Extensão da Cifra de César: Implementação de Cifras Baseadas em Chaves Numéricas Variáveis

Arthur Monteiro Pereira[1], **Filipe Brinati Furtado**[1]

[1]Depto de Ciência da Computação - Universidade Federal de Juiz de Fora (UFJF)
Caixa Postal 20010 – 36036-900 – Juiz de Fora – MG – Brasil

arthurmpereira2010@gmail.com, brinati@ice.ufjf.br

Abstract. *This work presents modifications to the traditional Caesar Cipher, with the aim of increasing its security and complexity. The new variants, called cifraAF and cifraAF2, use a variable numeric key applied to the complete ASCII table, making encryption more robust. The implementation includes adapting to multiple digits of the key and creating a brute force function to test the strength of the ciphers. The study demonstrated that these changes can make simple attacks, such as brute force, more difficult, suggesting the exploration of improvements and future applications in modern security systems.*

Resumo. *Este trabalho apresenta modificações na tradicional Cifra de César, com o objetivo de aumentar sua segurança e complexidade. As novas variantes, denominadas cifraAF e cifraAF2, utilizam uma chave numérica variável aplicada sobre a tabela ASCII completa, tornando a criptografia mais robusta. A implementação inclui a adaptação para múltiplos dígitos da chave e a criação de uma função de força bruta para testar a resistência das cifras. O estudo demonstrou que essas alterações podem dificultar ataques simples, como os de força bruta, sugerindo a exploração de melhorias e aplicações futuras em sistemas de segurança modernos.*

1. Introdução

Nos últimos dez anos, os meios de controle de informações se desenvolveram de maneira exponencial, tornando-se cada vez mais integrados ao nosso cotidiano. Inicialmente restritas a contextos militares ou confidenciais, essas tecnologias agora permeiam diversos setores, demonstrando sua eficácia em proteger dados sensíveis. No entanto, apesar de suas vantagens inegáveis, a adoção dessas tecnologias trouxe novos desafios, como a necessidade de garantir o sigilo e a autenticidade das informações transmitidas e armazenadas [Terada 2008].

Diante desse cenário, diversas metodologias de segurança foram revisitadas e aprimoradas para atender às demandas contemporâneas de proteção de dados. Entre essas metodologias, destaca-se a criptografia, definida por [Mishra 2013] como "a arte e a ciência de criar dados não legíveis ou cifras, de forma que apenas a pessoa pretendida seja capaz de decifrar e ler os dados."

Um dos métodos de criptografia mais básicos é o de substituição, que consiste em substituir cada elemento da mensagem original por outro de acordo com uma regra predefinida [Terada 2008]. Este método é amplamente utilizado devido à sua simplicidade e facilidade de manutenção, já que a mesma chave é usada tanto para criptografar quanto para descriptografar a mensagem. Um exemplo clássico deste tipo de criptografia é a cifra de César, na qual cada letra do texto é deslocada por um número fixo de posições no alfabeto.

Embora a cifra de César seja um método histórico e de importância fundamental para o estudo da criptografia, ela possui limitações significativas em termos de segurança, especialmente no contexto das tecnologias modernas. Este trabalho propõe uma modificação da cifra de César e da chave utilizada, com o intuito de superar as vulnerabilidades inerentes a essa técnica tradicional [Gowda 2016].

Este artigo está organizado da seguinte forma: a Seção 2 apresenta as características da cifra de César, explorando trabalhos relacionados na Seção 3. A Seção 4 discute as modificações propostas para a cifra, enquanto a Seção 5 analisa a eficácia do método modificado. Finalmente, as ameaças à validade são apresentadas na Seção 6 e as conclusões são apresentadas na Seção 7.

