UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB
CENTRO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUANA MACÊDO CAVALCANTE CHACON DE ALMEIDA

UMA REVISÃO SOBRE OS TIPOS DE CRIPTOGRAFIA E SUA APLICAÇÃO NA PROTEÇÃO DE DADOS EM TRÂNSITO

JOÃO PESSOA/PB
2025

LUANA MACÊDO CAVALCANTE CHACON DE ALMEIDA

UMA REVISÃO SOBRE OS TIPOS DE CRIPTOGRAFIA E SUA APLICAÇÃO NA PROTEÇÃO DE DADOS EM TRÂNSITO

Trabalho de conclusão de curso apresentado ao Curso de Ciência da Computação da Universidade Federal da Paraíba, como requisito final para obtenção do título de Bacharel em Ciência da Computação.

Orientadora: Profª. Dra. Josilene Aires Moreira

JOÃO PESSOA/PB

2025

Catalogação na publicação
Seção de Catalogação e Classificação

A447r Almeida, Luana Macedo Cavalcante Chacon de.
Revisão sobre os tipos de criptografia e sua aplicação na proteção de dados em trânsito / Luana Macedo Cavalcante Chacon de Almeida. - João Pessoa, 2025.
28 f. : il.

Orientação: Josilene Aires Moreira.
TCC (Graduação) - UFPB/CI.

1. Criptografia. 2. Algoritmos. 3. Desempenho. 4. Segurança da informação. I. Moreira, Josilene Aires. II. Título.

UFPB/CI CDU 004.6

Elaborado por MICHELLE DE KASSIA FONSECA BARBOSA - CRB-738

Dedico àqueles que me acompanharam nessa jornada, vocês me deram forças para chegar até aqui.

AGRADECIMENTOS

À minha mãe, Suênia, por nunca deixar de me apoiar e me fazer seguir em frente, hoje estou aqui porque você nunca me permitiu desistir.

Ao meu pai, Fabiano, que me incentivou em primeiro lugar a seguir o caminho da computação, obrigada por acreditar em mim e em meu sonho.

À minha irmã, Larissa, que sempre está ao meu lado, obrigada por sua companhia, cumplicidade e auxílio nesta etapa e em todas as outras.

À minha orientadora Dra. Josilene Aires Moreira, agradeço sua orientação, ajuda e paciência, não só nesse trabalho, mas no decorrer do curso.

UMA REVISÃO SOBRE OS TIPOS DE CRIPTOGRAFIA E SUA APLICAÇÃO NA PROTEÇÃO DE DADOS EM TRÂNSITO

Luana M. C. C. de Almeida[1], Josilene A. Moreira[2]

[1]Aluna do Curso Ciência da Computação – Universidade Federal da Paraíba (UFPB)

[2]Orientadora – Centro de Informática - Universidade Federal da Paraíba (UFPB)

{luanamchacon@gmail.com, josilene@ci.ufpb.br}

Resumo: *Este trabalho tem como objetivo descrever diferentes algoritmos criptográficos e avaliá-los dentro do contexto de sua aplicação prática na proteção de dados em trânsito. Através de uma revisão bibliográfica, foram comparados diferentes algoritmos quanto à eficiência energética, tempo de execução e segurança. Os resultados indicam que o algoritmo ChaCha20 apresenta o melhor desempenho geral em dispositivos limitados, em que não é possível rodar aceleração por hardware, enquanto o AES mostra excelente performance quando utilizado com instruções nativas (AES-NI), já dentre os algoritmos de assinatura digital o DSA se sobressai. A pesquisa destaca a importância da escolha dos algoritmos criptográficos e como essa escolha deve levar em conta a capacidade computacional do dispositivo, sem comprometer a segurança. O estudo contribui para decisões mais eficazes no projeto de sistemas seguros em ambientes com e sem restrição de recursos.*

Palavras-chave*: Criptografia; Algoritmos; Desempenho; Segurança da Informação.*

Abstract. *This work aims to gather and analyze different cryptographic algorithms and evaluate them within the context of their practical application in the protection of data in transit. Through a literature review, different algorithms were compared in terms of energy efficiency, execution time and security. The results indicate that the ChaCha20 algorithm presents the best overall performance in limited devices, where there is no possible way to run hardware acceleration, while AES shows excellent performance when used with native instructions (AES-NI). Among digital signature algorithms, DSA stands out. The research highlights the importance of choosing cryptographic algorithms and how this choice should take into account the computational capacity of the device, without compromising security. The study contributes to more effective decisions in the design of secure systems in environments with and without resource constraints.*

Keywords: *Cryptography. Algorithms. Performance. Data Security.*

SUMÁRIO

1. INTRODUÇÃO

A criptografia, a ciência de tornar dados indecifráveis para terceiros adversários, que historicamente teve como principal objetivo ocultar o significado de mensagens, hoje também é usado para outros objetivos como a autenticidade e integridade das mensagens (PAAR; PELZL; GÜNEYSU 2024). Hoje, ela se situa como um dos atuais pilares da segurança da informação na era digital, que é dependente do armazenamento e troca de dados entre sistemas.

Apesar de sua eficácia na proteção de informações, falhas na implementação, vulnerabilidades em algoritmos e o avanço da capacidade computacional representam desafios constantes para a segurança digital.

Este artigo descreve os conceitos básicos de criptografia, e apresenta os tipos de algoritmos de criptografia mais utilizados na atualidade. Descreve ainda alguns dos protocolos mais utilizados para dados em trânsito e conclui com uma comparação das características destes algoritmos para diferentes funções e tipos de dispositivos.

2. METODOLOGIA

Este trabalho foi desenvolvido por meio de uma revisão bibliográfica com uma abordagem de caráter qualitativo, com o objetivo de reunir, analisar e discutir os diferentes tipos e métodos criptográficos, avaliando sua relevância, eficiência e desafios na sua aplicação em ambientes digitais. A pesquisa foi conduzida com base em materiais acadêmicos, científicos e técnicos que versam desde os fundamentos da criptografia até seus tipos e diferentes finalidades.

No processo de levantamento do material bibliográfico, as referências contemplaram tanto obras fundamentais que constituem referência consagrada na área de segurança da informação quanto publicações recentes que refletem os avanços mais atuais em criptografia.

