Algoritmos DES e AES: Comparação dos processos de geração de chaves e cifração

José Gladistone da Rocha
Ministério da Defesa
Secretaria de Planejamento Baseado em Capacidade
Brasília, DF, Brasil
jgladistone@gmail.com

Carlo Kleber da Silva Rodrigues
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC (UFABC)
Santo André, São Paulo, Brasil
carlo.kleber@ufabc.edu.br

ABSTRACT
Cryptographic algorithms are the basis of information security, whether in symmetric or asymmetric systems. The objective of this article is to compare the symmetric algorithms DES and its replacement AES, regarding the security and speed of the encryption and key generation processes. The aim of this work is to indicate which cryptographic algorithm is the safest and fastest based on the indicated criteria. The result obtained is that some points led to AES being more secure, particularly because its cryptographic process has a very complex architecture and uses 128, 192 and 256-bit keys, which are considered strong today.

CCS Concepts
• **Security and Privacy→Cryptography**

Keywords
Cryptography; Symmetric; Algorithm; Security.

RESUMO
Os algoritmos criptográficos são a base da segurança da informação, seja nos sistemas simétricos como nos assimétricos. O objetivo desse artigo é fazer uma comparação entre os algoritmos simétricos DES e seu substituto AES, quanto à segurança e rapidez dos processos de cifração e geração de chaves. Pretende-se com esse trabalho apontar qual algoritmo criptográfico é o mais seguro e o mais rápido com base nos critérios indicados. O resultado obtido é que alguns pontos levaram a ter o AES mais seguro particularmente pelo seu processo criptográfico ter uma arquitetura bastante complexa e utilizar-se de chaves de 128, 192 e 256 bits sendo consideradas fortes nos dias atuais.

Palavras-chave
Criptografia; Simétrica; Algoritmo; Segurança.

1. INTRODUÇÃO
A criptografia na atualidade é essencial para a manutenção da Segurança da Informação [4]. Existem dois tipos básicos de algoritmos, os simétricos, objeto desse estudo, e os assimétricos – ambos se complementam para imprimir um sistema criptográfico seguro [17] [19] [8].

O sistema criptográfico simétrico é aquele no qual o processo de cifragem e decifragem utiliza a mesma chave secreta para realizar essas operações. Já o sistema assimétrico utiliza-se de um par de chaves, chamadas de chave pública e chave privada para realizar essas mesmas operações [17] [19] [8].

O sistema simétrico possui suas vantagens e desvantagens em relação ao assimétrico. Como exemplo, uma desvantagem que se pode apontar é a dificuldade de gerenciamento e distribuição de chaves e como uma vantagem é sua velocidade em executar a cifração e decifração. Dessa feita, os algoritmos simétricos são bastante utilizados para criptografar grandes volumes de dados [20] [1]. Existem vários deles disponíveis como o DES, 3DES, AES, *Blowfish*, *Twofish*, RC4, dentre outros.

Algoritmos criptográficos normalmente são de padrões abertos e públicos para serem verificados junto à comunidade científica e profissional na área para averiguarem a sua força, tanto no processo de geração de chaves, função *Feitel*, como no processo de cifração/decifração [7]. A segurança dos algoritmos criptográficos reside basicamente em dois processos fundamentais que são a geração das chaves secretas e o processo de cifração e decifração, ambos trabalhando em conjunto para resultar na criptografia do documento, texto ou fluxo de dados.

O que se pretende aqui é fazer um estudo detalhado do DES e AES para se ter elementos para executar o comparativo entre os dois algoritmos e assim atingir o objetivo de fazer um estudo entre o algoritmo simétrico *Data Encryption Standard* (DES) e o *Advanced Encryption Standard* (AES), por serem os mais conhecidos e utilizados [13] [14] [22], concluindo qual deles é o mais seguro e mais rápido, em relação ao seu processo de geração de chaves e processo criptográfico. Para tanto foi realizada uma pesquisa bibliográfica para a compilação de material da literatura para se obter elementos técnicos que compusessem esse trabalho.

O restante deste artigo está dividido da seguinte forma: a Seção inicial tratou da introdução; a Seção dois faz um apanhado da literatura que trata do assunto estudado; a Seção três apresenta os elementos que caracterizam o algoritmo DES; a Seção quatro trata da caracterização do algoritmo AES; a Seção cinco apresenta o estudo comparativo entre os dois algoritmos, concluindo qual deles é o melhor; e finalmente a Seção seis trata das considerações finais e trabalhos futuros.

2. REVISÃO DA LITERATURA

Em [11], os autores criaram um novo método de criptoanálise linear de cifra do DES, que é essencialmente um ataque de texto simples conhecido. Como resultado, foi possível quebrar cifra DES de oito rodadas com textos simples conhecidos de 2^{21} e cifra DES de 16 rodadas com textos simples conhecidos de 2^{47}, respectivamente. Seu método é aplicável a um ataque somente de texto cifrado em certas situações. Por exemplo, se os textos simples consistem em sentenças naturais em inglês representadas por códigos ASCII, a cifra DES de 8 rodadas é quebrável apenas com textos cifrados 2^{29}.

Esse seu trabalho é um exemplo de como a cifra do DES é vulnerável a ataques por criptoanálise linear.

Em [3], os autores realizaram um trabalho onde apresenta detalhes do processo de cifração e decifração do algoritmo do DES. Explana o processo de criptoanálise no DES e retratam ainda da expansão do DES (triplo DES e DES-X).

Em [6], os autores apresentam em seu trabalho uma técnica usada para proteger dados, por meio de criptografia. Discorrem os autores que a mensagem secreta é criptografada por uma cifra de bloco com base em dois algoritmos criptográficos, o *Data Encryption Standard* e o *Triple Data Encryption Algorithm* (TDEA), que podem ser usados por organizações federais para proteger dados confidenciais. Este algoritmo define exclusivamente as etapas matemáticas necessárias para transformar dados em uma cifra criptográfica e também para transformar a cifra de volta à forma original com comprimento de bloco de 128 bits e comprimento de chave de 256 bits, relatam os autores. Seu trabalho ainda fornece uma comparação de desempenho entre os algoritmos de criptografia mais comuns: DES, 3DES, AES e *Blowfish*.

Em [24], os autores afirmam que o DES é um dos procedimentos de criptografia/descriptografia de cifra de bloco mais preferidos usados atualmente. Seu trabalho apresenta uma implementação de *hardware* reconfigurável de alto rendimento do algoritmo de criptografia DES. Isso foi conseguido usando uma nova implementação proposta do algoritmo DES usando o conceito de *pipeline*. A implementação do *design* proposto é apresentada usando FPGAs da família Spartan-3E (XC3S500E) e é uma das implementações de *hardware* mais rápidas com muito mais segurança. Em uma frequência de *clock* de 167,448 MHz para criptografia e 167,870 MHz para descriptografia, ele pode criptografar ou descriptografar blocos de dados a uma taxa de 10.688 Mbps.

Em [12], os autores anunciaram seleção da família de cifras de bloco Rijndael como vencedora da competição AES. As cifras de bloco são a base para muitos serviços criptográficos, especialmente aqueles que fornecem garantia da confidencialidade dos dados. Três membros da família Rijndael foram especificados neste Padrão: AES-128, AES-192 e AES-256. Cada um deles transforma dados em blocos de 128 bits, e o sufixo numérico indica o comprimento de bits das chaves criptográficas associadas.

Em [23], os autores retratam que há muitas violações cibernéticas como acesso ilegal, indivíduo não autorizado, manipulações, destruição ou vulnerabilidades diariamente. Essa deficiência motivou os autores a trabalharem na avaliação comparativa dos algoritmos de criptografia simétrico *Advanced Encryption Standard* (AES) e do algoritmo assimétrico Rivest Shamir *Algorithm* (RSA) para determinar o mais confiável. O resultado mostrou que o AES é melhor entre as duas técnicas de criptografia. Os autores recomendam o AES como um bom algoritmo para medida de segurança, especialmente no desenvolvimento de aplicativos móveis.