2. Cifra de César

A cifra de César é uma das formas mais antigas e simples de criptografia. Este método de criptografia é um tipo de cifra de substituição monoalfabética, na qual cada letra do alfabeto é substituída por outra letra que se encontra um número fixo de posições à frente ou atrás no alfabeto. A cifra de César se destaca por sua simplicidade e facilidade de implementação, o que a torna um exemplo clássico para introdução ao estudo da criptografia [Terada 2008].

O funcionamento da cifra de César é bastante direto: cada letra do texto original, chamado de texto claro, é deslocada um número fixo de posições no alfabeto, determinado pela chave de criptografia. Por exemplo, com uma chave de deslocamento de 3, a letra 'A' é substituída pela letra 'D', 'B' pela letra 'E', e assim por diante. Este deslocamento é aplicado a todas as letras do texto claro para produzir o texto cifrado. Se a substituição ultrapassar o final do alfabeto, o deslocamento continua a partir do início, criando um

ciclo. O processo de criptografia e descriptografia são inversos um do outro: enquanto na criptografia cada letra é deslocada para a direita, na descriptografia cada letra é deslocada para a esquerda pelo mesmo número de posições.

Para ilustrar o funcionamento da cifra de César, considere a criptografia da palavra ”HELLO”usando uma chave de deslocamento de 3. Nesse caso, a letra ’H’ é deslocada três posições para a direita, resultando em ’K’, ’E’ torna-se ’H’, ’L’ torna-se ’O’, e ’O’ se transforma em ’R’. Assim, o texto claro ”HELLO”é transformado no texto cifrado ”KHOOR”. Para reverter o processo e decifrar ”KHOOR”com a mesma chave, cada letra é deslocada três posições para a esquerda, devolvendo ”HELLO”. Esse processo simples de deslocamento revela tanto a facilidade de uso da cifra de César quanto suas limitações, especialmente em termos de segurança.

Apesar de sua importância histórica, a cifra de César apresenta várias fraquezas significativas quando considerada para uso moderno. Sua simplicidade a torna vulnerável a ataques de força bruta, nos quais todas as possíveis chaves de deslocamento são testadas até que o texto claro seja recuperado. Com um alfabeto de 26 letras, existem apenas 25 chaves possíveis (excluindo o deslocamento de 0, que não alteraria o texto), o que significa que é relativamente fácil para um atacante descobrir a chave correta. Além disso, a cifra de César não altera a frequência das letras; as letras mais comuns no texto cifrado correspondem às letras mais comuns no idioma original. Isso a torna suscetível a ataques de análise de frequência, onde um atacante pode identificar a chave com base na frequência de ocorrência das letras no texto cifrado.

Por essas razões, a cifra de César é considerada insegura para proteger informações sensíveis nos dias de hoje. No entanto, ela continua a ser um excelente ponto de partida para entender os princípios básicos da criptografia e a evolução dos métodos de segurança da informação. Ao aprender sobre a cifra de César, podemos apreciar os fundamentos da criptografia e entender a necessidade de métodos mais avançados e robustos para garantir a confidencialidade e a integridade dos dados em um mundo digital cada vez mais complexo.

3. Trabalhos Relacionados

Com o objetivo de apresentar como outras pesquisas têm modificado a cifra de César para melhorá-la, esta seção explora as alterações realizadas nesses estudos.

Um dos trabalhos relacionados propõe uma nova abordagem para a cifra de César, onde a chave de criptografia é fixada em um valor de 1. Tradicionalmente, a cifra de César utiliza um valor inteiro como chave para determinar o número de posições a serem deslocadas no alfabeto para substituir as letras do texto claro. Este método se baseia na aritmética módulo 26, garantindo que o valor inteiro seja envolvido no caso de a chave ser maior que 26. O processo de decriptação segue a lógica inversa da encriptação, utilizando uma chave complementar para restaurar o texto original. No entanto, neste trabalho, o autor introduz uma modificação: ao invés de um deslocamento fixo, a substituição de caracteres é determinada pelo índice alfabético de cada letra. Se o índice for par, o valor é aumentado em uma unidade; se for ímpar, é reduzido em uma unidade. Além disso, os caracteres do texto cifrado são embaralhados, dificultando a decriptação. O algoritmo de encriptação começa com a inserção do texto claro e, após verificar e ajustar os índices das letras, gera o texto cifrado. O processo de decriptação segue passos semelhantes,

