CONCILIUM, Vol. 23, Nº 5, 2023
DOI: 10.53660/CLM-1064-23C67
ISSN: 0010-5236

Protection of Personal Data in Health Using Symmetric Encryption: A Comparative Study Between Different Algorithms

Proteção de Dados Pessoais na Saúde Utilizando Criptografia Simétrica: Um Estudo Comparativo Entre Diferentes Algoritmos

Received: 2023-02-10 | Accepted: 2023-03-20 | Published: 2023-03-31

Inaê Karine Maziero Marchese
ORCID: https://orcid.org/0009-0004-2558-111X
Instituto Federal Catarinense, Concórdia, Brasil
E-mail: inaekarine02@gmail.com
Yasmin Maria Zerbielli
ORCID: https://orcid.org/0009-0005-0648-2906
Instituto Federal Catarinense, Concórdia, Brasil
E-mail: zyasminmaria@gmail.com
Maíra Amélia Mafessoni Herpich
ORCID: https://orcid.org/0009-0009-7128-9298
Instituto Federal Catarinense, Concórdia, Brasil
E-mail: mairaherpich@gmail.com
Walter Priesnitz Filho
ORCID: https://orcid.org/0000-0002-8999-4843
Universidade Federal de Santa Maria, Brasil
E-mail: walter@redes.ufsm.br
Heitor Scalco Neto
ORCID: https://orcid.org/0000-0002-5961-5013
Instituto Federal Catarinense, Concórdia, Brasil
E-mail: heitor.scalco@ifc.edu.br

ABSTRACT

The LGPD (Lei Geral de Proteção de Dados) aims to protect the right to privacy of personal data of Brazilians. A challenging impasse for several institutions, mainly in the health area, is the process of evolving their systems to the new requirements imposed by the LGPD. The imposition of items such as data encryption and its impact on the performance of these systems brings a discussion about how this additional protection should be provided. This article analyzes several symmetric encryption algorithms available in the PyCryptodome library, such as DES, 3DES, Blowfish, CAST-128 and RC2 to identify which of these would be most suitable for the type of attributes most commonly used in these environments. For the experiments, an application was developed in Python 3 that generates volumes of predefined data, compatible with data from personal attribute management systems in the health area. This data is also applied to the encryption algorithms, where time measurements and function calls are performed during the data encryption and decryption process. The results show the disparity in performance between the different encryption algorithms, as well as the analyzes using different data volumes.

Keywords: LGPD; Cryptography; Privacy; Health.

RESUMO

A LGPD (Lei Geral de Proteção de Dados) tem como principal objetivo proteger o direito de privacidade dos dados pessoais dos brasileiros. Um impasse desafiador para diversas instituições, principalmente da área da saúde, é o processo de adequação dos seus sistemas para os novos requisitos impostos pela LGPD.

A imposição de itens como a criptografia dos dados, e o seu impacto no desempenho desses sistemas, traz a discussão sobre como deve ser feita essa proteção adicional. Este artigo analisa vários algoritmos de criptografia simétrica disponíveis na biblioteca PyCryptodome, como o DES, 3DES, Blowfish, CAST-128 e RC2 para identificar qual destes seria mais adequado para o tipo de atributos mais comumente utilizado nestes ambientes. Para os experimentos, foi desenvolvida uma aplicação em Python 3 que gera volumes de dados pré-definidos, compatíveis com os dados de sistemas de gestão de atributos pessoais da área da saúde. Estes dados são aplicados igualmente aos algoritmos de criptografia, onde são realizadas as medições de tempo e chamadas de função durante o processo de encriptação e decriptação dos dados. Os resultados mostram a disparidade no desempenho entre os diversos algoritmos de criptografia, bem como as análises utilizando diferentes volumes de dados.

Palavras-chave: LGPD; Criptografia; Privacidade; Saúde.

INTRODUÇÃO

O aumento de casos de vazamento de dados sensíveis relatados nos últimos anos no Brasil (BISSO *et al.*, 2020) trouxe a necessidade de uma forma de regulamentação em busca de proteção para dados envolvendo saúde, finanças, família e padrões de comportamento. A LGPD (Lei Geral de Proteção de Dados), aprovada em 14 de agosto de 2018, tem como objetivo proteger direitos como: liberdade, privacidade e livre desenvolvimento da pessoa natural ou jurídica, pública ou privada, inclusive nos meios digitais. Seu órgão regulador é a ANPD (Autoridade Nacional de Proteção de Dados), que é responsável por cuidar da proteção de dados pessoais de entidades regidas pela lei e por sua fiscalização (PRESIDÊNCIA DA REPÚBLICA, 2018). São considerados dados pessoais as informações referentes a nome, data de nascimento, CPF, RG, carteira nacional de habilitação (CNH), carteira de trabalho, passaporte, título de eleitor, sexo, endereço, e-mail, telefone, origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou organização, informações sobre saúde ou orientação sexual e dados genéticos ou biométricos (DA SILVEIRA, 2022).

Com o advento da lei, um impasse desafiador para diversas instituições da área da saúde, tais como: hospitais, clínicas, consultórios e laboratórios, é o processo de adequação de sistemas legados para os novos requisitos impostos pela LGPD. A imposição de itens como a criptografia dos dados traz a discussão sobre como deve ser feita essa proteção adicional, levando em consideração o desempenho dos algoritmos de criptografia.

