LGPD: Levantamento de Técnicas Criptográficas e de Anonimização para Proteção de Bases de Dados

Thiago R. Sousa[1] **Murilo Coutinho**[1] **Lilian Coutinho**[2] **Robson Albuquerque**[3]

[1]Centro de Pesquisa e Desenvolvimento para a Segurança da Comunicações (CEPESC)

[2]Escola de Inteligência (Esint), Agência Brasileira de Inteligência, Brasília, Brasil

[3]Programa de Pós-Graduação Profissional em Engenharia Elétrica (PPEE), Departamento de Engenharia Elétrica - Universidade de Brasília, Brasília, Brasil

Resumo. *A Lei nº 13.709, de 14 de agosto de 2018, conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD), veio para instituir princípios e regras para a proteção das pessoas naturais no que diz respeito ao tratamento de seus dados, principalmente no formato digital. Por essa razão, surge a necessidade de se estabelecer soluções tecnológicas capazes de atender às imposições da lei. Neste trabalho, apresentamos um levantamento de técnicas e ferramentas de anonimização e de criptografia que demonstram potencial para auxiliar no cumprimento da LGPD, no caso específico da proteção de bases de dados. Dentre as técnicas comparadas, percebe-se que não há nenhuma que atenda perfeitamente a todas as situações, seja por questões de desempenho ou por considerações de segurança. Ainda assim, conclui-se que, quando possível, essas soluções devem ser utilizadas, pois têm o potencial de aumentar significativamente a segurança dos sistemas e auxiliar no cumprimento da lei.*

1. Introdução

O uso da criptografia sempre foi importante para garantir a segurança das informações e das comunicações, proteger o sigilo de conhecimentos e de documentos sensíveis e, na era da informação, tornou-se fundamental para a preservação da privacidade de indivíduos no mundo digital. Ainda assim, tal importância tende a aumentar em virtude das novas leis para a proteção de dados que surgiram, primeiro na Europa com a *General Data Protection Regulation* (GDPR) e, posteriormente, em vários países ao redor do mundo.

No Brasil, em 14 de agosto de 2018, foi sancionada a Lei nº 13.709, conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD), com o objetivo de proteger o livre desenvolvimento da personalidade da pessoa natural e os direitos fundamentais de liberdade (CF, art. 5º, inciso IV) e de privacidade (CF, art. 5º, incisos X, XI e XII)[1]. Conforme o art. 1º da LGPD, a Lei dispõe sobre o tratamento de dados pessoais[2], inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado. Nesse sentido, destina-se também aos órgãos e entidades da Administração Pública e, por tratar-se de lei nacional, deve ser observada por todos os entes federados – União, Estados, Distrito Federal e Municípios.

[1]Está em trâmite a PEC nº 17/2019, que visa incluir a proteção de dados pessoais entre os direitos e garantias fundamentais. Além disso, no julgamento das ADIs 6387, 6388, 6389 e 6390, o STF a reconheceu como um direito fundamental autônomo.

[2]Conforme o art. 5º, inciso X da LGPD, "tratamento"deve ser compreendido como toda operação realizada com dados pessoais, como coleta, acesso, armazenamento etc.

Mesmo não estando em vigor, muitos órgãos e empresas passaram a buscar novos métodos e tecnologias visando a adequação do processamento de dados pessoais em conformidade à lei. Tais soluções tecnológicas têm largo escopo dentro da área de segurança da informação e das comunicações, incluindo-se aspectos da segurança cibernética. Naturalmente, a criptografia se destaca nesse cenário já que consiste em técnicas matemáticas que, se implementadas corretamente, garantem a impossibilidade de acesso não autorizado, mesmo por adversários com grande poder de processamento. Destaca-se, contudo, que segurança não é a mesma coisa que privacidade, e que apesar de ser difícil garantir a privacidade em sistemas inseguros, é possível não ter privacidade em sistemas com boas práticas de segurança.

Historicamente, a criptografia se concentrou em resolver o problema de comunicação clássico em que dois entes (Alice e Bob) buscam se comunicar de forma segura mesmo na presença de um adversário (Eve) capaz de acessar, ler, ou mesmo modificar o canal de comunicação. Hoje, pode-se afirmar que a criptografia é plenamente capaz de resolver esse problema, através de algoritmos de sigilo, troca de chaves, resumo e assinatura digital, que garantem a autenticidade, confidencialidade e integridade da comunicação. Portanto, soluções de comunicação por texto ou voz, videoconferências, acesso remoto via redes privadas virtuais (VPN) e similares podem, se bem implementadas, prover segurança suficiente para proteger tanto a privacidade dos usuários quanto a própria empresa que se responsabiliza pelos dados, garantindo a conformidade com a lei.

Há, entretanto, outras situações que não se encaixam no paradigma clássico da criptografia. Por exemplo, suponha a existência de uma base contendo dados privados que esteja armazenada em um servidor e uma série de analistas que necessitam acesso aos dados para realizarem trabalhos estatísticos. Em algumas situações, cada analista pode ser considerado como um potencial adversário, já que nem sempre podemos ter certeza que não irão acessar informações privadas que não deveriam ou mesmo vazá-las. Há também casos em que o próprio servidor pode ser visto como adversário, como quando dados privados são armazenados na nuvem com o intuito de utilizar o poder computacional dos servidores de grandes empresas.