aplicando o ajuste inverso aos índices para recuperar o texto original. Essa abordagem visa aumentar a complexidade da cifra e dificultar ataques de força bruta e tentativas de decriptação sem o conhecimento da chave [Goyal and Kinger 2013].

Outro trabalho relacionado propõe uma versão aprimorada da cifra de César, utilizando uma técnica chamada ”Transposição de Linhas Multinível com chaves iguais e diferentes”. Essa abordagem tem várias vantagens em relação à cifra de César simples. Ao aplicar o método de transposição dupla, a cifra resultante apresenta uma permutação muito menos estruturada, o que aumenta significativamente a dificuldade de criptoanálise. Isso significa que o padrão original das letras é obscurecido de tal forma que é difícil reconstruir o texto original, mesmo que se conheça parte da mensagem. Como resultado, ataques de força bruta se tornam inviáveis, pois a complexidade adicional introduzida pela transposição multinível exige um número muito maior de tentativas para quebrar a cifra. Além disso, essa técnica supera todas as limitações da cifra de César tradicional, tornando-a uma opção muito mais segura para proteger informações [Mishra 2013].

Outro trabalho relevante propõe um método que combina a cifra de César com a cifra de Vigenère, introduzindo modificações adicionais para aumentar a segurança. Neste método, são geradas aleatoriamente duas chaves: uma chave numerada e uma chave em letras. Primeiro, aplica-se a cifra de César na chave em letras usando o valor de deslocamento definido pela chave numerada. Em seguida, o texto claro a ser criptografado é processado usando a cifra de Vigenère, utilizando a nova chave gerada pela aplicação inicial da cifra de César. Essa transformação das chaves tem como objetivo dificultar a análise de frequência, especialmente em mensagens curtas, pois a estrutura simples das chaves é modificada. Além disso, ao final do processo, o binário da primeira letra do texto cifrado resultante é XORado com o binário da chave numerada, e o resultado é criptografado com o texto cifrado subsequente. Este resultado é então convertido para o seu valor binário na tabela ASCII. Como resultado final, é gerado um texto cifrado que mistura números, símbolos e letras, tornando a cifra final consideravelmente mais complexa e resistente a métodos tradicionais de criptoanálise [Omolara et al. 2014].

Outro trabalho relacionado propõe um método que combina uma troca de chaves simplificada, similar ao Diffie-Hellman, com a cifra de César. No processo, dois usuários trocam valores baseados em suas chaves privadas e uma chave secreta compartilhada para calcular uma chave criptográfica comum. Esta chave é então usada para cifrar o texto. A cifra de César é ajustada utilizando a chave compartilhada para determinar o deslocamento das letras. Se a operação módulo 26 da chave compartilhada resulta em zero, a chave é incrementada para evitar deslocamentos nulos. Além disso, espaços no texto são substituídos por caracteres baseados na operação módulo. Esse método visa aumentar a segurança da cifra de César, combinando a troca de chaves com ajustes adicionais na criptografia [Gowda 2016].

Outro trabalho relacionado apresenta a Cifra Monoalfabética Legível, uma modificação da cifra de César projetada para aumentar a legibilidade e a adequação ao idioma indonésio. Neste método, vogais são substituídas apenas por outras vogais, e consoantes por outras consoantes. Letras como Q, V, X, Y, Z, N e G não são substituídas para evitar resultados incomuns. As fórmulas de substituição são $C_v = (p_v + b_k) mod 5$ para vogais e $C_c = (p_c + b_k) mod 14$ para consoantes. A cifra combina as substituições de vogais e consoantes, totalizando 70 possibilidades para cada tipo, e resulta em um texto

cifrado que é mais difícil de quebrar enquanto mantém uma estrutura legível e prática para o idioma [Purnama and Rohayani 2015].

