Aplicação de Técnicas de Encriptação e Anonimização em Nuvem para Proteção de Dados

Matheus M. Silveira[1], **Ariel L. Portela**[1], **Michael S. Souza**[1], **Danielle S. Silva**[1], **Maria C. Mesquita**[1], **Douglas A. Silva**[1], **Rafael A. Menezes**[1], **Rafael L. Gomes**[1]

[1]Universidade Estadual do Ceará (UECE)

{matheus.monteiro,ariel.portela,michael.souza,danielle.santos,

clara.mesquita,douglas.araujo,almeida.menezes}@aluno.uece.br,

rafa.lopes@uece.br

Abstract. *The current trend of deploying online services can expose existing systems to hacking attempts and data leakage. In addition, it is necessary to deploy security solutions that do not alter customers' legacy systems. Within this context, this paper presents a system to protect the sensitive data of existing databases (legacy systems of clients)Based on two techniques designed and adapted to our solution: Searchable Symmetric Encryption for Databases (SSE-DB) and (2) Permutation and Properties Maintenance Anonymization (PPM-Anon). The proposed system prevents problems of data leakage and privacy breaches, attaching a security solution to the existing databases (without any change in these legacy systems). Results from real experiments using a real cloud environment suggest that the proposed solution is suitable for protecting the data without harming the performance of the existing services.*

Resumo. *A atual tendência de implantar serviços online pode expor os sistemas existentes a tentativas de invasão e vazamento de dados. Além disso, são necessárias soluções de segurança que não alterem os sistemas legados dos clientes. Dentro deste contexto, este artigo apresenta um sistema para proteger os dados sensíveis de bancos de dados existentes (sistemas legados de clientes) com base em duas técnicas, as quais adequamos à nossa solução: Busca em Criptografia simétrica para bancos de dados (SSE-DB) e (2) Permutação e manutenção de propriedades anonimização (PPM-Anon). O sistema proposto evita problemas de vazamento de dados e violação de privacidade, anexando uma solução de segurança aos bancos de dados existentes (sem nenhuma alteração nesses sistemas legados). Resultados de experimentos reais usando um ambiente de nuvem real sugerem que a solução proposta é adequada para proteger os dados sem prejudicar o desempenho dos serviços existentes.*

1. Introdução

Atualmente, muitas empresas e instituições governamentais tendem a implantar serviços online a fim de modernizar seus modelos de negócios, onde a maioria desses modelos de negócios é aprimorada devido às vantagens do Cloud Computing, como abordagens sob demanda, escalabilidade, confiabilidade, elasticidade, medição serviços, recuperação de

desastres, acessibilidade e muitos outros [Gomes et al. 2020, GUPTA and SINGH 2020]. No entanto, essa abordagem de serviços online pode ser exposta devido a vulnerabilidades tecnológicas existentes e partes envolvidas comprometidas, resultando em um cenário onde essas empresas, instituições governamentais e usuários finais estão sujeitos a tentativas de intrusão e possíveis situações de vazamento de dados.

Vários casos de vazamento de dados ocorreram em todo o mundo nos últimos anos. Por exemplo, os hotéis Marriott tiveram os dados de 500 milhões de clientes acessados por hackers [Yu et al. 2022]. Entre outras vítimas dos ataques estão empresas como T-Mobile, Quora, Google, Orbitz e Facebook, que enfrentaram grandes violações e incidentes que afetaram mais de 100 milhões de usuários.

O vazamento de dados compromete a confidencialidade das empresas e impacta diretamente as leis de privacidade existentes [Gong et al. 2022]. Atualmente, é necessário seguir os regulamentos de proteção de dados das leis de privacidade, como o Regulamento Geral de Proteção de Dados (GDPR) na Europa e a Lei Geral de Proteção de Dados (LGPD) no Brasil, para evitar problemas de privacidade e possíveis taxas. Além disso, os dados são considerados um dos ativos mais importantes das empresas e é crucial protegê-los. Dessa forma, micro, pequenas e grandes empresas, bem como instituições governamentais, precisam se adequar aos pontos elencados pelas leis de privacidade, pois isso pode impactar nos negócios, ao lidar com dados de seus clientes e funcionários, no momento de fazer a portabilidade dos dados, ao cooperar internacionalmente, etc. Assim, é necessário implantar soluções de segurança para proteger os dados de forma eficiente [Gupta et al. 2022].

Uma abordagem existente para proteger os dados é o uso de técnicas de criptografia, que convertem os dados de entrada (originais) em saída (criptografados). A conversão é baseada em uma chave, onde apenas entidades autorizadas a possuem e podem descriptografar os dados [Aparajit et al. 2022]. No entanto, as técnicas de criptografia de dados demandam tempo de processamento, prejudicando a possibilidade de realizar criptografia e descriptografia frequentes de grandes quantidades de dados. Consequentemente, aumentará o tempo para realizar um processo de busca e recuperação dos dados desejados [Gupta et al. 2022].

Outra abordagem de segurança para este cenário são as técnicas de anonimização. Essas técnicas visam tornar os dados sensíveis transmitidos pela Internet não identificáveis, preservando a privacidade dos usuários [Mosca et al. 2023]. As técnicas de anonimização surgem como uma abordagem crucial para atender aos aspectos mencionados das leis de privacidade, pois permitem que os usuários sejam protegidos de maneira irreversível. No entanto, as técnicas de anonimização existentes permitem diferentes níveis de anonimização, o que pode alterar o contexto dos dados, impossibilitando a aplicação de técnicas de soluções inteligentes (como Inteligência Artificial) para identificar padrões, o que pode prejudicar a gestão do serviço [Costa et al. 2021].