Para superar esses desafios, uma série de técnicas e ferramentas têm sido desenvolvidas nos últimos anos. Porém, é justo afirmar que ainda não há nada estabelecido no mercado ou na academia como um padrão a ser seguido ou uma solução que resolva todos esses problemas de forma plenamente segura e eficiente. Sendo assim, este trabalho apresenta um levantamento de um subconjunto de tais técnicas, dentre elas Anonimização, Pseudonimização, Privacidade Diferencial, *Fully Homomorphic Encryption, Property Preserving Encryption* e *Oblivious Random Access Memory*. Adicionalmente, elencamos algumas ferramentas de busca e gerenciamento de dados cifrados baseadas nessas técnicas, discutindo sua utilização, segurança e ataques.

Neste trabalho, também apresentamos uma discussão sobre situações em que cada uma dessas técnicas podem ser utilizadas, muitas vezes em conjunto, para incrementar a segurança dos sistemas para a proteção das informações armazenadas em bases de dados e também em situações de proteção de privacidade ou anonimato quando bases de dados devem ser disponibilizadas para analistas externos. Conclui-se que, apesar da potencial existência de ataques, a segurança agregada ainda é significativa, o que dificulta a recuperação de informações, bloqueando a maior parte dos adversários e contribuindo na

garantia da manutenção da privacidade e do cumprimento da lei.

O restante do artigo é organizado da seguinte forma: na Seção 2, discutimos a relação entre a LGPD e a criptografia. Na Seção 3, descrevemos as principais técnicas de anonimização e criptografia que podem ser utilizadas para a proteção de bases de dados. Na Seção 4, apresentamos exemplos de ferramentas utilizadas nesse contexto e, na Seção 5, abordamos ataques que demonstram as limitações das técnicas e soluções atuais. Na Seção 6, discutimos algumas situações em que as técnicas apresentadas podem ser utilizadas, fazendo um comparativo entre elas e na seção 7 concluímos o trabalho.

2. LGPD e Criptografia

O modelo para a criação da lei de proteção de dados pessoais brasileira foi a *General Data Protection Regulation* (GDPR). Tal regulamentação tem aplicação nos países que integram a União Europeia (UE), impondo obrigações a quaisquer organizações que utilizem e coletem dados pessoais nos Estados membros.

O art. 5(1) da GDPR delimita os princípios relacionados ao processamento de dados pessoais, dentre os quais destaca-se o princípio da segurança, que determina o uso de técnicas apropriadas e de medidas organizacionais que garantam um nível de segurança apropriado aos riscos envolvidos, incluindo a proteção contra processamento ilegais ou não autorizados e contra perda, destruição ou danos. Para tanto, a criptografia dos dados pessoais é prevista em um rol exemplificativo de medidas que podem ser utilizadas de acordo com a natureza, o escopo, o contexto e as finalidades do processamento, bem como com os riscos aos direitos e liberdades individuais (GDPR, art. 32).

Nesse contexto, cada controlador ou processador deve avaliar os riscos que estão presentes no processamento de dados pessoais, como vazamentos, perda, destruição, alteração, acesso ou divulgação não autorizados, e outros eventos potencialmente causadores de danos físicos, materiais ou imateriais, devendo mitigar tais riscos e garantir um nível apropriado de segurança a partir da implementação de medidas, a exemplo da criptografia.

Assim como a GDPR, a LGPD inclui a segurança como um de seus dez princípios gerais[3] (art. 6º, VII), determinando que os agentes de tratamento devem adotar medidas técnicas, administrativas e de segurança aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas, como destruição, perda, alteração, comunicação, difusão ou qualquer outra forma de tratamento inadequado ou ilícito (art. 46). Se, por um lado, a lei brasileira não dispõe expressamente sobre o uso da criptografia como medida técnica sugerida, por outro é clara no sentido de que, em caso de incidente de segurança que possa acarretar risco ou dano relevante aos titulares, será avaliada eventual comprovação de que foram adotadas medidas técnicas adequadas para tornar ininteligíveis os dados pessoais afetados (art. 48, § 3º). Com efeito, as técnicas de criptografia permitem exatamente isso: tornar ininteligível o dado àqueles que não possuem as chaves criptográficas, ou seja, a qualquer pessoa que não está autorizada a acessá-lo.

Os controladores e processadores devem avaliar os riscos envolvidos nos tratamentos de dados pessoais que realizam, implementando medidas adequadas de proteção.

[3]O art. 6 da LGPD determina que as atividades de tratamento de dados pessoais deverão observar a boa-fé e os seguintes princípios: 1) finalidade; 2) adequação; 3) necessidade; 4) livre acesso; 5) qualidade dos dados; 6) transparência; 7) segurança; 8) prevenção; 9) não discriminação; e 10) responsabilização e prestação de contas.

Não foram poucos os incidentes de segurança ocorridos nas últimas décadas, e os danos gerados poderiam ter sido reduzidos ou mesmo evitados se os dados fossem criptografados. Cientes de suas responsabilidades, muitas empresas e órgãos públicos já estudam o uso de técnicas criptográficas para base de dados que atendam aos objetivos da LGPD já que a criptografia se apresenta, simultaneamente, como meio para garantir a conformidade à lei, para assegurar a proteção de direitos fundamentais e para a segurança dos sistemas.

3. Técnicas para a segurança de base de dados

3.1. Pseudonimização e Anonimização

Uma das formas mais simples para tentar proteger a privacidade de indivíduos em bases de dados é pela anonimização [Jain et al. 2016] ou pela pseudonimização [Stalla-Bourdillon and Knight 2016] que, em geral, não são consideradas técnicas criptográficas. Tais técnicas podem ser entendidas como a modificação ou destruição de informações em bases de dados de forma que não seja mais possível identificar os indivíduos. Isso é feito, em geral, de três maneiras: supressão, quando alguns tipos de dados sensíveis são eliminados da base de dados; substituição, quando dados sensíveis são substituídos por outros dados não sensíveis ou falsos; e generalização, quando dados específicos são substituídos por categorias mais genéricas, por exemplo, a idade de indivíduos podem ser substituídas por intervalos como "entre 20 e 30 anos".