A principal contribuição deste trabalho é trazer uma análise sobre o desempenho dos principais algoritmos de criptografia de chave simétrica com foco na proteção de dados pessoais de pacientes das instituições de saúde, tais como: informações sobre consultas, doenças pré-existentes e medicamentos de uso contínuo. Para a obtenção dos dados e realização da análise proposta, foi desenvolvida uma aplicação em Python 3, capaz de gerar um volume de dados pré-definido, compatível com os tamanhos dos atributos mencionados anteriormente. Estes atributos são transmitidos em sistemas de gestão da área da saúde, como o OpenMRS (*https://openmrs.org/*),

utilizando o JSON (*JavaScript Object Notation*). Após a geração dos dados, foram realizados testes utilizando os algoritmos de criptografia simétrica disponibilizados pela biblioteca PyCryptodome (*https://www.pycryptodome.org/*), tais como: DES, 3DES, Blowfish, CAST-128, AES, RC2.

O artigo está organizado da seguinte forma: Primeiramente são apresentados os trabalhos relacionados com um breve resumo de cada proposta. Em seguida, é apresentado um estudo sobre o *status* da adequação das empresas brasileiras à LGPD. Após, apresentam-se os conceitos fundamentais de algoritmos de chave simétrica, seguido dos tipos de algoritmos utilizados. A partir dos conceitos expostos, são apresentados os materiais e métodos utilizados para realizar as comparações de desempenho entre os algoritmos. Com base nos experimentos realizados, são apresentados os resultados obtidos e suas devidas discussões. Por fim, são apresentadas as conclusões e propostas de trabalhos futuros.

TRABALHOS RELACIONADOS

Com a imposição da LGPD brasileira e da GDPR *(General Data Protection Regulation*) na Europa, o contexto da segurança de dados e criptografia ganhou força no meio científico. Desta forma, foram encontradas algumas propostas de trabalhos correlatos, que serviram como base para o desenvolvimento deste artigo e serão apresentados no decorrer desta seção.

Em Vargas *et al*. (2015) são comparados três algoritmos de criptografia simétrica, sendo eles o DES, 3DES e AES. O desenvolvimento do artigo apresenta uma análise sobre as diferenças existentes dentre os algoritmos, analisando tamanho de bloco, comprimento de chave, tipo de cifragem, resistência à criptoanálise, entre outros fatores. Os autores não apresentam a metodologia que foi utilizada para comparar os algoritmos descritos, mas concluem apontando o AES como o melhor dentre eles.

Em Semwal *et al*. (2017) os autores abordam uma comparação de diferentes algoritmos de criptografia, tendo por objetivo ressaltar a sua importância nos sistemas atuais de dados. Alguns dos algoritmos que o desempenho é comparado são: DES, 3DES, IDEA, CAST128, AES, Blowfish, RSA, ABE e ECC. Como resultados o trabalho apresenta os pontos positivos dos principais algoritmos em teste, mostrando que o Blowfish é excelente em termos de consumo de memória e tempo de criptografia e descriptografia, enquanto o RSA é o seu oposto. Já o AES é tido como o com melhor desempenho em relação a todos, principalmente por sua prioridade ser a mensagem.

Em Pikulík (2019) os autores apresentam dois métodos de proteção de dados compatíveis a *General Data Protection Regulation* (GDPR), sendo estes a pseudoanonimização e as técnicas de criptografia. O objetivo do trabalho é identificar qual dos métodos é mais adequado para que

a GDPR seja cumprida. Os autores identificaram que ambos os métodos são possíveis opções, pois seu uso é encorajado pela GDPR, porém é necessário que esse uso seja recorrente e generalizado para que melhores resultados sejam vistos.

Em Sousa *et al.* (2020) são apresentadas comparações entre as técnicas para proteção de bases de dados, como: anonimização, privacidade diferencial e tipos de criptografia distintos. O trabalho buscou prover insumos para auxiliar no cumprimento da legislação vigente por parte de sistemas que devem se adequar e aumentar gradativamente a segurança em seus bancos de dados. Como resultado, percebeu-se que a adequação de sistemas à LGPD não é uma tarefa trivial. Cada sistema tem as suas particularidades de desempenho e implementação, sendo assim, validou-se as técnicas de criptografia e anonimização como possíveis candidatos para adequação.

Já em Nurgaliyev *et al.* (2021) é apresentado um estudo sobre algoritmos de criptografia de chave simétrica, abordando seus pontos fracos e fortes. Entre os algoritmos estudados estão: *Data Encryption Standard* (DES), *Triple Data Encryption algorithm* (3DES), *Advanced Encryption Standard* (AES), *Blowfish*, *MARS Algorithm*. Como resultado, os autores destacam o algoritmo AES com o maior número de pontos positivos.

Esta proposta traz como diferencial o foco na Lei Geral de Proteção de Dados (LGPD) aplicada a dados e sistemas relacionados à saúde que envolvam o tratamento de dados pessoais de pacientes. A pesquisa foi projetada considerando a troca e armazenamento de dados compatíveis com sistemas de gestão de instituições de saúde, como o OpenMRS (VERMA *et al.* 2021). O trabalho traz uma análise de algoritmos de chave simétrica com cifra de bloco, pois eles têm melhor desempenho se comparados aos algoritmos de chave assimétrica (LOGUNLEKO *et al.* 2020) e são recomendados para utilização no armazenamento e anonimização de dados.