Dentro desse contexto, este artigo apresenta um sistema de nuvem inovador para proteger dados privados de bancos de dados existentes (sistemas legados de clientes) com base em duas técnicas projetadas: (1) Criptografia Simétrica Pesquisável para Bancos de Dados (SSE-DB) e (2) Anonimização com Preservação de Permutação e Propriedades (PPM-Anon). SSE-DB é uma evolução do SSE [Li et al. 2019a] original que permite

SHA256, particionamento de criptografia e criptografia de várias tabelas em bancos de dados SQL. Este ponto de evolução é baseado nos direcionamentos da RFC 6151 [1] da *Internet Engineering Task Force* (IETF), a qual descreve diversas vulnerabilidades sobre o uso de MD5, especialmente no que se refere a resistência a colisões. Portanto, o SSE-DB supera a limitação de processamento lento, permitindo uma maior eficácia ao considerar grandes bancos de dados e dados dinâmicos. Da mesma forma, o PPM-Anon é uma extensão de uma técnica descrita na referência [Aleroud et al. 2016] e gera dados sintéticos mantendo propriedades matemáticas, como média e desvio padrão, permutando os autovetores ao invés de gerar novos, o que evita erros de precisão nessas medidas.

Inicialmente, os dados confidenciais originais são armazenados com segurança no ambiente de nuvem usando SSE-DB (permitindo a pesquisa e a recuperação dos dados confidenciais originais da nuvem) e, posteriormente, os dados confidenciais originais são anonimizados no banco de dados do cliente usando PPM-Anom (mantendo o contexto dos dados e, consequentemente, mantendo sua usabilidade). Assim, o objetivo do sistema é prevenir problemas de vazamento de dados e violação de privacidade dos clientes de empresas e organizações, agregando uma solução de segurança aos bancos de dados existentes, ou seja, sem nenhuma alteração nesses sistemas legados.

Resultados dos experimentos reais utilizando um ambiente de nuvem real sugerem que a solução proposta é adequada para proteger os dados por meio de criptografia e anonimização de um banco de dados, onde vários cenários foram avaliados (variando o tamanho do banco de dados e a carga solicitada na criptografia, pesquisa e processos de anonimização).

O restante deste artigo está organizado da seguinte forma. A seção 2 detalha as soluções existentes para proteção de dados. A seção 3 descreve o sistema projetado, enquanto a seção 4 discute os experimentos realizados e os resultados. Finalmente, a seção 5 conclui o artigo e apresenta trabalhos futuros.

2. Trabalhos Relacionados

Esta seção descreve os principais trabalhos relacionados, e recentemente publicados pela comunidade científica, sobre proteção de dados e soluções de segurança que utilizam técnicas de anonimização e encriptação de dados.

Thabit et al. [Thabit et al. 2021] projetaram uma técnica de criptografia para segurança de computação em nuvem usando duas camadas de criptografia, garantindo a segurança de dados sensíveis e confidenciais durante o transporte e armazenamento. Primeiro, divide o texto original e a chave em partes iguais por meio de operações lógicas como XOR, XNOR, deslocamento etc. Posteriormente, aplica uma abordagem de criptografia genética baseada no dogma central da biologia molecular (transcrição e tradução). Apesar do uso de criptografia, esta solução não considera a necessidade de usar os dados protegidos, bem como a possível necessidade de pesquisar e recuperar os dados protegidos.

Mann et al. [Mann et al. 2021] apresentam uma abordagem para garantir a proteção de dados em sistemas baseados em nuvem que mudam dinamicamente, que analisa a configuração do sistema baseado em nuvem automaticamente para detectar

[1] https://datatracker.ietf.org/doc/html/rfc6151

mudanças nas ameaças à proteção de dados ou na disponibilidade de mecanismos de proteção de dados. Além disso, combina a detecção baseada em padrões de configurações problemáticas do sistema com adaptações automáticas de tempo de execução baseadas em modelos. Da mesma forma, Huang et al. [Huang et al. 2022] descrevem um método baseado em aprendizado de imitação adversária generativa para descobrir os riscos de segurança de dados de privacidade em IoT, treinando agentes de proteção de privacidade usando uma grande quantidade de dados especializados em proteção de privacidade. Porém, ambas as soluções não avaliam a proteção dos dados no sistema existente, nem consideram o cenário de utilização dos dados protegidos.

Wang et al. [Wang et al. 2022] propõem uma tecnologia de recuperação de privacidade aprimorada para IoT assistida em nuvem, que é projetada por meio de um índice implícito mantido por servidores de ponta e um modelo de recuperação hierárquico que preserva a privacidade dos dados, ocultando as informações de transmissão de dados entre a nuvem e os servidores de borda. Esta proposta armazena dados parciais em servidores de ponta a fim de preservar a privacidade dos usuários, mas não considera a realização da recuperação de dados para o sistema final utilizado pelos usuários, possibilitando possíveis problemas de Qualidade de Serviço (QoS) e Experiência (QoE).

Suresha et al. [D and Karibasappa 2021] apresentam uma técnica para aprimorar a proteção de dados usando criptografia baseada em derivação de chave, com o objetivo de fornecer confidencialidade, autenticação e modificação para os dados armazenados na nuvem. Este mecanismo é projetado para derivar três chaves secretas separadas de uma única chave mestra e cada chave é usada para uma operação específica. Apesar do uso de criptografia, esta proposta não considera cenários em que seja necessário pesquisar e obter dados do banco de dados criptografado, limitando sua aplicabilidade em sistemas legados existentes.