A pseudonimização se difere da anonimização no sentido de que a pseudonimização permite reidentificar e recuperar os dados originais enquanto a anonimização não (ao menos teoricamente). Uma das técnicas de anonimização mais conhecidas é denominada k-anonimato [Sweeney 2002] e busca garantir que a informação para cada pessoa contida nos dados disponibilizados seja indistinguível de pelo menos $k - 1$ outros indivíduos cujas informações também existam nos dados disponibilizados. Outras técnicas foram desenvolvidas como evoluções do k-anonimato, em particular a l-diversidade [Machanavajjhala et al. 2007] e a t-proximidade [Li et al. 2007].

Existem situações em que essas técnicas simples podem ser utilizadas. Por exemplo, o processo de pseudonimização conhecido como *Tokenização*, que consiste na substituição de dados por informações aleatórias conhecidas como *tokens*, é utilizado com sucesso em algumas aplicações financeiras para proteger dados sensíveis como contas bancárias e números de cartões de crédito [Mattsson and Rozenberg 2013]. Entretanto, em grande parte dos casos, técnicas como o k-anonimato não podem ser consideradas verdadeiramente seguras nem capazes de garantir a privacidade. Isso porque existem ataques efetivos de deanonimização explorando propriedades da própria base de dados ou de outras bases cujos dados estão correlacionadas com as informações sobre os indivíduos que se deseja identificar. Um exemplo famoso neste sentido, ocorreu com dados anonimizados da empresa Netflix, quando um grupo de pesquisadores foi capaz de identificar usuários, desvendando até mesmo suas preferências políticas e outras informações sensíveis [Narayanan and Shmatikov 2008].

3.2. Privacidade Diferencial (PD)

A técnica de Privacidade Diferencial (*Differential Privacy*) pode ser utilizada em situações em que se deseja que um analista possa realizar análise de dados sem afetar a pri-

vacidade dos indivíduos que fazem parte da base de dados [Dwork et al. 2014][4]. A PD funciona a partir da adição de ruído aos dados de maneira que não seja possível ter certeza da informação particular de determinado indivíduo ao mesmo tempo em que se permite boas estimativas para características populacionais. Portanto, tais técnicas possibilitam um balanço entre utilidade e privacidade, sendo que quanto maior o ruído adicionado menor será a utilidade e maior a privacidade, e vice-versa.

As técnicas de PD podem ser úteis em diversos cenários para contribuir com as instituições e empresas a seguir a LGPD. Um possível caso de uso é por órgãos como o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), que possui dados privados de estudantes e instituições de ensino, mas que pode necessitar dar acesso a base de dados a pesquisadores internos ou externos. Neste caso, seria possível disponibilizar dados numéricos de interesse através da adição de ruído de forma a mascarar dados individuais, mas ainda possibilitando a detecção de tendências ou comparações entre instituições de ensino. Outra possibilidade é utilizar a PD para mitigar o risco de coleta de dados de usuários adicionando ruído já na origem da coleta, evitando maiores consequências no caso de comprometimento da base de dados da empresa, mas preservando a possibilidade de melhorar os produtos e serviços a partir da análise de comportamento dos usuários. Por exemplo, a empresa *Apple* já utiliza PD para coletar informações obtidas a partir de sugestões de palavras, sugestões de "emojis", gasto de energia do aparelho, entre outros[5]. No contexto da GPDR, o projeto SODA (`https://www.soda-project.eu`) foi criado com o objetivo de desenvolver um sistema seguro para o processamento de dados pessoais em grande escala em conformidade com a GDPR. Uma de suas vertentes de pesquisa é voltada para a utilização de PD em conjunto com computação multipartidária segura.

Apesar de ter potencial aplicabilidade em diversas áreas, deve-se levar em conta algumas limitações das técnicas de PD. Em primeiro lugar, a PD é uma definição e não um algoritmo. Algoritmos específicos são desenvolvidos para diferentes situações e seu funcionamento deve ser avaliado cuidadosamente em cada uma delas [Dwork and Nissim 2004, Blum et al. 2013]. De fato, a depender do tipo de estatística que se deseja computar, pode nem mesmo existir um método conhecido de se aplicar a PD. Outro problema é que a técnica não funciona bem para bases de dados pequenas já que, neste caso, o ruído necessita ser pequeno para que o modelo possua utilidade, o que reduz significativamente a privacidade. Adicionalmente, a técnica apresenta problemas quando existe alta variabilidade na distribuição dos dados, já que a quantidade de ruído necessária para a privacidade pode destruir completamente a utilidade dos dados.

Finalmente, em muitos casos há a necessidade de limitar o número de consultas feitas por um mesmo analista. Caso contrário, o analista poderia realizar repetidas consultas sobre o mesmo indivíduo, o que possibilita a eliminação do ruído e, como consequência, a exposição do valor que se busca proteger. Ainda, um analista mal intencionado, mesmo que limitado no número de consultas, poderia convencer outros analistas a fazer a mesma consulta ou mesmo criar contas falsas para executar este ataque [Dwork 2008].

[4]Outra técnica que pode ser utilizada com objetivos parecidos é chamada de *Quantitative Information Flow* e carrega muitas similaridades com a PD [Alvim et al. 2011].

