



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e de Informática

Análise Comparativa de Algoritmos de Hashing na Proteção de Senhas em Sistemas de Informação*

Gabriella Dantas de Abreu Fandim¹

Pedro Siqueira Pereira Bitarães²

Luciana Mara Freitas Diniz³

Resumo

A segurança de senhas constitui aspecto central em Sistemas de Informação, dado o cenário atual de crescente exposição de dados sensíveis a ataques cibernéticos. O problema que se apresenta está na escolha de algoritmos de *hashing* capazes de oferecer proteção adequada diante da sofisticação das técnicas de invasão. Justifica-se este estudo pela relevância de subsidiar organizações e profissionais de tecnologia com evidências práticas para a adoção de soluções mais seguras e em conformidade com legislações como a LGPD e normas como a ISO/IEC 27001. Portanto, o objetivo dessa pesquisa consiste em analisar e comparar algoritmos de *hashing*, como Argon2, bcrypt, SHA-256, identificando aqueles que conciliam robustez criptográfica e eficiência computacional. O desenvolvimento da pesquisa ocorreu por meio de experimentos controlados que avaliaram o desempenho dos algoritmos em cenários de autenticação simulados, utilizando ferramentas especializadas e técnicas complementares como *salting* e múltiplas iterações. Os resultados apontaram diferenças quanto ao tempo de processamento, uso de recursos computacionais e resistência a ataques de força bruta e de dicionário, demonstrando que algoritmos modernos, como bcrypt e Argon2, apresentam maior resiliência quando comparados a algoritmos criptográficos mais simples e menos robustos, como o SHA-256. Conclui-se que a adoção criteriosa desses algoritmos fortalece as práticas de segurança digital, contribuindo para a construção de sistemas de informação mais confiáveis e resistentes a violações.

Palavras-chave: segurança da informação; hashing; senhas; ataques cibernéticos; algoritmos criptográficos.

*Artigo apresentado ao Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas Gerais, campus Contagem, como pré-requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

¹ Aluno(a) do Programa de Graduação em Sistemas de Informação – gabriellaxdantas@gmail.com.

² Aluno(a) do Programa de Graduação em Sistemas de Informação – pedrosiqueirapb@gmail.com.

³ Orientador(a) do Programa de Graduação em Sistemas de Informação – lucianadiniz@pucminas.br.

Abstract

Password security is a central aspect of Information Systems, especially in a context of increasing exposure of sensitive data to cyberattacks. The main challenge lies in selecting hashing algorithms capable of providing adequate protection against increasingly sophisticated intrusion techniques. This study is justified by the need to support organizations and technology professionals with practical evidence for adopting more secure solutions aligned with regulations such as the LGPD and standards such as ISO/IEC 27001. Therefore, the objective of this research is to analyze and compare hashing algorithms such as Argon2, bcrypt, and SHA-256, identifying those that best combine cryptographic robustness and computational efficiency. The study was conducted through controlled experiments that evaluated the performance of the algorithms in simulated authentication scenarios, using specialized tools and complementary techniques such as salting and multiple iterations. The results revealed differences in processing time, resource consumption, and resistance to brute-force and dictionary attacks, showing that modern algorithms such as bcrypt and Argon2 offer greater resilience when compared to simpler and less robust cryptographic functions such as SHA-256. It is concluded that the careful adoption of these algorithms strengthens digital security practices, contributing to the development of more reliable and attack-resistant information systems.

Keywords: information security; hashing; passwords; cyber attacks; cryptographic algorithms.

1 INTRODUÇÃO

A segurança da informação é um tema central no cenário digital, no qual o volume crescente de dados sensíveis exige mecanismos robustos de proteção. Sistemas de autenticação baseados em senha permanecem como a principal mecanismo de defesa, e sua eficácia depende da aplicação de algoritmos de *hashing* para assegurar que credenciais não sejam armazenadas em formato simples (Sheketa et al., 2021). Esta é uma prática fundamental reforçada por legislações como a Lei Geral de Proteção de Dados (LGPD) e normas internacionais, como a ISO/IEC 27001 e 27002.

Neste sentido, o problema que motiva este estudo consiste em responder quais algoritmos de *hashing* oferecem o melhor equilíbrio entre eficiência computacional e robustez contra ataques cibernéticos. Esta questão surge da obsolescência de funções tradicionais (MD5, SHA-1), que se tornaram vulneráveis a colisões e ataques de força bruta, exigindo a análise de alternativas modernas como Argon2, bcrypt e SHA-256 (Sheketa et al., 2021). Assim, a justificativa desta pesquisa compreende a necessidade de fornecer evidências práticas para profissionais de tecnologia, visto que uma escolha equivocada de algoritmo pode comprometer a integridade de sistemas inteiros e expor dados críticos a riscos elevados.

O objetivo geral deste trabalho é analisar e comparar experimentalmente o desempenho dos algoritmos Argon2, bcrypt e SHA-256 na proteção de senhas em cenários de vulnerabilidade. Para atingir esta meta, o estudo adota uma abordagem experimental com os seguintes objetivos específicos: (i) realizar testes de desempenho (tempo de execução e uso de CPU/memória) e de resistência (ataques de força bruta e dicionário); (ii) analisar o impacto de técnicas como *salting* e múltiplas iterações; e (iii) propor critérios para a seleção de algoritmos, com base no equilíbrio entre desempenho e resistência, adequados a diferentes contextos de sistemas de informação.

