Métodos para Criptografia Homomórfica na Mineração de Dados Aplicados 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. *The diffusion of information technologies increasingly contributes to the growth of data generated, leveraging Data Mining opportunities. On the other hand, concerns about the privacy of such data are also growing. Homomorphic Cryptography has the power to put an end to this supposed dichotomy, allowing operations on encrypted data, without losing the privacy of such data. Therefore, this work created three different methods for generating Association Rules for distributed databases, in a manner to preserve the parties' privacy. Real data taken from the edge routers of the Rede-Rio/FAPERJ backbone are used to validate the proposal.*

Resumo. *A difusão das tecnologias da informação contribui cada vez mais com o crescimento de dados gerados, tornando evidente as oportunidades presentes com a Mineração de Dados. Por outro lado, cresce também a preocupação com a privacidade de tais dados. A Criptografia Homomórfica detêm o poder de acabar com esta suposta dicotomia, permitindo operações em dados cifrados, sem a perda da privacidade de tais dados. Com isso, este trabalho criou três métodos diferentes para a geração de Regras de Associação para base de dados distribuídos, respeitando a privacidade de cada participante. Dados reais retirados dos roteadores de borda do backbone da Rede-Rio/FAPERJ são usados para validar a proposta.*

1. Introdução

A evolução e a difusão dos dispositivos digitais e tecnologias de rede vem aumentando rapidamente o volume de dados gerados, ultrapassando o patamar de 95 zettabytes, com tendências de crescimento [Taylor 2022]. Com isso, há uma miríade de dados que tem o potencial de lançar luz sobre informações até então não reveladas, o que permite extrair conhecimento de maneira sem precedentes. A área responsável por essa extração de conhecimento é chamada de Mineração de Dados e dispõe de diversas técnicas para alcançar este propósito.

Por outro lado, também cresce a preocupação com a privacidade de dados, como evidenciado até mesmo por artefatos legais como a Lei Geral de Proteção de Dados (LGPD) [BRASIL 2020]. Tanto as pessoas, como as instituições, entendem que seus dados são valiosos e desejam ter controle sobre eles, seja por preocupação do uso malicioso de seus dados ou simplesmente para evitar que sejam utilizados indevidamente [Clifton et al. 2002].

A Criptografia Homomórfica se apresenta como uma solução para a obtenção de informações para evitar vazamento de dados, permitindo a manipulação de dados encriptados por terceiros por meio de um conjunto de operações. Assim, a Criptografia Homomórfica permite a utilização de dados codificados de modo que só agentes autorizados conseguem lê-los e aplicar operações básicas, como soma e produto, para gerar informações que somente o agente autorizado do dado é capaz de descriptografar.

Uma das áreas nas quais a Criptografia Homomórfica pode atuar diretamente é no Aprendizado Federado [Zhang et al. 2021], um aprendizado distribuído no qual participantes cooperam para construir um modelo sem a divulgação de seus dados, caracterizando um tipo de Computação de Múltiplas Partes [Evans et al. 2018]. Desta maneira, a Criptografia Homomórfica pode alavancar técnicas de Mineração de Dados em um ambiente distribuído, no qual participantes desejam cooperar sem revelar seus dados.

Este trabalho traz uma contribuição para solução do problema de computação de valores sigilosos entre partes que não desejam revelar suas informações individuais. Tal solução se baseia em uma técnica de Mineração de Dados que pode tomar proveito da estrutura da Criptografia Homomórfica. A técnica em questão é a geração de Regras de Associação, que se encaixa por depender apenas de operações fundamentais, como soma e produto, e apresentar uma base no contexto dado pelo trabalho de [Kaosar et al. 2012]. O problema base consiste então em um cenário no qual n participantes detêm diferentes partes de um mesmo conjunto de dados distribuído e desejam cooperar para criar Regras de Associação sobre este conjunto sem revelar suas partes individuais. Para isso, foram desenvolvidos três métodos diferentes.

Os métodos foram construídos para funcionar a partir do problema genérico de operações com dados sigilosos, isto é, se encaixam em qualquer cenário que parte da necessidade de criação de Regras de Associação de um conjunto distribuído para preservar a privacidade das partes. Como cenário de aplicação, foi escolhido o tema de fluxos de rede para a validação dos métodos. Fluxos de rede podem se beneficiar diretamente da aplicação, por esconderem diversos padrões que podem ser utilizados para previsão de tráfego [Macedo 2015], detecção de anomalias [da Silva 2015], entre outros. Além disso, muitas vezes há a necessidade de cuidado na manipulação de dados de rede para manter a privacidade dos usuários que geraram tais fluxos. Desta forma, a aplicação direta deste trabalho ocorre quando detentores de diferentes roteadores de borda entendem que juntos têm características em comum. Logo, os participantes decidem cooperar sem revelar os dados referentes a cada fluxo a fim de descobrir padrões nesta base de dados conjuntos, para assim obter padrões frequentes que podem ser utilizados em detecção de anomalias, engenharia de tráfego e outras necessidades.