4. Modificação da Cifra de César

A modificação feita na cifra de César tem como objetivo torná-la mais robusta e difícil de ser quebrada através de abordagens tradicionais de criptoanálise, como a força bruta. Enquanto a cifra de César clássica utiliza um único valor fixo para deslocar os caracteres de uma mensagem, a nova implementação se baseia em uma chave numérica variável, o que oferece maior segurança ao introduzir um padrão mais complexo de cifragem.

Na implementação original, a cifra de César simples é aplicada a caracteres maiúsculos e minúsculos separadamente, deslocando-os por um número fixo de posições no alfabeto, determinado pela chave fornecida. Essa abordagem, apesar de funcional, é vulnerável a ataques simples de força bruta, pois o número de combinações possíveis é limitado ao tamanho do alfabeto, no caso, 26 posições para letras.

Na nova versão da cifra, nomeada de cifraAF e apresentada no *Algorithm* 1, o deslocamento de cada caractere depende de uma sequência de dígitos fornecida pela chave, e não de um único número. Isso significa que o primeiro caractere será deslocado de acordo com o primeiro dígito da chave, o segundo pelo segundo dígito, e assim sucessivamente. Ao final da sequência, a iteração retorna ao início da chave, criando um ciclo. Além disso, a cifra se aplica a todos os caracteres imprimíveis com base na tabela ASCII, o que amplia o escopo de criptografia para incluir pontuações e símbolos, tornando a cifra ainda mais resistente.

Algorithm 1 cifraAF(texto, chave)

$chave\_str \leftarrow$ Converter chave para string
$tamanho\_chave \leftarrow$ comprimento de $chave\_str$
$resultado \leftarrow$ string vazia
for cada caractere c em $texto$ com índice i **do**
    $valor\_ascii \leftarrow$ código ASCII de c
    if $32 \leq valor\_ascii \leq 126$ **then**
        $deslocamento \leftarrow$ dígito $i \mod tamanho\_chave$ da $chave\_str$
        $novo\_valor\_ascii \leftarrow 32 + ((valor\_ascii - 32 + deslocamento) \mod 95)$
        $novo\_caractere \leftarrow$ caractere correspondente a $novo\_valor\_ascii$
    else
        $novo\_caractere \leftarrow c$
    end if
    Adicionar $novo\_caractere$ a $resultado$
end for
Retornar $resultado$

A versão cifraAF2, apresentada no *Algorithm* 2, refina ainda mais esse conceito, utilizando pares de dígitos da chave para realizar deslocamentos. Esse modelo torna o processo de cifragem menos previsível, já que para cada caractere a cifra considera dois dígitos consecutivos da chave, criando uma variação adicional no padrão de deslocamento.

Algorithm 2 cifraAF2(texto, chave)

$chave\_str \leftarrow$ Converter chave para string
$tamanho\_chave \leftarrow$ comprimento de $chave\_str$
$resultado \leftarrow$ string vazia
for cada caractere c em $texto$ com índice i **do**
    $valor\_ascii \leftarrow$ código ASCII de c
    if $tamanho\_chave > 1$ **then**
        $indice\_chave1 \leftarrow (i * 2) \mod tamanho\_chave$
        $indice\_chave2 \leftarrow (i * 2 + 1) \mod tamanho\_chave$
        if $indice\_chave2 < tamanho\_chave$ **then**
            $deslocamento \leftarrow$ valor concatenado de $chave\_str[indice\_chave1]$ e $chave\_str[indice\_chave2]$
        else
            $deslocamento \leftarrow chave\_str[indice\_chave1]$
        end if
    else
        $deslocamento \leftarrow chave\_str$
    end if
    if $32 \leq valor\_ascii \leq 126$ **then**
        $novo\_valor\_ascii \leftarrow 32 + ((valor\_ascii - 32 + deslocamento) \mod 95)$
        $novo\_caractere \leftarrow$ caractere correspondente a $novo\_valor\_ascii$
    else
        $novo\_caractere \leftarrow c$
    end if
    Adicionar $novo\_caractere$ a $resultado$
end for
Retornar $resultado$