O trabalho está estruturado da seguinte forma: a Seção 2 apresenta o referencial teórico sobre autenticação e funções de hash. A Seção 3 discute os trabalhos relacionados, comparando estudos sobre algoritmos modernos e adaptativos (Sheketa et al., 2021; Rohit et al., 2019; Bachtiar et al., 2018; Skanda; Srivatsa; Premananda, 2022; Bai; Blocki, 2021). A Seção 4 detalha os materiais e métodos do experimento, seguida pela Seção 5, que apresenta e discute os resultados obtidos. Por fim, a Seção 6 reúne as conclusões da pesquisa e seus trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Segurança da Informação

A segurança da informação constitui-se como a disciplina que visa proteger os ativos de informação de uma organização contra uma vasta gama de ameaças, a fim de garantir a

continuidade dos negócios, minimizar os riscos e maximizar o retorno sobre os investimentos. Em um cenário onde os dados são considerados um dos ativos mais valiosos, a sua proteção torna-se um pilar estratégico para a sobrevivência e a competitividade (Cunha, 2021).

2.1.1 Autenticação em Sistemas de Informação

A autenticação é o processo que verifica a identidade de um usuário, sendo a senha o método mais primário e difundido. Apesar do surgimento de alternativas, as senhas continuam sendo a primeira linha de defesa contra acessos não autorizados na maioria dos sistemas digitais (Islam et al., 2023). A eficácia dessa defesa, contudo, é desafiada pelo comportamento do usuário. A necessidade de gerenciar múltiplas contas leva à "fadiga de senhas", incentivando a reutilização de credenciais e a escolha de senhas fracas, o que representa um risco significativo (Pal et al., 2019). Um estudo abrangente sobre senhas vazadas, analisando mais de 3,9 bilhões de contas, confirma que os usuários seguem padrões previsíveis, o que facilita a adivinhação por parte de atacantes (Kanta et al., 2021).

2.1.2 Vulnerabilidades em Esquemas de Senhas

A segurança dos esquemas de autenticação baseados em senha é constantemente testada por diversos vetores de ataque e a compreensão de técnicas é importante para o desenvolvimento de defesas robustas. Nesse viés, os ataques de força bruta e os dicionário são métodos que consistem em testar sistematicamente combinações de caracteres (força bruta) ou uma lista de senhas prováveis (dicionário) até encontrar a credencial correta. A eficácia desses ataques é diretamente proporcional à complexidade e ao comprimento da senha alvo (Nyangaresi; Abeka; Rodrigues, 2020).

Além disso, um estudo com trinta métodos de adivinhação de senhas publicados entre 2016 e 2023, classificados em duas categorias: adivinhação de arrasto e adivinhação direcionada, na qual a primeira busca explorar padrões gerais de criação de senhas, enquanto a segunda utiliza informações específicas dos usuários para aumentar as chances de sucesso. Nesse estudo, é evidente que o uso combinado de métodos tradicionais com técnicas de aprendizado profundo tem potencializado a eficiência dos ataques, tornando-os mais rápidos e precisos. Essa constatação reforça a necessidade do uso de algoritmos de *hashing* robustos, acompanhados de estratégias adicionais como *salting* e múltiplas iterações, para mitigar os riscos crescentes nesse cenário de ameaças (Yu et al., 2023).

2.2 Funções de Hash

Para proteger as senhas armazenadas, a prática padrão é o uso de funções de hash criptográficas. Uma função de hash é um algoritmo que transforma uma entrada de tamanho arbitrário em uma saída de tamanho fixo, chamada hash (George; Scholar; Mathew, 2021). Para ser criptograficamente segura, a função deve possuir propriedades essenciais, como a unidirecionalidade (ser computacionalmente inviável de reverter), resistência a colisões (ser inviável encontrar duas entradas que gerem o mesmo hash) e o efeito avalanche, na qual uma pequena mudança na entrada causa uma mudança drástica na saída (Yu et al., 2023).

2.2.1 Algoritmos Clássicos (MD5, SHA-1)

Algoritmos como MD5 e SHA-1 foram projetados para serem rápidos, uma característica desejável para verificação de integridade de arquivos, mas que se torna uma vulnerabilidade fatal no armazenamento de senhas. A velocidade beneficia desproporcionalmente um atacante em um cenário de quebra de senhas offline (Luo et al., 2018). Além disso, ambos os algoritmos foram criptograficamente comprometidos com a descoberta de ataques de colisão práticos. Em reconhecimento a essas falhas, o *National Institute of Standards and Technology* (NIST) anunciou a descontinuação completa do uso do SHA-1 até o final de 2030 (Mouha; Celi, 2023).

2.2.2 Algoritmos Modernos (SHA-256, SHA-512, bcrypt, scrypt, Argon2)

A inadequação dos algoritmos clássicos levou ao desenvolvimento de funções de hash especializadas para senhas, projetadas para serem intencionalmente lentas e consumidoras de recursos. Membros da família SHA-2, são algoritmos de propósito geral considerados seguros para aplicações como assinaturas digitais. No entanto, sua eficiência em CPUs os torna vulneráveis à aceleração massiva por hardware especializado (GPUs), sendo uma escolha subótima para o armazenamento de senhas (Islam et al., 2023).

