Aplicação de Criptografia Homomórfica na Mineração de Dados em Fluxos de Roteadores de Borda na Internet

Felipe M. F. de Assis[1], **Evandro L. C. Macedo**[1], **Luís F. M. de Moraes**[1]

[1]Laboratório de Redes de Alta Velocidade (RAVEL) - PESC/COPPE/UFRJ

{assis,evandro,moraes}@ravel.ufrj.br

Abstract. *Homomorphic Cryptography appears as a solution for data manipulation in a private preserving manner. On the other hand, the rising amount of data being daily generated makes Data Mining techniques more and more attractive. Therefore, this work has the creation of two methods for association rules generation for distributed databases as the objective, in a way that every participant mantains their share private. Real data extracted from RedeRio/FAPERJ's backbone edge routers is used to validate the proposal.*

Resumo. *A Criptografia Homomórfica surge como solução para a manipulação de dados de maneira a respeitar a privacidade. Por outro lado, a crescente quantidade de dados sendo gerada diariamente torna técnicas de Mineração de Dados cada vez mais atraentes. Com isso, este trabalho tem como objetivo a criação de dois métodos para a geração de regras de associação para bases de dados distribuídas, de forma que cada participante tenha sua parte privada. Dados reais retirados dos roteadores de borda do backbone da RedeRio/FAPERJ são usados para validar a proposta.*

1. Introdução

Com a difusão de novas tecnologias de comunicação e o aumento do número de dispositivos conectados à rede, o volume de dados gerados ultrapassou o patamar de 95 zettabytes, mostrando ainda uma tendência de crescimento futuro [Taylor 2022]. Por outro lado, a noção da necessidade de privacidade de dados e sua importância é cada vez mais difundida, com essa noção sendo explicitada até mesmo por artefatos legais como a Lei Geral de Proteção de Dados Pessoais (LGPD).

A Mineração de Dados (*Data Mining*) é a ciência de extrair conhecimento útil de grandes repositórios de dados [Chakrabarti et al. 2006]. Uma das técnicas mais simples, porém eficientes, da Mineração de Dados é a geração de Regras de Associação, responsável por achar padrões em conjuntos de dados. A existência de diversas técnicas de Mineração de Dados, somada a grande disponibilidade de informação na atualidade evidenciam uma oportunidade para a extração de conhecimento útil de tais informações. Este trabalho toma como exemplo dados de roteadores de borda, que podem armazenar informações de uma grande quantidade de fluxos de rede. Entretanto, se quisermos preservar a privacidade de dados entre roteadores, encontra-se um problema, pois tradicionalmente a geração de Regras de Associação envolve manipulação dos dados em claro. Com isso, o problema proposto é definido como a geração de Regras de Associação distribuída de modo a manter a privacidade entre seus participantes.

Com o objetivo de extrair informações de dados privados, surge a Criptografia Homomórfica, que se trata de sistemas criptográficos na qual operações no texto puro são preservadas pela função de encriptação [Henry 2008]. A vantagem deste tipo de característica é a possibilidade de manipulação de dados criptográficos e a extração de características de forma a preservar a privacidade dos dados envolvidos.

O objetivo principal do trabalho é a combinação da Criptografia Homomórfica com a Mineração de Dados, com o intuito de criar dois sistemas nos quais dois ou mais participantes podem cooperar para gerar regras de associação em comum sem a necessidade de compartilhar suas bases de dados. Para a contextualização em um cenário real, dados de um roteador de borda são utilizados, onde o problema estabelecido se trata de diferentes roteadores cooperando de forma privada com o intuito de gerar regras de associação sobre seus fluxos.

O restante do artigo está organizado da seguinte maneira. A Seção 2 comenta os trabalhos relacionados ao tema do artigo. A Seção 3 explica o conceitos teóricos utilizados. Na Seção 4, são apresentados os métodos propostos. Os resultados são apresentados na Seção 5. Por fim, a Seção 6 comenta trabalhos futuros e conclui o artigo.

2. Trabalhos Relacionados

Apesar da aplicação na prática de Criptografia Homomórfica ser relativamente recente, já existem diversos trabalhos sobre o tópico. Em sua tese, Henry [Henry 2008] apresenta e discute extensamente aspectos teóricos e práticos da Criptografia Homomórfica e os mecanismos por trás de seu funcionamento. Frikken [Frikken 2007], em seu trabalho, usa Criptografia Homomórfica para calcular a união de dois conjuntos sem revelar o conteúdo individual de cada conjunto. Para isso, os conjuntos são codificados como polinômios, para então representar a união em função de operações homomórficas.

