RAPPOR: Preservação de privacidade agregável e aleatória Resposta Ordinal

Úlfar Erlingsson
Google, Inc.
ulfar@google.com

Vasyl Pihur
Google, Inc.
vpihur@google.com

Aleksandra Korolova
Universidade do Sul da Califórnia
korolova@usc.edu

RESUMO

Resposta Ordinal de Preservação de Privacidade Agregada Randomizada, ou RAPPOR, é uma tecnologia para estatísticas de crowdsourcing de software cliente de usuário final, anonimamente, com fortes garantias de privacidade. Em suma, os RAPPORs permitem estudar a floresta de dados do cliente, sem permitir a possibilidade de olhar para as árvores individuais. Ao aplicar a resposta aleatória de uma maneira inovadora, o RAPPOR fornece os mecanismos para essa coleta, bem como para uma análise eficiente e de alta utilidade dos dados coletados. Em particular, o RAPPOR permite a coleta de estatísticas sobre a população de strings do lado do cliente com fortes garantias de privacidade para cada cliente e sem vinculação de seus relatórios.

Este artigo descreve e motiva o RAPPOR, detalha sua privacidade diferencial e garantias de utilidade, discute sua implantação prática e propriedades diante de diferentes modelos de ataque e, finalmente, apresenta resultados de sua aplicação a dados sintéticos e do mundo real.

1. Introdução

O crowdsourcing de dados para tomar decisões melhores e mais informadas está se tornando cada vez mais comum. Para qualquer crowdsourcing, mecanismos de preservação da privacidade devem ser aplicados para reduzir e controlar os riscos de privacidade introduzidos pelo processo de coleta de dados e equilibrar esse risco com a utilidade benéfica dos dados coletados. Para este propósito, apresentamos a Resposta Ordinal de Preservação de Privacidade Agregada Randomizada, ou RAPPOR, um novo mecanismo prático e amplamente aplicável que fornece fortes garantias de privacidade combinadas com alta utilidade, mas não se baseia no uso de terceiros confiáveis.

O RAPPOR baseia-se nas ideias de resposta aleatória, uma técnica de pesquisa desenvolvida na década de 1960 para coletar estatísticas sobre tópicos delicados em que os entrevistados desejam manter a confidencialidade [27]. Um exemplo comumente usado para descrever essa técnica envolve uma pergunta sobre um tópico delicado, como "Você é membro do partido comunista?" [28]. Para esta pergunta, o respondente da pesquisa é pediu para jogar uma moeda honesta, em segredo, e responder "Sim" se der cara, mas dizer a verdade caso contrário (se a moeda der coroa). Usando este procedimento, cada respondente retém uma negação muito forte para quaisquer respostas "sim", uma vez que tais respostas são provavelmente atribuíveis à moeda que dá cara; como um refinamento, os entrevistados também podem escolher a resposta falsa jogando outra moeda em segredo e obter forte negação para respostas "Sim" e "Não".

Pesquisas baseadas em respostas aleatórias permitem cálculos fáceis de estatísticas populacionais precisas enquanto preservam a privacidade dos indivíduos. Assumindo conformidade absoluta com o protocolo de randomização (uma suposição que pode não ser válida para seres humanos e pode até ser não trivial para implementações algorítmicas [23]), é fácil ver que em um caso em que tanto "Sim" quanto "Não " respostas podem ser negadas (jogando duas moedas honestas), o número real de respostas "Sim" pode ser estimado com precisão por 2(Y ÿ 0,25), onde Y é a proporção de respostas "Sim". Na expectativa, os respondentes fornecerão a resposta verdadeira 75% das vezes, como é fácil de ver por uma análise de caso dos dois cara ou coroa justos.

É importante ressaltar que, para coleta única, o mecanismo de pesquisa aleatória acima protegerá a privacidade de qualquer entrevistado específico, independentemente do conhecimento prévio de qualquer invasor, conforme avaliado por meio da garantia de privacidade diferencial [12]. Especificamente, os entrevistados terão privacidade diferencial no nível = ln 0,75/(1 ÿ 0,75) = ln(3). Dito isso, essa garantia de privacidade diminui se a pesquisa for repetida — por exemplo, para obter estatísticas diárias atualizadas — e os dados forem coletados várias vezes do mesmo respondente. Nesse caso, para manter a privacidade e a utilidade diferenciadas, são necessários mecanismos melhores, como os que apresentamos neste artigo.

A Resposta Randomizada Agregada com Preservação da Privacidade, ou RAPPORs, é um novo mecanismo para coletar estatísticas do software do lado do cliente do usuário final, de uma maneira que fornece forte proteção de privacidade usando técnicas de resposta aleatória. O RAPPOR é projetado para permitir a coleta, em um grande número de clientes, estatísticas sobre valores e strings do lado do cliente, como suas categorias, frequências, histogramas e outras estatísticas definidas. Para qualquer valor informado, a RAPPOR oferece uma forte garantia de negação para o cliente relator, o que limita estritamente as informações privadas divulgadas, conforme medido por um limite de privacidade diferencial, e vale até mesmo para um único cliente que reporta frequentemente sobre o mesmo valor.