ADEQUAÇÃO DE INSTITUIÇÕES DE SAÚDE À LEI GERAL DE PROTEÇÃO DE DADOS

O principal desafio para adequação à LGPD em ambientes que armazenam dados pessoais críticos, como os da saúde, é a organização da equipe multidisciplinar para reprogramar processos. Um estudo realizado pelo CGI (Comitê Gestor de Internet no Brasil) em 2022 apresenta que o desafio mais citado entre os gestores foi o desenvolvimento de uma política de privacidade que informe como os dados pessoais são tratados pela instituição (32%). Em segundo lugar, nos desafios mais citados, 30% das empresas informaram que realizaram testes de segurança contra vazamentos de dados. O que evidencia uma preocupação em ter seus processos de tratamento de dados pessoais melhor definidos. Apenas 17% das instituições nomearam um encarregado de dados *Data Protection Officer*. Já a criação de um plano de adequação à LGPD,

que pode favorecer uma operação mais segura e em conformidade com a lei, foi citada por apenas 24% das instituições (CGI, 2022).

Na era da informação, o armazenamento de todo o tipo de dado, mesmo sem uma justificativa plausível, trouxe diversos problemas à sociedade. Exemplos disso são os casos de utilização de dados pessoais para a realização de golpes financeiros, extorsão, telemarketing, comércio de dados, roubo de credenciais, entre outros. A maioria destes problemas não ocorre por má índole da entidade mantenedora dos dados, mas pelo vazamento destes dados oriundos de práticas mal-intencionadas (BISSO *et al.*, 2020).

Para contornar este problema, uma das ações propostas pela ANPD (Autoridade Nacional de Proteção de Dados) e o CERT.br é a utilização de criptografia de bancos de dados e arquivos que contenham dados pessoais sensíveis, incluindo àqueles que contenham informações sobre a saúde dos pacientes. Porém, existe uma relação entre a quantidade de proteção adicionada a um sistema de informação e o impacto aceitável em seu desempenho. Teoricamente, quanto maior a quantidade de recursos de segurança adicionados ao tratamento dos dados, menor é o seu desempenho do sistema. Desta forma, um estudo sobre o desempenho dos algoritmos de criptografia que possam ser utilizados para estes fins torna-se relevante (CGI, 2022), como é o caso dos algoritmos de chave simétrica, que serão apresentados nas próximas seções.

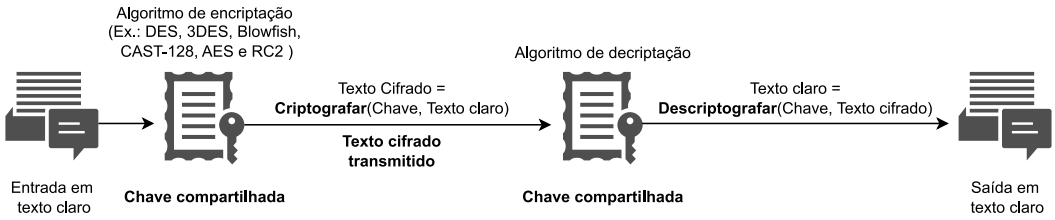
ALGORITMOS DE CHAVE SIMÉTRICA

Algoritmos de criptografia de chave simétrica são aqueles que utilizam da mesma chave tanto para criptografar quanto para descriptografar um dado de acordo com Logunleko *et al.* (2020). Esse tipo de criptografia necessita de uma autorização de acesso aos dados criptografados, ou seja, apenas entidades autorizadas a acessar a chave de criptografia compartilhada podem descriptografar o texto cifrado. Além disso, por utilizar uma única chave para ambos os processos, sua aplicação torna-se mais trivial. Sendo assim, em tese, a agilidade nos processos com este modelo de criptografia é maior em relação ao modelo assimétrico. No modelo assimétrico ocorre a utilização de duas chaves ao longo de todo o processo, uma pública e outra privada. Como exemplos de algoritmos de chave simétrica, pode-se citar: DES, 3DES, Blowfish, CAST-128, AES e RC2 (AL-SHABI, 2019; LOGUNLEKO *et al.*, 2020), os quais serão abordados nas próximas seções.

De acordo com a Figura 1, um esquema de encriptação possui no mínimo cinco componentes: (i) texto claro, (ii) algoritmo de encriptação, (iii) chaves, (iv) texto criptografado e (v) algoritmo de decriptação. O texto claro representa a mensagem a ser criptografada. Em seguida, o algoritmo realiza as transformações no texto claro, onde a chave é uma entrada para o algoritmo, produzindo uma saída de texto criptografada. Por fim, para realizar o processo reverso,

insere-se a mensagem criptografada e a chave utilizada na criptografia, produzindo uma saída de texto descriptografada (STALLINGS, 2006).

Figura 1 – Modelo de encriptação e decriptação com chave simétrica.

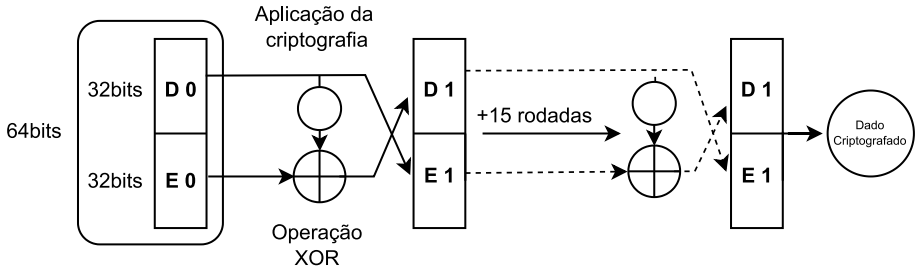


Fonte: Adaptado de Stallings (2006)

DATA ENCRYPTION STANDARD ALGORITHM (DES)

O DES é um algoritmo de criptografia que funciona por meio de cifras em bloco, o tamanho de cada bloco é de 64 bits, sendo 8 bits para verificar a paridade e depois serem descartados, enquanto os outros 56 bits são o tamanho efetivo da chave. Nele acontecem 16 estágios de processamentos idênticos que são chamados de rodadas (LOGUNLEKO *et al.*, 2020). Este processo é baseado no algoritmo de Feistel, conhecido como Rede Feistel - *Feistel Network* (SHEN *et al*., 2020).

