Gestão de Privacidade no Armazenamento de Dados do Paciente em Registros Médico-Hospitalares Eletrônicos

Fabio Demo da Rosa[1], **Walter Priesnitz Filho**[1]

[1]Colégio Técnico Industrial – Universidade Federal de Santa Maria (UFSM)
97.105-900 – Santa Maria – RS – Brasil

fabio.demo@redes.ufsm.br, walter@redes.ufsm.br

Abstract. *The increasing use of electronic records for users' control, permit to observe that any data, general or personal, interests to someone (including opponents or individuals with bad intentions). It is fundamental to ensure security and also the privacy of such data, so the owners of these informations won't be harmed in case third parties gain access to its private data. To ensure that, it is aimed to implement a service protecting stored personal information, through cryptographic methods that cypher sensitive information and turn it into readable information only for those who have permission to access it. By doing so, it minimizes the possible risks of an attack and/or the unauthorized use of this information by opponents.*

Resumo. *O aumento constante do uso de registros eletrônicos para controle de usuários, possibilita observar que quaisquer dados, sejam gerais ou pessoais, interessam a alguém (incluindo adversários ou indivíduos mal-intencionados). É fundamental garantir a segurança e a privacidade de tais dados, desse modo os proprietários destas informações não serão prejudicados caso terceiros obtenham acesso aos seus dados pessoais. Para garantir isto, busca-se implementar um serviço que atue na proteção de informações pessoais armazenadas, através de métodos criptográficos, que cifram informações sensíveis e as tornem legíveis somente a quem tiver permissão para acessá-las. Assim, minimizando os riscos de uma possível obtenção por meio de ataques e/ou uso não autorizado destas informações por adversários.*

1. Introdução

Devido aos baixos custos, e sendo uma opção mais amigável ao meio ambiente, os *Electronic Health Records* (EHR) ou Registros Médico-Hospitalares Eletrônicos vêm ocupando dos registros médicos feitos em papéis [Patrício et al. 2011]. Os ambientes médicos e/ou hospitalares contém uma coleção de dados sensíveis sobre cada um de seus pacientes. Estes dados podem estar sujeitos à vulnerabilidades [Win 2005], como por exemplo: acesso não autorizado (seja em nuvem, banco de dados ou servidores), no canal de comunicação, ou entre outras partes envolvidas no gerenciamento de dados.

Uma vez que informações importantes referentes a condições físicas, resultados de exames, entre outras, forem acessadas indevidamente, os proprietários de tais informações (que acabaram de serem comprometidas), poderão passar por problemas, como: personificação, danos morais e até mesmo extorsão [Landry et al. 2011]. Um ambiente, como o descrito pode tornar-se alvo para utilização indevida das informações, caso não apresente as proteções adequadas.

Buscando manter a privacidade, é possível recorrer a métodos e funções criptográficas, que devem ser implementadas para dificultar acessos indevidos a dados pessoais ou ao banco de dados [Kellaris et al. 2017]. Assim é possível manter informações íntegras e oferecer uma forma de segurança para todos os usuários do sistema. Porém, é possível que aumentar a segurança das informações, implique na perda de usabilidade [Kainda et al. 2010], logo é importante garantir que a usabilidade não se torne muito baixa ao garantir o aumento da segurança dos dados. Observando os fatores citados acima, propõe-se uma forma de manter a privacidade no armazenamento de registros eletrônicos médico-hospitalares, visando a melhoria da integridade e da confidencialidade do processo de armazenamento, protegendo os dados de forma eficaz.

O presente trabalho está divido da maneira a seguir: A Seção 2 apresenta o contexto geral e a importância deste trabalho. Os trabalhos relacionados, contemplando as principais tecnologias que podem ser aplicadas em um escopo semelhante ao proposto, são apresentados na Seção 3. O métodos utilizados no desenvolvimento do trabalho são apresentados na Seção 4. Na Seção 5 pode ser observada a implementação proposta no cenário médico-hospitalar. Os resultados obtidos pelos experimentos realizados são apresentados e discutidos na Seção 6. Na Seção 7 estão contidas as conclusões sobre o presente trabalho.