Essa rapidez é explorada em ataques baseados em GPU. Diferentemente das CPUs, que são otimizadas para executar poucas tarefas complexas sequencialmente, as GPUs possuem uma arquitetura de paralelismo massivo, projetada para executar milhares de operações simples (como cálculos matemáticos de um hash) ao mesmo tempo (Vu; Nguyen, 2014).

Em um ataque de força bruta ou de dicionário, um atacante utiliza o poder das GPUs para testar bilhões, ou até trilhões, de senhas candidatas por segundo. Como o SHA-256 exige apenas poder de processamento (CPU-bound) e não uma quantidade significativa de memória (memory-bound), ele é um alvo ideal para essa paralelização (Shaha; Raju, 2019).

Baseado no cifrador Blowfish, o bcrypt foi um dos primeiros algoritmos projetados para ser lento. Ele introduz um "fator de custo" que aumenta exponencialmente o tempo de compu-

tação, tornando-o resistente a ataques de força bruta. Sua natureza é intensiva em CPU (CPU-bound), o que oferece resistência moderada a GPUs (Santana, 2025).

O *scrypt* inovou ao ser uma função memory-hard, ou seja, intensiva em memória. Essa característica o torna mais resistente a ataques de hardware especializado (GPU/ASIC), pois a memória é um recurso mais caro e limitado nesses dispositivos do que o poder de processamento bruto (Choi; Kim; Seo, 2024).

Um grande avanço ocorreu com a criação do Argon2, vencedor da Password Hashing Competition em 2015, que rapidamente se consolidou como padrão de referência em segurança de senhas. O algoritmo apresenta três variantes principais: Argon2d, Argon2i e Argon2id, cada uma voltada a cenários específicos (Gregório; Goya, 2019). Neste estudo, analisaram-se essas versões considerando métricas de aleatoriedade, como entropia e monobit, concluindo que Argon2i e Argon2d apresentaram comportamento adequado para os contextos previstos, enquanto a versão híbrida Argon2id não demonstrou a mesma consistência.

Um estudo recente que combina modelos econômicos com uma análise empírica de sua adoção demonstra que, com a configuração correta, o Argon2id reduz a taxa de comprometimento de senhas em 42.5% em comparação com o SHA-256 para um orçamento de ataque fixo (Biryukov et al., 2021).

2.3 Técnicas de Fortalecimento de Senhas

As técnicas de fortalecimento de senhas representam um componente essencial na arquitetura de segurança de sistemas de informação, uma vez que os algoritmos de hash, por si só, não são suficientes para resistir a ataques sofisticados. Entre essas práticas, destacam-se o *salting*, o *key stretching* e o aumento do número de iterações, que têm como objetivo dificultar ataques de dicionário, de força bruta e o uso de tabelas pré-computadas.

2.3.1 *Salting*

O *salting* consiste na inserção de valores aleatórios antes da aplicação do algoritmo de hash, de modo a tornar cada senha única mesmo que dois usuários escolham a mesma combinação de caracteres. No estudo de Lustro (2019), é evidente que o uso de *salts* dinâmicos fortalece a proteção contra ataques baseados em tabelas de *rainbow* e dicionário. Por exemplo, em um estudo foi demonstrado que o emprego de *salts* gerados dinamicamente em conjunto com o bcrypt aumenta a resistência contra ataques de pré-computação, mitigando vulnerabilidades em ambientes de autenticação distribuída (Lustro, 2019).

Além disso, o *key stretching* surge como prática fundamental no fortalecimento das senhas, pois adiciona complexidade computacional ao processo de verificação. Algoritmos como PBKDF2, bcrypt, scrypt e Argon2 adotam essa técnica de forma nativa, exigindo maior custo

computacional por parte de um atacante que tente comprometer senhas armazenadas. No mais, a integração entre PBKDF2 e geradores criptograficamente seguros de números aleatórios (CS-PRNG), demonstram que o aumento controlado do número de iterações eleva substancialmente o tempo necessário para ataques de força bruta (Mustafa, 2024).

2.3.2 Key Stretching e Iterações

Os algoritmos modernos de derivação de chaves têm explorado combinações híbridas. Em um estudo é apresentado um modelo que combina Argon2i com criptografia simétrica AES, incorporando *salting* e *key stretching* para alcançar maior resiliência contra ataques baseados em hardware especializado (Modugula, 2020). De maneira semelhante, Bidhuri (2019) investigou a aplicação conjunta de script e AES, reforçando o papel da memória intensiva e do *salting* dinâmico como barreiras adicionais (Bidhuri, 2019).

O *key stretching* (ou fortalecimento de chave) é a prática de aplicar a função de hash repetidamente, milhares ou milhões de vezes. Cada iteração aumenta o tempo necessário para calcular um único hash. Para o servidor, o custo adicional é de milissegundos, mas para um atacante que precisa testar bilhões de senhas, o custo se multiplica drasticamente, tornando os ataques de força bruta offline computacionalmente inviáveis (Liu; Xu; Ma, 2020).

2.4 Normas e Legislação em Segurança da Informação

2.4.1 ISO/IEC 27001 e 27002

A família de normas ISO/IEC 27000 fornece uma estrutura reconhecida internacionalmente para a gestão da segurança da informação (Hintzbergen et al., 2018). A ISO/IEC 27001 especifica os requisitos para um Sistema de Gestão de Segurança da Informação (SGSI), sendo uma norma certificável (Hintzbergen et al., 2018).