Figura 2 – Funcionamento do algoritmo DES



Fonte: Adaptado de Stallings (2006)

Na prática, como pode-se observar na Figura 2, um bloco de 64 bits é dividido em dois outros blocos de 32 bits cada, esses denominados blocos de direita e esquerda. Na primeira rodada, o bloco da direita passa para a esquerda sem sofrer alterações, já o da esquerda sofre uma operação XOR[1] que se dá a partir do bloco direito, sofrendo a ação criptográfica do próprio DES, assim o código XOR é aplicado e o bloco passa para direita. Desta mesma forma, acontecerão todas as 16 rodadas, sempre invertendo os dois blocos até que se obtenha o texto cifrado no final (ADHIE *et al*., 2018).

[1] É uma porta lógica de duas entradas que produz em sua saída o nível lógico 1 quando suas entradas tiverem valores diferentes entre si, e o nível lógico 0 zero quando as entradas forem iguais.

Este algoritmo não é considerado um meio totalmente seguro para criptografia, pois pode ser quebrado em poucas horas utilizando ataques de força bruta - *brute force*. Todavia, há um modo de utilização mais seguro chamado Triple DES, que será apresentado na próxima seção (SEMWAL *et al.* 2017).

TRIPLE DATA ENCRYPTION ALGORITHM (3DES)

Sendo uma melhoria do algoritmo DES, o Triple DES também utiliza cifra em bloco, com o diferencial de possuir 3 chaves (K1, K2 e K3) de 56 bits. Se K1 ≠ K2, K2 ≠ K3 e K3 ≠ K1 há uma chave de comprimento total de 168 bits. Porém, se K1 for igual a K3, resulta em uma chave de comprimento total de 112 bits. A chave K1 irá criptografar, K2 irá descriptografar e K3 irá criptografar novamente, de acordo com Al-shabi (2019) e Barker *et al*. (2017). Em operação, ele irá implementar o algoritmo DES, 3 vezes em cada bloco de dados utilizando, uma chave diferente para cada um dos blocos (BARKER *et al*., 2017).

BLOWFISH

É um algoritmo de criptografia de bloco, com o tamanho de 64 bits, sendo que o tamanho de sua chave pode variar de 38 a 448 bits (passos de 8 bits). Sua estrutura é baseada na rede Feistel[2] constituída por 16 rodadas. Esse algoritmo é composto por duas fases: a expansão de chaves e a criptografia de dados (AL-SHABI, 2019; ELGELDAWI *et al.*, 2019). A expansão de chaves transforma uma chave de até 448 bits em uma matriz de subchaves denominada P-array, contendo 18 subchaves de 32 bits e 4 caixas de substituição (S-Box), com 256 entradas de também 32 bits cada segundo. Após a expansão de chaves, o texto plano passará pelo processo de cifragem. Primeiramente, o texto com 64 bits irá se dividir em dois blocos com 32 bits cada, esses 32 bits serão divididos em 4 bytes e esses 4 valores serão utilizados para pesquisa de tabela em sua S-Box[3] correspondente. Cada S-Box retornará uma saída de 32 bits, essas 4 saídas obtidas irão ser unidas para formar o texto cifrado (ELGELDAWI *et al.*, 2019).

CAST-128

O Cast-128 é um bloco de cifras pertencentes à família DES, que por sua vez utiliza substituições e permutações em cálculos de chaves e processos de encriptação e decriptação. O comprimento máximo de chave permitido no CAST é de 128 bits, ou 16 caracteres, ele também

[2] A rede de Feistel é uma estrutura de cifra de blocos, utilizada em importantes padrões de segurança, como o Data Encryption Standard (DES) e Blowfish (SHEN *et al*. 2020).

[3] Uma S-Box é um componente básico dos algoritmos de chave simétrica que realiza a substituição.

permite que os tamanhos de chave variem de 40 bits a 128 bits com a adição de 8 bits. Enquanto o comprimento do texto simples que pode ser encriptado e decriptado é de 64 bits (8 caracteres) e suporta todos os tipos de texto básico (L. ENAS *et al*. 2019; ISKANDAR *et al.* 2019).

ADVANCED ENCRYPTION STANDARD (AES)

O AES é um algoritmo de cifra em bloco que suporta uma chave de 128, 192 ou 256 bits. O número de rodadas do sistema varia de acordo com o tamanho da chave, esse número pode ser de 10, 12 ou 14 rodadas, para chaves de 128, 192 e 256 bits, respectivamente (ELGELDAWI *et al.*, 2019). A chave fornecida como entrada é expandida para um vetor de 44 palavras de 32 bits. Quatro palavras distintas (128 bits) servem como uma chave para cada rodada. A estrutura do algoritmo utiliza quatro estágios: *AddRoundKey*, *SubBytes*, *ShiftRows* e *MixColumns*. A cifra começa utilizando o estágio *AddRoundKey* que irá criptografar, por meio de um XOR, bit a bit do bloco atual com uma parte da chave expandida. Esse é o estágio que começa e finaliza o processo, pois é o único que faz a utilização da chave, ou seja, o processo só é reversível com o conhecimento da chave, trazendo mais segurança para o dado criptografado. Dentro desse estágio acontecem os outros três, o *SubBytes* que utiliza um "caixa-S" para substituir byte a byte do bloco, o *ShiftRows* que vai fazer uma permutação e o *MixColumns* um embaralhamento de colunas. Adicionalmente com esse processo pode-se acrescentar um *Salt*, que consiste em um conjunto de caracteres pré-definidos que serão implementados junto com a chave para passar pelo processo de encriptação (STALLINGS, 2006).