[5]`https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf`

3.3. Fully Homomorphic Encryption (FHE)

FHE é uma tecnologia promissora do ponto de vista de segurança. Ela permite realizar operações no texto em claro, utilizando-se somente os seus cifrados correspondentes sem a necessidade de decifração. O resultado dessa operação permanece cifrado e pode ser transmitido de forma segura ao usuário que requisitou a operação. A técnica de FHE tem o potencial de garantir a privacidade em diversas situações nas quais isso não era possível como, por exemplo, o processamento de dados privados na nuvem sem que a empresa que disponibiliza o serviço tenha ideia das informações que realmente estão sendo processadas. Dessa forma, elimina-se o risco de comprometimento desses dados pela própria empresa, garantindo-se o cumprimento da lei[6].

FHE vem se tornando viável à medida que os computadores se tornam mais poderosos. Em 2012, um esquema de FHE implementado em [Gentry et al. 2012] gastou 40 minutos para calcular um bloco de AES (o que é seis ordens de magnitude mais lento do que o correspondente cálculo não homomórfico). Não obstante, sistemas homomórficos com funcionalidades restritas têm sido utilizados na prática, ou seja, eles permitem certos tipos de manipulação em dados cifrados, ao invés de funções arbitrárias como em FHE, e têm a vantagem de ter performance elevada, o que os tornam práticos. Por exemplo, o esquema de Paillier [Paillier 1999] permite realizar somas homomórficas de maneira eficiente.

Recentemente, um projeto em parceria com algumas universidades europeias disponibilizou uma biblioteca, denominada TFHE [Chillotti et al. 2016], que permite o cálculo de 10 tipos de portas binárias (AND, OR, XOR, ...) e leva em torno de 13 milissegundos em cada cálculo. Apesar de ser em torno de um bilhão de vezes mais lenta do que o simples cálculo não homomórfico, os algoritmos implementados com esta biblioteca já têm potencial aplicações em problemas reais de física e estatística [Ducas and Micciancio 2015]. Por exemplo, Bourse et. al. [Bourse et al. 2018] utiliza a biblioteca TFHE para treinar redes neurais sobre imagens cifradas homomorficamente obtendo taxa de acerto de 96% na base de dados MNIST em menos de 1.7 segundos. Adicionalmente, outros trabalhos demonstram que é possível aumentar a performance em até 26 vezes utilizando GPUs no processamento [Dai and Sunar 2015].

A conclusão é que na maioria dos casos, utilizar FHE para tratar com bases de dados arbitrárias e de tamanhos moderados ainda é proibitivo do ponto de vista computacional, não sendo adequado para casos em que a velocidade de consulta e resposta seja um requisito. Por isso, os sistemas práticos utilizam técnicas menos seguras [Fuller et al. 2017] em troca de maior performance, sendo úteis para sistemas onde um vazamento específico de informação não irá comprometer a segurança dos dados.

3.4. Property Preserving Encryption (PPE)

Ao invés de focar em realizar operações arbitrárias em dados cifrados, uma alternativa é a utilização de esquemas criptográficos que preservem certa propriedade, conhecidos como PPE. Uma modalidade de PPE se chama OPE (*Order Preserving Encryption*) e serve para realizar comparações de ordem. Este esquema basicamente cifra valores numéricos

[6]Outra técnica promissora é chamada de Functional Encryption [Boneh et al. 2011]. Ela generaliza a noção de cifração de chave pública e permite que o detentor de uma chave de decifração associada a uma função obtenha somente o valor dessa função avaliada sobre o texto em claro e nada mais.

de forma que a ordem dos dados em claro seja preservada em seus cifrados correspondentes. Por exemplo, se objetivamos realizar somente comparações entre valores sem a necessidade de saber quais são estes valores em claro, então um esquema de cifração que preserve a ordem é ideal para essa situação. Neste caso, um sistema de cifração com essa propriedade deve vazar somente a ordem dos dados cifrados [Popa et al. 2011]. Outros sistemas OPE podem vazar outros tipos de informações, além da ordem dos dados.

Outro PPE bastante conhecido é a cifração determinística. As técnicas de cifração modernas garantem que blocos de textos iguais sejam mapeados em cifrados diferentes com alta probabilidade, o que garante a importante propriedade de segurança semântica [Boneh and Shoup 2015]. Em contrapartida, esse tipo de cifração não permite que sejam realizadas buscas que contenham critérios de igualdade na base de dados. Na cifração determinística, blocos de textos iguais devem ser mapeados para os mesmos cifrados, abrindo mão da segurança semântica pela funcionalidade de busca. Além de aplicações em sistemas de bases de dados, a cifração determinística pode também ser aplicada em buscas de palavras-chaves em documentos cifrados.

Por último, temos ainda o *Format Preserving Encryption* (FPE), que basicamente cifra mensagens mantendo a mensagem cifrada no mesmo formato que a mensagem em claro (por exemplo cifrando um número de cartão de crédito válido e obtendo outro número de cartão de crédito válido). A grande vantagem desse esquema é que ele mantém protocolos de comunicação e de acesso intactos, já que os formatos dos dados são preservados. Por exemplo, sistemas que manipulem bases de dados com variáveis em determinados formatos poderiam acessar os dados cifrados com FPE da mesma forma [Brightwell and Smith 1997]. Em tais esquemas, há que se precaver com o tamanho do espaço de mensagens utilizado no esquema de cifração para prevenir ataques (ver p.ex. os ataques sobre o esquema FPE baseado em redes *Feistel* em [Durak 2017]).

Os esquemas PPEs são aplicáveis em situações onde não se faz necessário disponibilizar o dado bruto para que o utilizador destes possa analisá-los, e isso deve ser sempre avaliado pelas instituições. Em várias situações, informações de frequência ou de ordenação dos dados já são suficientes. A desvantagem é que os PPEs vazam informações importantes que podem ser utilizadas por atacantes em determinados cenários, como veremos na sequência. Esquemas PPE são eficientes e têm performance comparada a sistemas que não utilizam cifração. Já os esquemas FHE têm performance ordens de magnitude inferior à esquemas com vazamento controlado como os PPEs [Popa et al. 2011].

