Algoritmo Eficiente para Quebra de Privacidade em Redes Sociais Anonimizadas

Pamela Tabak, Daniel Ratton Figueiredo

[1]Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro – RJ – Brasil

{pamelatabak, daniel}@cos.ufrj.br

Abstract. *Companies that control social networks often release data, including the users' relationship network, removing their identity to preserve their privacy. This work defines and evaluates a two-step methodology to reveal relationships between users of a social network. In the first phase, artificial nodes (users) are inserted and connected to a set of victim users. The second phase identifies the artificial nodes in the anonymized network and reveals relationships between the victim users. The algorithms were implemented and the results indicate that they are effective in revealing relationships between victims, breaking their privacy.*

Resumo. *Empresas que controlam redes sociais muitas vezes disponibilizam dados, incluindo a rede de relacionamento entre seus usuários, removendo a identidade deles para preservar sua privacidade. Este trabalho define e avalia uma metodologia de duas fases para revelar relacionamentos entre usuários de uma rede social. Na primeira, nós (usuários) artificiais são inseridos e conectados a um grupo de usuários vítimas. A segunda fase identifica os nós artificiais na rede anonimizada e revela os relacionamentos entre os usuários vítimas. Os algoritmos foram implementados e os resultados indicam que eles são eficientes em revelar os relacionamentos entre vítimas, quebrando sua privacidade.*

1. Introdução

Redes sociais vêm sendo usadas por uma quantidade crescente de serviços na *Web*. Um aspecto importante é a privacidade dos usuários e, em particular, os relacionamentos formados entre eles. Tal informação (arestas da rede social) geralmente não é pública e pertence às empresas que operam o serviço.

Entretanto, dados de usuários de redes sociais são ocasionalmente disponibilizados para a realização de estudos e melhorias dos serviços prestados (como o caso do Prêmio Netflix). Porém, para que a privacidade dos usuários seja preservada, os dados são anonimizados, removendo a identificação dos usuários (como trocar o nome dos usuários por um código aleatório) e gerando uma rede anonimizada isomórfica a rede original.

Neste contexto surgem ataques de quebra de privacidade, que visam desvendar a identidade de alguns usuários em redes sociais anonimizadas e, consequentemente, os relacionamentos entre eles. Este trabalho define e analisa uma metodologia de ataque ativo de duas fases, generalizando e avaliando empiricamente uma metodologia proposta recentemente [Backstrom et al., 2007], que vem sendo muito estudada. A primeira fase ocorre antes da rede ser disponibilizada, na qual o atacante cria nós (usuários) artificiais na rede social e conecta tais nós aos usuários vítimas, além de criar conexões entre os

próprios nós artificiais. Na segunda fase o atacante procura pelos nós criados na rede anonimizada, identificando os nós vítimas a partir de suas conexões com os nós criados e, consequentemente, revelando os relacionamentos (arestas) entre os nós vítimas. Existem, portanto, três conjuntos de nós importantes: nós vítimas, usuários da rede social que foram escolhidos para ter a privacidade violada; nós não vítimas, outros usuários da mesma rede social; nós atacantes, nós artificiais adicionados na rede para conduzir o ataque.

2. Criação e conexão dos nós atacantes

Nós atacantes são usados para a identificação dos relacionamentos entre nós vítimas na rede anonimizada, sendo criados e inseridos na rede social antes da anonimização. Iremos assumir que o atacante conhece a identidade de todos os nós antes da rede ser anonimizada, entretanto não conhece o grau destes nós ou as arestas incidentes a eles. Ou seja, o atacante conhece apenas as arestas que ele cria, que incidem sobre nós por ele criado (nós atacantes) e em outros nós (usuários) da rede. A Figura 1 ilustra os grafos após a criação dos nós e arestas artificiais, antes (esquerda) e depois (direita) da anonimização. O grafo da esquerda apresenta os nós atacantes $a_0, ..., a_3$, os nós vítimas v_0, v_1, vértices da rede não vítimas u_0, u_1 e as novas arestas, incidentes aos nós atacantes, em linhas pontilhadas.