Nesse sentido, em consonância com a quantidade de materiais relevantes para o tema, foram priorizadas as publicações em questão de relevância, como as de fontes de instituições de peso como a Intel e a NIST, artigos de publicação e documentação dos algoritmos criptográficos e livros acadêmicos de especialistas da área. Além disso, também foram considerados artigos recentes, escolhidos aqueles com dados relevantes e testes práticos que refletem a situação atual da criptografia de dados em

diferentes cenários. Os materiais escolhidos foram em inglês e português, com tradução própria fornecida às referências feitas aos textos em inglês.

Na análise do material selecionado foi adotada uma dupla perspectiva: descritiva e interpretativa. De forma que, foram explorados e sistematizados os principais conceitos, classificações, restrições e usos práticos concretos das diversas formas de criptografia.

Em vista disso, a estruturação do conteúdo foi organizada estabelecendo uma linha de raciocínio progressiva e lógica. O desenvolvimento inicia-se pelos fundamentos teóricos básicos da segurança da informação e da criptografia, avança para a exploração detalhada dos sistemas criptográficos, e finaliza no exame de implementações reais em protocolos e ferramentas de proteção de dados em movimento.

3. Fundamentação Conceitual

3.1 Conceitos Fundamentais de Segurança da Informação e Criptografia

A Segurança da Informação é um conjunto de princípios seguidos com o propósito de proteger dados contra ameaças, formada pelos pilares básicos da: confidencialidade, integridade, disponibilidade (CID) e ainda tem outros aspectos como autenticação, autorização, auditoria, autenticidade, não repúdio e legalidade. Dentro dos pilares CID, a confidencialidade assegura que apenas pessoas autorizadas acessem a informação, enquanto a integridade assegura que os dados sejam protegidos contra alterações indevidas, e a disponibilidade certifica que a informação esteja acessível onde e quando ela for necessária. Já em seus demais aspectos, a autenticação verifica a identidade de uma entidade, garantindo sua integridade e atribuição de responsabilidade, a autorização concede direitos e permissões a uma entidade para a realização de certas ações, a auditoria registra as ações realizadas pelas entidades de forma a corretamente atribuir responsabilidade em casos que sejam infringidas as regras de segurança, a autenticidade verifica a integridade da informação e a identidade dos que a proveram, o princípio do não repúdio vincula uma ação ou documento a entidade que os originou, de forma a prevenir que a autoria destes seja negada, e o princípio da legalidade verifica a conformidade a lei (MUNIZ et al., 2024).

A criptografia é utilizada há milhares de anos para ocultar o significado de mensagens: “A primeira evidência conhecida do uso da criptografia (de alguma forma) foi encontrada em uma inscrição esculpida por volta de 1900 a.C., na câmara principal do túmulo do nobre Khnumhotep II, no Egito” (SIDHPURWALA, 2023).

Na era moderna, com o advento da computação e grandes quantidades de dados que precisam ser protegidos, a criptografia atual é muito mais complexa do que as cifras de substituição utilizadas por boa parte da história, utilizando-se agora de teorias e operações matemáticas complexas e suposições de dureza computacional, projetando algoritmos computacionalmente seguros, em que apesar de em teoria seja possível quebrá-los, na prática, a demanda computacional para se realizar isso impossibilita que isso aconteça. Porém por serem projetados dessa maneira, esses algoritmos devem estar em um constante processo de revisão, já que avanços computacionais e teóricos da matemática podem colocar em risco sua viabilidade.

Como Aumasson (2025, p.82, tradução nossa) descreve,

> Existem dois tipos principais de aplicações de criptografia. A criptografia em trânsito protege os dados enviados de uma máquina para outra: os dados são criptografados antes de serem enviados e descriptografados após o recebimento, como em conexões criptografadas com sites de comércio eletrônico. A criptografia em repouso protege os dados armazenados em um sistema de informações.

O foco deste trabalho será na criptografia em trânsito, levando em conta sua importância nas conexões e comunicações entre sistemas e aparelhos, e por consequência, sua importância no mundo globalizado, tecnológico e conectado dos dias atuais.

3.2 Principais tipos de criptografia e sua aplicabilidade

Como Paar, Pelzl e Güneysu (2024) explicam, a criptografia se divide em três ramos principais: algoritmos simétricos (ou de chave privada), algoritmos assimétricos (ou de chave pública) e protocolos criptográficos.

Criptografia simétrica, o modelo mais antigo de criptografia, em que a chave, isto é, o elemento que dá acesso à mensagem oculta trocada entre duas partes, é igual (simétrica) para ambas as partes e deve permanecer em segredo (privada) (OLIVEIRA, 2012). A criptografia simétrica é uma categoria que inclui diferentes tipos de operações:

- A criptografia, em que a mesma chave secreta é usada para criptografar e descriptografar uma mensagem.

- Os *Message Authentication Codes* (MACs), que são códigos de autenticação de mensagem que tratam de um valor gerado a partir de uma mensagem e uma chave secreta, em que o receptor ao receber a mensagem original e o MAC, pode verificar se a mensagem foi alterada, segundo (KATZ & LINDELL, 2021, p.108, tradução nossa),

  > o objetivo de um código de autenticação de mensagem é impedir que um adversário modifique uma mensagem enviada de uma parte para outra, ou injete uma nova mensagem, sem que o receptor detecte que a mensagem não se originou da parte pretendida.

- Os *Pseudorandom Number Generators (PRNGs) (geradores de números pseudo aleatórios)*, tratam-se de funções determinísticas que em desconhecimento da chave secreta, os valores gerados devem ser indistinguíveis de uma saída verdadeiramente aleatória, já que caso ao contrário pode afetar a efetividade do seu uso, como exemplificado por (STALLINGS, 2014, p.161, tradução nossa)" se o fluxo de bits pseudo aleatório é usado em uma cifra de fluxo, então o conhecimento do fluxo de bits pseudo aleatório permitiria que o adversário recuperasse o texto claro a partir do texto cifrado".