3.5. *Oblivious Random Access Memory* (ORAM)

ORAM [Goldreich and Ostrovsky 1996] pode ser utilizada para esconder padrões de acesso em bases de dados disponibilizadas em fontes externas. Esses padrões de acesso vazam informações sobre: qual dado está sendo acessado, quando ele foi acessado pela última vez, se o mesmo dado está sendo acessado e se o acesso é sequencial ou aleatório. Essas informações podem auxiliar o atacante a recuperar dados em claro mesmo que armazenados em bases de dados cifradas [Grubbs et al. 2019]. O princípio básico de ORAM é que, para que esses acessos sejam escondidos, os blocos de dados que são lidos precisam se mover, e não ficarem estáticos, caso contrário haveria vazamento de informações de frequência sobre os dados armazenados. Dessa forma, cada vez que um bloco é lido ele precisa ser realocado em outro local de forma aleatória. ORAM também ofusca

o tipo de operação (leitura ou escrita) que está sendo realizada. Isso é atingido através de uma sequência de leituras e outra de escritas. Dessa forma, o servidor que armazena os dados observa sempre um par de leituras e um par de escritas, e não conseguirá distinguir qual operação está sendo feita.

Na prática, ORAM vem sendo utilizada para aumentar a segurança de sistemas de busca em bases de dados. O sistema de busca proposto em [Boneh et al. 2013] sugere utilizar ORAM para recuperar os dados de registros cujos índices são conhecidos, escondendo assim informações de acesso ao servidor. Uma variante de ORAM foi proposta em [Mishra et al. 2018] para implementação de um sistema de busca indexada, cujo acesso aos índices é feito de forma oblívia. Outras aplicações de ORAM envolvendo bases de dados pode ser encontrada em [Fuller et al. 2017, Seção III.A.5]. Em geral, soluções que implementam ORAM são mais lentas, porém oferecem maior grau de segurança em situações nas quais existe risco inerente de que informações sobre padrões de acesso possam ser utilizadas, o que é sempre o caso quando bases de dados sensíveis precisam ser acessadas em servidores inseguros.

4. Ferramentas de busca e gerenciamento de dados cifrados

Sistemas que permitem buscas mais diversas em bases de dados são, em geral, baseados na combinação das técnicas citadas na Seção 3, dentre as quais algumas vazam certos tipos de informação. Além disso, cada um deles tem sua segurança avaliada considerando cenários específicos que devem ser levados em consideração quando implementados em sistemas reais, já que, na prática, informações adicionais geradas por compiladores, metadados necessários para o funcionamento do sistema, ou outros subprodutos gerados pelos sistema, são informações passíveis de serem utilizadas por atacantes.

Existe uma diversidade de sistemas de bases de dados, cada um com o seu próprio conjunto de bases primitivas que podem ser combinadas para prover suas funcionalidades. Por isso, existe a demanda de uma variedade de ferramentas de buscas protegidas para realização dessas operações primitivas de maneira segura [Fuller et al. 2017]. No contexto de base de dados, geralmente focamos em realizar pesquisas em linguagem padrão SQL. Cada operação em SQL pode ser escrita como uma combinação de operações primitivas, tais como: verificações de igualdade, comparações de ordenação e agregações. Sendo assim, podemos utilizar os esquemas de FHE ou PPEs para realizar tais operações primitivas e as compor para formar operações SQL. Aplicações baseadas em esquemas PPEs são chamadas de BoPETs (*Building on Property-revealing EncrypTion*).

Uma plataforma que utiliza PPEs em sua construção é a CryptDB [Popa et al. 2011]. Adicionalmente, na CryptDB o sistema homomórfico aditivo de [Paillier 1999] é utilizado para calcular somas. O ponto mais importante no quesito segurança é que esta biblioteca tem o princípio de cifração em camadas, e revela ao servidor somente o necessário para que o cliente seja capaz de realizar a operação SQL desejada. Quando um cliente envia determinada busca SQL, esta passa por uma etapa de pré-processamento em um servidor proxy seguro. Dependendo das comparações existentes na busca SQL , a base de dados armazenada no servidor inseguro é ajustada para a camada de cifração adequada. A CryptDB tem a vantagem de ser de fácil integração a sistemas de bases de dados preexistentes, pois as consultas feitas pelos usuário são primeiro pré-processadas por um servidor proxy e enviadas ao servidor que possui a base de

dados cifrada, sem interferir no funcionamento do gerenciador da base. Apesar de prover confidencialidade dos dados com nível de segurança controlado, a CryptDB não provê integridade dos mesmos.

Um alternativa mais atrativa no quesito segurança é baseada na utilização de busca indexada. A indexação de registros é algo comum em bases de dados que diminui os tempos de busca. Neste caso, podemos cifrar a base de dados com um esquema de cifração seguro e as pesquisas SQL são feitas diretamente nos índices, e não na própria base. Essa estrutura de índices captura a informação dos dados na base por meio de resumos criptográficos de seus valores. Funções de resumo não são invertíveis com alta probabilidade, desde que espaço de entrada da função não seja pequeno o suficiente para permitir o adversário testar todas as entradas possíveis até encontrar o resumo correspondente. Portanto, elas asseguram que, de posse desses resumos, o atacante não possa reconstruir os dados em claro.