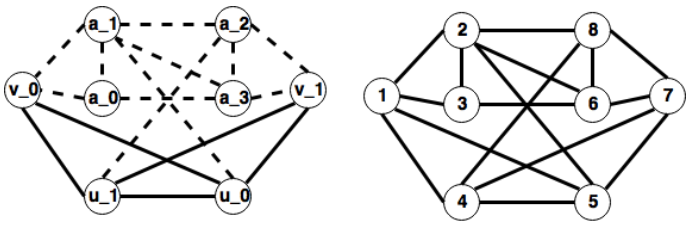

Figura 1. Grafo após a criação de nós e arestas artificiais, antes e depois da anonimização, que gera um identificador aleatório para cada nó.

2.1. Criação dos nós

Cada nó vítima precisa estar conectado a um subconjunto único de nós atacantes, composto por ao menos dois nós, para ser reconhecido. Ou seja, é preciso que nós vítimas se conectem a nós atacantes de maneira única, de forma que ao identificar o conjunto de nós atacantes identificamos também cada nó vítima. Considerando o caso em que todos os nós do grafo original, de tamanho n, estão sendo atacados, o número de nós atacantes precisa ser ao menos $\lceil \log_2 n \rceil + 1$, de forma a termos n subconjuntos distintos de nós atacantes, de tamanho mínimo 2. Neste trabalho, são criados $s = 2\lceil \log_2 n \rceil$ nós atacantes e o fator de 2 foi escolhido arbitrariamente para facilitar a busca por subconjuntos distintos.

2.2. Criação das arestas

É preciso identificar os nós atacantes no grafo anonimizado, formado pelo grafo original mais os nós e arestas artificialmente adicionados, para o ataque funcionar. Portanto, o subgrafo induzido pelos atacantes (a_0,..., a_{s-1}) será criado com propriedades específicas.

1. Criação das arestas entre nós atacantes: toda aresta do tipo (a_i, a_{i+1}) é criada. As demais arestas são criadas aleatoriamente: a aresta entre cada par (a_i, a_j), sendo $i < (j + 1)$, é criada com probabilidade 1/2. Essa configuração foi escolhida devido a propriedade do modelo de Erdõs-Rényi, pois com $p = 1/2$ todos os possíveis grafos têm a mesma probabilidade de serem gerados.

2. Criação das arestas entre nós atacantes e vítimas: um número binário b de s dígitos é criado e inicializado com 0, em que cada dígito representa um nó atacante. Para cada nó vítima é verificado se existem ao menos dois dígitos 1 em b. Caso positivo, para cada posição contendo 1, uma aresta é criada entre o nó atacante correspondente e o nó vítima em questão. Em seguida, b é acrescido em uma unidade binária. Caso não possua pelo menos duas posições com o valor 1, apenas a soma acontece. Esse processo é repetido para cada nó vítima. Dessa forma, é garantido que todo nó vítima estará conectado a pelo menos dois nós atacantes e que nenhum outro nó vítima estará conectado a exatamente os mesmos nós atacantes.
3. Criação das arestas entre nós atacantes e não vítimas: esse processo é feito para aumentar a aleatoriedade do subgrafo de nós atacantes, diminuindo as chances de encontrar um isomorfismo deste subgrafo induzido na rede divulgada. Para cada nó atacante, um número g maior ou igual ao seu grau atual é gerado aleatoriamente. Cada nó atacante se conecta a nós não vítimas até atingir grau igual a g. Entretanto, cada nó não vítima só pode se conectar a um nó atacante, de modo a jamais ser confundido com um nó vítima, que é conectado a pelo menos dois nós atacantes. Este processo é interrompido caso não existam mais nós não vítimas para conectar aos nós atacantes.

2.3. Detecção de Automorfismos

Ao final do procedimento acima, o algoritmo faz uma busca para verificar a possível existência de automorfismos no subgrafo de atacantes que foi criado, o que impossibilita a identificação correta destes nós no grafo anonimizado. Caso um possível automorfismo seja identificado, a criação de nós e arestas descrita acima é refeita.

O algoritmo para procurar por possíveis automorfismos possui uma complexidade polinomial e foi inspirado em ideias propostas recentemente [Hay et al., 2008]: utilizar o grau e a sequência parcial de grau dos vizinhos de cada nó atacante. Este algoritmo analisa se existem dois nós atacantes com mesmo grau e com sequência de grau dos vizinhos atacantes que seja majorada pela outra. Esta análise é uma condição necessária para a existência de um automorfismo entre os nós atacantes, mas não suficiente. Caso positivo, o processo de criação de arestas incidentes a nós atacantes é refeito.