Os fluxos utilizados foram retirados de roteadores de borda da Rede-Rio/FAPERJ [REDERIO 2023] e distribuídos em quatro conjuntos para atuar como diferentes participantes que desejam contribuir para criar Regras de Associação conjuntas, para preservar a privacidade individual. Os conjuntos foram então distribuídos em quatro instâncias de contêineres Docker e aplicados os três métodos propostos, assim como um método sem o uso de criptografia. Os resultados foram comparados, tanto no quesito prático, quanto teórico.

O restante do artigo está organizado da seguinte maneira. A Seção 2 comenta os

trabalhos relacionados ao tema do artigo. A Seção 3 explica os 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

Popularizado pelo trabalho de Gentry [Gentry 2009], o uso de Criptografia Homomórfica na prática é algo relativamente recente, mas que já encontra diversas aplicações. Por exemplo, temos o trabalho de [Frikken 2007] que mesmo antes da publicação de Gentry já propôs um método de calcular a união de conjuntos de maneira privada utilizando Criptografia Homomórfica. Para isso, Frikken representou os conjuntos como polinômios, a fim de poder utilizar operações homomórficas nestes polinômios.

Um exemplo mais prático ainda surge no trabalho de Drozdowski *et al.* [Drozdowski et al. 2019], onde os autores utilizam Criptografia Homomórfica para reconhecimento facial. Isto é, compara distâncias entre as biometrias encriptadas de um usuário com informações encriptadas em um servidor.

Para aplicações mais diretas em Mineração de Dados, podemos citar o trabalho de Mittal [Mittal et al. 2014] , que explora o uso de Criptografia Homomórfica para o cálculo de k-centros (*k-means*) em uma base de dados distribuída horizontalmente, sem os participantes divulgarem seus dados, e o de [Li and Huang 2020], que utiliza do poder distribuído da Criptografia Homomórfica para a realização de regressão logística com múltiplas partes para um cálculo distribuído ou até com múltiplos conjuntos de treino combinados sem revelar seu conteúdo.

Apesar de diversos trabalhos na área, nenhum deles investiga o de geração de Regras de Associação em um ambiente distribuído, com exceção do trabalho de [Kaosar et al. 2012]. Entretanto, os autores tratam apenas do caso particular com dois participantes. Neste artigo, propomos como contribuição e diferencial a extensão da geração de Regras de Associação em um ambiente distribuído para o caso mais geral, quando são permitidos um número indefinido de participantes. Uma comparação entre os trabalhos pode ser vista na Tabela 1. É importante notar que os trabalhos se tratam de diferentes modalidades de utilização de Criptografia Homomórfica, sendo o nosso o único que consegue fazer Mineração de Dados por Regras de Associação para um número qualquer de participantes.

Tabela 1. Comparação entre os Trabalhos Relacionados

Trabalho	Número máximo de participantes	Método Utilizado	Mineração de Dados
[Frikken 2007]	Qualquer	União de Conjuntos	✓
[Drozdowski et al. 2019]	Qualquer	Reconhecimento Facial	X
[Mittal et al. 2014]	Qualquer	k-centros	✓
[Li and Huang 2020]	Qualquer	Regressão Logística	✓
[Kaosar et al. 2012]	2	Regras de Associação	✓
Este Trabalho	Qualquer	Regras de Associação	✓

3. Fundamentação Teórica

Para compreender o restante do trabalho, são necessários conhecimentos de Criptografia Homomórfica, Criptografia de Limiar e Mineração de Dados, apresentados a seguir.

3.1. Criptografia Homomórfica e de Limiar

A Criptografia se trata da arte e a ciência da encriptação [Ferguson and Schneier 2003]. Desta forma, pode-se entender a Criptografia Homomórfica como um sistema no qual operações no texto puro são preservadas pela função de encriptação [Henry 2008]. Esta preservação é feita pelo que é chamado de homomorfismo. Uma função f é dita ser um homomorfismo se, para duas operações $\circ$ e $\bullet$ (não necessariamente distintas), temos:

$$f(x_1 \circ x_2) = f(x_1) \bullet f(x_2)$$

Um criptossistema com função de encriptação E e decriptação D é dito ser um homomórfico se D for um homomorfismo. Nota-se que se E é um homomorfismo, D também será, então se torna mais conveniente tratar do homomorfismo de E. A grande vantagem sobre a criptografia tradicional reside no fato de ser possível manipular os dados em seu estado encriptado, sem revelar suas informações. Esta extração de dados com segurança se mostrará útil na Mineração de Dados.

A Criptografia de Chave Pública, é um tipo popular de sistema criptográfico que utiliza um par de chaves para a encriptação e decriptação. Nela, uma chave é chamada de pública, necessária para a encriptação, permitindo que qualquer um possa encriptar um texto puro. Já a decriptação é realizada a partir de uma chave chamada de privada, disponível somente para quem é autorizado a recuperar o texto original.