A distribuição da chave é sua maior desvantagem, já que ambas as partes em comunicação precisam ter conhecimento da chave privada, isso é geralmente realizado em canais seguros ou por métodos de criptografia assimétrica.

Dentro da categoria de criptografia simétrica, os algoritmos são geralmente agrupados nas subcategorias: cifras de bloco ou cifras de fluxo. As cifras de bloco dividem o texto a ser criptografado em blocos de tamanho fixo de bits, cada um criptografado de forma separada pela mesma chave. As cifras de fluxo criptografam o texto bit a bit em um fluxo contínuo.

O modelo de criptografia assimétrica foi criado na década de 1970 - pelo matemático Clifford Cocks que trabalhava no serviço secreto inglês, o GCHQ, nele duas chaves diferentes são utilizadas, uma pública e a outra privada, formulando o que no futuro como o algoritmo RSA, que seria redescoberto e quatro anos depois por Riverst, Shamir e Adleman (SINGH, 2000). As operações realizadas na categoria assimétrica são: criptografia, na assimétrica o remetente utiliza a chave pública do destinatário para criptografar a mensagem, e o destinatário utiliza de sua chave privada para descriptografar a mensagem. Assinaturas digitais, em que o remetente utiliza de sua chave privada para assinar uma mensagem, e a chave pública do

remetente é utilizada para verificar se a assinatura é válida. Troca de chaves segura, nela, ambos os participantes trocam valores públicos e usam suas chaves privadas para obter o segredo compartilhado.

Existem três famílias de algoritmos assimétricos com relevância prática:

- Esquemas de Fatoração de Inteiros: Baseados na dificuldade de fatorar números inteiros grandes. Exemplo de algoritmo: RSA
- Esquemas de Logaritmo Discreto: Baseados no problema do logaritmo discreto em corpos finitos. Exemplo de Algoritmos: Diffie-Hellman, a criptografia Elgamal e o Algoritmo de Assinatura Digital (DSA).
- Esquemas de Curva Elíptica (EC): Uma generalização do algoritmo de logaritmo discreto são os esquemas de chave pública de curva elíptica. Exemplo de Algoritmos: Diffie-Hellman de Curva Elíptica (ECDH) e o Algoritmo de Assinatura Digital de Curva Elíptica (ECDSA)(PAAR; PELZL; GÜNEYSU, 2024).

A principal vantagem desse método é sua segurança, entretanto o tempo de processamento e consumo de recursos computacionais tende ser maior em comparação a criptografia simétrica, limitando o seu uso. Mesmo assim, ele ainda é utilizado certificados e assinaturas digitais, e troca de chaves seguras em sistemas de criptografia híbrida.

Os protocolos criptográficos, como caracterizados por (PAAR; PELZL; GÜNEYSU, 2024, p.4, tradução nossa)

> Em termos gerais, os protocolos criptográficos realizam funções de segurança mais complexas por meio do uso de algoritmos criptográficos. Algoritmos simétricos e assimétricos podem ser vistos como blocos de construção com os quais aplicações como a comunicação segura na internet podem ser realizadas.

Esses protocolos, como o TLS, IPsec e outros, são a forma em que a maior parte das comunicações entre dispositivos e transferências de dados são feitas, assegurando a segurança dos dados em trânsito.

3.3 ALGORITMOS CRIPTOGRÁFICOS MAIS UTILIZADOS

3.3.1 Advanced Encryption Standard (AES)

Criptografia simétrica e Cifra de bloco, o Rijndael foi desenvolvido em 1998 pelos belgas Joan Daemen e Vicent Rijmen como parte de uma competição para a proposta de um novo padrão para cifra de bloco considerando a quebra do DES, e foi escolhido como AES em 2001. Daemen e Rijmen (2003) explicam que no Rijndael, o tamanho de bloco e da chave poderiam ser escolhidos dentro de um conjunto de múltiplos de 32 bits, com o tamanho mínimo de 128 bits e máximo de 256, entretanto, no processo de ser publicado pelo FIPS e ter se tornado o AES, isso foi mudado, o tamanho de bloco utilizado pelo AES é fixo em 128 bits e suporta três diferentes tamanhos de chave 128,192 e 256 bits.

Os algoritmos criptográficos finalistas na competição para a escolha do AES foram, além do Rijndael, MARS, RC6, Serpent e TwoFish. Segundo Nechvatal et al. (2001), as razões citadas pela escolha do Rijndael foram sua boa performance em tanto hardware quanto software, simplicidade em relação aos outros algoritmos da competição, baixo custo de memória e flexibilidade, além de sua segurança contra diversas formas de ataque.

O AES, em alto nível, funciona como uma rede de substituição-permutação, em que o texto de entrada passa por várias rodadas de transformação,em cada uma dessas rodadas o AES utiliza diferentes chaves de 128 bits que são derivadas da chave principal e cada have da rodada é separada em colunas de 4 bytes. Cada rodada consiste em operações como:

- SubBytes - a caixa de substituição(S-box) é aplicada em cada byte do estado do AES, a S-box substitui um byte por outro em uma transformação não linear
- ShiftRows,- as três últimas linhas são deslocadas, a posição de seus bytes rotacionada para esquerda
- MixColumns- multiplica um polinômio fixo a cada coluna do estado
- AddRoundKey - junta os bytes da chave da rodada com o do estado utilizando de XOR bit a bit.

Além disso, o AES pode ser consideravelmente acelerado com o uso de instruções nativas de hardware, conhecidas como AES-NI (AES Native Instructions). Essas instruções foram introduzidas inicialmente pela Intel em 2008 e atualmente estão disponíveis na maioria dos processadores modernos, incluindo arquiteturas Intel, AMD e ARMv8. Em vez de codificar cada rodada do AES com operações básicas como XORs e multiplicações, o uso de AES-NI permite que o desenvolvedor

simplesmente invoque instruções específicas, como AESENC, que são executadas diretamente pelo processador (AUMASSON, 2025)

Essa aceleração é crucial, pois melhora tanto o desempenho quanto a segurança, minimizando o risco de ataques baseados no tempo (timing attacks) e no consumo de energia (power analysis attacks), já que a execução se torna mais previsível e rápida (INTEL, 2012)

Devido a esses avanços e a sua eficiência e segurança, até hoje o AES permanece um algoritmo confiável e seguro, e um dos de criptografia simétrica mais utilizados no mundo.