O problema reuperar o i-ésimo bit de informação de uma string em um banco de dados sem informar o valor de i para o banco é chamado de Recuperação Privada de Dados. Yi [Yi et al. 2012] utiliza de Criptografia Homomórfica para resolver este problema, se aproveitando das propriedades da operação XOR. Por fim, temos o trabalho de Drozdowski [Drozdowski et al. 2019] que utiliza da Criptografia Homomórfica de maneira bem mais palpável a um usuário comum. Ele usa essa técnica para gerar um método de reconhecimento facial que esconde do servidor as informações do rosto do usuário.

3. Fundamentação Teórica

A fim de compreender as soluções propostas, as próximas seções tratam de conceitos de criptografia e mineração de dados.

3.1. Criptografia Homomórfica

A ideia por trás da Criptografia Homomórfica reside em saber o que é um homomorfismo. Uma função f é dita um homomorfismo se, para duas operações $\circ$ e $\bullet$, temos $f(x_1 \circ x_2) = f(x_1) \bullet f(x_2)$.Um sistema criptográfico com função de encriptação E e função de decriptação D é dito homomórfico se D é um homomorfismo. Ou seja, para dois textos cifrados c_1 e c_2 , temos $D(c_1 \circ c_2) = D(c_1) \bullet D(c_2)$. Note que se E é um homomorfismo, D também será. Além disso, é possível que $\circ$ e $\bullet$ sejam a mesma operação.

3.2. Criptografia de Limiar

Na Criptografia Assimétrica é necessária a existência de chaves para a execução dos algoritmos de encriptação e decriptação. Uma é chamada de chave pública e tem esse nome por não ser uma informação secreta, podendo ser usada por qualquer um para a encriptação. A outra, é chamada de chave privada, onde somente quem é autorizado a detêm e pode realizar a decriptação.

A Criptografia de Limiar surge como uma variação dessa ideia, na qual múltiplos participantes possuem chaves privadas diferentes e a decriptação deve ser feita com um número mínimo de chaves diferentes. Esse tipo de sistema pode ser especialmente útil na Criptografia Homomórfica, para garantir que somente a informação gerada pelas operações homomórficas está sendo decriptada, e não os dados originais.

3.3. Mineração de Dados

Existem diversas técnicas de Mineração de Dados, porém uma das mais intuitivas é a geração de Regras de Associação [Agrawal et al. 1993]. Esse tipo de regra é direta, pois associa a frequência de certos conjuntos de itens com a frequência de outros conjuntos em um banco de dados transacional. Podemos usar um mercado como exemplo de geração de Regras de Associação. Por exemplo, se nos registros de um mercado temos que um cliente que compra banana e maça também leva junto uma uva em grande parte das vezes, podemos inferir a regra $\{\text{banana,maçã}\} \Rightarrow \{\text{uva}\}$.

Para formalizar a ideia de Regra de Associação, são necessários alguns conceitos. Primeiramente, *Transação* se trata (neste trabalho) de uma linha na base de dados. Uma *Regra* $S_1 \Rightarrow S_2$ diz que se o conjunto S_1 aparecer na Transação, S_2 também aparecerá. Note que o conjunto aparecer significa que todos os seus itens estão presentes na transação. O *Suporte* é definido como uma maneira de medir se o conjunto aparece um número de vezes o suficiente para ser significativo na criação de regras. O Suporte de um conjunto S é definido por $Suporte_S = \frac{\text{nº de transações onde o conjunto S aparece}}{\text{nº total de transações}}$. Por fim, temos a *Confiança*, que é necessária para dizer se a relação de dois conjuntos S_1 e S_2 é forte o suficiente para a geração da regra. É definida por $\text{Confiança}_{S_1 \Rightarrow S_2} = \frac{Suporte_{S_1 \cup S_2}}{Suporte_{S_1}} = \frac{\text{nº de transações onde o conjunto } S_1 \cup S_2 \text{ aparece}}{\text{nº de transações onde o conjunto } S_1 \text{ aparece}}$.

O algoritmo para a criação de regras acontece da seguinte forma: escolhe-se um valor de corte s para o Suporte e um valor de corte c para a confiança. São calculados os valores de Suporte para todos os conjuntos na base e avançam para a próxima fase os que tem suporte maior ou igual a s. Um conjunto S que passou para a próxima fase é desmembrado em todas as possíveis combinações de dois conjuntos S_1 e S_2 onde $S = S_1 \cup S_2$ e a Confiança é calculada de acordo com a separação. Se o valor obtido é maior ou igual a c, a regra $S_1 \Rightarrow S_2$ é aceita.