2. Contexto

Ao realizar buscas por sistemas ou métodos que são utilizados para preservar a segurança e/ou privacidade no ambiente proposto, não foi possível encontrar respostas concretas, visto que cada entidade pode operar nas informações da forma que achar melhor, desde que atenda requisitos presentes em leis, normas e boas práticas. As premissas especificadas para tais ambientes, são o *Integrating the Healthcare* [Schabetsberger et al. 2010] e o *Master Patient Index* [American Health Information Management Association 2018], sendo estes padrões de integração que não dispõem sobre a segurança de forma abrangente (i.e., não definindo o que deve ou não ser usado).

Os diversos sistemas presentes em hospitais, contam com arquivos contendo dados sensíveis de um número indeterminado de pacientes, onde estes dados podem ser eventualmente roubados, adulterados ou completamente deletados [Li et al. 2018]. Seja por estarem em formulários de papel, registros eletrônicos em texto plano ou até mesmo que tenham algum mecanismo de proteção falho. Isto faz com que estes apresentem vulnerabilidades por não serem gerenciados de forma adequada. Tal fato pode chamar a atenção de atacantes, que pretendam usar estes dados em benefício próprio e/ou com intuitos danosos ao proprietário das informações.

Pode-se observar que tanto leis estrangeiras, como a *General Data Protection Regulation* (GDPR) [European Union 2016], como sua equivalente nacional, a Lei Geral de Proteção de Dados (LGPD) [Senado Federal 2018], tem buscado adequar-se com o intuito de prover segurança e privacidade para o usuário final. Elas buscam estabelecer normas e primitivas que definem como os dados pessoais devem ser tratados, armazenados, entre outros tipos de manipulação destes. Baseado nestas normas, toda empresa ou instituição que armazenar dados pessoais ou realizar qualquer manipulação/gerenciamento sobre estes, deverá tratá-los conforme especificado nestas leis. Caso contrário, a empresa terá de responder judicialmente por não prover a privacidade/segurança mínima, as quais um

indivíduo tem direito.

Tendo em vista os fatores citados acima, propõe-se uma solução para atuar no armazenamento de informações sensíveis em ambientes médico-hospitalares. Para atingir este objetivo, foi realizada uma pesquisa de métodos criptográficos, buscando identificar e aplicar a melhor solução ao ambiente e condições propostos. Tais métodos transformam as informações sensíveis em texto cifrado, de forma que não revele nada que possa ser facilmente capturado/decifrado. Também atuando na preservação da privacidade dos indivíduos, já que as informações pessoais possuem valor inestimado para seus proprietários.

3. Trabalhos Relacionados

Entre as diversas tecnologias citadas em [Zhang et al. 2017], descreve-se noções e usos da *Searchable Asymmetric Encryption* (SAE) no contexto de aplicações médico-hospitalares, além de caracterizá-la e citar exemplos onde pode ser aplicada com o intuito de preservação da segurança dos dados.

No trabalho de [Zhang et al. 2018] é apresentado um esquema que usa busca de criptografia assimétrica (SAE) para ambientes em nuvem em uma arquitetura *multi-data owner* (MDO), permitindo que qualquer pessoa crie textos cifrados pesquisáveis sob a chave pública, enquanto a chave privada correspondente cria os *Trapdoors*.

No trabalho de [Chen et al. 2019] é apresentado um esquema, que atua em um servidor de dados médico-hospitalares em nuvem, definido como *Dynamic Searchable Symmetric Encryption*, visando alcançar a *forward privacy* (o servidor de nuvem pode deduzir que o novo documento adicionado tem uma palavra-chave que foi pesquisada no passado, mantendo a privacidade) e *backward privacy* (consultas não podem ser executadas em documentos excluídos).