Essas mudanças visam criar uma cifra que, embora baseada em um conceito tradicional como a cifra de César, seja mais adequada aos requisitos modernos de segurança, elevando a dificuldade de ataque e tornando-a menos vulnerável a técnicas de criptografia simples.

5. Resultados

As alterações realizadas introduziram várias vantagens significativas:

- **Aumento da complexidade na chave de criptografia:** A modificação da cifra de César para utilizar chaves numéricas de múltiplos dígitos (como nas versões cifraAF e cifraAF2) introduz um nível adicional de complexidade à chave, dificultando significativamente o processo de quebra por ataques de força bruta. A possibilidade de variação no comprimento da chave oferece maior flexibilidade e segurança ao processo de criptografia.
- **Criptografia adaptada a caracteres não alfabéticos:** Diferente da cifra de César tradicional, que se restringe ao alfabeto, as versões cifraAF e cifraAF2 permitem a criptografia de todo o espectro de caracteres imprimíveis com base na tabela ASCII. Essa adaptação amplia a versatilidade da cifra, permitindo a criptografia de textos que contenham caracteres especiais e números, além de letras.

- **Segurança aprimorada com múltiplos deslocamentos:** A incorporação de múltiplos deslocamentos derivados da chave na cifraAF2 adiciona uma camada extra de complexidade. O uso de dois dígitos consecutivos como base para o deslocamento de cada caractere melhora a aleatoriedade e a imprevisibilidade, reforçando a segurança da criptografia.

Para comprovar essas vantagens, foi implementada uma função de ataque por força bruta para testar a resiliência da cifra modificada. O ataque de força bruta consiste em testar todas as possíveis combinações de chaves até encontrar aquela que descriptografa corretamente o texto cifrado. O Algoritmo 3 apresenta o pseudocódigo utilizado.

Algorithm 3 forcaBruta(textoCriptografado, textoAlvo)

```
inicio ← tempoAtual()
Imprimir("Tentando força bruta...")
i ← 0
texto ← string vazia
while texto ≠ textoAlvo do
    texto ← descriptografarCifraAF2(textoCriptografado, i)
    i ← i + 1
end while
Imprimir("Chave encontrada: ", i − 1)
fim ← tempoAtual()
Imprimir("Tempo de execução força bruta: ", fim − inicio)
```

O algoritmo de força bruta apresentado acima busca encontrar a chave correta para descriptografar um texto criptografado pela cifra AF2. Ele funciona iterando sobre diferentes chaves, aplicando a função de descriptografia em cada tentativa e comparando o resultado com o texto alvo. Quando o texto correto é obtido, o algoritmo imprime a chave encontrada e o tempo total de execução. Esse método demonstra a resistência adicional da cifra modificada, pois exige mais tentativas em um ataque de força bruta, comprovando a maior complexidade introduzida.

Nas primeiras execuções, com chaves de 1 a 3 dígitos, o tempo de execução é relativamente rápido, variando entre 0,01 e 0,41 segundos. No entanto, à medida que o tamanho da chave aumenta, o tempo de execução cresce significativamente. Por exemplo, para uma chave de 6 dígitos, o tempo de execução é de aproximadamente 62,25 segundos, enquanto para uma chave de 8 dígitos, o tempo atinge impressionantes 5.871,18 segundos (ou cerca de 98 minutos). Os resultados de tempo com cada chave podem ser observados na Figura 1 a seguir.