Em [10], os autores propuseram um novo método de arquitetura para reduzir a complexidade da arquitetura do algoritmo AES quando ele está sendo implementado em *hardware* como telefone celular, PDAS e cartão inteligente etc. Este método consistiu em integrar o AES criptografado e o AES descriptografado para fornecer um mecanismo de criptografia AES funcional perfeito. Para fazer isso, eles se concentraram em alguns recursos importantes do AES, especialmente os módulos *(Inv)SubBytes* e *(Inv)Mixcolumn*.

Por fim, em [2], os autores avaliaram o desempenho de três algoritmos como AES, DES e RSA para criptografar arquivos de texto sob três parâmetros como tempo de computação, uso de memória e bytes de saída. O tempo de criptografia foi computado para converter texto simples em texto cifrado e, em seguida, comparar esses algoritmos para descobrir qual algoritmo leva mais tempo para criptografar o arquivo de texto. De acordo com os resultados obtidos, o RSA leva mais tempo em comparação com outros algoritmos. Para os segundos parâmetros, o RSA precisa de uma memória maior do que os algoritmos AES e DES. Finalmente, o byte de saída de cada algoritmo foi considerado. O DES e o AES produzem o mesmo nível de byte de saída, enquanto o RSA tem um baixo nível de byte de saída.

3. ALGORITMO DES

Esta seção trata do primeiro algoritmo criptográfico, sendo abordados os processos de geração de subchaves e cifração. Não é objeto de estudo o processo de decifração, pois ele é, basicamente, o processo invertido de cifração.

O DES foi desenvolvido nos anos 70 pela IBM e foi o padrão criptográfico por muitos anos. Utilizam-se chaves de 56 bits, que pelos padrões atuais é considerado bastante curto e suscetível a ataques de criptoanálise linear, devido ao aumento da capacidade computacional. Ele é a cifra de bloco que pega uma sequência de bits de texto simples de comprimento fixo e a transforma por meio de uma série de operações complicadas em outra sequência de bits de texto cifrado do mesmo comprimento [6].

Particularmente, o DES opera em blocos de texto simples de um determinado tamanho (64 bits) e retorna blocos de texto cifrado do mesmo tamanho. Assim, ele resulta em uma permutação entre os 264 possíveis arranjos de 64 bits, cada um dos quais pode ser 0 ou 1. Cada bloco de 64 bits é dividido em dois blocos de 32 bits cada, um bloco da metade esquerda L e a metade direita R [19].

A estrutura do DES é relativamente simples, o que pode tornar mais suscetível a ataques de análises criptográficas. Ele ainda é empregado em sistemas legados, mas não é mais utilizado em aplicações novas, sendo substituído pelo AES [3].

Por fim, o DES foi desenvolvido baseado em um design anterior de *Horst Feistel*, o algoritmo submetido ao *National Bureau of Standards* (NBS) para propor um candidato para a proteção de dados eletrônicos não classificados sensíveis do governo. Em janeiro 1999, o .net distribuído e a *Electronic Frontier Foundation* (EFF) colaboraram para quebrar publicamente uma chave DES em 22 horas e 15 minutos [6].

3.1 Geração de Subchaves

A chave consiste basicamente em 64 bits, no entanto, apenas 56 bits destes são realmente usados pelo algoritmo. Cada 8º bit da chave selecionada são usados apenas para verificar a paridade, ou seja, as posições 8, 16, 24, 32, 40, 48, 56, 64 são removidas da chave de 64 bits resultando na chave efetiva de 56 bits [7] [4] [22].

O processo de geração de chaves do DES é um processo complexo que envolve várias etapas. Na sequência está uma visão geral do processo de geração de chaves do DES [3] [17] descrito em etapas, enquanto a Figura 1 representa o algoritmo de geração de chaves do DES. Pode-se notar que o processo de geração de chaves do DES é complexo e envolve várias etapas.

Etapa 1: **Geração da chave de 64 bits**: a chave de 64 bits é gerada utilizando um algoritmo de geração de chaves. Esse algoritmo pode ser um gerador de números aleatórios ou um algoritmo de *hash*.

Etapa 2: **Permutação inicial**: a chave de 64 bits é permutada utilizando uma tabela de permutação inicial chamada de PC1. Essa tabela é fixa e é utilizada para misturar os bits da chave e reduzir a chave de 64 bits para 56 bits (ver parte de cima da Figura 1).

Etapa 3: **Divisão da chave**: a chave permutada de 56 bits é dividida em duas metades de 28 bits cada. Essas metades são chamadas de "chave esquerda" e "chave direita".

Etapa 4: **Rotação da chave**: a chave esquerda e a chave direita são rotacionadas utilizando um algoritmo de rotação. Esse algoritmo é utilizado para misturar os bits da chave. Cada metade é rotacionada para a esquerda por um número específico de posições, dependendo da rodada. Isso é feito para misturar os bits da chave. A rotação ocorre da seguinte forma:

- Na 1ª rodada, a metade esquerda é rotacionada para a esquerda por 1 posição;

- Na 2ª rodada, a metade esquerda é rotacionada para a esquerda por 2 posições;

- Na 3ª rodada, a metade esquerda é rotacionada para a esquerda por 4 posições;

- Na 4ª rodada, a metade esquerda é rotacionada para a esquerda por 8 posições; e

- Na 5ª rodada, a metade esquerda é rotacionada para a esquerda por 16 posições.

Etapa 5: **Substituição da chave**: a chave rotacionada é substituída utilizando uma tabela de substituição chamada de SBox. Essa tabela é fixa e é utilizada para substituir os bits da chave.

Etapa 6: **Permutação final**: a chave substituída é permutada novamente utilizando uma tabela de permutação final PC-2. Essa tabela é fixa e é utilizada para misturar os bits da chave.

Etapa 7: **Geração da chave de sessão**: a chave de 56 bits é utilizada para gerar a chave de sessão. A chave de sessão é utilizada para criptografar e descriptografar os dados.

3.2 Processo de Cifração

O processo de cifração do DES consiste em duas etapas distintas embutidas uma na outra, ou seja, a função f do DES e a cifra de Feistel que compõe a base do processo de cifração [16].

3.2.1 A Função f do DES

Inicialmente será apresentado o algoritmo da função f empregado no processo de cifragem do DES. Essa função é o coração do algoritmo e tem como entrada os 32 bits do bloco de bits da direita do processo de criptografia do DES que sofre uma expansão para 48 bits que em seguida é realizada uma operação de $\oplus$ (XOR) com os 48 bits da chave K [4].

Essa saída dá entrada ao processo de permutação de passagem pelos 8 SBox sendo 6 bits de entrada para cada SBox e tendo como saída 4 bits para cada SBox, perfazendo um total de saída de 32 bits da função f. Essa função do DES é representada pela Figura 2.

3.2.2 O Algoritmo de Cifração do DES

O processo de cifração é composto pela realização de duas permutações (P-Boxes), que são as permutações iniciais (IP) e finais (IP^{-1}), e dezesseis rodadas *Feistel*.

Inicialmente tem-se uma entrada de 32 bits que realizada uma permutação inicial. Em seguida os 32 bits permutados é dividido em duas parte L_0 e R_0 (ver Figura 4). Aplicando a função $\boldsymbol{f}$ na parte da direita tem-se as fórmulas (1) e (2) para L_n e R_n.

$$L_n = R_{n-1} \quad (1)$$

$$R_n = L_{n-1} \oplus \boldsymbol{f}(R_0,k_1) \quad (2)$$

Cada uma das 16 chaves de sessão que são utilizadas no algoritmo é gerada no processo de geração de chaves (indicado na Figura 1).

Quando termina a 16ª rodada do algoritmo é realizada uma inversa da permutação inicial e tem-se o dado criptografado pelo DES.

O esquema geral para criptografia DES é ilustrado nas figuras 3 e 4. Essas figuras diferem entre si pelo fato da Figura 3 apresentar o gerador de subchaves do DES para cada um dos rounds da cifra de *Feistel*. Esse gerador de chave foi estudado em Seção anterior.