Em [Li et al. 2017], é proposto um sistema no qual o índice de pesquisa das informações é adicionado ao *blockchain*. Os EHRs reais são armazenados em um servidor de nuvem pública em um formato criptografado. Assim, quando os usuários desejam acessar suas informações, eles precisam se autenticar para o proprietário dos dados e assim obter a autorização e a chave para decifrar seus dados.

Para proteger também a privacidade e segurança dos dados em sistemas de saúde móveis (o qual fornece serviços médicos e informações através de tecnologias modernas de comunicação móvel e de sensores), [Ma et al. 2018] desenvolveram uma eficiente *Searchable Encryption*. A análise de segurança indica o esquema desenvolvido pelos autores é considerado seguro, por resistir a diversos ataques aos quais foi submetido.

4. Metodologia

Para alcançar os objetivos propostos, a *Searchable Encryption* (SE) foi implementada para a proteção da privacidade dos dados dos pacientes, armazenando estas informações em um banco de dados cifrado. *Salt*s, foram concatenados a valores *hash* para evitar que as trocas de atributos de verificação revelassem informações sensíveis através de técnicas de pré-computação dos valores *hash*. Ainda com o intuito de proteger a privacidade dos dados do paciente no ambiente médico-hospitalar, propõe-se também um protótipo de operação neste cenário, onde haverão consultas dos registros/informações do paciente através das tecnologias citadas anteriormente.

As seguintes especificações técnicas referem-se à máquina utilizada na execução dos testes de desempenho: Sistema Operacional Linux Mint 18.3 Cinnamon 64-*bit*s, processador Intel® Core™ i5 4200U 1.6Ghz, 8GB DDR3L 1600 MHz.

O *framework* Flask[1] foi utilizado para estabelecer uma aplicação *web* com o intuito de fornecer uma maior interação entre possíveis pacientes e o seu acesso a suas informações contidas no banco de dados cifrados. O sistema de gerenciamento de banco de dados utilizado foi o PostgreSQL versão 9.5.17.

Além da utilização da linguagem de programação Python em sua versão 3, as bibliotecas utilizadas no desenvolvimento da implementação foram: `pyDH`[2] (biblioteca para o Diffie-Hellman), `Crypto`[3] (biblioteca para o AES, ElGamal e para função de geração aleatória de números), `psycopg2`[4] (biblioteca que realiza a conexão do Python ao PostgreSQL), `time` (biblioteca para fazer a cronometragem do tempo de execução), `datetime` (biblioteca para alterar o formato das datas), `binascii` (biblioteca de conversão dos valores para hexadecimal ou para binário), `os` (biblioteca para funções do sistema operacional), além de uma implementação existente do DES que foi importada como uma biblioteca/módulo para o programa.

A *Searchable Encryprion* (SE) permite terceirizar o armazenamento de um banco de dados em um servidor não confiável, permitindo pesquisas [Bost and Fouque 2019]. Assim, os dados serão armazenados em texto cifrado, podendo haver buscas sem revelar informações sensíveis (a menos que as informações precisem ser decifradas/acessadas, como em ambientes médico-hospitalares). Em sua definição, a SE necessita de um *trapdoor* para permitir a busca dos dados. Ao utilizá-lo, as buscas não revelam informações sobre seus conteúdos [Curtmola et al. 2011], adicionando assim uma camada extra de proteção e dificultando sua reversão.

5. Implementação

Para realizar a cifra na SE, foi utilizada o *Advanced Standard Encryption* (AES). O AES é uma especificação que realiza a cifra de dados eletrônicos, estabelecido pelo *National Institute of Standards and Technology* (NIST), tendo um bloco de cifra com 16 *bytes* de tamanho, porém três diferentes tamanho de chave de acordo com o método de operação: 128 (AES-128), 192 (AES-192) e 256 (AES-256) *bit*s [Rijmen and Daemen 2001].

Já para criação do *trapdoor* da SE foi utilizado o *Data Encryption Standard* (DES). Sendo um sistema criptográfico que opera com codificação de bloco, projetada para cifrar e decifrar blocos de dados de 64 *bit*s usando uma chave de mesmo tamanho [Singh 2013].