4. Métodos de Criptografia Homomórfica na Mineração de Dados

Com o arcabouço teórico explicado, é possível então combinar os conceitos em prol da solução do problema proposto.

4.1. Regras de Associação Distribuídas

Baseado no trabalho de [Kaosar et al. 2012], é possível obter os valores de suporte e confiança de determinados conjuntos em uma base de dados distribuída de maneira

a preservar a privacidade das partes. Sejam n participantes desejando cooperar e seja $i \in 1, \ldots, n$ seu índice. Seja também $|DB_i|$ o número de transações na base de dados do participante i e seja c_i a contagem de vezes que o conjunto S aparece nessa base de dados. Assim, o suporte do conjunto de todos os participantes será:

$$Suporte_S = \frac{\text{nº de transações onde o conjunto S aparece}}{\text{nº total de transações}} = \frac{\sum_{i=1}^{n} c_i}{\sum_{i=1}^{n} |DB_i|}$$

Com uma pequena manipulação algébrica, se quisermos ter que isso seja maior que um suporte mínimo $s/100$, temos que:

$$Suporte_S \geq s/100 \Leftrightarrow \frac{\sum_{i=1}^{n} c_i}{\sum_{i=1}^{n} |DB_i|} \geq s/100 \Leftrightarrow 100 \sum_{i=1}^{n} c_i \geq s \sum_{i=1}^{n} |DB_i|$$

$$\sum_{i=1}^{n} 100c_i - s|DB_i| \geq 0 \tag{1}$$

Da mesma forma, a confiança de uma regra $S_1 \Rightarrow S_2$, com l_i sendo a contagem de ocorrência de $S_1 \cup S_2$ e L_i a contagem de ocorrência de S_1, é:

$$\text{Confiança}_{S_1 \Rightarrow S_2} = \frac{\text{nº de transações onde o conjunto } S_1 \cup S_2 \text{ aparece}}{\text{nº de transações onde o conjunto } S_1 \text{ aparece}} = \frac{\sum_{i=1}^{n} l_i}{\sum_{i=1}^{n} L_i}$$

Para este valor ser maior que uma confiança mínima $c/100$, temos da mesma maneira que acima:

$$\sum_{i=1}^{n} 100l_i - cL_i \geq 0 \tag{2}$$

Note que o valor dentro de cada somatório depende somente do participante i, logo pode ser calculado pelo participante i e a soma total pode ser feita por operações homomórficas, ou seja, soma nos textos cifrados.

4.2. Métodos Propostos

Os dois métodos desenvolvidos foram criados com o auxílio da biblioteca OpenFHE [Badawi et al. 2022], que provê primitivas de criptografia homomórfica. O criptossistema utilizado se trata do Brakerski/Fan-Vercauteren (BFV) [Halevi et al. 2019]. Para a simulação de um ambiente real, foram criadas 4 instâncias de *container* Docker, cada uma com uma das fatias determinadas na seção anterior.

4.2.1. Método Padrão

O Método Padrão foi implementado como *baseline* para comparação com os métodos propostos. Este método se baseia em uma versão distribuída sem privacidade, no qual um dos participantes é escolhido para receber os dados das frequências dos conjuntos dos outros participantes, afim gerar regras de associação e transmiti-las para os demais.

4.2.2. Primeiro Método

Assim como no método padrão, um dos participantes é escolhido para ser o responsável por realizar as operações. Desta vez, outro participante também é necessário pela criação do criptossistema, além de ser o dono das chaves pública e privada. O responsável pelas chaves envia os parâmetros do sistema, assim como a chave pública para os demais. Todos os participantes encriptam o arquivo com seu termo da Inequação (1) e mandam para o participante responsável pelas operações. O participante soma todos os valores recebidos assim como na Seção 4.1. Os valores são enviados de volta ao detentor da chave privada e comparados com 0, para a verificação se o suporte de cada conjunto é maior que o valor mínimo pré-estabelecido. Com o intuito de mascarar o valor total somado, para evitar que o detentor das chaves tente descobrir algo sobre a soma, o operador multiplica os valores por inteiros aleatórios entre 1 e 5, o que mantém a propriedade de ser maior ou igual a 0. Assim, os conjuntos com suporte mínimo são encontrados e o detentor das chaves envia essa informação de volta aos outros participantes. Da mesma forma, as regras com confiança mínima são calculadas.