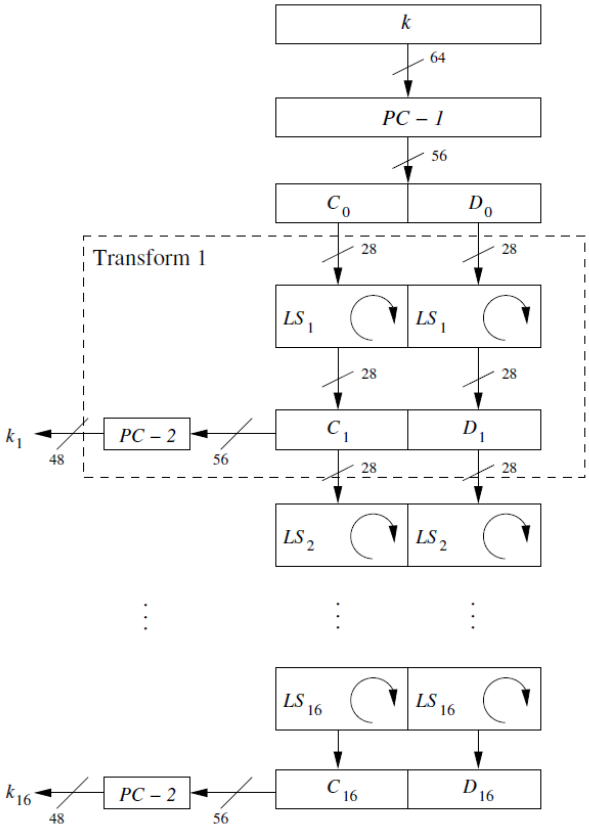

Figura 1. Geração de Subchaves do DES [15].

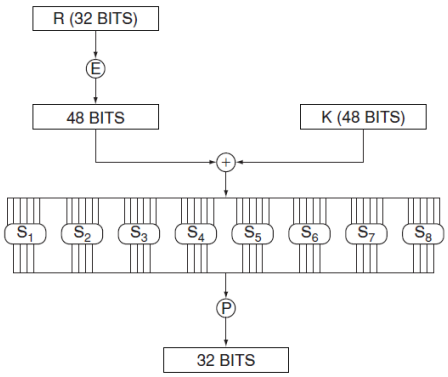

Figura 2. Função *f* do DES [3].

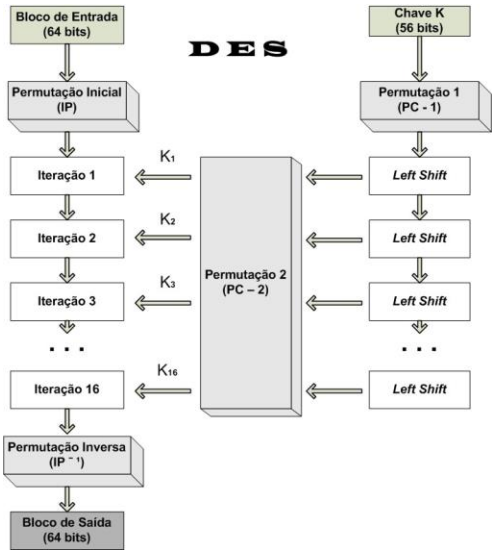

Figura 3. Estrutura geral do DES. Fonte: os autores.

Para tornar o algoritmo mais complexo, a cada rodada *Feistel* é realizada uma inversão dos blocos da esquerda L com os da Direita R (ver Figura 4) [24]. A função ***f*** é aplicada apenas no bloco da direita da entrada de bits.

Na 16ª e última rodada no *PREOUTPUT* não é feita a inversão de lados dos blocos da direita com o da esquerda e em seguida aplica-se a inversa da permutação inicial a IP^{-1} para se obter a saída em texto cifrado.

Essa inversão de metade dos blocos e a consequente aplicação da função ***f*** na parte direita do bloco de bits, aliado ao processo de geração de chaves do DES, imprime o grau de complexidade do algoritmo.

O DES é forte, mas não é ótimo contra criptoanálise linear ou ataque de Davies aprimorado [5], por exemplo, a simples reordenação das caixas S tornaria a cifra menos vulnerável a esses ataques sem prejudicar sua força contra o ataque diferencial. Isso pode indicar que os projetistas do DES não sabiam sobre tais ataques [3].

4. ALGORITMO AES

Nessa Seção do trabalho será abordado o AES, seu processo de geração de chaves e de cifração. Pelas mesmas razões da Seção anterior não será abordado o processo de decifração do algoritmo.

O AES foi publicado em 2001 devido a sua grande segurança e flexibilidade. Suporta chaves de 128, 192 e 256 bits, oferecendo níveis de segurança muito mais altos que o DES [14] [21] [26]. Possui uma estrutura mais complexa e versátil, tornando-o mais resistentes a ataques. É amplamente utilizado em diversas áreas como segurança de rede sem fio (Wi-Fi), criptografia de discos rígidos, VPN e protocolos de comunicação seguros [21].

Os três tamanhos de chaves podem ser usados para diversos fins criptográficos, por exemplo, se for criptografar um arquivo muito grande (terabytes para cima) seria interessante usar uma chave de 128 bit por ser muito mais rápida que a de 256 bits.

O processo de criptografia AES consiste em várias etapas principais, incluindo substituição de bytes, deslocamento de linha, ofuscação de coluna e adição de chave [25]. No processo, a matriz de estado inicial e a chave realizam uma série de operações para gerar o texto cifrado. A fase de substituição de bytes usa uma S-box para substituir os elementos na matriz de estado, chamada de *state*. O processo de construção desta S-box passa pela operação inversa e operação afim. A operação de deslocamento de linha move a posição dos elementos na matriz de estado de acordo com uma regra predefinida, enquanto a operação de ofuscação de coluna trata as colunas na matriz de estado como polinômios. Finalmente, por meio da operação de adição de chave, a matriz de estado após a confusão de coluna e a chave estendida são OR individualmente, e o texto cifrado final é gerado. Essas etapas não envolvem apenas cálculo matemático, mas também incluem alguns conceitos básicos em criptografia, o que é de grande importância para a compreensão e aplicação do algoritmo de criptografia AES [9].

Comparada a cifras de chave pública como RSA, a estrutura de AES e da maioria das cifras simétricas é bastante complexa e não pode ser explicada tão facilmente quanto muitos outros algoritmos criptográficos [20].

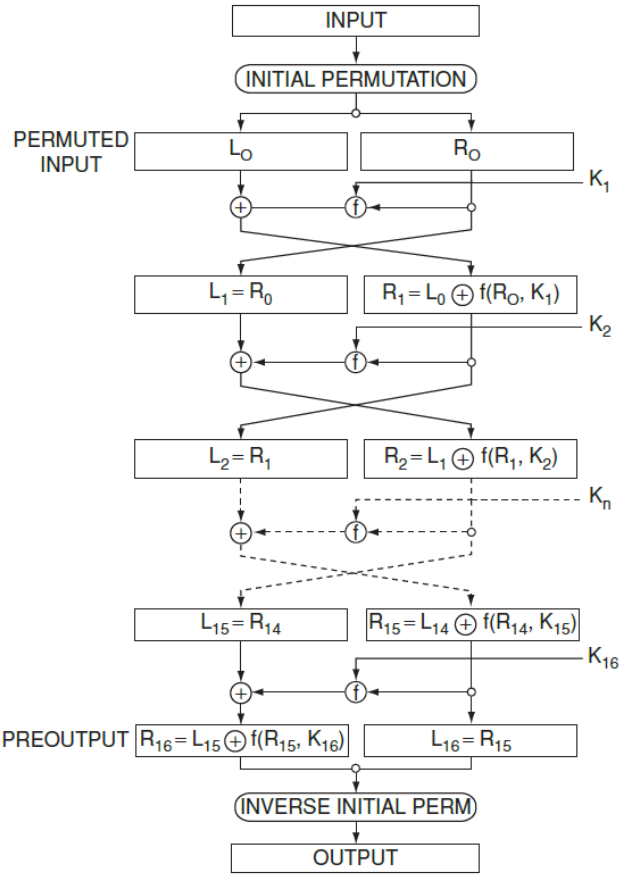

Figura 4. Algoritmo de criptografia do DES [3].