Da Criptografia Assimétrica surge a Criptografia de Limiar. A ideia central da Criptografia de Limiar se trata da existência de não apenas uma, mas múltiplas chaves privadas, distribuídas por todos os agentes autorizados. Para a decriptação de um texto, é necessário que um número mínimo de chaves privadas sejam utilizadas. Este tipo de funcionalidade pode ser obtido, por exemplo, por interpolação das chaves privadas. Este 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.

3.2. Mineração de Dados

Mineração de Dados (*Data Mining*) é a ciência de extrair conhecimento útil de grandes repositórios de dados [Chakrabarti et al. 2006]. Uma das técnicas importantes da Mineração de dados, apesar de simples, é chamada de Mineração por Regra de Associação (*Association Rule Mining* [Agrawal et al. 1993]). Esta técnica se baseia na repetição de padrões frequentes em determinadas operações. No contexto fluxos de redes, estas regras podem sugerir o comportamento padrão de determinada parte de uma rede.

Para entender a criação de Regras de Associação, são necessários alguns conceitos. *Transação* é definido como um registro na base de dados. Um conjunto S aparece em uma transação se todo elemento de S está na transação. 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. É necessário também uma forma de saber se um conjunto é frequente o suficiente para ser analisado. Para isso, é definido o $Suporte$:

$$Suporte_S = \frac{\text{nº de transações onde o conjunto S aparece}}{\text{nº total de transações}}$$

Se um conjunto S tem um valor maior ou igual a um valor mínimo pré-estabelecido m, ele é dito frequente e pode ser investigado pela próxima etapa. Desta forma, S é separado em diferentes conjuntos disjuntos S_1 e S_2 para investigar regras do tipo $S_1 \Rightarrow S_2$. Para saber se cada uma das regras geradas é aceita como válida ou não, é calculada a frequência relativa dos conjuntos, chamada de $Confiança$:

$$\textit{Confiança}_{S_1 \Rightarrow S_2} = \frac{Suporte_{S_1 \cup S_2}}{Suporte_{S_1}} = \frac{\text{nº de transações onde o conjunto } S_1 \cup S_2 \text{ aparece}}{\text{nº de transações onde o conjunto } S_1 \text{ aparece}}$$

Se a Confiança da regra $S_1 \Rightarrow S_2$ é maior que um valor c escolhido, a regra é aceita.

4. Métodos Propostos de Criptografia Homomórfica na Mineração de Dados

Nesta seção, apresentamos e exploramos as soluções propostas, tendo como base a formalização dos conceitos teóricos e os fundamentos apresentados na Seção 3.

4.1. Métricas Propostas para Regras de Associação Distribuídas

Baseado no trabalho de [Kaosar et al. 2012], definimos a criação de Regras de Associação para n participantes, o que fundamentalmente se reduz ao cálculo do $Suporte$ e da $Confiança$ em um ambiente distribuído. Sejam n participantes diferentes com uma base de dados distribuída com a intenção de criar Regras de Associação mantendo a privacidade individual. Seja também i o índice de um participante, $|DB_i|$ o número de transações na base do participante i, S um determinado conjunto a ser investigado e c_i as contagens do participante i para as ocorrências do conjunto S. Então definimos o $Suporte_S$ para nosso contexto como:

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

Se queremos comparar isto com um suporte mínimo $m/100$, temos:

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

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

Chamemos esta última linha de Inequação do Suporte Distribuído. Se esta é válida, então o conjunto S passa pela etapa do $Suporte$. Nota-se que cada parcela da soma depende somente do participante i . Assim, temos a soma como operação homomórfica em diversos criptossistemas, de valores que dependem apenas de cada participante. Logo pode ser realizado em um contexto encriptado.

O mesmo vale para a $Confiança$. Para uma regra $S_1 \Rightarrow S_2$ sejam l_i e L_i as contagens de ocorrências de $S_1 \cup S_2$ e S_1 respectivamente em um participante i. Temos:

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

Para a $Confiança$ ser maior que a mínima $c/100$, temos:

$$Confiança_{S_1 \Rightarrow S_2} \geq c/100 \qquad \frac{\sum_{i=1}^{n} l_i}{\sum_{i=1}^{n} L_i} \geq c/100$$

$$100 \sum_{i=1}^{n} l_i \geq c \sum_{i=1}^{n} L_i \qquad \sum_{i=1}^{n} 100 l_i - c L_i \geq 0$$

De maneira semelhante ao $Suporte$, a última parte depende apenas de soma de parcelas individuais. Chamamos essa última linha de Inequação da Confiança Distribuída e é o último critério para uma regra ser aceita ou não.

4.2. Métodos Propostos de Comunicação entre Participantes

Neste trabalho, foram desenvolvidos três métodos para a geração de Regras de Associação distribuída de n participantes. Além disso, foi implementado um Método Padrão, sem a preocupação de privacidade, para apresentar uma *baseline* de comparação com os outros métodos. O criptossistema escolhido foi o Brakerski/Fan-Vercauteren (BFV) [Fan and Vercauteren 2012], implementado pela biblioteca OpenFHE [Badawi et al. 2022]. Tal biblioteca dispõe de diversos criptossistemas e necessitou de configurações de parâmetros, como o tamanho do módulo do texto puro, quantas multiplicações poderiam ser feitas e a ativação do modo de limiar.