O *Locality Sensitive Hashing* é uma tecnologia que permite transformar um objeto específico em um valor *hash*, utilizada para comparações e/ou similaridade entre valores [Chierichetti and Kumar 2015]. O *Salt*, é uma tecnologia que garante segurança contra ataques pré-computados (como *Birthday Paradox*, *rainbow tables*, entre outros), por meio da concatenação de um valor aleatório ao valor *hash* [Gauravaram 2012].

[1]https://flask.palletsprojects.com/
[2]https://pypi.org/project/pyDH/
[3]https://pypi.org/project/pycrypto/
[4]https://pypi.org/project/psycopg2/

Após passar pela etapa de confirmação, onde o usuário autoriza o acesso aos seus dados privados, e considerando que este é um usuário legítimo, a comunicação com o servidor deve ser restrita aos dois pontos da comunicação, seja o usuário que deseja verificar seus registros. ou um terceiro que aguarda autorização para ler os dados disponíveis.

Tendo como exemplo a Figura 1, se uma requisição for feita por um médico (ou outro profissional médico) para um atributo que possa identificar um único usuário, como o número do Registro Geral (RG) ou um número de identificação único. Se os dados desse usuário puderem ser acessados por este profissional, uma solicitação é feita diretamente ao usuário para liberar o acesso aos outros atributos solicitados (condição médica, doenças e outras condições físicas semelhantes).

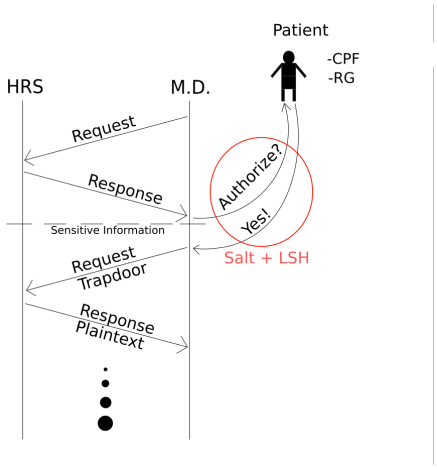

Figura 1. Processo de comunicação do protótipo.

Se o usuário consentir em liberar esses dados, a comunicação continuará, o que acabará revelando informações sensíveis/confidenciais. A primeira parte da comunicação, assim como a autorização do paciente, será feitas através de valores *hashes* concatenados ao *Salt*, assim evita-se possíveis ataques pré-computados (como *rainbow tables*). Se ocorrer uma interferência ou um ataque, a finalidade dos dados que passam por esse canal é para autenticação em comparação à similaridade.

Ao ultrapassar o ponto da comunicação onde informações sensíveis são reveladas, todas buscas e pesquisas no banco de dados cifrado são feitas através da *Searchable Symmetric Encryption*, onde os *trapdoors* são enviados aos HRS (*Health Record System*), e os valores em texto planos são retornados ao M.D. (*Medicinae Doctor*), assim a comunicação continua enviando todos os atributos solicitados, até que um dos elementos da troca de dados, HRS ou M.D., finalize esta comunicação.

A geração de 1.000 registros (contendo: nome, data de nascimento, gênero, cidade, nacionalidade e nome da mãe/pai) foi realizada através de um gerador aleatório, para que tais registros passem pela função SE, cifrando todos dados e armazenando-os.

6. Resultados

Em geral, não há ataques práticos que permitam que um oponente, sem o conhecimento da chave usada na criptografia, decifre as informações criptografadas pelo AES, considerando que as funções estão corretamente implementadas [Rijmen and Daemen 2001]. Em

função disso, optou-se pelo uso do AES-256, devido à maior quantidade de operações realizadas durante o ato de cifrar todas as informações contidas nos registros dos pacientes.

Na Tabela 1, é possível observar os tempos de cifra de todos os campos dos 1.000 registros, utilizando o AES-256, onde o maior tempo observado foi relacionado aos nomes (nomes completos), devido ao seu tamanho maior que as informações de outros campos.