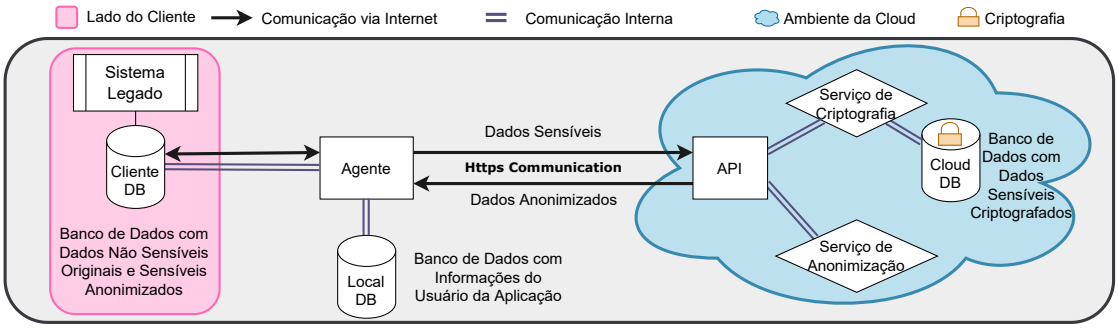
Rafique et al. [Rafique et al. 2021] descrevem um sistema distribuído de proteção de dados para provedores de SaaS, que suporta criptografia de dados e implantação de consultas agregadas no mecanismo de bancos de dados NoSQL heterogêneos. Apesar da proteção de dados e capacidade de busca desta solução, ela precisa construir anotações que representem requisitos específicos da aplicação para realizar o processo de busca, limitando sua aplicabilidade para cenários automatizados e prejudicando sua integração com sistemas legados da indústria.

Com base na revisão da literatura, essas propostas existentes não realizam uma proteção de dados adequada quando é necessário pesquisar e recuperar dados para usuários ou sistemas finais. Além disso, estes trabalhos não consideram cenários onde os dados protegidos são usados como entrada para outra solução, sendo necessário anonimizar esses dados para serem usados, enquanto as questões de privacidade são preservadas.

3. Proposta

O sistema proposto é composto por: Agente, API, Serviço de Criptografia, Serviço de Anonimização e Cloud DB. Além disso, existem duas entidades externas: sistema legado e banco de dados do cliente. Uma visão geral do sistema proposto e do contexto de implantação é ilustrada na Figura 3.

O *Sistema Legado*, do lado do cliente, é o serviço existente que faz parte de uma empresa, instituição governamental etc. O *Sistema Legado* possui um banco de dados,



chamado *Cliente DB*, que armazena os dados sensíveis a serem protegidos e os dados normais. Assim, o *Cliente DB* é o alvo a ser protegido, pois pode estar vulnerável a ameaças vindas do *Sistema Legado* (devido a problemas de codificação ou quebra de frameworks) ou tentativas de invasão direta (como SQL Injection, exploração de configuração incorreta e outros) [Kalugina et al. 2020].

O *Agente* interage diretamente com o Cliente DB, sendo responsável pela comunicação entre Cliente e API. Possui um *Local DB*, que armazena todas as informações necessárias para se comunicar com o ambiente de nuvem do sistema proposto através da API definida. Essas informações incluem token de acesso à API, endereço IP e porta do banco de dados cliente, conjunto de campos e colunas do banco de dados cliente que contém dados confidenciais (que precisam ser protegidos) e outras configurações. A *API* permite que o sistema conecte as técnicas SSE-DB e PPM-Anom ao banco de dados que está salvo no Cloud Service protegido, bem como é gerado um token para permitir que o cliente solicite seu dado sensível a ser protegido na nuvem. Da mesma forma, o *Cloud DB* é o banco de dados em nuvem que armazena os dados sensíveis com o SSE.

Finalmente, os *Serviço de Criptografia* e *Serviço de Anonimização* executam as técnicas Searchable Symmetric Encryption for Databases (SSE-DB) e Permutation and Properties Maintenance Anonymization (PPM-Anon), respectivamente. Assim, o *Serviço de Criptografia* criptografa os dados confidenciais originais, bem como realiza a busca e a recuperação dos dados confidenciais originais do *Cloud DB*. Da mesma forma, o *Serviço de Anonimização* anonimiza os dados originais e retorna esses dados anonimizados para o *Client DB*, permitindo seu uso pelo *Sistema Legado* enquanto as questões de privacidade são preservadas.

O comportamento do sistema proposto pode ser definido nas seguintes etapas: (1) os dados confidenciais originais são armazenados de forma segura no ambiente de nuvem usando SSE-DB; (2) posteriormente os dados sensíveis originais são anonimizados no banco de dados do cliente usando o PPM-Anom (mantendo o contexto dos dados e, consequentemente, mantendo sua usabilidade); e, (3) a API recebe as solicitações para pesquisar e recuperar os dados confidenciais originais da nuvem, conforme mostrado na Figura 1.

Assim, o objetivo do sistema é prevenir problemas de vazamento de dados e violação de privacidade, agregando uma solução de segurança aos bancos de dados existentes, ou seja, sem nenhuma alteração nesses sistemas legados. A seguir, detalharemos as técnicas SSE-DB e PPM-Anon, nas Subseções 3.1 e 3.2, respectivamente. Durante o trabalho, a notação aplicada é resumida na Tabela 1.