4.2.1. Método Padrão

Neste método, um dos participantes recebe a contagem de todos os conjuntos de todos os participantes e as une, atuando como uma só base, sem a preocupação com a privacidade. Calcula então as Regras de Associação e as envia aos demais participantes.

4.2.2. Método Proposto 1: Sistema Homomórfico com Soma e Produto

Assim como no método padrão, um dos integrantes é escolhido para ser o responsável para os cálculos necessários para a geração das Regras de Associação. Por conveniência, chamemos este responsável de *Operador*. De maneira semelhante, definimos o *Contextualizador*, que é o responsável por criar o contexto criptográfico que será utilizado no método. Essa terminologia será também utilizada nos métodos propostos seguintes.

O processo começa com o membro *Contextualizador* gerando o contexto criptográfico, isto é, a instância do sistema BFV utilizado, assim como um par de chaves pública e privada. O processo da primeira metade do método pode ser visto na Figura 1. O Passo 1 consiste no envio do contexto criptográfico e da chave pública para os demais participantes.

Após isso, todos os participantes encriptam seus respectivos arquivos com suas parcelas da Inequação do Suporte Distribuído usando a chave pública recebida e enviam os textos cifrados ao *Operador* no Passo 2. O *Operador* pode então calcular o valor do lado esquerdo da Inequação do Suporte Distribuído no espaço encriptado para cada um

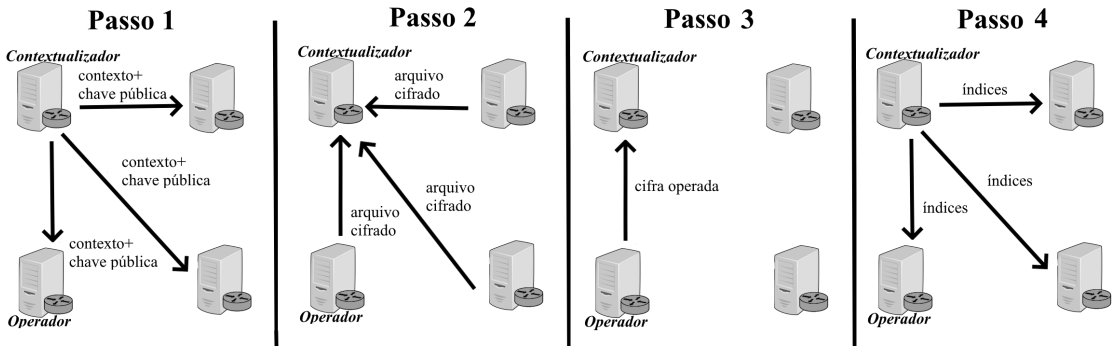

Figura 1. Comunicações na primeira metade do Método 1 ilustrado com quatro participantes.

dos conjuntos criptografados. Por questões de segurança, cada valor calculado é multiplicado por um inteiro aleatório inteiro maior que zero. Essa nova sequência de valores cifrados é devolvida ao *Contextualizador* no que consiste o Passo 3.

Por fim, o *Contextualizador* em posse da chave privada decripta os valores recebidos do *Operador* e os compara individualmente com zero. Note que multiplicar o lado esquerdo da Inequação do Suporte Distribuído por um inteiro maior que zero não muda seu sinal, isto é, o valor obtido após o produto realizado pelo *Operador* é maior que zero se e somente se a parte esquerda da Inequação do Suporte Distribuído for maior que zero. Assim, o *Contextualizador* descobre os índices dos valores descriptografados maiores que zero e descobre os conjuntos com suporte mínimo. Como Passo 4, envia esses índices para os demais participantes.

Para cada índice recebido, os participantes descobrem os conjuntos de suporte mínimo e criam um arquivo com todas as Regras de Associação possíveis a partir destes conjuntos. Além disso, criam outro arquivo, seguindo a mesma ordem deste primeiro, com os valores de sua parcela da Inequação da Confiança Distribuída. A segunda metade ocorre da mesma maneira que a primeira, mas desta vez com o cálculo da confiança. A comunicação ocorre então da mesma forma, porém sem a criação do contexto.

Caso todos os participantes executem suas funções corretamente, o único participante que terá acesso a algum texto em claro que não seja o seu original será o *Contextualizador*. Porém, o texto em claro será a soma dos textos de todos, multiplicado por um número aleatório, ou seja, um dado que não diz nada sobre os dados individuais. Além disso, o único participante que tem acesso aos dados encriptados dos outros participantes é o *Operador*, que não possui a chave privada. Ou seja, a única maneira de um participante descobrir informações de outro ocorre caso o *Operador* e o *Contextualizador* cooperem, com o *Operador* enviando dados sigilosos encriptados dos outros participantes para o *Contextualizador* descriptografar.