Um dos sistemas baseados em busca indexada é o Arx [Poddar et al. 2019], que cifra os dados sensíveis no servidor com segurança semântica. Uma alternativa ao Arx é o BlindSeer [Pappas et al. 2014], um sistema que implementa busca de forma indexada, mas utilizando uma ferramenta diferente do Arx, baseada em *Bloom Filters*. A indexação é feita de forma individual, porém pode-se realizar indexação de colunas simultaneamente, com o custo de aumentar-se o armazenamento. O BlindSeer não tem suporte nativo a vários clientes, mas uma solução natural para esse problema é que os clientes compartilhem uma chave única desconhecida pelo servidor inseguro. Ao contrário do Arx, no BlindSeer as informações sobre a busca também são protegidas. Por outro lado, ele perde no quesito performance e funcionalidade em relação ao Arx.

Os sistemas acima estão sujeitos a ataques que exploram padrões de acesso, pois cada um deles apresenta vazamentos específicos que propiciam diferentes vulnerabilidades. Em tais sistemas, uma segurança indireta é a detecção da presença de um atacante online o mais rápido possível por meio de análise de tráfego malicioso [Pimenta Rodrigues et al. 2017], o que, neste caso, é facilitado pela característica específica dos padrões de acesso necessários para executar tais ataques. Existem outros sistemas com funcionalidades similares/complementares que podem ser consultados em [Fuller et al. 2017, Tabelas II e V], que inclui um sumário com a performance de cada sistema, sua segurança e funcionalidades.

Para se proteger contra ataques de padrões de acesso, vários sistemas de busca em bases de dados utilizam a técnica ORAM ou tecnologias baseadas nessa (ver p.ex. [Fuller et al. 2017, Seção III.A.5]). Eles provém segurança adicional, porém com a contrapartida de o sistema prover menos funcionalidades e/ou ter maior custo computacional. Em particular, o sistema Oblix [Mishra et al. 2018], que implementa buscas indexadas, utiliza ORAM de maneira a não somente esconder padrões de acesso do cliente acessando o servidor, como também deste quando acessando sua própria memória interna. No artigo, ele é aplicado para sistemas de busca em dados cifrados, porém também pode ser combinado com sistemas de bases de dados com pesquisas indexadas, pois o sistema de índices é dinâmico, ou seja, é passível de inserções e deleções de registros. As buscas em dados cifrados realizadas no Oblix retornam uma quantidade fixa de resultados, mesmo que existam mais ou menos resultados que satisfaçam o critério de busca. A escolha dos resultados retornados é feita através de uma métrica que quantifica o quão próximo o re-

sultado está da busca (o que é comum em aplicações de busca de palavras-chaves). Isso esconde do atacante o tamanho do conjunto de dados retornados, o que é algo atrativo no quesito segurança. Por outro lado, dependendo da aplicação em mente, o fato de existir um limite na quantidade de registros retornados em pesquisas pode ser uma limitação.

5. Ataques

Uma série de ataques práticos contra sistemas que implementam busca em bases de dados pode ter sucesso em recuperar informações em claro sobre os dados armazenados. Os objetivos são, em geral, a recuperação dos próprios dados cifrados e/ou do tipo de busca que está sendo realizada. Os ataques exploram informações de frequência, metadados armazenados no servidor, informações parcialmente conhecidas, padrões de acesso, entre outros. Em [Fuller et al. 2017, Tabela III], pode ser encontrado um sumário com tipos de ataque, seus objetivos, tempos de execução e condições necessárias para implementação. Vale ressaltar que os sistemas apresentados na seção anterior focam em resolver o problema de confidencialidade e podem não implementar integridade, como é o caso da CryptDB. Além da integridade ser uma propriedade essencial para quem utiliza os dados, sua ausência pode ser explorada por atacantes, como nos ataques realizados por administradores maliciosos de bases de dados [Akin and Sunar 2014].

Os ataques de inferência objetivam recuperar dados cifrados por meio do vazamento de informação inerente ao sistema de cifração utilizado e pelo uso adicional de informações públicas. Estes dados públicos são, em geral, provenientes de censo ou de redes sociais, que estejam correlacionados de alguma forma com a informação visada. Para avaliar a segurança de um sistema vulnerável a ataques de inferência, é necessário avaliar o impacto da utilização de informações públicas correlacionadas com os dados protegidos. Isso deve ser feito até mesmo em cenários em que o atacante obtém acesso não autorizado a bases de dados de outras instituições que estejam correlacionados de alguma forma com os dados que se pretende obter.

Em [Naveed et al. 2015] ataques reais contra sistemas de cifração que utilizam PPEs são discutidos e aplicados. Dois ataques são explorados: um baseado em informações de frequência das observações e outro na sua distribuição. Ataques de inferência são propostos de forma geral, ainda assim foram aplicados ao sistema CryptDB [Popa et al. 2011] para demonstrar sua praticidade. O ataque implementado foca em recuperar informações confidenciais em bases de dados contendo informações hospitalares e obteve sucesso significativo quando o atributo em questão possuía um número moderado de valores possíveis (p. ex. risco de mortalidade). Portanto, deve-se ter cuidado em se proteger bases de dados utilizando PPEs sem que se avalie os riscos associados. Além dos vazamentos inerentes em sistemas que utilizam PPEs, há também que se precaver de ataques que focam nos padrões de acesso existentes nas comunicações entre os clientes e o servidor que armazena a base de dados (ver Seção 3.5). Informações contidas em logs de transação quando o Arx é utilizado em conjunto com gerenciadores de bases de dados na prática podem quebrar a segurança semântica dos dados [Grubbs et al. 2017], o que exige que a gravação de logs de transação ou de outros metadados sejam desativados [Poddar et al. 2019]. Um resultado mais geral é que informações sobre padrões de acesso, quando unidas com o conhecimento da distribuição dos dados contidos na base ou com o vazamento de alguns registros, podem permitir ao atacante reconstruir boas aproximações da base de dados original [Grubbs et al. 2019].