A ISO/IEC 27002, por sua vez, funciona como um código de prática, detalhando centenas de controles de segurança. No contexto deste trabalho, ela é particularmente relevante por dois motivos principais:

- **Controle A.10 (Criptografia):** Este grupo de controles estabelece a necessidade de uma política sobre o uso de controles criptográficos para proteger a confidencialidade, autenticidade e integridade da informação.
- **Controle A.9 (Controle de Acesso):** De forma mais específica, o controle A.9.4.2 (Gerenciamento de informação de autenticação secreta) define que as senhas dos usuários devem ser protegidas. A norma estabelece que senhas não devem ser armazenadas em texto claro, mas sim protegidas por mecanismos criptográficos robustos, como as funções

de *hashing* com *salt*, tema central desta pesquisa.

Dessa forma, a adoção de algoritmos de *hashing* modernos (como bcrypt ou Argon2) é uma medida técnica essencial para que uma organização esteja em conformidade com as boas práticas de segurança recomendadas pela família ISO 27000.

2.4.2 Lei Geral de Proteção de Dados (LGPD)

A Lei Geral de Proteção de Dados (LGPD), Lei nº 13.709/2018, estabelece um marco legal para o tratamento de dados pessoais no Brasil. O Artigo 46 da lei determina que os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados (Minghelli et al., 2024).

No que tange especificamente ao armazenamento de senhas, a LGPD exige que as organizações adotem mecanismos de segurança compatíveis com o risco envolvido, o que inclui o uso de técnicas robustas de criptografia e *hashing*. A não observância dessas práticas pode resultar em sanções administrativas, como multas de até 2% do faturamento da empresa, além de danos reputacionais significativos (Carloto et al., 2021).

3 TRABALHOS RELACIONADOS

Um estudo (Bachtiar et al., 2018) aborda os desafios da proteção de dados na era digital, destacando a vulnerabilidade de sistemas criptográficos frente a ataques de criptoanálise. Os autores propõem o uso de criptografia dinâmica como alternativa para aumentar a segurança, implementando uma versão modificada do algoritmo AES de 256 bits. Nesse esquema, é introduzido um *salt* gerado por meio de um algoritmo Linear Congruential Generator (LCG), que é combinado ao texto simples antes da aplicação do algoritmo One Time Pad (OTP). Essa estratégia gera padrões imprevisíveis, reduzindo a probabilidade de ataques de dicionário ou quebra sistemática da chave. O método apresenta ainda a vantagem de ser aplicável em sistemas distintos, ampliando sua versatilidade prática. Os testes realizados incluíram o Teste de Ferramentas do NIST, no qual o sistema obteve taxa de sucesso de 72,5% em oito parâmetros avaliados.

Pesquisadores analisaram diferentes algoritmos de hash (Skanda; Srivatsa; Premananda, 2022), enfatizando o aumento expressivo de violações de dados que expuseram contas e informações financeiras, destacando a importância de fortalecer os mecanismos de armazenamento de senhas. Os autores analisam o desempenho de diferentes algoritmos de hash, como MD5, SHA-256 e bcrypt, ressaltando a superioridade deste último em cenários de ataques reais. A pesquisa mostra que o bcrypt, por incorporar técnicas como *salting* obrigatório e *key stretching*, dificulta a exploração de credenciais comprometidas, reduzindo riscos como fraude, apropriação de contas e ataques de preenchimento de credenciais. A metodologia incluiu a tentativa de reversão dos hashes por meio de tabelas rainbow, permitindo evidenciar as diferenças de resis-

tência entre os algoritmos. Enquanto SHA-256 e MD5 demonstraram maior suscetibilidade a ataques por serem algoritmos rápidos, o bcrypt apresentou desempenho mais lento, porém mais seguros.

Outro estudo (Sheketa et al., 2021) discute a relevância das funções hash na proteção de informações sensíveis em ambientes digitais voltados à educação. Os autores ressaltam que, diante do volume crescente de dados críticos, é essencial empregar algoritmos que assegurem tanto a integridade quanto a autenticidade das informações. Nesse contexto, o estudo explora a aplicação do SHA-512 em sistemas de gestão acadêmica, mostrando como sua utilização contribui para a proteção de dados pessoais de estudantes, além de garantir confiabilidade em processos de autenticação e autorização. A pesquisa destaca ainda que a eficiência das funções hash depende de parâmetros como taxa de transferência, fator decisivo para o processamento de grandes volumes de dados. Além disso, o trabalho avalia o efeito avalanche, que evidencia a robustez do SHA-512 ao gerar mudanças na saída mesmo com pequenas variações na entrada.

4 METODOLOGIA

A pesquisa teve uma abordagem experimental quantitativa, estruturada em etapas sucessivas para possibilitar a comparação entre os algoritmos de *hashing* Argon2, bcrypt e SHA-256 em diferentes condições de complexidade de senhas. Os experimentos foram realizados em um computador equipado com processador AMD Ryzen 5 4500, GPU AMD RX 6650 XT e 16 GB de memória RAM DDR4 3200 MHz, executando diretamente no Windows 11.