4.2.3. Método Proposto 2: Sistema Homomórfico de Limiar com Soma

O segundo método utiliza da Criptografia de Limiar para evitar a colaboração do *Operador* e do *Contextualizador* que juntos têm todos os textos e a capacidade de descriptografá-los. A criação das chaves com o criptossistema escolhido se torna mais complexa caso necessitado o uso da operação de multiplicação, então criamos este Método para este

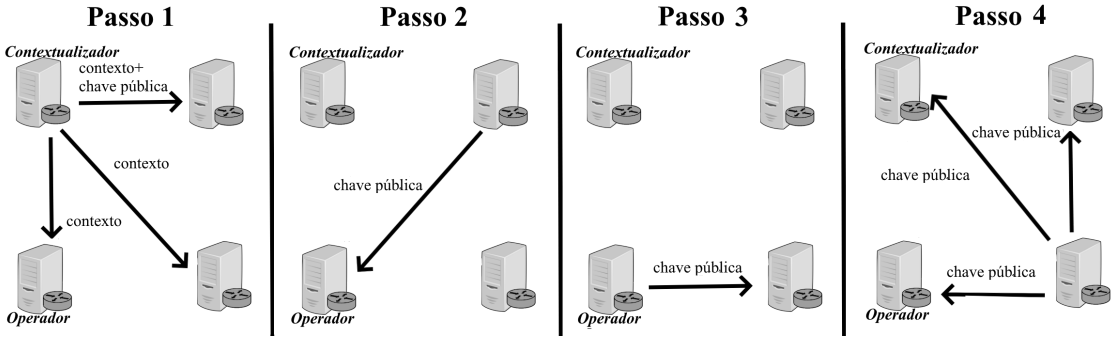

Figura 2. Comunicações para a geração de chave pública no Método 2 ilustrado com quatro participantes.

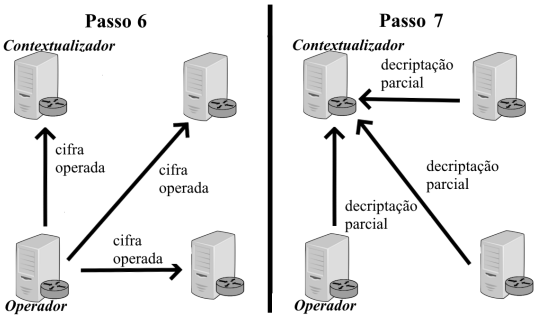

Figura 3. Comunicações para a decriptação no Método 2 ilustrado com quatro participantes.

caso mais simples. As diferenças deste método para o anterior são a criação das chaves e a decriptação, além de não ocorrer um produto por parte do *Operador*.

Na criação de chaves, o *Contextualizador* cria seu par de chaves e o contexto, enviando-o para os demais. Além disso, envia a chave pública para um segundo participante, que cria seu próprio par, enviando também sua chave pública para um terceiro. Isto continua até o último participante gerar seu par, enviando a chave pública final para todos os participantes, como descrito na Figura 2. O resto do processo permanece os mesmo, até o processo de decriptação. Nele, o *Operador* envia a cifra operada para todos os participantes, que realizam uma decriptação parcial, enviando seus resultados ao *Contextualizador*, que as une seguindo o resto do processo como no Método 1, tanto para a etapa do $Suporte$ quanto para a $Confiança$. A decriptação é ilustrada na Figura 3.

4.2.4. Método Proposto 3: Sistema Homomórfico de Limiar com Soma e Produto

Este é o método mais completo que reúne características dos dois métodos anteriores, tendo tanto a operação de produto por um natural não nulo, quanto a Criptografia de Limiar. A diferença deste método em relação ao Método 2, com exceção é claro do produto, reside na geração de chaves. No Método 2, é necessário ocorrer uma sequência de transmissão de chaves passando por cada participante uma única vez. Para a criação da parte da chave pública necessária para o produto, é necessário passar duas vezes por cada integrante.

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 de redes extraídos de roteadores 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. Comparação Teórica

Antes de comparar os métodos, é importante distinguir dois tipos de adversário [Evans et al. 2018]. O adversário semi-honesto (também conhecido como honesto-mas-curioso) executa o algoritmo corretamente, mas pode guardar informações e tentar descobrir mais informações depois. Enquanto isso, o adversário malicioso pode desviar o quanto quiser do processo.

Para o caso de um adversário semi-honesto que tenha controle apenas de um participante, ele não descobrirá nada, pois nunca terá, ao mesmo tempo, um texto encriptado que não é dele e a chave privada necessária para decifrá-lo. O primeiro problema surge quando este adversário controla o *Contextualizador* e o *Operador* no Método 1, pois assim terá ambos e poderá decriptá-los. Outro problema surge no Método 2, quando o *Contextualizador* e mais $n - 2$ participantes se voltam contra um específico, pois juntos eles tem o total da soma calculada pelo *Operador* e os valores de todas as parcelas individuais, com exceção de uma. Isso pode ser usado para descobrir a parcela do participante restante. Sendo assim, o Método 3 não apresenta problemas no caso semi-honesto.