4.1 Geração de Subchaves

O exemplo que será trabalhado nesse artigo é o AES de 128 bits do tamanho da chave para explanação de todo o processo de expansão da chave de 128 bits para 176 bits que serão utilizados nas 10 rodadas do processo criptográfico do AES.

Para uso em criptografia AES, uma única chave inicial pode ser expandida em uma série de *rounds keys* usando a técnica de expansão de chaves do AES. Essas *rounds keys* são necessárias para cada rodada de criptografia e descriptografia do AES.

O método de expansão de chave AES recebe uma chave de quatro palavras (16 bytes) e retorna uma matriz linear de 44 palavras (176 bytes). Isso é suficiente para fornecer a etapa AddRoundKey inicial e uma chave de rodada de quatro palavras para cada uma das dez rodadas da cifra.

O uso da chave no algoritmo de criptografia e o processo de mistura de parte da chave (uma palavra) através da função g é apresentado na Figura 5.

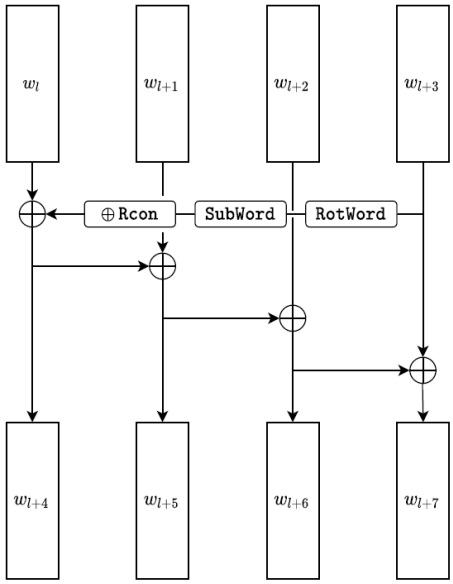

Figura 5. Expansão de chave do AES [20].

No processo de uso da chave pelo algoritmo do AES, conforme consta na Figura 5, é mostrado nas equações em (3).

$$\begin{aligned} w_4 &= w_0 \oplus g(w_3) \\ w_5 &= w_1 \oplus w_4 \\ w_6 &= w_2 \oplus w_5 \\ w_7 &= w_3 \oplus w_6 \end{aligned} \qquad (3)$$

Percebe-se pelas equações em (3) que apenas a palavra w3 é submetida a função ***g***, nessa rodada. Para as demais rodadas as palavras w7, w11..., w39, são aplicadas também à função ***g*** para as rodadas 1,2 ..., 10, respectivamente.

A Figura 6 trás a *KeyExpansion*, expansão de chaves do AES128 para gerar as palavras w[i] para $4 \leq i < 44$, onde l varia entre os múltiplos de 4, entre 0 e 36.

Olhando para a parte b da Figura 6 tem-se a apresentação da função ***g*** onde a mistura das palavras de w3, w7, w11 a w39, como indicadas acima. Inicialmente há uma rotação *ShiftLeft*1 dos bytes da palavra. Em seguida a nova ordem dos bytes, ou seja, B1, B2, B3, B0, sofre uma substituição com S-Box de 16x16, para cada um dos quatro bytes da palavra de entrada, resultando em B'1, B'2, B'3 e B'0. Depois da substituição os valores obtidos para cada byte sofrem uma operação de $\oplus$, com um array de Constantes de rodada, conforme o Quadro 1 a seguir.

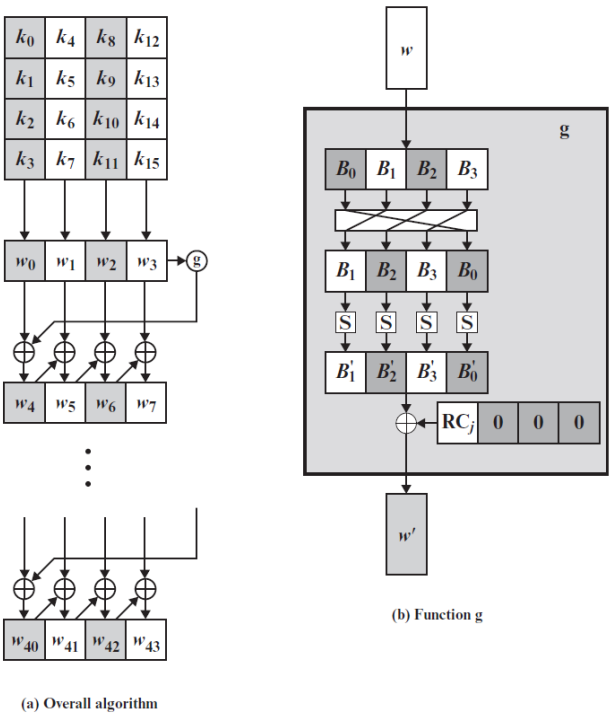

Figura 6. Expansão de chave do AES128 [12].

Quadro 1. Constantes de rodada da função g do AES [20].

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

A chave original é expandida em um número de *rounds keys* usando a técnica de expansão de chave AES. Cada round key é criada a partir da anterior usando constantes de rodadas e operações como XOR, *RotWord* e *SubWord*. Essas chaves de rodadas são então usadas em cada rodada de criptografia AES para aumentar a segurança e evitar ataques criptográficos [10].

O método de expansão de chave AES cria uma chave inicial e uma coleção de chaves de rodadas, que são necessárias para criptografia e descriptografia no AES. Em operações como *RotWord* realizada pela Fórmula 4, *SubWord* realizada pela Fórmula 5 e XOR, ele faz uso de constantes de rodadas. A segurança é aumentada, pois cada chave de rodada garante uma chave única para cada rodada de criptografia. O processo melhora a segurança e protege contra ataques criptográficos.

A *KeyExpansion* é uma rotina aplicada à chave para gerar 4 ∗ (Nr+1) palavras. Assim, quatro palavras são geradas para cada uma das Nr+1 aplicações de *AddRoundKey* dentro da especificação do algoritmo. A saída da rotina consiste em uma matriz linear de palavras, denotada por w[i], onde i está no intervalo $0 \leq i < 4 * (Nr+1)$ [12]. Onde Nr equivale ao número de rodadas do algoritmo (10, 12 ou 14 rodadas).

KeyExpansion invoca 10 palavras fixas denotadas por RC[j] para $1 \leq j \leq 10$. Essas 10 palavras são as constantes de rodadas. Para AES-128, uma constante de rodada distinta é utilizada na geração de cada uma das 10 chaves de rodada. Os valores de RC[j] são dados em notação hexadecimal no Quadro 1.

$$\text{ROTWORD}([a0,a1,a2,a3]) = [a1,a2,a3,a0] \quad (4)$$

$$\text{SUBWORD}([a_0,\ldots,a_3]) = [\text{SBOX}(a_0),\text{SBOX}(a_1),\text{SBOX}(a_2),\text{SBOX}(a_3)] \quad (5)$$

Os desenvolvedores do AES projetaram o algoritmo de expansão de chave para ser resistente a ataques criptoanalíticos conhecidos. A inclusão de uma constante de rodada dependente da rodada elimina a simetria, ou similaridade, entre as maneiras pelas quais as chaves de rodada são geradas em diferentes rodadas [20].

4.2 Processo de Cifração

O núcleo do algoritmo para cifragem é uma sequência de transformações fixas do estado chamada rodada. Cada rodada requer uma entrada adicional chamada chave de rodada; a chave de rodada é um bloco que é geralmente representado como uma sequência de quatro palavras, ou seja, 16 bytes [12].

As rodadas, na especificação do algoritmo, são compostas das quatro transformações orientadas a bytes chamada de *state*, conforme apresentado pela Figura 7 [19] [22] [26]:

• *Substitute Bytes ou SubBytes* aplica uma tabela de substituição (S-box) a cada byte;

• *ShiftRows* desloca linhas da matriz de state por diferentes deslocamentos;

• *MixColumns* mistura os dados dentro de cada coluna da matriz de state; e

• *AddRoundKey* combina uma chave de rodada com o state.