6. Discussão

Uma das grandes dificuldades na questão da proteção de base de dados é que existem diversos tipos diferentes de sistemas. Cada um desses sistemas demanda tipos diferentes de proteção, não havendo, portanto, solução única para todos os casos. Primeiro, é necessário entender o contexto, verificar quais informações devem ser protegidas e qual o custo computacional aceitável. Mesmo assim, podem existir casos em que não tenhamos uma resposta satisfatória. De fato, as técnicas apresentadas neste trabalho nem sequer podem ser consideradas concorrentes, pois servem a propósitos distintos e, muitas vezes, podem ser utilizadas de forma complementar. De maneira geral, podemos identificar 4 tipos de usos diferentes:

- **Uso 1:** Situações em que se deseja proteger a privacidade ou anonimato dos usuários quando a base de dados for utilizada em análises por estatísticos ou outros profissionais.
- **Uso 2:** Situações em que se deseja proteger a base de dados armazenada em um servidor considerado inseguro, mas mantendo a possibilidade de a empresa (ou instituição) realizar buscas e/ou manipular dados.
- **Uso 3**: Proteção contra ataques baseados em padrões de acesso em bases de dados mantidas em um servidor inseguro e manipuladas por uma empresa (ou instituição).
- **Uso 4:** Proteção criptográfica de base de dados cujas informações de cada usuário são cifradas com chaves que estão em sua posse, protegendo a privacidade do usuário contra a própria empresa (ou instituição) que armazena ou processa seus dados. Neste contexto, cada usuário só terá acesso e poderá manipular os dados que detém.

Apesar das técnicas FHE e ORAM serem mais seguras, em alguns casos, a baixa performance acaba sendo uma característica impeditiva do seu uso na prática. Alternativas, como PPE, apresentam vulnerabilidade que as deixam sujeitas a certos tipos de ataques, porém têm potencial de aumentar a segurança de bases de dados existentes.

Conclui-se que, devido ao compromisso inerente entre segurança e funcionalidade, a escolha e o uso de tais ferramentas devem ser feitos com cautela e com o devido entendimento das limitações das ferramentas atuais. De forma geral, a utilização de tais esquemas deve sempre levar em consideração o vazamento de informação que eles possuem, avaliando quais riscos ainda serão ameaças para a instituição.

Em um cenário como esse, muitos podem ter a tentação de não utilizar nenhum tipo de defesa, porém essa não é a melhor estratégia. Em face da LGPD, empresas e órgãos que negligenciarem a proteção dos dados pessoais podem ser responsabilizados. Lembramos que segurança é um processo composto por diversos elementos que se somam, e não uma solução mágica. Assim, recomendamos o uso das técnicas de FHE e ORAM em situações nas quais seja possível aceitar o custo computacional associado. Já nos casos em que a performance é importante, recomendamos o uso de técnicas como PD ou PPEs. Em particular, como vimos na Seção 4, existem algumas ferramentas disponíveis utilizando PPEs. Apesar da potencial existência de ataques, a segurança agregada ainda é significativa, o que dificulta a recuperação de informações, bloqueando a maior parte dos adversários e contribuindo na busca da manutenção da privacidade e do cumprimento da lei.

Técnica	Anonimização	PD	FHE	PPE	ORAM
Desempenho	Alto	Alto	Baixo	Alto	Baixo
Segurança	Baixa	Média	Alta	Média	Alta
Usos	1	1	2,4	2	3

Tabela 1. Comparação entre as técnicas apresentadas neste trabalho. Deve-se destacar que as características definidas para cada técnica não refletem todos os casos. Portanto, podem existir exceções a depender da situação e do uso. Adicionalmente, algumas dessas técnicas podem ser utilizadas em conjunto para prover soluções mais robustas. Em particular, ORAM é geralmente combinada com outras técnicas quando utilizada nos protocolos de sistemas de busca em bases de dados para esconder padrões de acesso. O item segurança objetiva comparar as técnicas em relação a quantidade de informação revelada sobre o texto em claro através dos dados cifrados.

7. Conclusão

Neste trabalho foram apresentadas algumas técnicas e ferramentas de criptografia e anonimização que podem ser utilizadas na proteção de bases de dados. Pela discussão apresentada, concluímos que a escolha e implementação de tais técnicas não é uma tarefa fácil, principalmente pela inexistência de soluções padronizadas que possam ser utilizadas facilmente por diferentes instituições. Existem alternativas que possuem alto grau de segurança, como FHE e ORAM, porém o custo de performance a ser pago pode tornar o sistema inviável na prática. Outras opções, como a anonimização, PD e PPEs, são mais eficientes, mas acabam vazando certos tipos de informação, o que gera vulnerabilidades que podem ser exploradas por atacantes. Apesar das dificuldades e limitações, as técnicas de proteção de bases de dados mencionadas neste trabalho são altamente recomendadas, pois têm o potencial de aumentar significativamente a segurança dos sistemas e auxiliar no cumprimento da LGPD.

Referências

[Akin and Sunar 2014] Akin, I. H. and Sunar, B. (2014). On the difficulty of securing web applications using CryptDB. In *IEEE Fourth International Conference on Big Data and Cloud Computing*, pages 745–752.

[Alvim et al. 2011] Alvim, M. S., Andrés, M. E., Chatzikokolakis, K., and Palamidessi, C. (2011). On the relation between differential privacy and quantitative information flow. In *International Colloquium on Automata, Languages, and Programming*, pages 60–76. Springer.