Em relação ao adversário malicioso, diversos problemas ocorrem em todos os métodos. Os problemas podem ser simples como o simples envio de dados incorreto para causar uma saída errada quanto o envio incorreto de chaves, para transformar a Criptografia de Limiar em um Criptografia de uma só chave privada. Os outros dois casos se tratam de quando o *Operador* envia os dados de um participante ao invés da soma para descobrir regras específicas deste participante e de quando todos os participantes enviam parcelas nulas ao *Operador* para as regras serem exclusivas dos dados do *Operador*.

5.3. Comparação Prática

Todos os testes foram executados em um computador Acer Nitro 5, com 24 GB de RAM, processador Ryzen 7 4800H e Ubuntu 22.04 como sistema operacional. Cada método foi

executado 100 vezes e medidas foram tomadas. Foi também acrescentado um atraso para a comunicação entre os participantes, a fim de se aproximar ao ambiente real, onde há um tempo maior de transferência de arquivos. O atraso de comunicação foi oriundo de uma distribuição normal com média 2 e desvio padrão 0.5, ambos em segundos. Os resultados das medições podem ser vistos na Tabela 2 e na Figura 4. O significado de cada métrica é explicado a seguir:

M1. Memória Máxima Média Usada Pelo *Contextualizador* (kB): Se trata da média do uso máximo de memória pelo *Contextualizador* dos 100 testes para o determinado método. Isto é, foi calculada a média aritmética do uso máximo de memória para o *Contextualizador* executar o método para as 100 repetições.

M2. Tempo Médio de Comunicação do *Contextualizador* (s): Tempo médio utilizado pelo *Contextualizador* recebendo e enviando arquivos.

M3. Tempo Médio de Processamento do *Contextualizador* (s): Tempo médio utilizado pelo *Contextualizador* para a execução das etapas de processamento. Por exemplo, encriptação e decriptação.

M4. Memória Máxima Média Usada Pelo *Operador* (kB): Se trata da média do uso máximo de memória pelo *Operador* dos 100 testes para o determinado método. Semelhante ao feito para o *Contextualizador*.

M5. Tempo Médio de Comunicação do *Operador* (s): Tempo médio utilizado pelo *Operador* recebendo e enviando arquivos.

M6. Tempo Médio de Processamento do *Operador* (s): Tempo médio utilizado pelo *Operador* para a execução das etapas de processamento. Por exemplo, soma homomórfica e produto homomórfico.

M7. Memória Máxima dos Participantes (kB): Média do uso máximo de memória por todos os quatro participantes nos 100 testes para o determinado método. Isto é, foi calculada a média aritmética do uso máximo de memória para o cada participante em um método para as 100 repetições.

M8. Tempo Médio de Comunicação dos Participantes (s): Tempo médio utilizado pelos participantes para as etapas de comunicação, ou seja, enviando e recebendo arquivos.

M9. Tempo Médio de Processamento dos Participantes (s): Tempo médio utilizado pelos participantes para as etapas de processamento, como decriptação, soma, etc.

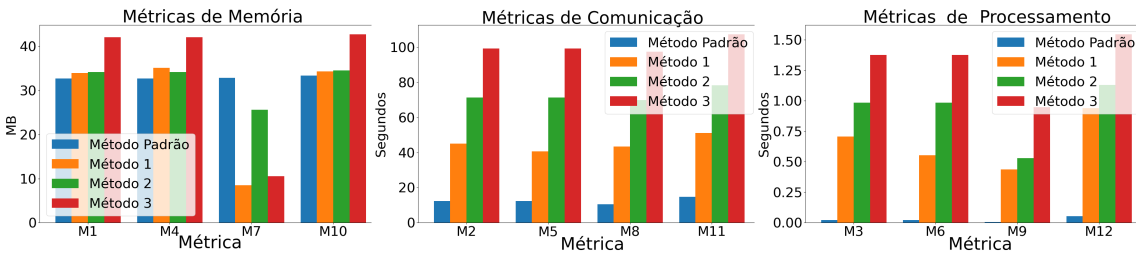

Figura 4. Métricas de Execução.

M10. Memória Máxima dos Participantes (kB): É o máximo de memória pelos participantes nos 100 testes para o determinado método.

M11. Tempo Máximo de Comunicação dos Participantes (s): Tempo máximo utilizado pelos participantes para as etapas de comunicação, isto é, receber e enviar arquivos.

M12. Tempo Máximo de Processamento dos Participantes (s): Tempo máximo utilizado pelos participantes para as etapas de processamento, como encriptação e soma.