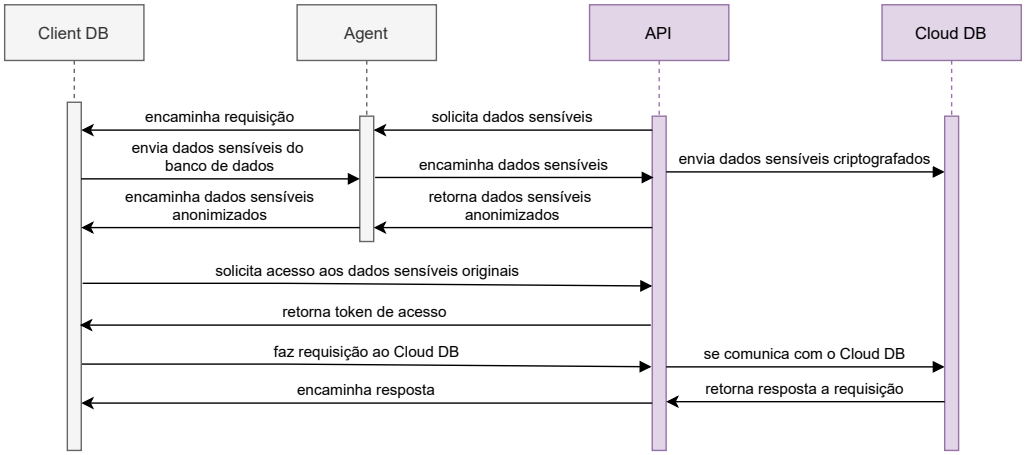


Figura 1. Armazenamento Seguro, Pesquisa e Recuperação.

Tabela 1. Notação

Simbolo	Descrição
SK	Esquema de função para executar a busca.
Gen	Função para gerar uma chave secreta aleatória.
Enc	Função para criptografar uma mensagem.
Dec	Função para descriptograr uma mensagem.
K	Chave secreta aleatória gerada.
m	Mensagem original.
w	Solicitação de uma trapdoor.
c	Mensagem criptografada.
T_w	Trapdoor de uma palavra w.
R	Conjunto de registros a serem criptografados.
I	Conjunto de registros criptografados.
S	Conjunto de palavras a serem pesquisadas.
B	Banco de dados.
MK	Masterkey definida.
D	Dataset a ser anonimizado.
A	Dataset anonimizado.
CM	Matriz de Covariância.
E	Conjunto de autovalores e autovetores.

3.1. SSE-DB para Proteção de Dados

Nossa nova abordagem SSE-DB evoluiu o SSE original nos seguintes aspectos: (I) Uso de SHA256 em vez de MD5, para melhorar o nível de segurança pela geração de uma saída de 256 bits expressa em 64 caracteres hexadecimais (enquanto MD5 gera 128 bits com 32 caracteres hexadecimais); (II) Particionamento da criptografia, divisão da criptografia dos dados para evitar problemas de sobrecarga de memória; e, (III) Criptografia de Múltiplas Tabelas em bancos de dados SQL, permitindo criptografia e descriptografia mais rápidas.

Um modelo geral de funcionamento do SSE consiste na requisição de um Trapdoor w feito pelo cliente (proprietário dos dados) ao provedor de serviço Cloud (servidor)

que retornará uma lista do índice dos documentos que contém w. Como fornecer uma chave de descriptografia para cada usuário não é uma abordagem segura, ter um método de pesquisa eficiente é essencial para manter esse modelo funcional.

Um dos principais modelos de SSE-DB é chamado de pesquisa em dados criptografados de chave privada. Consiste em dar ao utilizador que encriptou os dados uma chave de acesso $Token$ que, depois de os dados requeridos serem encriptados e armazenados na Cloud, permite a ele e a cada utilizador a quem ele dá a chave fazer pedidos nessa base de dados sem a decifrar. Segundo a referência [Li et al. 2019a], este esquema é feito pelo conjunto de três algoritmos $SK = (Gen, Enc, Dec)$, os dois primeiros são algoritmos probabilísticos e o último é um algoritmo determinístico. Inicialmente, Gen é usado para gerar uma chave secreta aleatória K usando um parâmetro de segurança arbitrário como entrada. O algoritmo Enc usará K e uma mensagem m para gerar a criptografia c da mensagem. Por fim, Dec usa K e c para fazer o trabalho reverso e gerar m novamente, funcionando como um método de descriptografia.

Um esquema geral de algoritmos de criptografia SSE, que são [Li et al. 2019b]: Keygen(s), Trapdoor(MK, w), BuildIndex(R, MK) e Search(T, I).

1. **Keygen(s):** é um algoritmo que deve ser executado no lado do cliente para gerar uma chave mestra MK com base em um parâmetro de segurança.
2. **Trapdoor(MK, w):** é um algoritmo executado pelo cliente, que recebe MK e uma palavra-chave w como entrada e gera o alçapão T_w da palavra w.
3. **BuildIndex(R, MK):** é um algoritmo que deve ser executado pelo cliente tomando MK e um registro R como entrada, e gera o índice IR para o registro R.
4. **Search(T, I):** é um algoritmo que deve ser executado pelo servidor usando um alçapão T_w e um índice de documento IR como entrada, e retorna 1 se w ϵ R ou 0 caso contrário.