3.3.2 Salsa20 e ChaCha

As cifras de fluxo têm ganhado cada vez mais adesão, como Stallings (2023, p.260) justifica, por serem úteis em situações em que os dispositivos têm memória e poder de processamento limitado, os chamados dispositivos restritos, como os IoT(*Internet of Things*, Internet das Coisas) ou em casos em que há necessidade de criptografar grandes quantidades de dados em fluxo rápido. Entre os algoritmos modernos desse tipo, destacam-se o Salsa20 e sua variação mais avançada, o ChaCha. Ambos foram projetados por Daniel J. Bernstein com o objetivo de fornecer uma alternativa mais segura, eficiente e confiável a cifras anteriores como o RC4, que se tornaram obsoletas devido a diversas vulnerabilidades.

O Salsa20 foi apresentado em 2005 como parte da competição eSTREAM, organizada pelo projeto europeu eCRYPT. Como o próprio criador da cifra descreve a versão padrão, Salsa20/20, A cifra de fluxo Salsa20/20 expande uma chave de 256 bits em 264 fluxos acessíveis aleatoriamente, cada um contendo 264 blocos de 64 bytes acessíveis aleatoriamente. Salsa20/20 é um design mais conservador do que o AES, e a comunidade parece ter rapidamente adquirido confiança na segurança da cifra (BERNSTEIN, 2008, p. 1).

Sua estrutura baseia-se em uma função de mistura que opera sobre um estado de 64 bytes, composto por uma chave secreta, um nonce (valor único por mensagem) e um contador. O algoritmo utiliza apenas três operações simples: soma modular, XOR e rotações circulares, todas com baixo custo computacional e altamente eficientes em diversas arquiteturas. A versão padrão, Salsa20/20, executa 20 rodadas de transformação sobre o estado e gera blocos de 64 bytes de keystream, que são combinados com o texto simples via operação XOR para produzir o texto cifrado.

A partir do sucesso do Salsa20, Bernstein propôs, em 2008, uma variante chamada ChaCha, que mantém os princípios de design do Salsa20, porém sua maior experiência com o algoritmo permitiu que (BERNSTEIN, 2008, p. 1), em suas próprias palavras, fizesse o ChaCha projetado para melhorar a difusão por rodada, aumentando conjunturalmente a resistência à criptoanálise, ao mesmo tempo que preserva — e frequentemente melhora — o tempo por rodada. Assim como no caso do Salsa20, a versão mais utilizada atualmente é a ChaCha20, que executa 20 rodadas e mantém o mesmo formato de entrada e tamanho de bloco. Atualmente, é utilizado em protocolos modernos como TLS 1.3, SSH, WireGuard, QUIC e com suporte a ele em telefones Android e sistemas operacionais Apple.

Embora ambos os algoritmos tenham se mostrado seguros até o momento, foi o ChaCha, mais especificamente o ChaCha20, por sua superioridade em relação ao Salsa20 em questão de rapidez e segurança, que se consolidou como a alternativa preferida em aplicações que demandam rapidez, portabilidade e segurança robusta, especialmente quando o AES não é ideal por limitações de hardware.

3.3.3 RSA

O algoritmo RSA é de criptografia de chave pública/assimétrico, criado em 1977 e nomeado seguindo as iniciais dos sobrenomes dos seus criadores, Ron Rivest, Adi Shamir e Len Adleman. A sua segurança depende da dificuldade de fatoração de inteiros, mais especificamente de números semiprimos, já que até hoje nenhum algoritmo de fatoração de inteiros em tempo polinomial foi encontrado, nem sequer foi comprovado se é possível ou não a sua existência em computadores não quânticos. O algoritmo funciona da seguinte maneira, dois números primos grandes são escolhidos, o produto desses dois primos, n, será parte das chaves pública e privada, em seguida, o totiente de Euler, $\varphi(n)=(p-1)(q-1)$, é calculado, o totiente é o número de inteiros menores que n que são coprimos a ele, ou seja, não tem divisores em comum além de 1. Utilizando o totiente e n, são gerados os outros números que irão formar pares com n para compor as chaves pública e privada

A segurança do RSA reside na dificuldade de fatorar semiprimos. Mesmo com o avanço da computação, nenhum algoritmo clássico (não quântico) é capaz de fatorar tais números em tempo polinomial. No entanto, a computação quântica, com algoritmos como o de Shor, representa uma potencial ameaça futura à segurança do RSA, o que tem incentivado o desenvolvimento de criptografia pós-quântica.

O RSA é utilizado tanto para criptografar dados, realização de assinaturas digitais e troca de chaves. De ampla utilização em ambientes de rede, o RSA é principalmente usado para o gerenciamento de chaves em sistemas de criptografia híbrida, auxiliando e protegendo a troca de chaves simétricas que serão utilizadas na comunicação, e em assinaturas digitais. É menos utilizado para diretamente criptografar dados, já que é menos eficiente em tempo e custo que as alternativas de criptografia simétrica, como AES ou ChaCha20.

3.3.4 Digital Signature Algorithm (DSA)

O DSA é um algoritmo assimétrico utilizado para a geração e verificação de assinaturas digitais, promovendo pilares da segurança digital como a integridade, autenticidade e o não repúdio. Ele foi proposto em 1991 pela NSA (National Security Agency) dos Estados Unidos e padronizado pelo NIST (National Institute of Standards and Technology) como parte do Digital Signature Standard (DSS), sob a especificação FIPS PUB 186.

Diferente do RSA, que pode exercer funções de assinatura digital, criptografar dados e troca de chaves, o DSA exclusivamente lida com a assinatura digital.

O funcionamento do algoritmo pode ser dividido em três partes: geração de chaves, assinatura e verificação. Um par de chaves é gerado, a chave privada com a função de assinar as mensagens e a pública para a verificação das assinaturas. Para assinar a mensagem, o remetente primeiro aplica uma função de hash nela, gerando um resumo criptográfico e, com base nesse resumo, um valor aleatório k e a chave privada, são calculados os valores r e s, que formam a assinatura digital. Na verificação da assinatura, o receptor calcula o hash da mensagem recebida e o utiliza juto aos valores r e s, e a chave pública do remetente para realizar cálculos, que se coincidirem com o valor esperado (r), a assinatura é considerada válida, comprovado que a mensagem não foi alterada e que o remetente é quem diz ser.