4.1 Etapas da Pesquisa

- a) Preparação de um conjunto de dados em formato CSV gerado por meio de um script *Python* denominado *generate_passwords.py*. Através da biblioteca *pandas*, estruturou-se os dados em um *DataFrame*, exportando-os para o arquivo *passwords.xlsx*;
- b) Implementação de scripts com uso das bibliotecas *bcrypt*, *argon2-cffi*, *psutil*, *pandas* e *matplotlib* com o *Python v.3.12.10*, para aplicação dos algoritmos de *hashing* e registro de métricas como tempo de processamento, uso de CPU, uso de memória e parâmetros empregados em cada função;
- c) Realização de testes de resistência com a ferramenta *John the Ripper*, para simular ataques de dicionário, foi utilizada a *wordlist_test.txt*. Nesses testes, foram analisados: (a) o número e o percentual de hashes quebrados, (b) a taxa média de tentativas por segundo (hashes/s), e (c) o uso de CPU e memória durante a execução do ataque;
- d) Consolidação de todos os dados coletados, provenientes tanto das execuções em *Python* quanto dos testes com o *John the Ripper*, em planilhas CSV para análise quantitativa;

- e) Elaboração de tabelas e gráficos comparativos com o auxílio das bibliotecas `pandas` e `matplotlib`, possibilitando avaliar o equilíbrio entre desempenho computacional e resistência prática aos ataques.

5 RESULTADOS

Nesta seção são apresentados os resultados obtidos nos experimentos de geração e quebra de hashes. Foram gerados arquivos correspondentes às três funções de hash escolhidas: **SHA-256**, **bcrypt** (com *cost* = 12) e **Argon2** (com parâmetros *time_cost* = 2, *memory_cost* = 65536 KB e *parallelism* = 1).

Para a etapa de validação inicial do ambiente experimental, utilizou-se uma amostra de mil senhas do arquivo `data/passwords.xlsx`. Nos testes de resistência realizados com o John the Ripper, embora tenha sido utilizada uma wordlist específica para o experimento, é importante destacar que wordlists empregadas em ataques reais geralmente variam de dezenas de milhares a milhões de senhas. Vazamentos amplamente conhecidos, como RockYou e Collection Series, frequentemente ultrapassam 1 milhão de entradas. Os hashes resultantes foram armazenados nos arquivos `hashes/sha256_test.txt`, `hashes/bcrypt_test_r12.txt`, `hashes/argon2_test.txt`. Os arquivos completos utilizados na pesquisa permanecem disponíveis no repositório do projeto⁴, conforme especificado na metodologia.

5.1 Amostra de Hashes Gerados

A Tabela 1 apresenta um exemplo simplificado dos resultados obtidos para a amostra inicial. Por questões de segurança, as senhas foram substituídas por valores representativos.

Tabela 1 – Comparação de formatos de hash (exemplos abreviados)

SHA-256	bcrypt	Argon2
dbb211...7bccbc	\$2b\$12\$HQ...3Tm	\$argon2id\$v=19\$m=16384...+lE
77db3a...b40989	\$2b\$12\$le...zxO	\$argon2id\$v=19\$m=16384...RU4
bcb15f...09802a	\$2b\$12\$Jn...qz.	\$argon2id\$v=19\$m=16384...C/k
fb1ec0...fe34fa	\$2b\$12\$u5...Moa	\$argon2id\$v=19\$m=16384...GI4
a742bc...2a39bd	\$2b\$12\$sL...Uym	\$argon2id\$v=19\$m=16384...vgg
8d969e...dc6c92	\$2b\$12\$4l...zIu	\$argon2id\$v=19\$m=16384...+FY
1c8bfe...b63032	\$2b\$12\$9u...2u6	\$argon2id\$v=19\$m=16384...hd4
280d44...9f2d02	\$2b\$12\$Wf...p6q	\$argon2id\$v=19\$m=16384...j4c
5c7b1f...16c3d2	\$2b\$12\$ca...w2C	\$argon2id\$v=19\$m=16384...1T0

A estrutura dos exemplos apresentados na Tabela 1 evidencia algumas diferenças entre

⁴Disponível em: <https://github.com/pedrosiqueirapb/analysis_of_hashing_algorithms/>

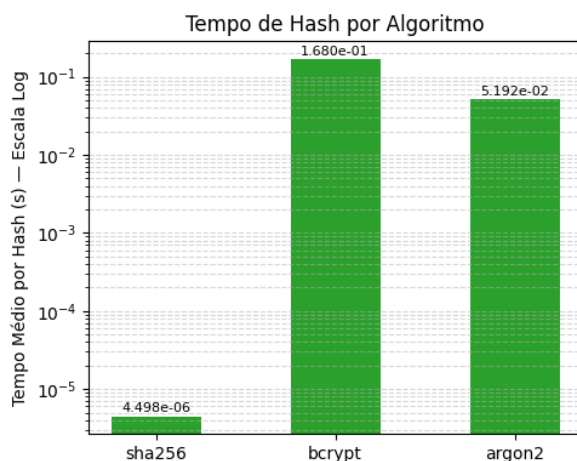
os algoritmos analisados. O **SHA-256** gera um hash hexadecimal de comprimento fixo, composto por 64 caracteres, e não incorpora um *salt* em sua estrutura. Trata-se de um algoritmo extremamente rápido, o que o torna eficiente para verificações, mas também mais vulnerável a ataques em larga escala quando utilizado isoladamente, especialmente se não forem aplicadas técnicas complementares de *salting* e *key-stretching*.