Para analisar o tempo de força bruta necessário para quebrar diferentes chaves criptográficas, foi realizada uma modelagem dos dados com uma função exponencial, dada a clara tendência de crescimento exponencial observada. A fórmula ajustada

$$y = 0.0098 \cdot e^{1.5465 \cdot x}$$

foi obtida a partir dos dados experimentais, onde x representa o comprimento da chave e y o tempo de força bruta em segundos. O parâmetro $a = 0.0098$ indica o valor

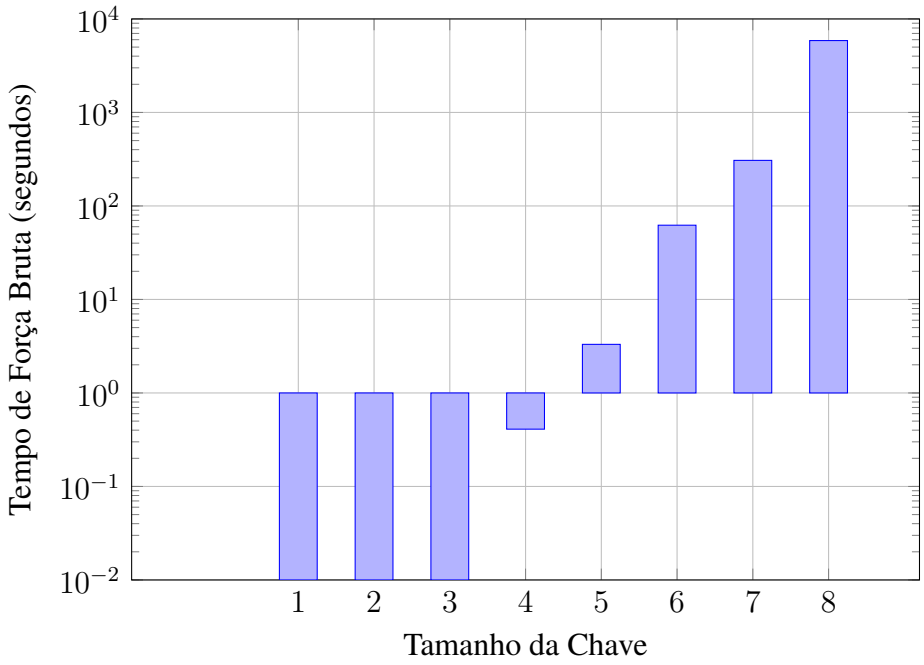

Figura 1. Tempo de força bruta para diferentes chaves em escala logarítmica

inicial do tempo de força bruta quando o comprimento da chave é pequeno, enquanto $b = 1.5465$ determina a taxa de crescimento exponencial do tempo necessário à medida que o comprimento da chave aumenta. Esse ajuste revela que o tempo de força bruta aumenta de forma exponencial com o comprimento da chave, o que está de acordo com a teoria de que a dificuldade de quebra de uma chave aumenta exponencialmente com seu tamanho. Essa função proporciona uma compreensão mais profunda da escalabilidade da complexidade na criptografia e pode ser útil para avaliar a eficácia e a segurança de diferentes métodos criptográficos em cenários práticos.

Isso demonstra que, embora a cifra de César tradicional e a cifraAF apresentem tempos de execução constantes e praticamente nulos, a implementação da cifraAF2 com múltiplos deslocamentos introduz uma complexidade significativa, especialmente quando sujeita a ataques de força bruta. A complexidade aumenta exponencialmente conforme o tamanho da chave se expande, comprovando a robustez da cifra modificada para proteger dados de forma mais eficaz contra esse tipo de ataque.