4.3. Segundo Método

Este método também utiliza as inequações desenvolvidas na Seção 4.1. Porém, ao invés de usar uma só chave privada como no método anterior, utiliza também de Criptografia de Limiar para a geração de chaves. Por isso, é necessária uma etapa anterior de comunicação para a geração e distribuição de chaves. Além disso, não utiliza do artifício de multiplicação por um inteiro aleatório, pois necessita de geração de chaves especiais para o produto, o que será explorado em um método futuro. De resto, o método segue como o anterior.

5. Avaliação Experimental

Para a avaliação dos métodos propostos utilizou-se uma base de dados de fluxos de roteadores de borda, os quais são descritos a seguir.

5.1. Dados Utilizados

Os dados utilizados no trabalho para a geração de regras de associação são fluxos extraídos de um roteador de borda da Rede-Rio/FAPERJ . Tais fluxos são tratados e disponibilizados pela plataforma IPTraf [de Assis et al. 2021]. Um fluxo tratado é composto pelos seguintes campos: endereço de origem, endereço de destino, porta de origem, porta de destino, protocolo, flags tcp, número de pacotes, número de bytes e horário de início.

Foram separados 20GB destes fluxos processados para este estudo, divididos em arquivos que representam intervalos de 5 minutos. Para suprir a necessidade de um ambiente distribuído, os fluxos foram divididos em 4 bases de dados com tamanhos distintos.

Cada intervalo foi alocado em uma destas 4 bases, sendo seus valores de aproximadamente 46%, 28%, 18% e 8% do total original de intervalos. Note que os campos dos fluxos devem ser trabalhados antes de serem analisados. Por exemplo, o número de portas pode ir de 1 a 65535. Esse valor é muito elevado, fazendo nenhuma ocorrência ter frequência grande o suficiente para passar pela etapa do suporte. Por isso, alguns campos foram agrupados e alguns descartados.

5.2. Resultados

Por se tratar de uma versão inicial, apenas uma parte dos dados coletados foi utilizada para na qual foi aplicado o método padrão para estabelecer uma *baseline*. Após isso, ambos os métodos desenvolvidos foram aplicados nesta mesma porção de dados, gerando o mesmo conjunto de regras, assim mostrando sua validade.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou duas soluções distintas para o problema da geração de Regras de Associação em bases de dados distribuídas, para tomarem proveito de grandes quantidades de dados, ao mesmo tempo preservando a privacidade dos envolvidos. Os métodos foram validados em uma base de dados real composta por fluxos dos roteadores de borda da Rede-Rio/FAPERJ. Como passos futuros, pretende-se evoluir o trabalho para uma versão ampliada, apontando as diferenças entre os métodos, novas avaliações e uma implementação de um terceiro método que combina os dois métodos apresentados.

Referências

Agrawal et al. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD*, pages 207–216.

Badawi, A. A. et al. (2022). OpenFHE: Open-Source Fully Homomorphic Encryption Library. Cryptology ePrint Archive, Paper 2022/915.

Chakrabarti et al. (2006). Data mining curriculum: A proposal (Version 1.0). *Intensive working group of ACM SIGKDD curriculum committee*, 140:1–10.

de Assis et al. (2021). Iptraf: Coleta e detecção de anomalias em fluxos de rede. In *Anais do XXVI Workshop de Gerência e Operação de Redes e Serviços*, pages 96–109. SBC.

Drozdowski et al. (2019). On the application of homomorphic encryption to face identification. In *2019 BIOSIG*, pages 1–5. IEEE.

Frikken, K. (2007). Privacy-preserving set union. In *ACNS*, pages 237–252. Springer.

Halevi et al. (2019). An improved rns variant of the bfv homomorphic encryption scheme. In *Topics in Cryptology–CT-RSA 2019*, pages 83–105. Springer.

Henry, K. J. (2008). The theory and applications of homomorphic cryptography. Master's thesis, University of Waterloo.

Kaosar et al. (2012). Fully homomorphic encryption based two-party association rule mining. *Data & Knowledge Engineering*, 76:1–15.

Taylor, P. (2022). Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025.

Yi et al. (2012). Single-database private information retrieval from fully homomorphic encryption. *IEEE Trans on Knowledge and Data Engineering*, 25(5):1125–1134.