Por outro lado, o **bcrypt** produz uma string mais complexa, que inclui o identificador do algoritmo, o custo de processamento (por exemplo, “12”) e o *salt* embutido. Por fim, o **Argon2** apresenta um formato ainda mais robusto e configurável, contendo parâmetros explícitos de consumo de memória e tempo, como $v=19$, $m=16384$.

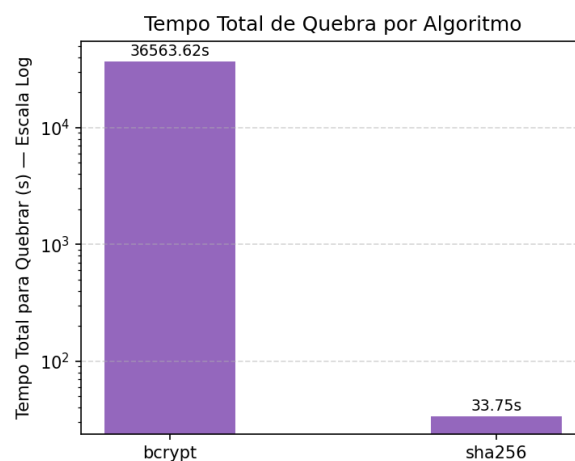
5.2 Desempenho e Resistência

Cabe ressaltar que, durante a execução dos testes de resistência, foram identificadas limitações na versão da ferramenta *John the Ripper* utilizada. Embora o *benchmarking* de uso de memória tenha sido realizado com sucesso, a ferramenta não reportou de forma consistente a taxa de tentativas por segundo (hashes/s) para o Argon2. Por essa razão, algumas métricas de ataque para este algoritmo podem não estar presentes nas análises a seguir. A Figura 1 ilustra a comparação do tempo médio de geração de hashes por algoritmo, já a Figura 2 mostra o tempo total necessário para a quebra dos hashes.

Figura 1 – Tempo para Gerar Cada Hash **Figura 2 – Tempo de Quebra por Algoritmo**



Fonte: Elaborado pelos autores (2025).



Fonte: Elaborado pelos autores (2025).

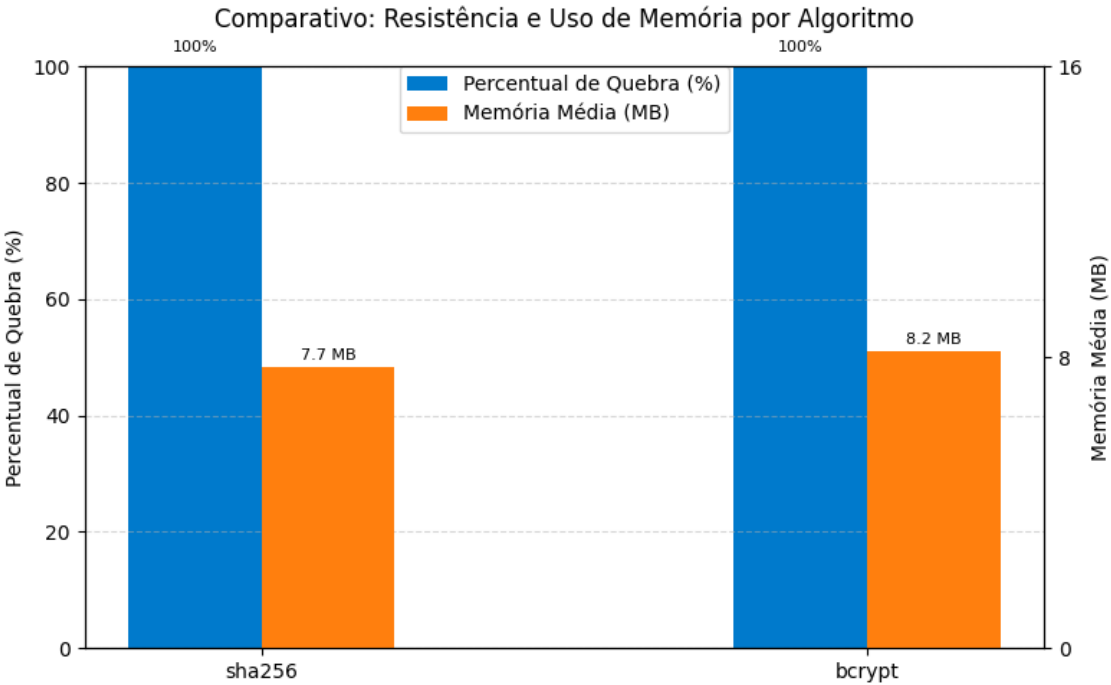
A análise dos dados evidencia um claro *trade-off* entre desempenho e segurança. Algoritmos como o **SHA-256** apresentam excelente desempenho em termos de velocidade, sendo capazes de processar milhares de hashes por segundo. No entanto, essa rapidez implica menor segurança no contexto de armazenamento de senhas, pois permite que um atacante realize um grande número de tentativas em um curto intervalo de tempo.

Por outro lado, o **bcrypt** é consideravelmente mais lento, uma vez que utiliza um fator de custo configurável (por exemplo, 12) que aumenta exponencialmente o tempo necessário

para gerar cada hash. Essa lentidão é intencional e representa um mecanismo eficaz de defesa, pois reduz drasticamente a taxa de tentativas possíveis em ataques de força bruta.