Observe que o esquema da Figura 7 apresenta 10 rodadas, indicando que se trata do algoritmo AES de 128 bits. A chave expandida possui 44 palavras, sendo empregada em cada rodada 4 palavras de 4 bits da chave, parte central da Figura 7.

A rodada final de criptografia e descriptografia consiste em apenas três estágios. Novamente, isso é uma consequência da estrutura particular do AES e é necessário para tornar a cifra reversível.

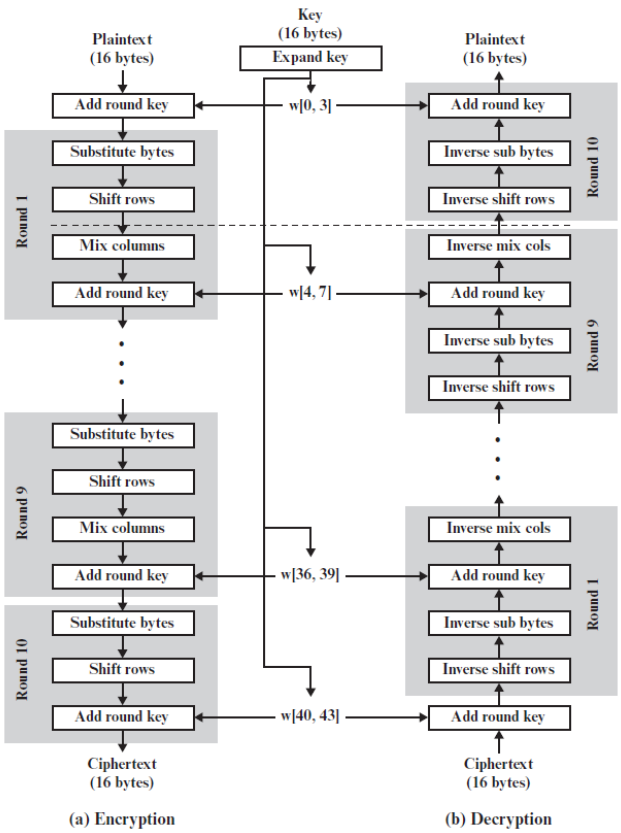

Figura 7. Cifração e decifração do AES-128 [20].

A seguir, nas quatro Subseções seguintes serão apresentadas explicações para cada uma das operações internas do Cipher, que são: *SubBytes*; *ShiftRows*; *MixColumns*; e *AddRoundKey*.

4.2.1 SubBytes

Essa implementação utiliza a S-Box do AES para substituir cada byte do *state* por um outro *byte*.

A operação *SubBytes* é uma substituição não linear de cada byte do estado por um outro *byte*, utilizando uma tabela de substituição chamada S-Box. A S-Box é uma tabela de 256 entradas, onde cada entrada é um byte ou 8 *bits*.

Cada *byte* do *state* é considerado separadamente. Cada *byte* é utilizado como índice para acessar a S-Box. O *byte* correspondente na S-Box é utilizado como substituto para o byte original.

A operação *SubBytes* é importante para a segurança do AES, pois ajuda a: introduzir não linearidade no algoritmo; difundir os *bits* de entrada de forma uniforme; e resistir a ataques de criptanálise. A Figura 8 ilustra a aplicação do *SubBytes*.

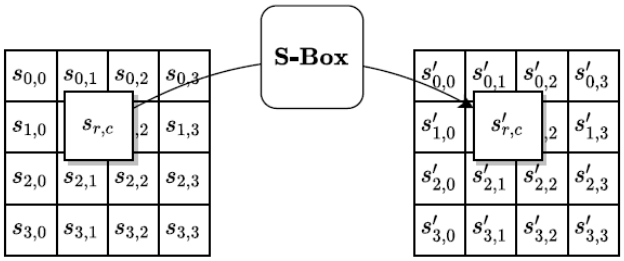

Figura 8. Aplicação do *SubBytes* [12].

4.2.2 ShiftRows

O estágio ShiftRows resume em rotações ShiftLeft da matriz 4x4 de entrada ou *state*, da seguinte forma: a primeira linha não é rotacionada; a segunda linha sofre uma rotação *ShiftLeft* 1; a terceira linha é rotacionada em *ShiftLeft* 2 e finalmente, a quarta linha é rotacionada em *ShiftLeft* 3.

A operação *ShiftRows* é importante para a segurança do AES, pois ajuda a misturar os bytes do state e a evitar ataques de criptoanálise. A Figura 9 apresenta o *ShiftRows* e seus deslocamentos.

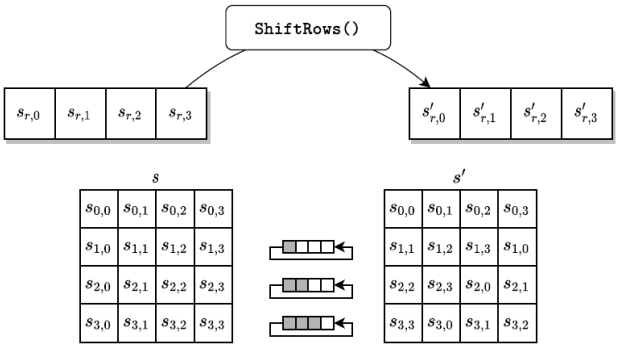

Figura 9. Deslocamentos com *ShiftRows* [12].

Observe na parte inferior da Figura 9 que a 1ª linha da matriz 4x4 não sofre nenhuma rotação ao passo que as três linhas subsequentes sofrem rotação de 1, 2 e 3 shiftleft, respectivamente.

4.2.3 MixColumns

O *MixColumns* faz a mistura das colunas do state (matriz 4x4) da seguinte forma: é realizada uma multiplicação de matrizes 4x4, sendo a primeira matriz formada por um padrão definido e a segunda matriz é o próprio *state*.

Uma forma de visualização dessa multiplicação é apresentada nas quatro equações em (6), onde ***s'*** representa a saída da matriz 4x4, que é o novo *state* para entrada do próximo estágio. A Figura 10 ilustra essa multiplicação de matrizes do *MixColumns*.

$$
\begin{aligned}
s'_{0,j} &= (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\
s'_{1,j} &= s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j} \\
s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j}) \\
s'_{3,j} &= (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})
\end{aligned}
\tag{6}
$$

Como exemplo da aplicação das equações em (6) será apresentado a seguir como ficaria o byte da primeira linha e primeira coluna da matriz de saída:

$$S'_{0,0} = (\{02\} \cdot \{S_{0,0}\}) \oplus (\{03\} \cdot \{S_{1,0}\}) \oplus S_{2,0} \oplus S_{3,0} \quad (7)$$

Onde $S_{0,0}$, $S_{1,0}$, $S_{2,0}$, $S_{3,0}$ são os *bytes* da primeira coluna do *state*.

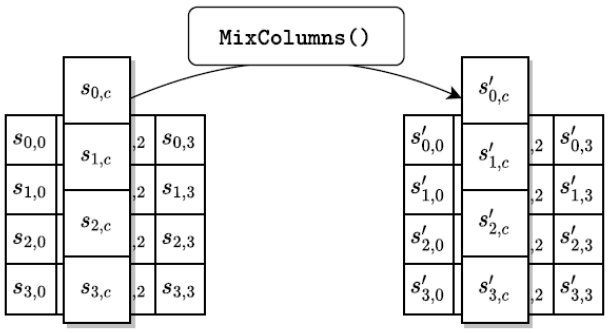

Figura 10. Ilustração do *MixColumns* [12].