RC2

O RC2 é uma cifra de bloco que trabalha com chave simétrica, utilizando uma chave fornecida pelo usuário, que pode ter o tamanho de um byte até 128 bytes (SELVANAYAGAM *et al*., 2018). Em operação, o algoritmo RC2 é dividido em duas partes. Primeiro acontece a expansão de chaves, na qual utiliza a chave fornecida pelo usuário e um parâmetro que especifica o comprimento de chave efetivo da criptografia. Em seguida, utilizando um vetor K de 64 chaves de 16 bits, é criptografado um bloco simples de 64 bits (KNUDSEN et al., 1998).

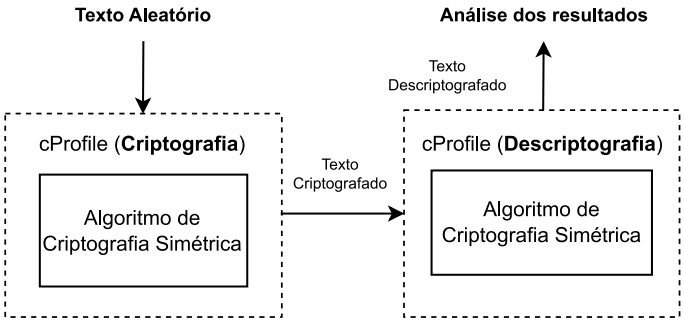
Essa criptografia ocorre por dois meios: (i) embaralhamento e (ii) repartição. No processo de criptografia, uma matriz contém quatro palavras de 16 bits r0, r1, r2, r3. Essas palavras são modificadas e retornadas no mesmo vetor. Para a inicialização do processo de criptografia, inicializa-se as palavras r0, r1, r2, r3, para obter o texto claro de 64 bits. Após isso, é necessário expandir a chave para que comece a realização das rodadas. Primeiramente realiza-se 5 rodadas de embaralhamento (i), uma de repartição (ii), mais 6 rodadas de embaralhamento (i), uma rodada

de repartição (ii) novamente e, por fim, 6 rodadas de embaralhamento (i), finalizando o processo de criptografia (KNUDSEN et al., 1998).

MATERIAIS E MÉTODOS

Para a realização dos experimentos de comparação de desempenho dos algoritmos de criptografia, utilizou-se um computador equipado com processador Intel® Core i5 de 7ª geração, com velocidade de *clock* de 3.00GHz (modelo 7400), 8GB de memória RAM DDR3 e Windows 10 Pro. Para ter uma maior confiabilidade nos resultados obtidos e minimizar o efeito de possíveis variações de tempo, os testes foram repetidos por 10 vezes, utilizando as mesmas versões de sistema, e com o computador desconectado da rede - *offline*.

Figura 3 – Organização dos *Scripts* de Teste



Fonte: do Autor

A metodologia deste trabalho consiste no desenvolvimento de uma aplicação em Python 3 que gera textos pseudoaleatórios, compatíveis com os dados trocados em sistemas da área da saúde, em volumes de dados pré-definidos (Ex.: 10KB, 20KB, 40KB [...]) e realiza a encriptação com diversos algoritmos de chave simétrica. A Figura 3 apresenta, em alto nível, o passo a passo para a obtenção dos resultados. Inicia-se com a geração do texto aleatório com o volume de dados especificado, em seguida, aplica-se o algoritmo de criptografia e descriptografia. A aplicação desenvolvida utiliza a biblioteca cProfile (*https://docs.python.org/3/library/profile.html\#module-cProfile*) para obtenção dos tempos de processamento e quantidade de chamadas de sistema durante o processo de criptografia e descriptografia dos dados. Cabe ressaltar que a biblioteca cProfile monitora apenas o tempo de encriptação e decriptação, desconsiderando as chamadas de função e outros elementos relativos à geração de textos e pré-processamento dos dados, representados pelos retângulos pontilhados na Figura 3.

Também utilizou-se a biblioteca PyCryptoDome, versão 3.16.0, que é uma coleção de funções de *hash* seguras (como SHA-256) e vários algoritmos de criptografia (AES, DES, etc.). Esta biblioteca tem suporte a todos os algoritmos de criptografia utilizados neste trabalho. Para melhor visualização dos resultados dos tempos de criptografia e descriptografia de algoritmos de

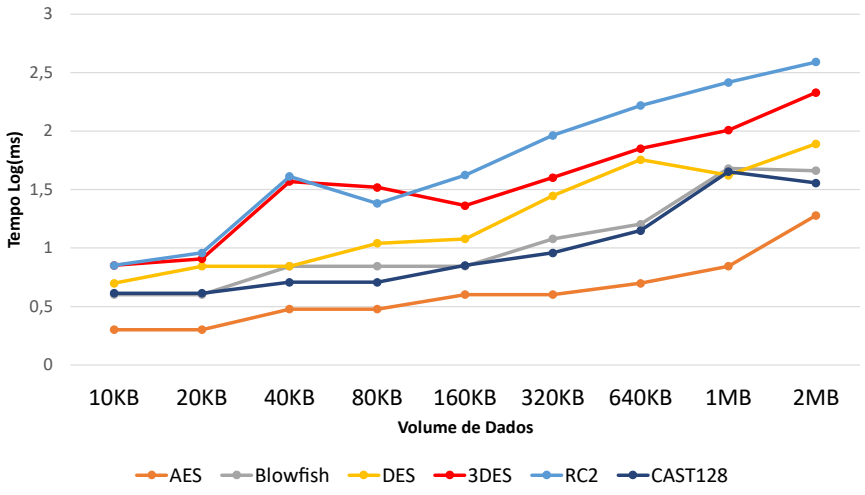
chave simétrica, medidos em milissegundos (ms), aplicou-se uma função logarítmica aos resultados.