O DSA, como a maior parte dos outros algoritmos criptográficos baseados no logaritmo discreto, possui uma variante de curva elíptica, o ECDSA (Elliptic Curve DSA), que é mais utilizada que a versão clássica, já que possui o mesmo nível de segurança com chaves menores e maior eficiência computacional, ideal para dispositivos móveis ou com baixa banda larga.

Apesar de ser seguro se implementado corretamente, ele facilmente pode ser quebrado com pequenas falhas na implementação. Uma característica crítica que se

destaca é a importância da escolha do valor aleatório k, que deve seguir os requerimentos de ser único, secreto e imprevisível. Caso alguma dessas propriedades não seja cumprida, como em implementações que o valor é reutilizado ou foi comprometido, a chave privada é facilmente descoberta, como já ocorrido em casos práticos como em 2010, em que a sony utilizava um valor estático de k na sua implementação do ECDSA que era usada na assinatura de software do PS3 e consequentemente teve sua chave privada obtida por hackers, ganhando acesso irrestrito, como relatado por Bendel (2010).

O DSA continua sendo suportado por vários padrões de segurança e protocolos, como o SSL/TLS, SSH e PGP, embora, na prática, RSA e ECDSA sejam mais utilizados devido à sua maior flexibilidade e desempenho.

3.3.5 Diffie-Hellman

Proposto em 1976 por Whitfield Diffie e Martin Hellman, este algoritmo marcou a história da criptografia ao ser o primeiro trabalho publicado de um algoritmo criptográfico que faz utilização de duas chaves, uma pública e outra privada, introduzindo ao mundo um novo tipo de cifra, a assimétrica. Além disso, é também um dos primeiros protocolos de troca de chaves, permitindo retificar o maior problema da criptografia simétrica, a troca segura da chave compartilhada.

Diffie-Hellman se baseia em conceitos da aritmética modular e na dificuldade computacional do problema do logaritmo discreto. O algoritmo se inicia com dois números públicos: um primo grande p e um número g, ambos conhecidos por todas as partes. Cada participante escolhe um número secreto e calcula um valor público com base em módulos.

Com esses valores trocados, ambos conseguem calcular a mesma chave compartilhada secreta, mas sem que seus valores secretos sejam revelados. Graças às propriedades da exponenciação modular, o resultado de ambos é o mesmo. Essa chave pode então ser usada para a comunicação usando criptografia simétrica.

Entretanto, o protocolo em sua forma básica é vulnerável a ataques do tipo "man-in-the-middle", definido por (STALLINGS, 2023) como um tipo de ataque de escuta(wiretapping) no qual o invasor intercepta e modifica seletivamente os dados comunicados para se disfarçar como uma ou mais das entidades envolvidas em uma

comunicação. Para mitigar esse risco, o Diffie-Hellman é usado em protocolos junto a métodos de autenticação como MACs ou assinaturas digitais.

Assim como o DSA, esse algoritmo também possui uma versão baseada em curvas elípticas, o ECDH (Elliptic Curve Diffie-Hellman), que é a sua versão mais utilizada por proporcionar os benefícios citados anteriormente de maior eficiência e precisar de chaves menores.

4. APLICAÇÃO DA CRIPTOGRAFIA EM DADOS EM TRÂNSITO

Os algoritmos criptográficos, como os citados anteriormente, quando utilizados na prática na criptografia em trânsito são implementados dentro de protocolos, definidos por (STALLINGS, 2023, p.36, tradução nossa) como

> o formato e procedimentos que regem a transmissão e o recebimento de dados entre pontos em uma rede. Um protocolo define a estrutura das unidades de dados individuais (por exemplo, pacotes) e os comandos de controle que gerenciam a transferência de dados.

Neles, uma combinação de criptografia simétrica, assimétricas são utilizados em etapas predefinidas como troca de chaves, criptografia de dados, autenticação e troca de mensagens. Esses protocolos regulamentam e asseguram a proteção da transmissão de dados, permitindo que haja privacidade e segurança fundamentais para a comunicação segura entre sistemas e dispositivos. Com o exponencial aumento do uso e importância das redes na atualidade, a importância desses protocolos aumenta proporcionalmente, responsáveis por proteger os mais diversos tipos de dados em trânsito, de transações bancárias a aplicativos de mensagem, em dispositivos igualmente variados, dos computadores aos dispositivos IoT.

4.1 PRINCIPAIS PROTOCOLOS

4.1.1 *TLS (Transport Layer Security)*

O TLS é um dos protocolos criptográficos mais utilizados atualmente para proteger conexões em redes TCP/IP. Ele é sucessor do SSL (Secure Sockets Layer) e é usado em aplicações como HTTPS, e-mail (SMTP, IMAP, POP3), VPNs e muito mais. Como descrito por Stallings (2020, p.536, tradução nossa) há duas opções de implementação do TLS:

> Para total generalidade, o TLS poderia ser fornecido como parte do conjunto de protocolos subjacente e, portanto, ser transparente para os aplicativos. Alternativamente, o TLS pode ser incorporado em pacotes específicos. Por exemplo, a maioria dos navegadores vem equipada com TLS, e a maioria dos servidores Web implementou o protocolo.

O TLS proporciona confidencialidade por meio de criptografia simétrica (após uma troca de chaves segura), integridade com códigos de autenticação de mensagem (MACs) e autenticação opcional por meio de certificados digitais.

Durante o handshake TLS, algoritmos assimétricos como RSA ou ECDHE são usados para estabelecer uma chave de sessão simétrica, geralmente usada com algoritmos como AES ou ChaCha20 para criptografar os dados. Desde a versão TLS 1.3, melhorias de desempenho e segurança foram introduzidas, incluindo a eliminação de algoritmos obsoletos e a redução do número de mensagens no handshake.