Tabela 2. Comparação Prática dos Métodos

Métrica	Padrão	Método 1	Método 2	Método 3
Memória Máxima Média Usada Pelo *Contextualizador* (kB)	32684.60	33907.36	34162.68	42080.12
Tempo Médio de Comunicação do *Contextualizador* (s)	12.3109	45.1172	71.2764	99.2884
Tempo Médio de Processamento do *Contextualizador* (s)	0.0208	0.7058	0.9844	1.3758
Memória Máxima Média Usada Pelo Operador (kB)	32684.60	35128.60	34162.68	42080.12
Tempo Médio de Comunicação do Operador (s)	12.3109	40.6229	71.2764	99.2884
Tempo Médio de Processamento do Operador (s)	0.02082	0.5537	0.9844	1.3758
Memória Máxima Média dos Participantes (kB)	32817.45	8477.17	25569.84	10520.78
Tempo Médio de Comunicação dos Participantes (s)	10.4462	43.3150	69.7988	97.4703
Tempo Médio de Processamento dos Participantes (s)	0.0052	0.4381	0.5304	0.9481
Memória Máxima dos Participantes (kB)	33320	34300	34548	42716
Tempo de Comunicação Máximo dos Participantes (s)	14.7424	51.0247	78.2757	107.3357
Tempo de Processamento Máximo dos Participantes (s)	0.05272	0.9389	1.1297	1.5456

Primeiro, deve se observar que os valores para o *Operador* e para o *Contextualizador* são os mesmos para os Métodos 2 e 3. Isso ocorre pois foi escolhido o mesmo participante para realizar os dois papéis. Isto só é possível nesses dois métodos. Esta escolha foi tomada por facilitar a implementação, reduzir as etapas de comunicação e permitir estudar um participante com o máximo de processamento possível.

Um fator importante para analisar estes valores se trata do contexto no qual os métodos propostos são utilizados. Regras de Associação procuram descobrir características de um certo conjunto de dados. Espera-se que as características sejam inerentes ao estado do conjunto de dados e que este conjunto não altere sua essência muito rapidamente. Por este motivo, não há uma necessidade de compromisso com o desempenho em termos de tempo de execução. Uma loja *online* pode, por exemplo, necessitar que regras sejam desenvolvidas mensalmente, para se adequarem a padrões de consumo. No ambiente de Redes de Computadores ao qual esse trabalho se propõe, a janela de repetição da criação de Regras pode ser muito variada, como de uma semana para procurar maiores padrões de rede ou como 5 minutos para caracterizar a rede em um momento específico. De qualquer forma, no caso mediano não há a necessidade de agilidade na ordem de dezenas de segundos. Desta forma, é possível perceber que todos os métodos apresentaram um desempenho aceitável, visto que o pior caso de todos demorou menos de 2 minutos.

É importante perceber que a maioria do tempo despendido está relacionada à comunicação, sendo praticamente desprezível o tempo consumido pelas partes referentes às operações criptográficas. Como a comunicação é inevitável em um método distribuído, tal fator em geral é considerado um gargalo. Desta forma, para a obtenção de outro método mais eficiente na questão de tempo despendido, seria necessária a redução do número de comunicações realizadas. Se a prioridade é velocidade, pode ser utilizado o Método 1 ou o Método 2 por serem mais velozes que o Método 3.

Há uma diferença do *Contextualizador* e do *Operador* em relação aos outros participantes, mas de pequeno impacto quando comparado ao tempo de comunicação e ao nível de urgência. Também há uma diferença significativa na memória utilizada. Porém, da mesma forma que nos tempos, o máximo de memória utilizada não passa dos 42.716MB, valor relativamente baixo para um processo deste tamanho.

Outro ponto relevante a ser abordado é o tamanho dos arquivos gerados, como os textos cifrados e as chaves. Os textos encriptados em todos os métodos possuíam tamanhos menores que 700kB, enquanto as chaves podiam chegar a até 3.3MB no caso da chave responsável pelo produto. Em geral, devido novamente a falta de urgência para aplicação desse tipo de computação e por não estar sob restrições de armazenamento, os tamanhos dos arquivos são considerados relativamente pequenos.

Sobre os parâmetros para o método em si, foram utilizados os valores de 50% tanto para o Suporte quanto para a Confiança. As regras encontradas são coerentes, pois a saída de todos os métodos em si convergiram ao mesmo valor encontrado no Método Padrão. Alguns exemplos de regras encontradas são "1 pacote $\Rightarrow$ fluxo com menos de 100 bytes" e "fluxo sem flags $\Rightarrow$ 1 pacote".

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou métodos de Mineração de Dados por meio da Criptografia Homomórfica para resolver o conflito entre as oportunidades geradas pela crescente quantidade de dados sendo criados e pelo avanço da preocupação com a privacidade.

A técnica estudada foi a de criação de Regras de Associação em um ambiente distribuído, seguindo e expandindo a ideia proposta em [Kaosar et al. 2012]. Desta forma, foram criados três métodos diferentes para um cenário no qual diferentes participantes detêm diferentes partes de um conjunto de dados distribuído e desejam descobrir padrões neste conjunto sem revelar suas partes.