A escolha do volume de dados a ser criptografado deu-se a partir do tamanho médio de arquivos que, comumente, transportam atributos de usuários i.e.: JSON e/ou SAML, sendo eles de sistemas públicos (SUS, Dataprev, Tribunais) ou privados (Hospitais particulares, Clínicas, Bancos). Ainda, os tamanhos selecionados são compatíveis com conjuntos de campos de informações pessoais que, de acordo com a LGPD, devem ser criptografados, tais como: nome, CPF, RG, endereço, histórico de consultas, alergias e imagem de perfil. Após realizados experimentos de criptografia e descriptografia dos algoritmos selecionados, analisou-se as saídas da biblioteca cProfile para ambos os processos e obteve-se os resultados que serão discutidos na próxima seção.

RESULTADOS E DISCUSSÕES

Iniciando a análise dos algoritmos apresentados, pode-se observar na Figura 4 que o algoritmo AES mostrou-se o mais eficiente, em relação ao tempo, entre os algoritmos testados no processo de criptografia. Em relação ao algoritmo RC2, percebe-se uma crescente degradação no desempenho ao criptografar um volume de dados maior que 80KB, tendendo a piorar conforme o crescimento do volume dos dados. Os algoritmos DES, Blowfish e Cast-128 possuíram desempenhos similares, pois suas maiores degradações também ocorreram a partir de 80KB, o que indica que o aumento no volume de dados a ser criptografado influencia no aumento de período de tempo para a criptografia. O algoritmo 3DES mostrou um aumento gradativo em seu tempo conforme o aumento do volume dos dados, sendo o que mais se aproximou dos resultados obtidos pelo RC2.

Figura 4 – Tempo de Criptografia de Algoritmos de Chave Simétrica em Função do Volume de Dados

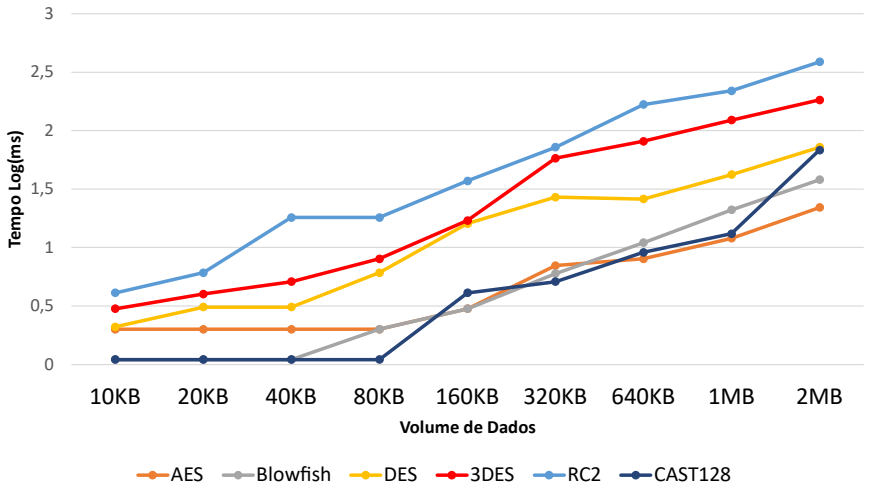


Fonte: do Autor

Em relação ao desempenho no processo de descriptografia de algoritmos de chave simétrica, observa-se na Figura 5 que, assim como no processo de criptografia, o algoritmo RC2 teve o pior desempenho em volume de dados maiores que 10KB. O algoritmo 3DES obteve um aumento gradativo desde os 10KB, conforme o aumento do volume de dados, sendo o que mais se aproximou do RC2. Os algoritmos DES e AES apresentaram um desempenho mediano em relação aos outros algoritmos, com volume de dados até 80KB. O CAST-128 apresenta uma elevada degradação a partir de 2MB, atingindo o mesmo tempo do DES, que inicia sua degradação em 160KB. Com volume de dados até 80KB, percebe-se que o CAST-128 e o Blowfish obtiveram melhor desempenho.

Ainda na Figura 5 o destaque vai para o algoritmo AES, que apesar do aumento no volume de dados, o tempo de decriptação se manteve com bons resultados, apresentando o melhor desempenho de todos os outros algoritmos testados. Os algoritmos Blowfish, DES e Cast-128 obtiveram os resultados mais aproximados ao AES. Porém DES e Cast-128 em 2MB possuem aumento significativo no tempo em comparação ao AES. O Cast-128 apresenta-se como uma boa opção em pequenos volumes de dados de até 1MB. Ainda o algoritmo AES, que não apresentou aumento significativo em consequência do aumento do volume de dados, também é uma boa opção para processos que demandam eficiência na descriptografia.

