muito mais bai
ao, õ enregecana terra ,elar.