4.2.4 AddRoundKey

AddRoundKey é uma transformação do estado em que uma chave de rodada é combinada com o state aplicando a operação $\oplus$ bit a bit. Em particular, cada chave da rodada consiste em quatro palavras da programação de chaves, cada uma das quais é combinada com uma coluna do *state* como segue na Fórmula (7):

$$[s_{0,c},\ s_{1,c},\ s_{2,c},\ s_{3,c}] = [s_{0,c},\ s_{1,c},\ s_{2,c},\ s_{3,c}] \oplus [w_{(4*round+c)}] \text{ para } 0 \leq c < 4 \quad (7)$$

Onde *round* é um valor no intervalo $0 \leq round \leq Nr$, e w[i] é o array de palavras-chave de agendamento. Na especificação de *Cipher*, *AddRoundKey* é invocado Nr + 1 vezes — uma vez antes da primeira aplicação da função round e uma vez dentro de cada das Nr rodadas, quando $1 \leq round \leq Nr$. A ação desta transformação é ilustrada na Figura 11, onde $l = 4 * round$.

A operação *AddRoundKey* basicamente funciona da seguinte forma:

1. O *state* atual é dividido em 16 bytes (ou 128 bits);

2. A chave da rodada é dividida em 16 bytes (ou 128 bits);

3. Cada byte do *state* é realizado um $\oplus$ com o byte correspondente da chave da rodada; e

4. O resultado da operação de $\oplus$ é o novo *state*.

A operação *AddRoundKey* é importante para a segurança do AES, pois ajuda a: introduzir a chave de rodada no state; misturar os bits do state com os bits da chave de rodada; e evitar ataques de criptanálise que dependem da estrutura do state.

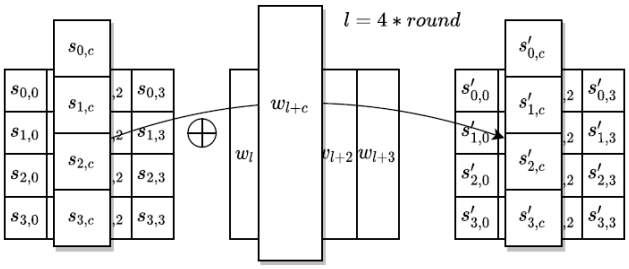

Figura 11. Transformação *AddRoundKey* [12].

No caso que está sendo apresentado, ou seja, o AES128, na sua 10ª e última rodada da criptografia não existe este estágio, conforme dito anteriormente.

Observa-se que, no início de todo o processo criptográfico, os bits de entrada (*state*) do texto claro é embaralhado de variadas formas e esse processo se repete de 10 vezes para a chave de 128 bits. Isso imprime ao algoritmo AES uma segurança bastante robusta, sendo resistente ao ataque por força bruta.

A Figura 12 é outra visão da Figura 7 e mostra a estrutura geral do processo de criptografia AES. A cifra pega um tamanho de bloco de texto simples em claro de 128 bits, ou 16 bytes, que é a entrada do algoritmo é de tamanho fixo, com uma saída de bits criptografados também de 128 bits fixos. O bloco de entrada é representado como uma matriz quadrada de 4 x 4 bytes. Em seguida, Este bloco é copiado para a matriz *State*, que é modificada em cada estágio de criptografia ou descriptografia. Após o estágio final, o *State* é copiado para uma matriz de saída.

A Figura 13 apresenta uma maior granularidade da rodada de encriptação do algoritmo AES onde é possível perceber a dinâmica que envolve cada uma das etapas das rodadas do algoritmo. Com essa imagem é possível se ter uma ideia do grau de complexidade que envolve o processo de cifração do AES para cada uma das 10 rodadas que perfazem a cifração com o AES-128 bits.

Agora, se fosse expandido cada uma das quatro operações (*SubBytes*, *ShiftRows*, *MixColumns* e *AddRoundKey*) que compõe a rodada ter-se-ia uma maior visão dessa complexidade da arquitetura criptográfica do AES. As figuras de 8 a 11 dão uma visão singular dessas operações matemáticas.

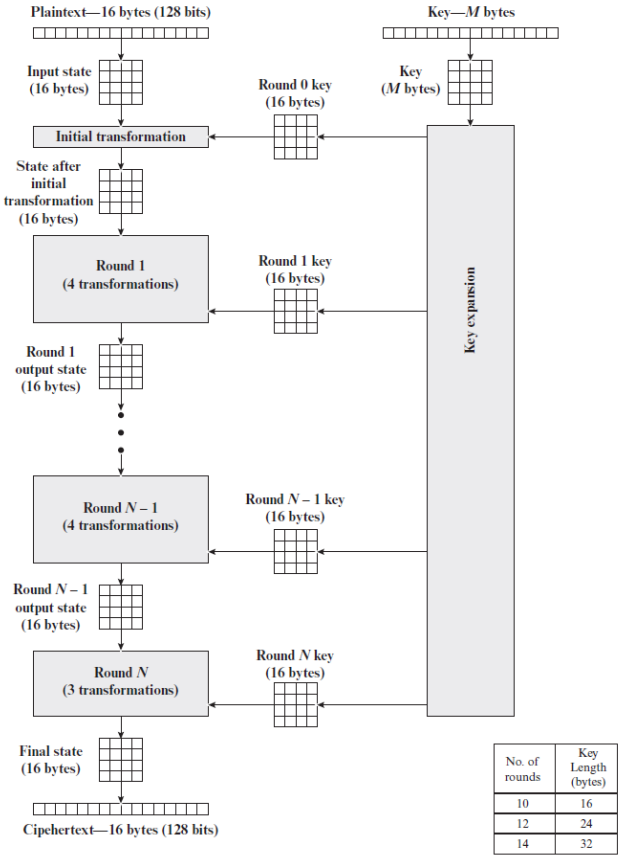

Figura 12. Estrutura do processo de cifração do AES [20].

Observe as quatro palavras (4 bits cada) utilizadas pelo *state* ao longo de cada bloco do algoritmo (*SubBytes*, *ShiftRows*, *MixColumns* e *AddRoundKey*) que engloba a rodada.

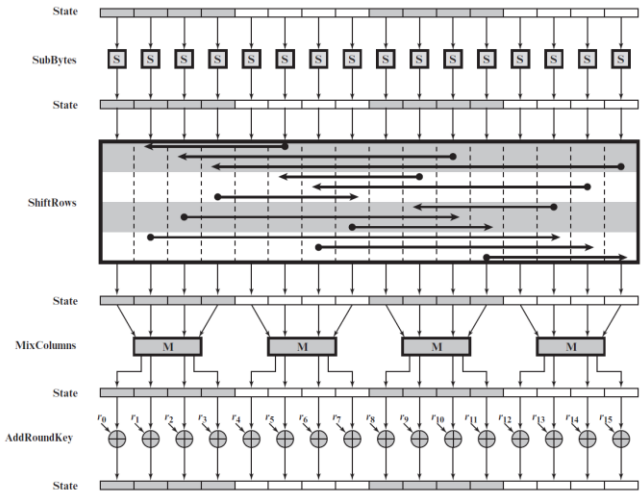

Figura 13. Rodada de encriptação do algoritmo AES [20].

5. COMPARAÇÃO DOS ALGORITMOS

Os algoritmos AES e DES são dois dos mais conhecidos e utilizados na criptografia simétrica. Embora ambos tenham como objetivo cifrar e decifrar dados, apresentando diversas diferenças em termos de segurança, desempenho e aplicações [16].

O AES utiliza uma rede de substituição-permutação para realizar a criptografia isso lhe proporciona uma arquitetura de criptografia mais avançada. Ele é considerado seguro e resistente a ataques de força bruta e criptoanálise e é amplamente utilizado em aplicações modernas, incluindo SSL/TLS, IPsec e criptografia de disco [21]. Já o DES utiliza uma chave efetiva de 56 bits tornando-o suscetível a ataque de força bruta e criptoanálise linear, portanto não é considerado seguro para aplicações modernas. Diferentemente de seu primo o TriploDES é seguro pois potencializa a criptografia de cifragem e decifragem, usando uma chave de 168 bits, já considerada segura [18].

Quanto a questão de velocidade dos algoritmos, o DES mostrou-se mais rápido tanto para criptografar quanto para descriptografar em relação ao AES [8]. A razão para isso pode se dar pela menor complexidade da estrutura do DES em relação ao AES e ao seu tamanho de chave ser menor que o do AES. A Figura 14 mostra um exemplo dos testes realizados entre eles.