Já o **Argon2** demonstra um equilíbrio entre tempo de execução e segurança, combinando parâmetros ajustáveis de custo computacional e uso de memória. Esse design permite adaptabilidade a diferentes contextos: pode ser configurado para operar de forma mais leve em sistemas com restrições de desempenho ou de modo mais rigoroso em ambientes que priorizam a segurança. Dessa forma, o Argon2 se destaca como uma solução moderna e eficiente para o armazenamento seguro de credenciais, oferecendo resistência aprimorada a ataques baseados em hardware paralelo, sem comprometer excessivamente o desempenho do servidor. A Figura 3 apresenta o percentual de senhas quebradas por cada função e mostra o consumo médio de memória durante os ataques.

Figura 3 – Percentual de Senhas Quebradas e Uso Médio de Memória por Algoritmo



Fonte: Elaborado pelos autores (2025).

Os resultados mostraram que tanto o **SHA-256** quanto o **bcrypt** tiveram 100% das senhas quebradas pelo *John the Ripper*. Esse comportamento ocorreu porque o ataque foi baseado em um dicionário contendo todas as senhas utilizadas nos experimentos, o que faz com que a ferramenta encontre correspondências exatas independentemente do algoritmo ou do nível de custo configurado. Dessa forma, o percentual de quebra reflete não a “força” criptográfica dos algoritmos, mas as características do cenário experimental. Ainda assim, o tempo necessário para a quebra foi significativamente maior no caso do **bcrypt** — consequência direta do aumento do custo computacional imposto pelo parâmetro de *rounds*. Por fim, observou-se que o **Argon2** não teve suas senhas avaliadas pelo *John the Ripper* devido a limitações de suporte ao formato de hash utilizado, embora o monitoramento de recursos tenha mostrado que ele foi o algoritmo com maior consumo de memória durante os testes (~160 MB), enquanto o bcrypt

apresentou maior carga de CPU e o SHA-256 manteve consumo reduzido.

Os resultados apontaram que o algoritmo SHA-256 apresentou maior vulnerabilidade, permitindo elevadas taxas de tentativa e, conseqüentemente, um maior percentual de senhas comprometidas. Em contrapartida, o bcrypt e o Argon2 apresentaram comportamento consistente com sua proposta de segurança, impondo maior custo computacional e retardando significativamente as tentativas de quebra.

6 CONCLUSÃO

Este estudo comparativo demonstrou a importância crítica da escolha de algoritmos de *hashing* na segurança de senhas. Os resultados experimentais confirmaram que funções de hash de propósito geral, como o SHA-256, são inadequados para o armazenamento de credenciais devido à sua alta velocidade de processamento, que permite ataques de força bruta e dicionário em tempo inviável para algoritmos modernos. Em contraste, algoritmos projetados para consumo intencional de recursos, como bcrypt e Argon2, oferecem proteção superior ao elevar o custo computacional e de memória para um atacante.

Os testes de desempenho e resistência evidenciaram que o bcrypt, configurado com um fator de custo adequado, e o Argon2, com seus parâmetros de memória e tempo, estabelecem um equilíbrio robusto entre a usabilidade para operações legítimas de autenticação e a inviabilidade de ataques em larga escala. Embora o SHA-256 tenha se mostrado eficiente em tempo de geração, seu uso no armazenamento de senhas é inadequado, pois sua alta velocidade de processamento favorece taxas maiores de tentativas por segundo em ataques de força bruta e dicionário. O bcrypt apresentou maior resistência temporal, exigindo significativamente mais tempo para que as senhas fossem quebradas, enquanto o Argon2 não teve senhas quebradas devido à limitação da ferramenta, embora tenha apresentado alto consumo de memória, coerente com sua natureza *memory-hard*.

Algumas limitações técnicas foram observadas ao longo da condução dos experimentos. A primeira refere-se ao suporte parcial do *John the Ripper* ao formato de hash utilizado para o Argon2, o que impediu a realização dos testes de quebra para esse algoritmo. A segunda diz respeito à impossibilidade de executar o *Hashcat* no ambiente experimental, o que limitou a análise de ataques por GPU. Embora essas limitações não comprometam os resultados obtidos, elas apontam oportunidades para aprofundamento em trabalhos futuros.

Por fim, recomenda-se ampliar a amostra de senhas e a diversidade das *wordlists* utilizadas para fortalecer a validade estatística dos resultados, explorar a performance dos algoritmos em diferentes plataformas e configurações de hardware (incluindo testes com aceleração via GPU, com o Hashcat, para uma análise mais completa) e investigar a otimização dos parâmetros de custo (como o fator de *rounds* do bcrypt e os parâmetros de memória/tempo do Argon2) para diferentes cenários de risco e capacidade de servidor.