4.1.2 *IPsec (Internet Protocol Security)*

O IPsec é um conjunto de protocolos desenvolvido para proteger comunicações em nível de rede (camada 3 do modelo OSI). Exemplos de seu uso incluem: Conectividade segura de filiais pela Internet, acesso remoto seguro pela Internet, estabelecimento de conectividade de extranet e intranet com parceiros e aprimoramento da segurança do comércio eletrônico (STALLINGS, 2020). Ele é amplamente utilizado em VPNs (redes privadas virtuais) para criar túneis seguros entre dois pontos na Internet. O IPsec pode operar em dois modos: modo transporte, que protege apenas o payload do pacote IP, e modo túnel, que encapsula todo o pacote original, adicionando uma nova camada de segurança.

O IPsec utiliza criptografia simétrica (como AES) e troca de chaves baseada em Diffie-Hellman ou IKE (Internet Key Exchange), garantindo tanto a confidencialidade quanto a integridade dos dados.

4.1.3 *SSH (Secure Shell)*

O protocolo SSH é amplamente utilizado para acesso remoto seguro a sistemas, transferência de arquivos (SCP, SFTP) e tunelamento de tráfego. O SSH é organizado em três protocolos que normalmente são executados sobre o TCP: Protocolo da Camada de Transporte, Protocolo de Autenticação do Usuário e Protocolo de Conexão (STALLINGS, 2020).

Ele proporciona uma comunicação criptografada ponto a ponto, combinando algoritmos assimétricos para autenticação inicial com criptografia simétrica para a sessão. O SSH também oferece autenticação por senha ou por chaves públicas, sendo essa última preferível por razões de segurança.

5. Comparação de Algoritmos de Criptografia na Prática

5.1 Comparação entre os algoritmos em dispositivos IoT

A Tabela 1 mostra os resultados obtidos pelo monitor de energia (Power Monitor) para cada um dos algoritmos AES e ChaCha20, comparando os seus diferentes níveis de consumo de energia.

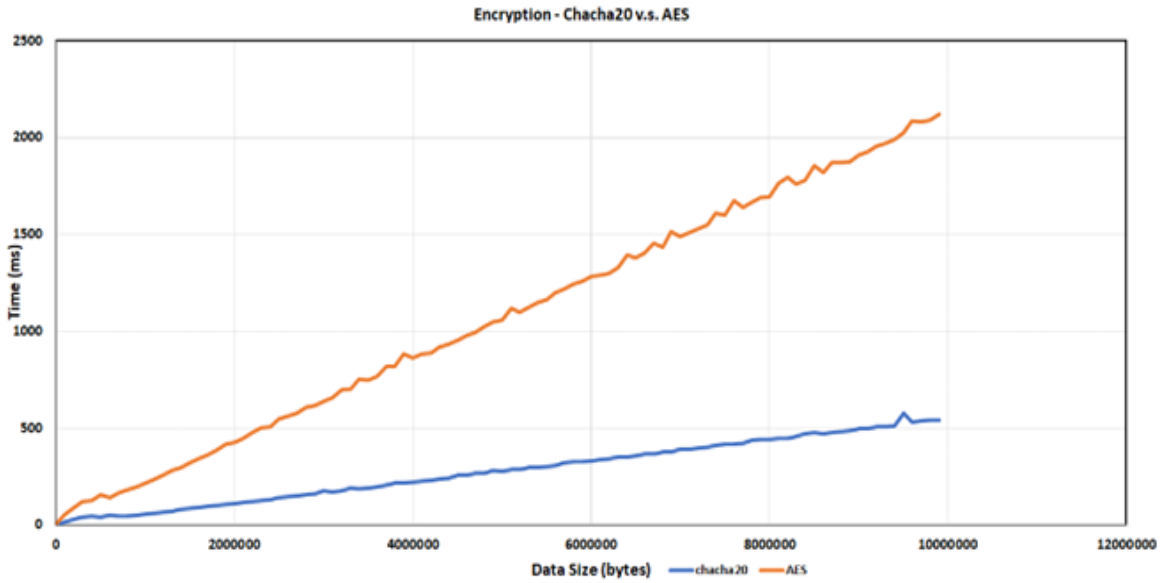
Tabela 1 - comparação entre os algoritmos de criptografia em dispositivos IoT com relação à eficiência energética

Algoritmo	Tensão (V)	Corrente (A)	Potência (W)	Consumo de Corrente (mA)
Ocioso	5.19	0.40	2.08	388
Sem Criptografia	5.21	0.48	2.49	475
ChaCha20	5.21	0.51	2.71	783
AES	5.21	0.50	2.61	757
Twofish	5.21	0.49	2.55	553
RSA	5.21	0.49	2.57	620

Fonte: Anaya et al. (2020)

Na Figura 1, o tempo de processamento da encriptação dos algoritmos AES e ChaCha20 são postos em função do tamanho dos arquivos a serem encriptados e comparados entre si.

Figura 1 – Tempo de Encriptação vs Tamanho do arquivo



Fonte: Anaya et al. (2020)

Anaya et al. (2020) analisam o desempenho de algoritmos criptográficos aplicados a dispositivos de IoT, destacando os compromissos entre segurança e consumo de recursos. Com base nas tabelas e gráficos apresentados no artigo e representados acima, o ChaCha20 foi considerado o algoritmo mais adequado para esse tipo de dispositivo limitado, por ser mais rápido que o AES em equipamentos que não possuem aceleração de hardware.

Comparado aos algoritmos RSA e Twofish, que são *data-dependent* (dependentes dos dados), o ChaCha20 também se mostra mais seguro contra *timing attacks* (ataques de temporização) e de canal lateral, que exploram variações no tempo de execução ou no consumo de energia para tentar descobrir a chave secreta. Apesar disso, em termos de eficiência energética, o algoritmo Twofish apresentou superioridade no cenário avaliado.