[Blum et al. 2013] Blum, A., Ligett, K., and Roth, A. (2013). A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25.

[Boneh et al. 2013] Boneh, D., Gentry, C., Halevi, S., Wang, F., and Wu, D. J. (2013). Private database queries using somewhat homomorphic encryption. In *International Conference on Applied Cryptography and Network Security*, pages 102–118. Springer.

[Boneh et al. 2011] Boneh, D., Sahai, A., and Waters, B. (2011). Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference*, pages 253–273. Springer.

[Boneh and Shoup 2015] Boneh, D. and Shoup, V. (2015). A graduate course in applied cryptography. *Draft 0.2*.

[Bourse et al. 2018] Bourse, F., Minelli, M., Minihold, M., and Paillier, P. (2018). Fast homomorphic evaluation of deep discretized neural networks. In *Annual International Cryptology Conference*, pages 483–512. Springer.

[Brightwell and Smith 1997] Brightwell, M. and Smith, H. (1997). Using datatype-preserving encryption to enhance data warehouse security. In *20th National Information Systems Security Conference Proceedings (NISSC)*.

[Chillotti et al. 2016] Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. (2016). TFHE: Fast fully homomorphic encryption library. https://tfhe.github.io/tfhe/.

[Dai and Sunar 2015] Dai, W. and Sunar, B. (2015). cuhe: A homomorphic encryption accelerator library. In *International Conference on Cryptography and Information Security in the Balkans*, pages 169–186. Springer.

[Ducas and Micciancio 2015] Ducas, L. and Micciancio, D. (2015). FHEW: bootstrapping homomorphic encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 617–640. Springer.

[Durak 2017] Durak, F. (2017). *Cryptanalytic study of property-preserving encryption*. PhD thesis, Rutgers University-School of Graduate Studies, New Brunswick, NJ, USA.

[Dwork 2008] Dwork, C. (2008). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

[Dwork and Nissim 2004] Dwork, C. and Nissim, K. (2004). Privacy-preserving datamining on vertically partitioned databases. In *Annual International Cryptology Conference*, pages 528–544. Springer.

[Dwork et al. 2014] Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

[Fuller et al. 2017] Fuller, B., Varia, M., Yerukhimovich, A., Shen, E., and Hamlin, A. (2017). SoK : Cryptographically Protected Database Search. *IEEE Symposium on Security and Privacy (SP)*, pages 172–191.

[Gentry et al. 2012] Gentry, C., Halevi, S., and Smart, N. P. (2012). Homomorphic evaluation of the AES circuit. In *Annual Cryptology Conference*, pages 850–867. Springer.

[Goldreich and Ostrovsky 1996] Goldreich, O. and Ostrovsky, R. (1996). Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 43(3):431–473.

[Grubbs et al. 2019] Grubbs, P., Lacharité, M.-S., Minaud, B., and Paterson, K. G. (2019). Learning to reconstruct: Statistical learning theory and encrypted database attacks. In *IEEE Symposium on Security and Privacy (SP)*, pages 1067–1083.

[Grubbs et al. 2017] Grubbs, P., Ristenpart, T., and Shmatikov, V. (2017). Why your encrypted database is not secure. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, pages 162–168.

[Jain et al. 2016] Jain, P., Gyanchandani, M., and Khare, N. (2016). Big data privacy: a technological perspective and review. *Journal of Big Data*, 3(1):25.

[Li et al. 2007] Li, N., Li, T., and Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering*, pages 106–115.

[Machanavajjhala et al. 2007] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es.

[Mattsson and Rozenberg 2013] Mattsson, U. and Rozenberg, Y. (2013). Tokenization in payment environments. US Patent App. 13/761,009.

[Mishra et al. 2018] Mishra, P., Poddar, R., Chen, J., Chiesa, A., and Popa, R. A. (2018). Oblix: An efficient oblivious search index. In *IEEE Symposium on Security and Privacy (SP)*, pages 279–296.

[Narayanan and Shmatikov 2008] Narayanan, A. and Shmatikov, V. (2008). Robust deanonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125.

[Naveed et al. 2015] Naveed, M., Kamara, S., and Wright, C. V. (2015). Inference attacks on property-preserving encrypted databases. *Proceedings of the ACM Conference on Computer and Communications Security*, 2015:644–655.

[Paillier 1999] Paillier, P. (1999). Public-Key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer.

[Pappas et al. 2014] Pappas, V., Krell, F., Vo, B., Kolesnikov, V., Malkin, T., Choi, S. G., George, W., Keromytis, A., and Bellovin, S. (2014). Blind Seer: A scalable private DBMS. In *IEEE Symposium on Security and Privacy*, pages 359–374.

[Pimenta Rodrigues et al. 2017] Pimenta Rodrigues, G. A., de Oliveira Albuquerque, R., Gomes de Deus, F. E., De Oliveira Júnior, G. A., García Villalba, L. J., Kim, T.-H., et al. (2017). Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection. *Applied Sciences*, 7(10):1082.

[Poddar et al. 2019] Poddar, R., Boelter, T., and Popa, R. A. (2019). Arx: an encrypted database using semantically secure encryption. *Proceedings of the VLDB Endowment*, 12(11):1664–1678.

[Popa et al. 2011] Popa, R. A., Redfield, C. M., Zeldovich, N., and Balakrishnan, H. (2011). CryptDB: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 85–100.

[Stalla-Bourdillon and Knight 2016] Stalla-Bourdillon, S. and Knight, A. (2016). Anonymous data v. personal data-false debate: An eu perspective on anonymization, pseudonymization and personal data. *Wis. Int'l LJ*, 34:284.

[Sweeney 2002] Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570.