3. Identificação dos nós atacantes e vítimas

Para encontrar os relacionamentos entre nós vítimas na rede anonimizada basta identificar todos os nós atacantes, uma vez que cada nó vítima está conectado a um subconjunto distinto de nós atacantes. Consequentemente, nós vítimas também serão identificados, assim como seus relacionamentos, presentes na rede anonimizada.

O algoritmo procura nós atacantes respeitando o ciclo de arestas (a_i, a_{i+1}), iniciando por a_0 e utilizando duas informações de cada nó atacante: seu grau e sequência de grau dos vizinhos que são nós atacantes. Uma lista de árvores é utilizada e cada árvore representa um possível subgrafo formado por nós atacantes. Para encontrar a_0, o processo identifica todos os nós u do grafo anonimizado que tem o mesmo grau de a_0 e sequência de grau dos nós atacantes adjacentes a a_0 contida na sequência de grau dos nós vizinhos de u. Caso positivo, uma nova árvore com raiz u é criada. Os filhos v de u são todos os vértices vizinhos de u que tem o mesmo grau de a_1 e sequência de grau dos vizinhos

atacantes de a_1 contida na sequência de grau dos vizinhos de v. A busca então avalia cada nó folha f, que representa um possível a_i, sendo i sua altura na árvore. Para cada f, o processo segue as regras de busca definidas nos passos anteriores nos vizinhos de f, procurando por a_{i+1}. Se não encontrar nos vizinhos de f um possível a_{i+1}, f é removido da árvore. Uma vez removido, o algoritmo analisa se o pai deste nó na árvore virou folha. Caso positivo, ele também é removido e o processo continua subindo na árvore até encontrar um nó não folha ou a raiz.

O processo é bem sucedido se ao final da busca existe uma única árvore com s nós, na forma de um grafo linha, identificando assim os nós atacantes $a_0, \ldots, a_{s-1}$. Caso contrário, o subgrafo de nós atacantes possui um automorfismo ou isomorfismo com outro subgrafo induzido do grafo e o processo de identificação dos nós atacantes falha.

4. Resultados

Os algoritmos descritos neste artigo foram testados em redes geradas a partir de dois modelos de grafos aleatórios, Erdös-Rényi e Barabási-Albert, e em três redes reais [Leskovec and Krevl, 2018], de tamanhos variando entre 1000 nós e 5000 arestas até 7115 nós e 100762 arestas. Foram definidos diversos conjuntos de nós vítimas, de tamanhos entre 2 e o número de nós na rede. Para cada rede e conjunto de vítimas, 50 execuções do algoritmo foram realizadas. No total, o procedimento de quebra de privacidade foi executado 20800 vezes em um Macbook Pro, com processador Intel core i7 e 8 Gb de memória RAM, sendo apresentadas algumas médias de tempos de execução na tabela 1. Em apenas 28 casos não foi possível identificar os relacionamentos entre os nós vítimas, tendo sido todas as falhas causadas por isomorfismos entre o subgrafo de nós atacantes e subgrafos do grafo anonimizado, isto é, nós do grafo original foram confundidos com nós atacantes. Esse resultado indica que o processo de detecção de automorfismos foi capaz de evitar este tipo de falha, uma vez que não aconteceu em nenhuma das execuções feitas.

Número de Nós no Grafo Original - Nós Vítimas	Média do Tempo de Execução
1000 - 500	1.7 segundos
1000 - 900	1.9 segundos
4039 - 1000	1.3 segundos
4039 - 4039	7.5 segundos
6301 - 1000	0.8 segundos
6301 - 4250	7.2 segundos
6301 - 6301	14.2 segundos

Tabela 1. Tempo de execução do algoritmo proposto em diferentes cenários

Referências

Backstrom, L., Dwork, C., and Kleinberg, J. (2007). Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *Proc. of Intern. Conf. WWW*.

Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P. (2008). Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114.

Leskovec, J. and Krevl, A. (2018). SNAP Datasets: Stanford large network dataset collection.