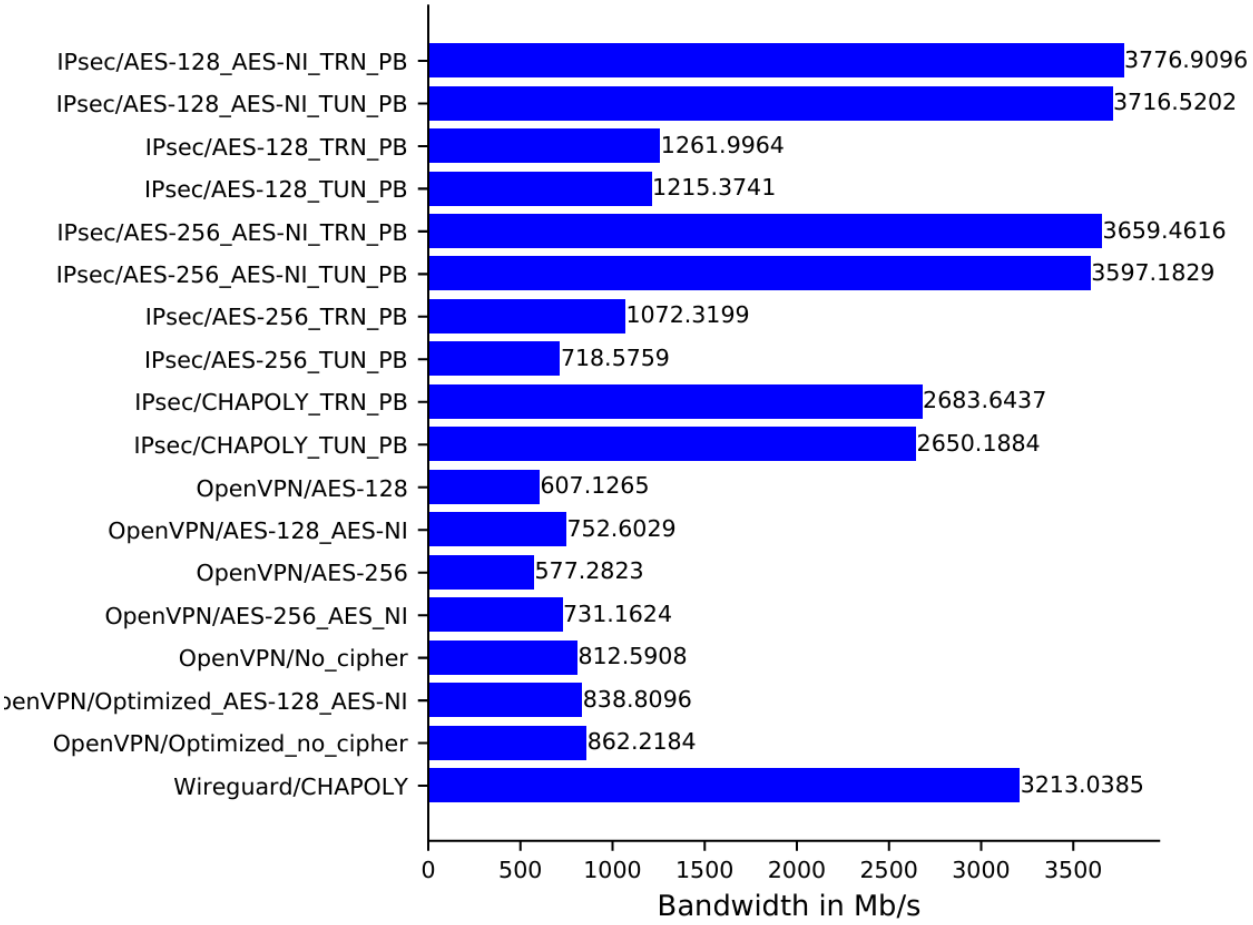
5.2 Comparação entre os algoritmos AES e ChaCha20 em diferentes protocolos VPN

A Figura 2 apresenta como alguns dos protocolos mais utilizados para VPNs (IPsec, OpenVPN, Wireguard), se comportam em relação a sua banda larga média quando utilizam do AES, com e sem desaceleração, e quando utilizam ChaCha20.

Segundo Osswald, Haeberle e Menth (2023), diferentes soluções de VPN apresentam variações significativas de desempenho, especialmente em termos de

latência e consumo de CPU. Como no gráfico acima aponta, em seu artigo foi concluído que nas condições testadas, o protocolo IPsec se destaca em questão de performance, em especial quando utilizando algoritmos baseados em AES com aceleração, mas sem as instruções AES-NI, algoritmos baseados em ChaCha20 tanto em Wireguard quanto no IPsec performam melhor.

Figura 2 – Diferentes protocolos e configurações vs Largura de Banda



Fonte: Osswald, Haeberle e Menth (2023)

5.3 Comparação de assinaturas digitais

Ahmed e Ahmed (2022) realizaram uma análise comparativa sistemática entre algoritmos criptográficos, considerando seu desempenho em diferentes contextos de comunicação. Nessa tabela de comparação entre algoritmos de assinatura digitais, o DSA se destaca tanto em seu tempo de execução total, quanto em tempo de execução da assinatura (Tabela 2).

Tabela 2 - Comparação entre diferentes tempos de execução de algoritmos de assinatura digital

Nome do Algoritmo	Assinatura: Tempo de Execução (média de 10 rodadas)	Verificação: Tempo de Execução (média de 10 rodadas)	Tempo Total de Execução
RSA	4.90917 s	0.01136 s	4.92053 s
DSA	0.00143 s	0.00536 s	0.00679 s
ElGamal	0.00347 s	0.00426 s	0.00773 s
Schnorr multi-sig	0.33162 s	0.47101 s	0.80721 s

Fonte: Ahmed e Ahmed (2022)

6. CONSIDERAÇÕES FINAIS

Como todos os outros campos da computação, a criptografia está em um estado de constante evolução e mudanças, e algumas tendências e futuras ameaças já começaram a moldar o seu desenvolvimento futuro.

Neste trabalho, foi analisado o estado atual da criptografia em trânsito, e feito um estudo bibliográfico comparativo entre diferentes algoritmos utilizados. observado Dentro da criptografia simétrica, o algoritmo ChaCha20 demonstra superioridade em performance e se destaca como escolha em dispositivos limitados que não conseguem rodar as instruções AES-NI para aceleração do AES, mas em dispositivos que possuem essas instruções o AES é superior. Já nos algoritmos de assinatura digital, o DSA se provou a escolha superior em questão de tempo de execução.