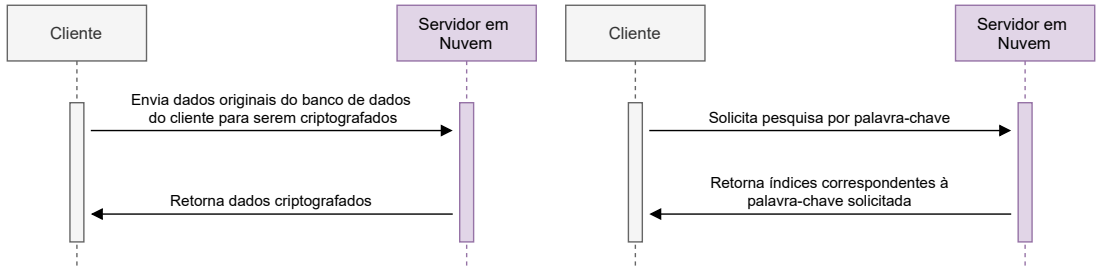
Os dados criptografados gerados, juntamente com os índices associados de todas as palavras-chave, são geralmente mantidos em segurança pelo servidor. Assim, usando o método de pesquisa de dados criptografados com chave privada, o servidor só será acessado usando a chave de acesso fornecida a ele.

Utilizamos $R = \{R_1, R_2, ..., R_n\}$ para representar um conjunto de n registros que serão criptografados para que seja possível realizar o processo de busca sem descriptografar os dados. Os dados criptografados são representados por $I = \{I_1, I_2, ..., I_m\}$. Além disso, $R_{i,j}$ representa a j-ésima palavra-chave no i-ésimo registro. Usamos $S = \{S_1, S_2, ..., S_k\}$ para representar o conjunto de k palavras a serem pesquisadas na tabela criptografada.

Além do esquema geral, o resumo de cada linha é armazenado com o banco de dados. Essas informações são usadas para executar com eficiência operações como atualizar, excluir e inserir. Os dados criptografados com os índices associados serão armazenados no servidor em SSE-DB. Para realizar o processo de busca, o cliente gera o alçapão T_w para uma palavra w e envia T_w para o servidor que realiza o processo de busca para cada registro em R.

O esquema de criptografia usa cifras e funções de hash no processo de construção e pesquisa e também precisa acessar o token para funcionar. Os dados criptografados finais serão mantidos na nuvem pelo servidor. Neste esquema, o servidor não poderá

deletar, modificar ou compartilhar os dados armazenados com terceiros, sendo apenas uma forma segura de armazenar dados sensíveis do cliente. O servidor pode conter vários bancos de dados e a cada requisição feita pelo cliente ele precisa especificar qual deles está sendo consultado.



(a) BuildIndex(R,MK). (b) Search(T,I).

Figura 2. Ilustração das Funções

O processo de construção, conforme indicado na Figura 2(a), consiste em uma transferência de dados entre o cliente e o servidor em nuvem. O cliente, que possui um banco de dados com dados sensíveis, os enviará para a nuvem para serem criptografados e os dados originais agora serão armazenados em um ambiente seguro (servidor em nuvem). O banco de dados original do cliente será substituído por uma cópia do banco de dados em nuvem com dados confidenciais criptografados que serão devolvidos ao cliente. Essa abordagem garante que, em caso de qualquer problema de segurança no lado do cliente, nenhum dado confidencial armazenado no banco de dados criptografado vaze porque a criptografia SSE-DB não permite descriptografar os dados armazenados se o invasor não tiver os dados originais.

Isso ocorre porque o processo de criptografia consiste em usar Secure Hash Algorithms (SHA) como a função de hash e, em seguida, usar a cifra de bloco simétrico Advanced Encryption Standard (AES) como o modo de criptografia principal. O AES é usado com duas chaves diferentes e duas cifras diferentes para gerar a criptografia de um único registro. A primeira chave utilizada é a palavra-chave do registro e é utilizada com a chave mestra MK para gerar um Trapdoor T_w do registro w com AES. O Trapdoor consiste em uma função que calcula o hash da palavra-chave e, em seguida, gera o resultado após o uso de uma função pseudo-aleatória. Em seguida, esse alçapão é usado para gerar a palavra-código final w com outra chamada de função AES, juntamente com o ID original do registro. Finalmente, esta palavra-código é armazenada com um índice seguro gerado com base no número de colunas do banco de dados.

Para cada índice I gerado a partir da criptografia de um Banco de Dados B, cada registro R_i é criptografado separadamente e armazenado em uma tabela diferente. Este processo é uma função polinomial e requer um tempo considerável para funcionar, mas será necessário fazer este processo apenas uma vez para cada banco de dados solicitado pelo cliente. Esta função também permite a criptografia de várias tabelas ao mesmo tempo.

O processo de pesquisa, conforme indicado na Figura 2(b), será solicitado pelo cliente quando ele fornecer uma consulta a uma tabela necessária do banco de dados crip-

tografado contendo uma informação de palavra-chave. Depois disso, o servidor receberá essa consulta e executará o algoritmo de pesquisa SSE-DB para responder com uma lista com todos os identificadores correspondentes no banco de dados necessário que contém essa palavra-chave. É a forma mais segura de garantir que o cliente receberá o que procura e ainda não comprometerá a segurança dos dados sigilosos armazenados.

Para pesquisar no banco de dados criptografado gerado usando a função build, o algoritmo recalculará as mesmas chaves usadas para criptografar os dados anteriormente. Isso significa que, na função de pesquisa, teremos as mesmas funções de criptografia usadas para construir o banco de dados para possibilitar a geração novamente da cifra original para a palavra-chave fornecida. Após adquirir esta cifra AES, a função irá procurar correspondências em todas as tabelas solicitadas, salvando os índices das correspondências encontradas. Por fim, uma lista com todos os identificadores correspondentes será retornada ao cliente.