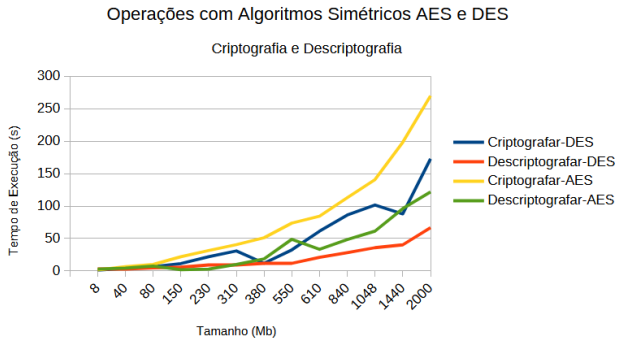

Figura 14. Comparação entre DES e AES. Fonte: os autores.

Outro estudo comparativo chegou ao seguinte resultado, se empregado o *Cipher Block Chaining* (CBC). O resultado encontra-se na Figura 15. CBC é um modo de operação de criptografia que é utilizado para criptografar dados em blocos. É um dos modos de operação mais comuns e utilizados em criptografia.

Observando tanto as Figuras 14 e 15 percebe-se o processo de cifração e decifração do DES é mais rápido que do AES [18]. Agora se for imaginada a cifração de arquivos com *terabytes* de tamanho, a diferença de tempo entre os dois algoritmos seria considerável. Por isso, o AES pode operar com três tamanhos de chaves para acelerar esse processo, além de ser otimizada sua aplicação em *hardware* específico, obtendo-se ótimas performances de velocidade de cifração e decifração [21] [16].

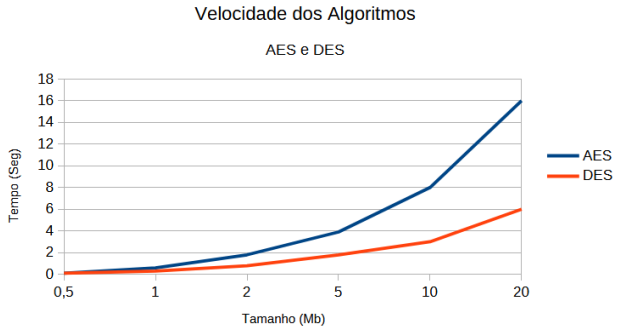

Figura 15. Cifragem do AES e DES. Fonte: os autores.

O Quadro 2 apresenta dados comparativos entre o DES e o AES 128. Percebe-se que o AES é bem mais seguro que o DES.

Pode-se notar no Quadro 2 que o tempo de decifração da chave requerido pelos dois algoritmos é extremamente diferente, sendo que para o DES possui um tempo viável para ser quebrado por força bruta ou criptoanálise.

Quadro 2. Comparativo de quebra dos Algoritmos. Fonte: adaptado de [20].

Tamanho da chave (bits)	Cifra	Número de chaves alternativas	Tempo requerido a 10^9 decifrações/seg	Tempo requerido a 10^{13} decifrações/seg
56	DES	$2^{56} \sim 7{,}2 \times 10^{16}$	2^{55} ns = 1,125 anos	1 hora
128	AES	$2^{128} \sim 3{,}4 \times 10^{38}$	2^{127} ns = $5{,}3 \times 10^{21}$ anos	$5{,}3 \times 10^{17}$ anos

Ambos os processos de criação de subchaves são bastante complexos proporcionando segurança ao sistema. O que ocorre é que o DES opera com um tamanho de chave muito pequeno que pode ser facilmente quebrado (ver Quadro 2).

Quanto ao processo de cifragem o AES envolve quatro estágios, *SubBytes*, *ShiftRows*, *MixColumns*, *AddRoundKey* em cada rodada ao passo que no DES isso é realizado pela única cifra de Feistel, que é complexa mas não comparado aos estágios do AES, que isso lhe dá uma maior robustez e segurança, apesar de ser mais lento que o DES [21].

O AES substituiu o DES devido a alguns aspectos como: a) a **segurança** – o tamanho da chave de 56 bits do DES tornou-o vulnerável a ataques de força bruta. O AES, com chaves muito maiores, oferece uma segurança muito mais robusta; b) a **flexibilidade** – o AES suporta diferentes tamanhos de chave, permitindo escolher o nível de segurança mais adequado para cada aplicação; e c) o **desempenho** – embora o DES seja relativamente rápido [8], o AES, especialmente em *hardware* otimizado, oferece um desempenho excelente. Ou seja, o AES é considerado o padrão atual em criptografia simétrica devido à sua segurança, flexibilidade e desempenho [18]. O DES, por sua vez, é um algoritmo histórico que, apesar de ter sido importante em sua época, não é mais recomendado para novas aplicações. O Quadro 3 apresenta as características básicas do DES e AES.

Quadro 3. Características do DES e AES. Fonte: o autor

Características	DES	AES
Tamanho da chave	56 bits	128, 192 ou 256 bits
Segurança	Considerado fraco pelos padrões atuais	Considerado muito seguro
Desempenho	Relativamente rápido	Rápido, especialmente em *hardware* otimizado
Estrutura	Simples	Complexa e versátil
Aplicações	Sistemas legados	Amplamente utilizado em diversas áreas

Agora será medida a arquitetura criptográfica dos algoritmos em estudo avaliando-se a quantidade de estruturas matemáticas existentes tanto para o processo de geração de Subchaves quanto ao processo de cifração. Os Quadros 4 e 5 apresentam as quantidades dessas estruturas para avaliação, respectivamente.

Quadro 4. Estruturas do Processo de Subchaves dos Algoritmos DES e AES.

Estruturas	DES	AES
Permutação	17	-
Substituição (SBox)	16	10
Rotação	32	10
Operações com XOR	-	60
Divisão de bloco	16	-
Inversão de lados	-	-
Expansão	-	-
TOTAL	81	80

No quesito quantidade de estruturas matemáticas dos processos de Subchaves percebe-se que os algoritmos possuem valores aproximados tendo o DES uma estrutura a mais que o AES.

Quadro 5. Estruturas do Processo de Cifração dos Algoritmos DES e AES.

Estruturas	DES	AES
Permutação	18	122
Substituição (SBox)	128	50
Rotação	-	40
Operações com XOR	40	141
Divisão de bloco	-	11
Inversão de lados	15	-
Expansão	16	-
TOTAL	217	364

O resultado para esse quesito tem ao todo certa vantagem para o AES, são 147 estruturas a mais que o DES. Além do mais, levando-se em consideração que o processo de cifração do DES envolve 16 rodadas ao passo que o do AES é de 10 rodadas, percebe-se que, por rodada, o DES emprega 14 estruturas ao passo que o AES emprega 37 estruturas, portanto o processo de cifração do AES é mais complexo, neste quesito.

Certamente a quantidade de estruturas matemáticas adotadas pelos algoritmos interfere diretamente na sua complexidade e principalmente como essas estruturas estão imbricadas umas nas outras perfazendo um verdadeiro emaranhado de funções e procedimentos para imprimir complexidade aos algoritmos. Considera-se que no AES são utilizados quatro processos distintos,

SubBytes, *ShiftRows*, *MixColumns* e *AddRoundKey* para a cifração ao passo que o DES utiliza a cifra de *Feistel* contendo uma função ***f***, que é o coração do algoritmo, isso representa por si só uma maior complexidade para o AES.

Somente para ilustrar, o AES foi assim retratado quando estava em processo de aceitação pelo Instituto Nacional de Padrões e Tecnologia (NIST) em Mar 1999: "Excelente desempenho em todas as plataformas; boa margem de segurança; adequado para cartões inteligentes devido aos baixos requisitos de memória de acesso aleatório (RAM) e memória somente leitura (ROM); as operações defendem bem contra ataques em implementações de cartões inteligentes; configuração rápida de chaves; suporta tamanhos de chaves e blocos expandidos; bom paralelismo em nível de instrução" [21].