Figura 5 – Tempo de Descriptografia de Algoritmos de Chave Simétrica em Função do Volume de Dados



Fonte: do Autor

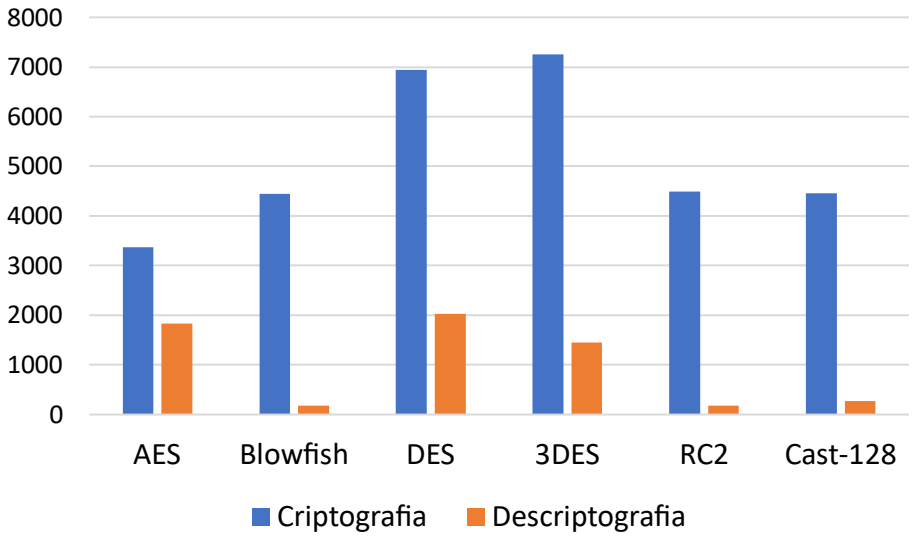
Em relação as análises das quantidades de chamadas de função, observa-se na Figura 6 que os valores, tanto para a criptografia quanto para a descriptografia, possuem evidentes variações. Os algoritmos DES e 3DES são os que apresentam o maior número de chamadas para a criptografia, chegando a 7.000 chamadas. Entretanto, na descriptografia o DES apresenta um resultado maior do que o algoritmo AES que atinge aproximadamente 1.800 chamadas, já o 3DES

é ultrapassado por ambos. O algoritmo AES apresenta um número baixo de chamadas de função para criptografia em comparação com os outros algoritmos, sendo o algoritmo com o menor número de chamadas.

Em relação aos algoritmos RC2, Blowfish e Cast-128, é possível perceber uma nítida variação na quantidade de chamadas de função entre os processos de criptografia e de descriptografia em cada algoritmo. Os algoritmos apresentam aproximadamente 4.500 chamadas de função para a criptografia, porém há uma evidente diferença para a descriptografia, que apresenta os menores números registrados entre todos os outros algoritmos testados.

Comparando as Figuras 4 e 5, o algoritmo AES se mostrou o mais eficiente entre os algoritmos testados. Apesar de não apresentar o melhor desempenho nas chamadas de função Figura 6, obteve resultados semelhantes aos algoritmos em destaque de ambos os processos nestes aspectos.

Figura 6 – Quantidade de Chamadas de Função para Criptografia e Descriptografia



Fonte: do Autor

O algoritmo que mais se assemelha aos resultados do AES é o Blowfish, que obteve um bom tempo no processo de encriptação e teve destaque no tempo de decriptação. Ainda, o algoritmo RC2 obteve resultados bons no quesito número de chamadas de função, porém foi o algoritmo que apresentou pior desempenho no tempo de criptografia e descriptografia. Pode-se concluir que, em alguns casos, o número de chamadas de função não possui grande influência nos tempos de processamento, pois o RC2 obteve um bom resultado nas chamadas de função da descriptografia, entretanto foi o algoritmo com os piores tempos em relação a criptografia e descriptografia.

CONCLUSÕES E TRABALHOS FUTUROS

O avanço da tecnologia no decorrer dos anos trouxe inúmeros benefícios para toda a população. No entanto, com esse crescimento tornaram-se constantes os casos de vazamento de dados sensíveis em todos os ramos em que se utilizam sistemas e redes computacionais. Tendo conhecimento de tal situação, é de grande importância manter a confidencialidade dos dados em sistemas computacionais, buscando recursos para garantir que apenas pessoas devidamente autorizadas tenham acesso a esses dados.

Este artigo abordou o uso de algoritmos de chave simétrica para criptografia e descriptografia usando textos pseudoaleatórios, compatíveis com dados pessoais utilizados em sistemas de gestão da saúde, em volumes de dados pré-definidos, permitindo a análise e comparação entre algoritmos distintos. A análise e comparação entre os algoritmos de criptografia simétrica mostrou resultados significativos, visíveis para os gestores de sistemas da área da saúde definirem qual algoritmo utilizar para a adequação de seus sistemas à LGPD. O algoritmo AES apresentou o melhor desempenho dentre todos os experimentados para a criptografia e descriptografia dos dados, o que acaba por mostrar o AES como uma excelente opção de algoritmo de chave simétrica.

Como trabalhos futuros, sugere-se desenvolver uma plataforma do tipo *compliance* para adequação de sistemas de gestão de saúde à LGPD, utilizando criptografia simétrica para proteção dos dados. Ainda, estabelecer a relação entre os níveis de desempenho com os níveis de segurança que cada algoritmo de criptografia simétrica oferece.