6. Ameaças à Validade

Ao avaliar a eficácia e a segurança das cifras modificadas, é crucial considerar possíveis ameaças à validade dos resultados obtidos. As duas principais ameaças identificadas são:

- **Melhores Computadores podem Apresentar Diferentes Resultados:** O desempenho dos algoritmos de força bruta pode variar significativamente com a capacidade de processamento do hardware utilizado. Computadores mais poderosos têm a capacidade de realizar operações de forma mais rápida e eficiente, o que pode afetar o tempo de execução observado. Assim, o tempo de força bruta registrado em experimentos realizados em máquinas de menor desempenho pode ser substancialmente maior do que o tempo obtido com hardware mais avançado. Isso

pode levar a uma subestimação da real complexidade e segurança da cifra modificada, uma vez que a eficiência do hardware pode influenciar os resultados e as conclusões obtidas.

- **Códigos de Força Bruta Mais Robustos Podem Apresentar Resultados Diferentes:** O desenvolvimento de algoritmos de força bruta mais eficientes e robustos pode alterar a maneira como os resultados são obtidos. Algoritmos mais sofisticados podem utilizar otimizações que reduzem o tempo necessário para encontrar a chave correta, como técnicas de paralelização ou heurísticas para reduzir o espaço de busca. Como resultado, o tempo de execução pode ser significativamente menor do que o estimado com algoritmos mais simples. Isso representa uma ameaça à validade, pois os resultados obtidos com métodos básicos podem não refletir a real dificuldade de quebra da cifra quando técnicas avançadas são empregadas.

7. Conclusões Finais

As modificações realizadas na Cifra de César demonstram que, com ajustes simples, é possível aumentar significativamente a complexidade e a segurança da criptografia clássica. Ao incorporar uma chave variável baseada em uma sequência numérica e utilizar o intervalo completo de caracteres imprimíveis da tabela ASCII, o novo método proposto oferece uma alternativa mais robusta e resistente a ataques por força bruta. As variantes implementadas, como a cifraAF e cifraAF2, mostram diferentes níveis de complexidade e a capacidade de manipular múltiplos dígitos da chave, tornando o processo de criptografia e descriptografia mais dinâmico e flexível.

A adição da função de força bruta também ilustra a vulnerabilidade potencial de qualquer cifra, destacando a importância de uma chave suficientemente longa e complexa para impedir a quebra por tentativa e erro. Embora o método proposto traga avanços em relação à Cifra de César original, o uso de técnicas de força bruta ainda pode ser explorado para testar os limites de segurança das variantes desenvolvidas.

Trabalhos futuros podem incluir a análise detalhada da resistência dessas cifras contra ataques criptográficos mais sofisticados, como análise de frequência e ataques de conhecimento prévio. Além disso, a aplicação das cifras em contextos de segurança modernos, como comunicação segura em redes, poderia ser investigada, avaliando sua eficácia e performance em sistemas com maiores volumes de dados. Outra vertente interessante seria a implementação de melhorias adicionais, como a combinação da cifra com técnicas mais avançadas de criptografia, como a cifra de bloco, para aumentar ainda mais a segurança e a integridade dos dados.

Referências

Gowda, S. N. (2016). Innovative enhancement of the caesar cipher algorithm for cryptography. In *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall)*, pages 1–4. IEEE.

Goyal, K. and Kinger, S. (2013). Modified caesar cipher for better security enhancement. *International Journal of Computer Applications*, 73(3):0975–8887.

Mishra, A. (2013). Enhancing security of caesar cipher using different methods. *International Journal of Research in Engineering and Technology*, 2(09):327–332.

Omolara, O., Oludare, A., and Abdulahi, S. (2014). Developing a modified hybrid caesar cipher and vigenere cipher for secure data communication. *Computer Engineering and Intelligent Systems*, 5(5):34–46.

Purnama, B. and Rohayani, A. H. (2015). A new modified caesar cipher cryptography method with legibleciphertext from a message to be encrypted. *Procedia Computer Science*, 59:195–204.

Terada, R. (2008). *Segurança de dados: criptografia em rede de computador*. Editora Blucher.