Tabela 1. Tempo da realização de cifra com o AES-256 e do *Trapdoor* de cada campo de todos os 1.000 registros em milissegundos

	Nome	Data de Nasc.	Gênero	Cidade	Nacionalidade	Nome da Mãe/Pai
Mínimo	8,216	8,131	8,086	8,08	8,087	8,078
Máximo	18,396	11,252	11,244	12,891	11,869	14,814
Média	10,396	8,519	8,358	8,527	8,346	9,295

Já na Figura 2, constam os tempos de busca sequencial do AES-256 em segundos, realizada através da SE, a cada 10 posições da base de dados (para então obter-se uma melhor análise do que ocorre em cada ponto no banco de dados). Nota-se que quanto mais avançada a posição no banco de dados cifrado, maior será o tempo em relação a posições iniciais. Os picos observados em alguns trechos, são uma decorrência da maneira como o Python executa o gerenciamento dos *buffers* na execução das funções utilizadas.

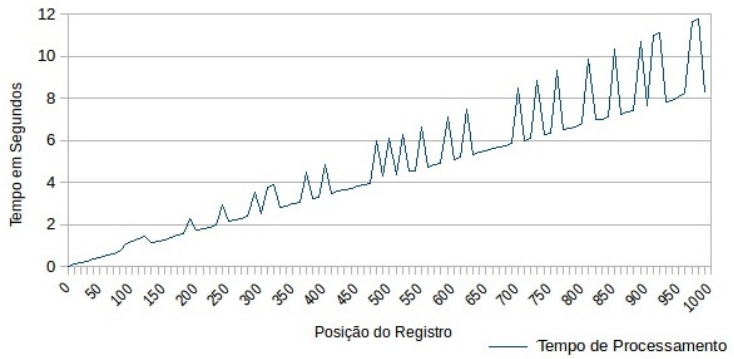

Figura 2. Tempos de busca (s) pelos valores cifrados de 100 nomes

A Tabela 2 demonstra os tempos de busca em texto plano e em texto cifrado (através da SE), no banco de dados PostgreSQL. Foram analisados 100 valores, do início ao fim do banco de dados. Embora os tempos sejam superiores na busca cifrada e busca da SE, esta demora trará benefícios como: segurança, confidencialidade e confiabilidade dos dados, já que estes serão armazenados de forma a impossibilitar sua fácil compreensão.

Tabela 2. Tempos (s) de busca de 100 nomes cifrados (AES-256) e em texto plano

	Texto Plano	Cifrado
Mínimo	0,000075	0,010183
Máximo	0,000217	11,79726
Média	0,00009978	4,71638807

7. Considerações Finais

A criptografia mostra ser uma técnica eficiente de proteção de dados ao criar uma barreira extra, dificultando o processo de reversão do texto cifrado caso não se tenha conhecimento das chaves utilizadas, da ordem dos processos. Também auxiliando em casos de tentativas de acessos não autorizados, mantendo assim a privacidade dos dados sensíveis.

As *rainbow tables* são utilizadas para atacar funções de *hash* ao tentar pré-calcular o texto original. Na fase da autenticação e autorização, ao concatenar o valor em *hash* com um valor (*Salt*), reduz-se significativamente a probabilidade de se obter o texto plano.

Notou-se que embora os tempos de execução de uma busca em um banco de dados cifrados possa demorar significantemente mais que uma simples busca em texto plano, o banco cifrado torna-se muito mais vantajoso ao garantir: a segurança, a confidencialidade e a privacidade, ao cifrar as informações armazenadas.

A *Searchable Encryption* utilizada neste trabalho pode ser considerada *Searchable Symmetric Encryption* (SSE), devido ao emprego do AES para realizar as cifras, sendo este considerado um algoritmo simétrico. A geração de *trapdoors* foi feita através da utilização do *Data Encryption Standard* (DES) para que, além de buscas nos dados cifrados, seja criada uma proteção extra até o ponto onde este possa ser decifrado.