Mesmo com a necessidade de reconstruir a cifra e pesquisar registro por registro na tabela criptografada do banco de dados, este é um processo muito rápido e pode ser facilmente utilizado em aplicações da vida real, que é o foco deste trabalho. Um único cliente pode fazer várias requisições ao servidor e ainda receber as respostas em um tempo aceitável. Isso será provado mais adiante na seção Experimentos.

3.2. Processo de Anonimização

O PPM-Anon proposto neste artigo modifica um método de anonimização baseado em condensação [Aleroud et al. 2016] existente que gera um conjunto de dados sintéticos por meio do uso de informações dos dados originais.

A ideia de gerar um dataset sintético é deslocar os dados para outro espaço criando componentes, como o processo realizado na Análise de Componentes Principais (PCA) para reduzir a dimensionalidade dos dados. No entanto, neste contexto, estamos interessados em preservar o máximo de informações possível sobre os dados. Portanto, quando os dados estão sendo deslocados para outro espaço, todos os autovetores são usados.

O processo mencionado no parágrafo anterior pode ser revertido, o que significa que podemos retornar os dados ao espaço original. O método original [Aleroud et al. 2016] faz o processo inverso usando autovetores gerados aleatoriamente que possuem medidas estatísticas como média e desvio padrão iguais às medidas dos autovetores originais. Dessa forma, os dados deslocados de volta ao espaço original são os dados sintéticos que compartilham propriedades matemáticas com os dados originais.

O processo de geração de novos autovetores pode gerar erros de precisão em propriedades matemáticas como média e desvio padrão durante sua geração. O PPM-Anon surge para mitigar esses erros de precisão através da permutação do eixo de cada autovetor ao invés de gerar novos. A permutação não alterará medidas estatísticas como média, desvio padrão e outras.

A informação acima mencionada nos permite introduzir o algoritmo PPM-Anon. Na notação usada $D = \{D_1, D_2, ..., D_n\}$ representa os dados originais e $A = \{A_1, A_2, ..., A_n\}$ é o conjunto de dados resultante após a aplicação do método de anonimização. Em seguida, no Algoritmo 1, é o PPM-Anon proposto, que gera permutações aleatórias de cada autovetor para evitar erros de precisão e ainda preservar

as propriedades matemáticas dos dados.

Algoritmo 1 PPM-Anon

Entrada Dataset D
Saída Dataset Anonimizado A

1: D' ← centeredData(D)
2: CM ← covarianceMatrix(D')
3: E ← eingvaluesAndEignvectors(CM)
4: P ← PCA(D', E)
5: E ← randomShuffle(E)
6: A ← reversePCA(P, E)
7: **Retorna** A

Na linha 1 os dados são centralizados (pela função $centeredData$), então retiramos a média de cada linha. Na linha 2, a matriz de covariância D dos dados é calculada (através da função $covarianceMatrix$) e na linha 3 seus valores/vetores próprios são calculados e usados para executar a Análise de Componentes Principais (PCA) para deslocar o dados para um novo espaço criando componentes. Na linha 5, uma permutação aleatória dos autovetores é gerada (pela função $randomShuffle$), e a ideia é usar essa permutação para deslocar os dados para o espaço original (pela função $reversePCA$).

A ideia de usar uma permutação de autovetores é que ela mudará apenas as posições dos números, o que significa que medidas como média, desvio padrão, mediana, moda, variância e outras medidas estatísticas ainda serão as mesmas. Assim, esta abordagem manterá o significado dos dados após a realização do processo de anonimização.

Manter o significado dos dados significa que ainda é possível usar os dados sintéticos para resolver tarefas complexas sobre o conjunto de dados original, como treinar modelos de aprendizado de máquina ou talvez resolver problemas de mineração de dados usando os dados anônimos.

4. Experimentos Realizados

Esta seção apresenta os experimentos realizados para avaliar o desempenho do sistema proposto para proteção de dados, que está disponível no repositório do projeto[2], ou seja, a ideia dos experimentos realizados é mostrar a viabilidade da proposta em um ambiente real. Para a realização dos experimentos, foi definido um cenário realista com informações de um banco de dados e um ambiente de nuvem real, possibilitando uma avaliação adequada do sistema e seu impacto. A subseção 4.1 apresenta a configuração testbed dos experimentos, enquanto a subseção 4.2 discute os resultados.

4.1. Configuração dos Experimentos

Para garantir que o método proposto funcione mesmo em aplicações da vida real, vários experimentos foram feitos variando o tamanho dos bancos de dados utilizados e o número de consultas de palavras-chave feitas em cada um. Como ambiente de nuvem, utilizamos um Elastic Cloud Server (ECS) em Huawei Cloud[3] com a seguinte configuração: 12 vCPUS, 16GB de Memória RAM e Disco SSD de 40GB.

[2]A ser disponibilizado na versão final em caso de aceitação
[3]huaweicloud.com

Em relação ao banco de dados a ser protegido, foi implantado um banco de dados PostgreSQL e o Python Faker Package[4] para gerar os dados para preencher esse banco de dados. Esses dados são gerados acessando as propriedades com o nome do tipo de dados no gerador inicializado. Assim, controlando o tamanho do banco de dados podemos criar várias situações diferentes e ver em quais casos o sistema ainda é eficiente. A população da base de dados considera que também são salvas todas as possíveis palavras-chave nela encontradas, que serão utilizadas para realizar as requisições dos clientes na nuvem durante a avaliação do processo de busca.