REFERÊNCIAS

- BACHTIAR, M. M. et al. Security enhancement of aes based encryption using dynamic salt algorithm. **Journal of Computer Science**, v. 14, n. 5, p. 693–701, 2018.
- BAI, W.; BLOCKI, J. Dahash: Distribution aware tuning of password hashing costs. *In: SPRINGER. International Conference on Financial Cryptography and Data Security*. [S.l.], 2021. p. 382–405.
- BIDHURI, V. **Enhancing Password Security Using a Hybrid Approach of SCrypt Hashing and AES Encryption**. Tese (Doutorado) — Dublin, National College of Ireland, 2019.
- BIRYUKOV, A. et al. Argon2 memory-hard function for password hashing and proof-of-work applications. **Internet Research Task Force (IRTF), RFC**, v. 9106, 2021.
- CARLOTO, S. et al. **Lei Geral da Proteção de Dados Comentada: Com enfoque nas relações de trabalho**. [S.l.]: LTr Editora, 2021. v. 1.
- CHOI, S.; KIM, D.; SEO, S. C. Parallel implementation of scrypt: A study on gpu acceleration for password-based key derivation function. Korea Institute of Information and Communication Engineering, 2024.
- CUNHA, J. F. **Gestão de segurança de informação para sistemas de Confiança Seguros**. Dissertação (Dissertação de Mestrado) — Universidade do Minho, 2021.
- GEORGE, A. T.; SCHOLAR, P.; MATHEW, J. Argon2: The secure password hashing function. *In: Proc. Natl. Conf. Emerg. Comput. Appl.* [S.l.: s.n.], 2021. v. 3, p. 81–84.
- GREGÓRIO, P.; GOYA, D. Hash criptográfico sobre senhas e aleatoriedade do argon2. *In: SBC. Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SB-Seg)*. [S.l.], 2019. p. 71–80.
- HINTZBERGEN, J. et al. **Fundamentos de Segurança da Informação: com base na ISO 27001 e na ISO 27002**. [S.l.]: Brasport, 2018.
- ISLAM, M. et al. {Arana}: Discovering and characterizing password guessing attacks in practice. *In: 32nd USENIX Security Symposium (USENIX Security 23)*. [S.l.: s.n.], 2023. p. 1019–1036.
- KANTA, A. et al. How viable is password cracking in digital forensic investigation? analyzing the guessability of over 3.9 billion real-world accounts. **Forensic Science International: Digital Investigation**, Elsevier, v. 37, p. 301186, 2021.
- LIU, H.; XU, Y.; MA, C. Chaos-based image hybrid encryption algorithm using key stretching and hash feedback. **Optik**, Elsevier, v. 216, p. 164925, 2020.
- LUO, W. et al. Authentication by encrypted negative password. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 14, n. 1, p. 114–128, 2018.
- LUSTRO, R. A. F. Ameliorating password security authentication using bcrypt algorithm with dynamic salt generation. **Journal of Advanced Research in Dynamical and Control Systems**, v. 11, n. 12-SPECIAL ISSUE, p. 1240–1245, 2019.
- MINGHELLI, M. et al. Lei geral de proteção de dados ea elaboração do relatório de impacto à proteção de dados pessoais. **Em Questão**, SciELO Brasil, v. 30, p. e–138249, 2024.

MODUGULA, R. S. R. **A Hybrid approach for Augmenting password security using Argon2i hashing and AES Scheme**. Tese (Doutorado) — Dublin, National College of Ireland, 2020.

MOUHA, N.; CELI, C. A vulnerability in implementations of sha-3, shake, eddsa, and other nist-approved algorithms. *In: SPRINGER. Cryptographers' Track at the RSA Conference*. [S.l.], 2023. p. 3–28.

MUSTAFA, N. A. A. Analysis attackers' methods with hashing secure password using csprng and pbkdf2. **Wasit Journal of Engineering Sciences**, v. 12, n. 2, p. 60–70, 2024.

NYANGARESI, V. O.; ABEKA, S. O.; RODRIGUES, A. J. Guti-based multi-factor authentication protocol for de-synchronization attack prevention in lte handovers. **Int. J. Cyber-Secur. Digit. Forensics**, v. 9, n. 1, p. 1–12, 2020.

PAL, B. et al. Beyond credential stuffing: Password similarity models using neural networks. *In: IEEE. 2019 IEEE Symposium on Security and Privacy (SP)*. [S.l.], 2019. p. 417–434.

ROHIT et al. Secure hashing algorithms and their comparison. *In: 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. [S.l.]: IEEE, 2019. p. 788–792.

SANTANA, B. L. d. N. Desenvolvimento de um chat seguro com autenticação de usuários e criptografia de dados usando rust. **PUC Goiás**, Pontifícia Universidade Católica de Goiás, 2025.

SHAHA, R. M.; RAJU, M. Security of password hashing in cloud. **Journal of Information Security**, Scientific Research Publishing, v. 10, n. 02, p. 92–106, 2019.

SHEKETA, V. et al. System analysis and example of using sha-512 hash functions to protect students' personal data on educational platforms. **International Journal of Advanced Computer Science and Applications (IJACSA)**, v. 12, n. 9, p. 85–91, 2021.

SKANDA, C.; SRIVATSA, B.; PREMANANDA, B. Secure hashing using bcrypt for cryptographic applications. *In: 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*. [S.l.]: IEEE, 2022. p. 1–5.

VU, D. H.; NGUYEN, D. T. A homogeneous parallel brute force cracking algorithm on the gpu. *In: IEEE. 2014 International Conference on Information Science and Application (ICISA)*. [S.l.], 2014. p. 1–4.

YU, W. et al. A systematic review on password guessing tasks. **Entropy**, MDPI, v. 25, n. 9, p. 1303, 2023.