REFERÊNCIAS

ADHIE, Roy Pramono et al. Implementation cryptography data encryption standard (DES) and triple data encryption standard (3DES) method in communication system based near field communication (NFC). In: **Journal of Physics: Conference Series**. IOP Publishing, 2018. p. 012009. DOI: https://doi.org/10.1088/1742-6596/954/1/012009

AL-SHABI, M. A. A survey on symmetric and asymmetric cryptography algorithms in information security. **International Journal of Scientific and Research Publications (IJSRP)**, v. 9, n. 3, p. 576-589, 2019. DOI: http://dx.doi.org/10.29322/IJSRP.9.03.2019.p8779

BARKER, Elaine; MOUHA, Nicky. **Recommendation for the triple data encryption algorithm (TDEA) block cipher**. National Institute of Standards and Technology, 2017. DOI: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf

BISSO, Rodrigo et al. Vazamentos de Dados: Histórico, Impacto Socioeconômico e as Novas Leis de Proteção de Dados. **Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação**, v. 3, n. 1, 2020. DOI: https://zenodo.org/record/3833275

CGI - COMITÊ GESTOR DA INTERNET NO BRASIL. PRIVACIDADE E PROTEÇÃO DE DADOS PESSOAIS. *In*: PRIVACIDADE e proteção de dados pessoais 2021: perspectivas de indivíduos, empresas e organizações públicas no Brasil. [*S. l.*: *s. n.*], 2022. p. 33-66. Disponível em: https://cetic.br/pt/publicacao/privacidade-e-protecao-de-dados-2021/

DA SILVEIRA, Kamilla Dória. Segurança em Banco de Dados para Adequação a LGPD. In: **Anais do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**. SBC, 2022. p. 278-287. DOI: https://doi.org/10.5753/sbseg.2022.223953

ELGELDAWI, Enas; MAHROUS, Maha; SAYED, Awny. A comparative analysis of symmetric algorithms in cloud computing: a survey. **International Journal of Computer Applications**, v. 975, p. 8887, 2019. DOI: https://doi.org/10.5120/ijca2019918726

ISKANDAR, Akbar et al. Utility Software Design to Comprehend The Cryptography Cast-128 Method. In: **Journal of Physics: Conference Series**. IOP Publishing, 2019. p. 012049. DOI: https://doi.org/10.1088/1742-6596/1364/1/012049

KNUDSEN, Lars R. et al. On the design and security of RC2. In: **Fast Software Encryption: 5th International Workshop, FSE'98 Paris, France, March 23–25, 1998 Proceedings 5**. Springer Berlin Heidelberg, 1998. p. 206-221. DOI: https://doi.org/10.1007/3-540-69710-1_14

L. Enas Tariq. Image Encryption and decryption using CAST-128 with proposed adaptive key. **مجلة المستنصرية للعلوم والتربية**, v. 20, n. 5, p. 89-100, 2019. Disponível em: https://edumag.uomustansiriyah.edu.iq/index.php/mjse/article/view/675/539

LOGUNLEKO, K. B.; ADENIJI, O. D.; LOGUNLEKO, A. M. A comparative study of symmetric cryptography mechanism on DES AES and EB64 for information security. **Int. J. Sci. Res. in Computer Science and Engineering**, v. 8, n. 1, 2020. Disponível em: https://www.isroset.org/journal/IJSRCSE/full_paper_view.php?paper_id=1690

NURGALIYEV, Alibek; WANG, Hua. Comparative study of symmetric cryptographic algorithms. In: **2021 International Conference on Networking and Network Applications (NaNA)**. IEEE, 2021. p. 107-112. DOI: https://doi.org/10.1109/NaNA53684.2021.00026

PIKULÍK, Tomáš. GDPR COMPLIANT METHODS OF DATA PROTECTION. **Business & Management**, 6th SWS International Scientific Conference on Social Sciences ISCSS 2019, p. 1-10, 20 ago. 2019.

PRESIDÊNCIA DA REPÚBLICA SECRETARIA-GERAL SUBCHEFIA PARA ASSUNTOS JURÍDICOS. **Lei Geral de Proteção de Dados nº 13.709, de 14 de agosto de 2018**. Dispõe sobre o tratamento de dados pessoais, [...] e o livre desenvolvimento da personalidade da pessoa natural. [*S. l.*], 14 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm

SELVANAYAGAM, Joseph et al. Secure file storage on cloud using cryptography. **Int. Res. J. Eng. Technol**, v. 5, n. 3, p. 2044, 2018. Disponível em: https://www.irjet.net/archives/V5/i3/IRJET-V5I3475.pdf

SEMWAL, Pradeep; SHARMA, Mahesh Kumar. Comparative study of different cryptographic algorithms for data security in cloud computing. In: **2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)**. IEEE, 2017. p. 1-7. DOI: https://doi.org/10.1109/ICACCAF.2017.8344738

SHEN, Yaobin; GUO, Chun; WANG, Lei. Improved security bounds for generalized Feistel networks. **IACR Transactions on Symmetric Cryptology**, p. 425-457, 2020. DOI: https://doi.org/10.13154/tosc.v2020.i1.425-457

SOUSA, Thiago R. et al. LGPD: Levantamento de Técnicas Criptográficas e de Anonimização para Proteção de Bases de Dados. In: **Anais do XX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**. SBC, 2020. p. 55-68. DOI: https://doi.org/10.5753/sbseg.2020.19227

STALLINGS, William. Criptografia e segurança de redes Princípios e práticas. *In*: CRIPTOGRAFIA e segurança de redes Princípios e práticas. [*S. l.*: *s. n.*], 2006.

VARGAS, Yuri Tatiana Medina; MNEDEZ, Haider Andrés Miranda. Comparación de algoritmos basados en la criptografía simétrica DES, AES y 3DES. **Mundo Fesc**, v. 5, n. 9, p. 14-21, 2015. Disponível em: https://dialnet.unirioja.es/servlet/articulo?codigo=5286657

VERMA, Neha et al. OpenMRS as a global good: Impact, opportunities, challenges, and lessons learned from fifteen years of implementation. **International Journal of Medical Informatics**, v. 149, p. 104405, 2021. DOI: https://doi.org/10.1016/j.ijmedinf.2021.104405