Durante os experimentos o tempo de busca do SSE-DB e o tempo de processamento do PPM-Anom são considerados como métrica de avaliação, pois são os maiores impactos do sistema para os clientes a serem protegidos em um cenário realista. Em ambos os experimentos variamos o tamanho do banco de dados $B = \{125, 250, 500, 1000, 2000, 4000\}$, cada B_k representa um número diferente de linhas no banco de dados e o número de colunas foi fixado em $C = 10$.

4.2. Resultados

Esta subseção discute os resultados do experimento realizado, onde a Figura 3 ilustra o tempo de busca do SSE-DB para proteção de dados e a Figura 4 apresenta o tempo de processamento do PPM-Anom para anonimização de um conjunto de dados de dados confidenciais.

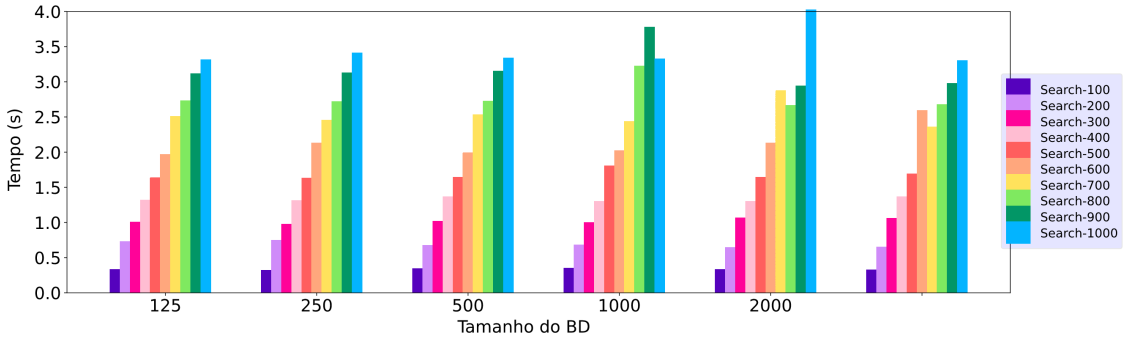


Figura 3. Tempo de pesquisa do SSE-DB

Conforme observado na Figura 3, o comportamento do tempo de busca é linear para a maioria das requisições realizadas pelo cliente, onde as diferenças de tempos de busca nos experimentos foram pequenas (cerca de 3 segundos considerando o número máximo de buscas) considerando o tamanho das bases de dados analisadas. É possível notar que uma quantidade maior de requisições resulta em um comportamento variável do tempo de busca, visto que a carga é alta quando comparada ao tamanho do banco de dados. Este ponto é ilustrado no caso de requisições de 600 quando o banco de dados é maior que 2000.

Além disso, é importante observar que o tempo médio de busca de um dado único é em torno de 3 milissegundos, ou seja, quando um dado específico é necessário para o sistema legado, ele pode ser buscado e recuperado sem impacto no desempenho do sistema quando uma comunicação ponta-a-ponta pela Internet é considerada [Sengupta et al. 2022, Flinta et al. 2020].

[4]pypi.org/project/Faker/

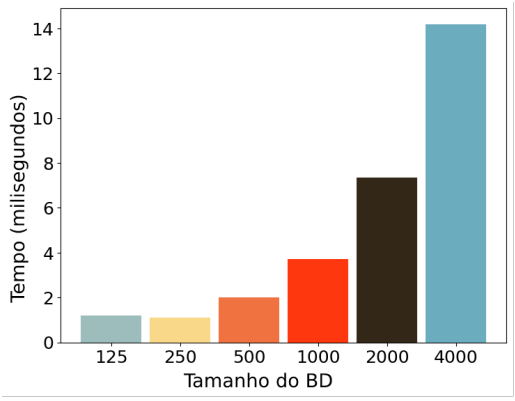


Figura 4. Tempo de processamento da PPM-Anon

Com relação aos resultados do PPM-Anom, mostrados na Figura 4, o tempo de processamento para realizar a anonimização cresce de acordo com o tamanho dos dados, já que o PPM-Anom executa diversas funções de alto custo computacional, como cálculo de matriz de covariância, eingvelues, eignvectors, PCA e permutação aleatória. Apesar do comportamento exponencial, o tempo de processamento é baixo quando o contexto do sistema é considerado. Por exemplo, no caso de banco de dados de tamanho maior, o tempo de processamento é de 14 milissegundos, muito menor do que um tempo de ida e volta usual na Internet [Sengupta et al. 2022, Flinta et al. 2020, Gomes et al. 2016], evitando um impacto negativo considerável na comunicação existente comportamento.

Com base nos experimentos realizados, os resultados indicam que o sistema proposto atinge seu objetivo de proteger, buscar e recuperar os dados em um ambiente de nuvem em um desempenho adequado. Dessa forma, permite questões de proteção de dados e preservação da privacidade.

5. Conclusão

Os sistemas existentes, que implementam serviços online como modelos de negócio, são alvo de tentativas de intrusão e, consequentemente, de possíveis situações de fuga de dados. Esta situação de vazamento de dados afeta a confidencialidade das empresas e impacta as leis de privacidade existentes (possivelmente resultando em taxas e sanções operacionais). Desta forma, estes sistemas existentes necessitam de ser protegidos por soluções de segurança que não interfiram no seu funcionamento normal.

Para lidar com esta situação, este trabalho apresenta um sistema para proteger dados sensíveis nos sistemas existentes, evitando problemas de vazamento de dados e violação de privacidade, sem qualquer alteração na mesma. O sistema proposto é baseado em técnicas de criptografia, denominada SSE-DB, e anonimização, denominada PPM-Anon. Assim, protege os sistemas existentes ao mesmo tempo em que possibilita a busca e recuperação de dados criptografados, bem como a disponibilização de dados anônimos que podem ser utilizados como entrada para outras soluções. Resultados de experimentos reais usando um ambiente de nuvem real sugeriram que a solução proposta é adequada para proteger os dados sem prejudicar o desempenho dos serviços existentes.