Ainda existem pontos a serem melhorados neste trabalho. O sistema de buscas no banco de dados pode ser otimizado nas buscas com a SE (através do uso de ordenação do banco de dados e busca binária, como observada em [Cormen et al. 2017]). Também há margem para verificar outras tecnologias que possam ser empregadas de forma a melhorar a segurança e privacidade das informações sensíveis em registros médico-hospitalares.

Referências

American Health Information Management Association (2018). Fundamentals for Building a Master Patient Index/Enterprise MasterPatient Index (Updated). *Journal of AHIMA*, pages 1–15.

Bost, R. and Fouque, P.-A. (2019). Security-efficiency tradeoffs in searchable encryption. *Proceedings on Privacy Enhancing Technologies*, 4:132–151.

Chen, L., Lee, W.-K., Chang, C.-C., Choo, K.-K. R., and Zhang, N. (2019). Blockchain based searchable encryption for electronic health record sharing. *Future Generation Computer Systems*, 95:420–429.

Chierichetti, F. and Kumar, R. (2015). Lsh-preserving functions and their applications. *Journal of the ACM (JACM)*, 62(5):33.

Cormen, T., Leiserson, C., and Rivest, R. (2017). *Algoritmos*. Elsevier Brasil.

Curtmola, R., Garay, J., Kamara, S., and Ostrovsky, R. (2011). Searchable symmetric encryption: improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934.

European Union (2016). Regulation 2016/679 of the European parliament and the Council of the European Union. *Official Journal of the European Communities*, 2014(March 2014):1–88.

Gauravaram, P. (2012). Security analysis of salt—— password hashes. In *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, pages 25–30. IEEE.

Kainda, R., Flechais, I., and Roscoe, A. (2010). Security and usability: Analysis and evaluation. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 275–282. IEEE.

Kellaris, G., Kollios, G., Nissim, K., and O'Neill, A. (2017). Accessing data while preserving privacy. *arXiv preprint arXiv:1706.01552*.

Landry, J. P., Pardue, J. H., Johnsten, T., Campbell, M., and Patidar, P. (2011). A threat tree for health information security and privacy. In *AMCIS*.

Li, H., Yang, Y., Dai, Y., Bai, J., Yu, S., and Xiang, Y. (2017). Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data. *IEEE Transactions on Cloud Computing*.

Li, H., Zhu, L., Shen, M., Gao, F., Tao, X., and Liu, S. (2018). Blockchain-based data preservation system for medical data. *Journal of medical systems*, 42(8):141.

Ma, M., He, D., Khan, M. K., and Chen, J. (2018). Certificateless searchable public key encryption scheme for mobile healthcare system. *Computers & Electrical Engineering*, 65:413–424.

Patrício, C. M., Maia, M. M., Machiavelli, J. L., and Navaes, M. d. A. (2011). O prontuário eletrônico do paciente no sistema de saúde brasileiro: uma realidade para os médicos? *Scientia Medica*, 21(3).

Rijmen, V. and Daemen, J. (2001). Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pages 19–22.

Schabetsberger, T., Wozak, F., Katt, B., Mair, R., Hirsch, B., Hörbst, A., et al. (2010). Implementation of a secure and interoperable generic e-health infrastructure for shared electronic health records based on ihe integration profiles. In *Medinfo*, pages 889–893.

Senado Federal (2018). Lei Geral Brasileira de Proteção de Dados. *PROJETO DE LEI DA CÂMARA Nº 53, DE 2018 (nº 4.060/2012, na Câmara dos Deputados)*, pages 1–226.

Singh, G. (2013). A study of encryption algorithms (rsa, des, 3des and aes) for information security. *International Journal of Computer Applications*, 67(19).

Win, K. T. (2005). A review of security of electronic health records. *Health Information Management*, 34(1):13–18.

Zhang, R., Xue, R., and Liu, L. (2017). Searchable encryption for healthcare clouds: a survey. *IEEE Transactions on Services Computing*, 11(6):978–996.

Zhang, Y., Zheng, D., and Deng, R. H. (2018). Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet of Things Journal*, 5(3):2130–2145.