6. CONCLUSÃO

O presente trabalho teve como objetivo realizar uma análise comparativa entre os algoritmos criptográficos simétricos DES e AES, com foco nos processos de geração de chaves e cifração, a fim de identificar qual deles apresenta maior segurança e eficiência. Para atingir esse propósito, foi conduzida uma pesquisa bibliográfica baseada em livros, artigos e normas técnicas reconhecidas na área de criptografia, como as publicações do NIST e autores clássicos. A metodologia adotada envolveu a descrição detalhada de cada algoritmo, desde a estrutura matemática de suas funções até as etapas de transformação dos dados durante a cifragem.

Os resultados obtidos demonstraram que, embora o DES apresente um processo de cifração relativamente rápido, sua chave efetiva de apenas 56 bits o torna vulnerável a ataques de força bruta e criptoanálise linear. Em contrapartida, o AES, ao empregar chaves de 128, 192 e 256 bits e uma estrutura de cifração composta por múltiplas transformações (SubBytes, ShiftRows, MixColumns e AddRoundKey), revelou-se significativamente mais robusto e seguro. A análise quantitativa das estruturas matemáticas mostrou ainda que o AES possui maior complexidade interna, garantindo melhor difusão e resistência criptográfica, ainda que com custo computacional ligeiramente superior.

A principal contribuição deste estudo consiste em oferecer uma visão técnica e comparativa abrangente sobre os dois algoritmos, permitindo compreender não apenas suas diferenças estruturais e de desempenho, mas também os motivos que levaram o AES a substituir o DES como padrão mundial de criptografia simétrica. O trabalho contribui, portanto, para o entendimento aprofundado dos mecanismos de segurança criptográfica e fornece uma base teórica útil para estudantes, pesquisadores e profissionais que atuam na área de segurança da informação.

Como perspectivas de continuidade, esta pesquisa pode ser ampliada em diferentes direções: (1) realizar uma comparação detalhada entre o AES e o Triple DES (3DES) para avaliar ganhos de desempenho e segurança; (2) investigar implementações otimizadas em hardware dos dois algoritmos, considerando consumo energético e velocidade; (3) aplicar testes empíricos de desempenho em diferentes plataformas e linguagens de programação; e (4) explorar o uso de algoritmos simétricos em conjunto com técnicas de criptografia quântica ou híbrida, avaliando sua adequação aos novos paradigmas de segurança digital.

7. REFERÊNCIAS

[1] Abdullah, A. M. (2017) Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data. Cryptography and Network Security.

[2] Berent, A. (2013) Advanced Encryption Standard by Example. Disponível em: http://www.networkdls.com/Articles/AESbyExample.pdf Acesso em 01 Jan 25.

[3] . Biryukov, A. e Canniére, C. (2010) Data Encryption Standard (DES). Springer Fachmedien Wiesbaden GmbH. DOI: https://doi.org/10.1007/978-1-4419-5906-5_568 .

[4] Chen, M. (2021) Accounting Data Encryption Processing Based on Data Encryption Standard Algorithm, Complexity, vol. 2021, Article ID 7212688, 12 pages.

[5] Davies, D. W. A. (1987) New Attack on the DES. Advances in Cryptology - CRYPTO 87. Lecture Notes in Computer Science, vol. 293, pp. 267-276. Springer-Verlag.

[6] Karthik, S. e MURUGANANDAM, A. (2014) Data Encryption and Decryption by Using Triple DES and Performance Analysis of Crypto System. International Journal of Scientific Engineering and Research (IJSER). V. 2 N. 11.

[7] Katz, J. e Lindell, Y. (2024) Introduction to Modern Cryptography - The Practical Guide. 425 p., ISBN 978-1-4932-2566-8.

[8] Laia, O.; Zamzami, E. M. e Sutarman. (2021) Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS). 5 th International Conference on Computing and Applied Informatics (ICCAI 2020). DOI: https://doi.org/10.1088/1742-6596/1898/1/012017.

[9] Lia, L.; SONG e Yaxun. (2024) Intelligent Logistics Management System Based On Improved AES Algorithm. The 4th International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy. Procedia Computer Science n. 243 p. 882–890.

[10] Lu, C. C. e Tseng, S. Y. (2002) Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application-Specific Systems, Architectures and Processors, Proceedings. The IEEE International Conference on (p. 277-285).

[11] Matsui, M. (1994) Linear Cryptanalysis Method for DES Cipher. Advances in Cryptology - EUROCRYPT '93, LNCS 765, pp. 386-397. 0 Spnnger-Verlag Berlin Heidelberg.

[12] NIST – National Institute of Standards and Technology. (2023) Advanced Encryption Standard (AES). Federal Information Processing Standards Publication.

[13] Nugroho, A. R. e Pramusinto, W. (2018) Implementasi Kriptografi Dengan Algoritma Caesar Cipher, AES 192 dan DES untuk Aplikasi Pesan Instan Berbasis Android Jurnal Sistem Komputer dan Teknik Informatika (SKANIKA), 1(1), pp 14-19.

[14] . Oladoyinbo, T. O.; Oladoyinbo, O. B. e Akinkunmi, A. I. (2024) The Importance Of Data Encryption Algorithm In Data Security. IOSR Journal of Mobile Computing & Application (IOSR-JMCA) e-ISSN: 2394-0050, P-ISSN: 2394-0042. V. 11, N. 2, p. 10-16.

[15] Paar, C. e Pelzl. (2010) Understanding Cryptography - A Textbook for Students and Practitioners. Springer. DOI: https://doi.org/10.1007/978-3-642-04101-3.

[16] Sabeena, S. J. e VIJILA S. A. (2023) Identification of Better Encryption Algorithm in Securing Data. Journal of Survey in Fisheries Sciences V. 10 N.4S. p. 889-896.

[17] Schneier, B. (1996) Applied Cryptography: Protocols, Algorithms, and Source Code in C. (2nd ed.). John Wiley & Sons. ISBN 0-471-11709-9.

[18] Selent, D. (2010) ADVANCED ENCRYPTION STANDARD. InSight: RIVIER ACADEMIC JOURNAL, VOLUME 6, NUMBER 2, ISSN 1559-9388.

[19] SOOD, R. e Kaur, H. (2023) A Literature Review on RSA, DES and AES Encryption Algorithms. Emerging Trends in Engineering and Management, 57–63. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-955020-3-5-07.

[20] Stallings, W. (2017) Criptografia e Segurança de Redes: Princípios e Práticas. 7ª Ed, São Paulo, Editora Pearson Education do Brasil, ISBN: 978-85-430-1450-0.

[21] Smid, M. E. (2021) Development of the Advanced Encryption Standard. Journal of Research of the National Institute of Standards and Technology. Volume 126, Article No. 126024. DOI: https://doi.org/10.6028/jres.126.024. .

[22] . TENG, L.; Li, H.; Yin, S. e Sun, Y. (2020) A Modified Advanced Encryption Standard for Data Security. International Journal of Network Security, Vol.22, No.1, PP.112-117, DOI: https://doi.org/10.6633/IJNS.202001, 22(1).11.

[23] Olutola, A. e Olumuyiwa, M. (2023) Comparative Analysis of Encryption Algorithms. European Journal of Technology ISSN 2520-0712 (online) Vol.7, n. 1, p. 1 - 9.

[24] Zeebaree, S. R. M. (2020) DES encryption and decryption algorithm implementation based on FPGA. Indonesian Journal of Electrical Engineering and Computer Science Vol. 18, No. 2, May 2020, pp. 774~781, ISSN: 2502-4752, DOI: https://doi.org/10.11591/ijeecs. v18. N2. pp774-781.

[25] Zhao, X. (2010) The Implementation and Application of AES Encryption Algorithm Journal of Changshu Institute of Technology, 2010, v.24 n.2. p. 105-110.

[26] Zodpe, H. e Sapkal, A. (2020) An efficient AES implementation using FPGA with enhanced security features. Journal of King Saud University– Engineering Sciences V. 32 p. 115–122. DOI: https://doi.org/10.1016/j.jksues.2018.07.002.