Como trabalho futuro, pretende-se avaliar outras abordagens de criptografia pesquisável e outras técnicas de anonimização, ampliando o pool de soluções de segurança

que podem ser implantadas pelo sistema e, consequentemente, melhorando o nível de segurança das empresas. Adicionalmente, tem-se por objetivo evoluir a ferramenta no que se refere a autenticação e atestação do agente em relação ao ambiente de nuvem, uma abordagem promissora é a utilização de TPM durante este processo.

Agradecimentos

Este projeto foi apoiado pelo programa PPI Softex, Acordo de Parceria nº 126/2022, financiado pelo Ministério da Ciência, Tecnologia e Inovações com recursos da Lei nº 8.248, de 23 de outubro de 1991. Adicionalmente, os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) do Brasil (N^o 303877/2021-9), Centro de Informação da Rede da América Latina e Caribe (LACNIC) através do FRIDA Grant pelo apoio financeiro.

Referências

Aleroud, A., Chen, Z., and Karabatis, G. (2016). Network trace anonymization using a prefix-preserving condensation-based technique (short paper). In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pages 934–942. Springer.

Aparajit, S., Shah, R., Chopdekar, R., and Patil, R. (2022). Data protection: The cloud security perspective. In *2022 3rd International Conference for Emerging Technology (INCET)*, pages 1–5.

Costa, W. L., Portela, A. L., and Gomes, R. L. (2021). Features-aware ddos detection in heterogeneous smart environments based on fog and cloud computing. *International Journal of Communication Networks and Information Security*, 13(3):491–498.

D, S. and Karibasappa, K. (2021). Enhancing data protection in cloud computing using key derivation based on cryptographic technique. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 291–299.

Flinta, C., Yan, W., and Johnsson, A. (2020). Predicting round-trip time distributions in iot systems using histogram estimators. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9.

Gomes, R. L., Bittencourt, L. F., Madeira, E. R., Cerqueira, E., and Gerla, M. (2016). A combined energy-bandwidth approach to allocate resilient virtual software defined networks. *Journal of Network and Computer Applications*, 69:98–106.

Gomes, R. L., Bittencourt, L. F., and Madeira, E. R. M. (2020). Reliability-aware network slicing in elastic demand scenarios. *IEEE Communications Magazine*, 58(10):29–34.

Gong, X., Chen, Y., Wang, Q., Wang, M., and Li, S. (2022). Private data inference attacks against cloud: Model, technologies, and research directions. *IEEE Communications Magazine*, 60(9):46–52.

GUPTA, I. and SINGH, A. K. (2020). An integrated approach for data leaker detection in cloud environment. *Journal of Information Science Engineering*, 36(5):993 – 1005.

Gupta, I., Singh, A. K., Lee, C.-N., and Buyya, R. (2022). Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions. *IEEE Access*, 10:71247–71277.

Huang, C., Chen, S., Zhang, Y., Zhou, W., Rodrigues, J. J. P. C., and de Albuquerque, V. H. C. (2022). A robust approach for privacy data protection: Iot security assurance using generative adversarial imitation learning. *IEEE Internet of Things Journal*, 9(18):17089–17097.

Kalugina, O., Barankova, I., and Mikhailova, U. (2020). Development of a tool for modeling security threats of an enterprise information system. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–5.

Li, J., Huang, Y., Wei, Y., Lv, S., Liu, Z., Dong, C., and Lou, W. (2019a). Searchable symmetric encryption with forward search privacy. *IEEE Transactions on Dependable and Secure Computing*, 18(1):460–474.

Li, J., Niu, X., and Sun, J. S. (2019b). A practical searchable symmetric encryption scheme for smart grid data. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Mann, Z. , Kunz, F., Laufer, J., Bellendorf, J., Metzger, A., and Pohl, K. (2021). Radar: Data protection in cloud-based computer systems at run time. *IEEE Access*, 9:70816–70842.

Mosca, E. E. P., Ribeiro, S., Urbano, A., Silva, D. S., and Gomes, R. L. (2023). Evaluation of security techniques in heterogeneous iot devices. LADC '22, page 91–94, New York, NY, USA. Association for Computing Machinery.

Rafique, A., Van Landuyt, D., Heydari Beni, E., Lagaisse, B., and Joosen, W. (2021). Cryptdice: Distributed data protection system for secure cloud data storage and computation. *Information Systems*, 96:101671.

Sengupta, S., Kim, H., and Rexford, J. (2022). Continuous in-network round-trip time monitoring. In *Proceedings of the ACM SIGCOMM 2022 Conference*, SIGCOMM '22, page 473–485, New York, NY, USA. Association for Computing Machinery.

Thabit, F., Alhomdy, S., and Jagtap, S. (2021). A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions. *International Journal of Intelligent Networks*, 2:18–33.

Wang, T., Yang, Q., Shen, X., Gadekallu, T. R., Wang, W., and Dev, K. (2022). A privacy-enhanced retrieval technology for the cloud-assisted internet of things. *IEEE Transactions on Industrial Informatics*, 18(7):4981–4989.

Yu, J., Moon, H., Chua, B.-L., and Han, H. (2022). Hotel data privacy: strategies to reduce customers' emotional violations, privacy concerns, and switching intention. *Journal of Travel & Tourism Marketing*, 39(2):213–225.