Uma das atuais preocupações para a criptografia é o desenvolvimento dos computadores quânticos. Apesar de atualmente os computadores quânticos existentes serem limitados e não apresentarem ameaças, seu avanço pode tornar complexo esse cenário. Por exemplo, algoritmos como o de Shor, que trivializa os problemas matemáticos que o RSA e ECC dependem para funcionar, ou como o de Grover, que como (TAMBE-JAGTAP, 2023) explica, reduz o tempo de verificação das chaves. Na prática, isso significa que uma chave de 128 bits considerada segura hoje forneceria segurança comparável a uma chave de 64 bits no mundo quântico.

Portanto, consideramos que a área de criptografia é relevante e apresenta desafios importantes para que possa garantir a segurança da informação em um contexto de um mundo cada vez mais conectado.

REFERÊNCIAS

AHMED, Shahzad; AHMED, Tauseef. **Comparative analysis of cryptographic algorithms in context of communication: a systematic review.** *International Journal of Scientific and Research Publications*, v. 12, n. 7, p. 165-173, jul. 2022. DOI: 10.29322/IJSRP.12.07.2022.p12720. Disponível em: http://dx.doi.org/10.29322/IJSRP.12.07.2022.p12720. Acesso em: 04 maio 2025.

ANAYA, J.; PATEL, J.; SHAH, P.; SHAH, V.; CHENG, Y. **A performance study on cryptographic algorithms for IoT devices.** In: *ACM Conference on Data and Application Security and Privacy*, 10., 2020. Proceedings... [S.l.]: ACM, 2020. Disponível em: https://doi.org/10.1145/3374664.3379531. Acesso em: 30 abril 2025.

AUMASSON, Jean-Philippe. **Serious cryptography: a practical introduction to modern encryption.** San Francisco: *No Starch Press, Inc*, 2024.

BENDEL, Mike. **Hackers describe PS3 security as "epic fail," gain unrestricted access.** *Exophase*, 29 dez. 2010. Disponível em: https://www.exophase.com/20540/hackers-describe-ps3-security-as-epic-fail-gain-unrestricted-access/. Acesso em: 21 abril 2025.

BERNSTEIN, Daniel J. **ChaCha, a variant of Salsa20.** Chicago: University of Illinois at Chicago, 2008. Disponível em: https://cr.yp.to/chacha/chacha-20080128.pdf. Acesso em: 15 abril 2025.

DAEMEN, Joan; RIJMEN, Vincent. **The Rijndael Block Cipher.** Gaithersburg, MD: National Institute of Standards and Technology, 2003. Disponível em: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf. Acesso em: 16 abril 2025.

KATZ, Jonathan; LINDELL, Yehuda. **Introduction to modern cryptography: principles and protocols.** 3. ed. Boca Raton: *CRC Press*, 2021. ISBN 978-0-8153-5436-9 (impresso); 978-1-351-13303-6 (e-book).

MUNIZ, Antonio et al. (Curadores). **Jornada Segurança da Informação: unindo visão executiva e técnica para estratégia, comportamento, inovação e tendências.** Rio de Janeiro: *Brasport*, 2024. Edição Kindle. ISBN 978-65-6096-009-1.

NECHVATAL, James; BARKER, Elaine; BASSHAM, Lawrence; BURR, William; DWORKIN, Morris; FOTI, James; ROBACK, Edward. **Report on the development of the Advanced Encryption Standard (AES).** *Journal of Research of the National Institute of Standards and Technology*, Gaithersburg, MD, v. 106, n. 3, p. 511-577, maio/jun. 2001. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4863838/. Acesso em: 16 abril 2025.

OLIVEIRA, Ronielton Rezende. **Criptografia simétrica e assimétrica-os principais algoritmos de cifragem.** *Segurança Digital* [revista online], v. 31, p. 11-15, 2012.

OSSWALD, Lukas; HAEBERLE, Marco; MENTH, Michael. **Performance comparison of VPN solutions.** [S.l.]: University of Tübingen, 2023. Disponível em: http://dx.doi.org/10.15496/publikation-41810. Acesso em: 4 maio 2025.

PAAR, Christof; PELZL, Jan; GÜNEYSU, Tim. **Understanding cryptography: from established symmetric and asymmetric ciphers to post-quantum algorithms.** 2. ed. Berlin: *Springer-Verlag*, 2024. ISBN 978-3-662-69006-2. DOI: https://doi.org/10.1007/978-3-662-69007-9.

ROTT, Jeffrey Keith. **Intel® Advanced Encryption Standard Instructions (AES-NI).** *Intel Developer Zone*, 2 fev. 2012. Disponível em: https://www.intel.com/content/www/us/en/developer/articles/technical/advanced-encryption-standard-instructions-aes-ni.html?wapkw=aes-ni. Acesso em: 4 maio 2025.

SIDHPURWALA, Huzaifa. **Uma breve história da criptografia.** *Red Hat*, 12 jan. 2023. Disponível em: https://www.redhat.com/pt-br/blog/brief-history-cryptography. Acesso em: 04 maio 2025.

SINGH, Simon. **The code book: the science of secrecy from ancient Egypt to quantum cryptography.** New York: *Anchor Books*, 2000.

STALLINGS, William. **Criptografia e segurança de redes: princípios e práticas.** 6. ed. Tradução: Daniel Vieira. Revisão técnica: Paulo Sérgio Licciardi Messeder Barreto; Rafael Misoczki. São Paulo: *Pearson Education do Brasil*, 2015. ISBN 978-85-430-1450-0.

STALLINGS, William. **Cryptography and network security: principles and practice.** 8. ed. Global edition. Harlow, United Kingdom: *Pearson Education Limited*, 2023. ISBN 978-1-292-43748-4.

TAMBE-JAGTAP, Swapnali N. **A survey of cryptographic algorithms in cybersecurity: from classical methods to quantum-resistant solutions.** *SHIFRA*, [S.l.], v. 2023, p. 43-52, 2023. ISSN 3078-3186. Disponível em: https://peninsula-press.ae/Journals/index.php/SHIFRA/article/view/30/124. Acesso em: 5 abril 2025.