Os métodos propostos foram implementados e testados, cada um tendo suas vantagens e desvantagens, com a comparação abrangendo questões de segurança e eficiência. Além disso, os métodos foram comparados com uma versão que não tem como premissa a garantia da privacidade. Para a validação, foram utilizados fluxos reais de roteadores de borda, com o mesmo conjunto de regras gerado em todos os métodos, estando conforme a base que não considera o aspecto de privacidade. Em resumo, foram criados com sucesso diferentes métodos para resolver o problema de geração de Regras de Associação em um ambiente distribuído, segundo o proposto.

É importante lembrar como estes resultados podem ser úteis no contexto escolhido. As oportunidades geradas pela descoberta de padrões em fluxos de rede abrangem diversos tópicos, principalmente no sentido de engenharia de tráfego e detecção de anomalias. Um exemplo concreto na qual a privacidade se torna evidente é em um caso onde diferentes provedores decidem cooperar para encontrar padrões em comum em seus dados, fazendo isso sem revelar tais dados.

Como trabalhos futuros, a comparação entre os valores obtidos pelas operações homomórficas e o número zero, no que consistem a Inequação do Suporte Distribuído e a Inequação da Confiança Distribuída, pode ser melhorada em termos do uso de mecanismos que permitem a comparação de inteiros no domínio cifrado. Tais mecanismos não

foram utilizados por não estarem disponíveis na biblioteca OpenFHE, que foi escolhida por sua robustez e sua Criptografia de Limiar. Porém, os criadores no presente momento já investigam métodos para a troca de criptossistema, com um de seus sistemas já tendo comparação numérica.

Outra proposta relacionada à anterior é a mudança do modo de implementação. Há outros criptossistemas e outras bibliotecas que podem ser utilizadas para a implementações. Tais variações podem prover diferenças práticas e teóricas na execução dos métodos. Uma ideia promissora é o uso de alguma forma de Encriptação Parcialmente Homomórfica, já que o número de operações utilizado neste trabalho é limitado e isso diminuiria a complexidade da criptografia utilizada.

Um desafio que pode ser investigado é a questão de tamanho do domínio e *overflows*. Se não houver certo controle prévio, o método pode falhar ou até mesmo apresentar respostas incorretas. Pode se dividir os valores das inequações propostas por um valor fixo dependendo do caso, por exemplo, 10000 para assim diminuir sua ordem de grandeza e diminuir sua chance de *overflow*. Note que existe a necessidade de argumentação sobre aproximações, pois a divisão proposta é entre inteiros com saída inteira, pois o BFV só trabalha com inteiros.

Por fim, é importante apontar que o estudo de Regras de Associação não se limita somente aos conceitos apresentados. Podemos citar, por exemplo, o conceito de Convicção [Brin et al. 1997] utilizado no lugar da confiança, tendo suas próprias vantagens e desvantagens. Neste sentido, se mostra útil a investigação futura de outros conceitos de geração de Regras de Associação para o contexto distribuído alavancados por Criptografia Homomórfica.

Referências

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216.

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

BRASIL (2020). Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Disponível em: `https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm`. Acesso em: 18 jan. 2023.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 255–264.

Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., and Wang, W. (2006). Data mining curriculum: A proposal (version 1.0). *Intensive working group of ACM SIGKDD curriculum committee*, 140:1–10.

Clifton, C., Kantarcioglu, M., and Vaidya, J. (2002). Defining privacy for data mining. In *National science foundation workshop on next generation data mining*, volume 1, page 1. Citeseer.

da Silva, V. L. P. (2015). Identificacão de anomalias em fluxos de rede utilizando o método de previsão em séries temporais de holt-winters. Dissertação de mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

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, P., Buchmann, N., Rathgeb, C., Margraf, M., and Busch, C. (2019). On the application of homomorphic encryption to face identification. In *2019 international conference of the biometrics special interest group (biosig)*, pages 1–5. IEEE.

Evans, D., Kolesnikov, V., Rosulek, M., et al. (2018). A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3):70–246.

Fan, J. and Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*.

Ferguson, N. and Schneier, B. (2003). *Practical cryptography*, volume 141. Wiley New York.

Frikken, K. (2007). Privacy-preserving set union. In *Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007. Proceedings 5*, pages 237–252. Springer.

Gentry, C. (2009). *A fully homomorphic encryption scheme*. Stanford university.

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

Kaosar, M. G., Paulet, R., and Yi, X. (2012). Fully homomorphic encryption based two-party association rule mining. *Data & Knowledge Engineering*, 76:1–15.

Li, J. and Huang, H. (2020). Faster secure data mining via distributed homomorphic encryption. In *Proceedings of the 26th ACM SIGKDD*, pages 2706–2714.

Macedo, E. L. C. (2015). Previsão de tráfego em enlaces de redes utilizando séries temporais. Dissertação de mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

Mittal, D. et al. (2014). Secure data mining in cloud using homomorphic encryption. In *CCE)*, pages 1–7. IEEE.

REDERIO (2023). Rederio de computadores/faperj. Disponível em `https://rederio.br/`. Acessado em Maio de 2023.

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

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems*, 216:106775.