Uma contribuição distinta é a capacidade do RAPPOR de coletar estatísticas sobre um conjunto arbitrário de strings, aplicando respostas randômicas a filtros Bloom [5] com fortes garantias diferenciais de privacidade. Outro contributo é a forma elegante como a RAPPOR protege a privacidade dos clientes

CCS'14, 3 a 7 de novembro de 2014, Scottsdale, Arizona, EUA.
ACM 978-1-4503-2957-6/14/11, http://dx.doi.org/10.1145/2660267.2660348.

de quem os dados são coletados repetidamente (ou mesmo com frequência infinita) e como o RAPPOR evita a adição de externalidades de privacidade, como aquelas que podem ser criadas ao manter um banco de dados de respondentes contribuintes (que podem ser violados) ou repetir um único, resposta memorizada (que seria vinculável e poderia ser rastreada). Em comparação, a resposta aleatória tradicional não fornece qualquer privacidade longitudinal no caso em que múltiplas respostas são coletadas do mesmo participante. Ainda outra contribuição é que o mecanismo RAPPOR é executado localmente no cliente e não requer um terceiro confiável.

Finalmente, o RAPPOR fornece uma nova estrutura de decodificação de alta utilidade para aprender estatística com base em uma combinação sofisticada de testes de hipóteses, resolução de mínimos quadrados e regressão LASSO [26].

1.1 O Domínio de Aplicação Motivadora RAPPOR é uma tecnologia geral para coleta de dados de preservação da privacidade e crowdsourcing de estatísticas, que pode ser aplicada em uma ampla gama de contextos.

Neste artigo, no entanto, focamos no domínio de aplicação específico que motivou o desenvolvimento do RAPPOR: a necessidade dos operadores de serviços Cloud coletarem estatísticas atualizadas sobre a atividade de seus usuários e seu software do lado do cliente. Neste domínio, o RAPPOR já viu implantação limitada no navegador Google Chrome, onde tem sido usado para melhorar os dados enviados por usuários que optaram por relatórios estatísticos [9]. A Seção 5.4 descreve resumidamente esse aplicativo do mundo real e os benefícios que o RAP POR forneceu ao esclarecer o seqüestro indesejado ou malicioso das configurações do usuário.

Por uma variedade de razões, entender as estatísticas populacionais é uma parte fundamental de uma operação eficaz e confiável de serviços on-line pelos operadores de plataformas de software e serviços em nuvem. Esses motivos geralmente são tão simples quanto observar a frequência com que certos recursos de software são usados e medir seu desempenho e características de falha. Outro importante conjunto de razões envolve o fornecimento de melhor segurança e proteção contra abusos para os usuários, seus clientes e o próprio serviço. Por exemplo, para avaliar a prevalência de botnets ou clientes sequestrados, um operador pode querer monitorar quantos clientes - nas últimas 24 horas - tiveram preferências críticas substituídas, por exemplo, para redirecionar as pesquisas dos usuários na Web para o URL de um conhecido provedor de pesquisa -para-ser-malicioso.

A coleta de estatísticas atualizadas de crowdsourcing levanta um dilema para os operadores de serviços. Por um lado, provavelmente será prejudicial à privacidade dos usuários finais coletar diretamente suas informações. (Observe que mesmo as preferências do provedor de pesquisa de um usuário podem identificar, incriminar ou de outra forma comprometer esse usuário.) Por outro lado, não coletar essas informações também prejudicará os usuários: se os operadores não podem reunir as estatísticas corretas, não podem fazer muitas melhorias de software e serviço que beneficiem os usuários (por exemplo, detectando ou impedindo atividades maliciosas do lado do cliente). Normalmente, os operadores resolvem esse dilema usando técnicas que derivam apenas as estatísticas necessárias de alta ordem, usando mecanismos que limitam os riscos de privacidade dos usuários - por exemplo, coletando apenas dados de granularidade grosseira e eliminando dados que não são compartilhados por um determinado número de usuários.

Infelizmente, mesmo para operadores cuidadosos, dispostos a utilizar técnicas de ponta, existem poucos mecanismos práticos existentes que ofereçam privacidade e utilidade, e ainda menos que fornecem garantias claras de proteção de privacidade. Portanto, para reduzir os riscos de privacidade, os operadores dependem em grande medida de meios e processos pragmáticos, que, por exemplo, evitam a coleta de dados, removem identificadores exclusivos ou, de outra forma, depuram dados sistematicamente, executam a exclusão obrigatória de dados após um determinado período período e, em geral, aplicar políticas de controle de acesso e auditoria sobre o uso de dados. No entanto, essas abordagens são limitadas em sua capacidade de fornecer garantias de privacidade comprovadamente fortes. Além disso, podem surgir externalidades de privacidade de coletas de dados individuais, como carimbos de data/hora ou identificadores vinculáveis; o impacto na privacidade dessas externalidades pode ser ainda maior do que o dos dados coletados.

O RAPPOR pode ajudar os operadores a lidar com os desafios significativos e as possíveis armadilhas de privacidade levantadas por esse dilema.

1.2 Estatísticas de Crowdsourcing com RAPPOR

Os operadores de serviços podem aplicar o RAPPOR às estatísticas de crowdsourcing de forma a proteger a privacidade de seus usuários e, assim, enfrentar os desafios descritos acima.

Como simplificação, as respostas do RAPPOR podem ser consideradas cadeias de bits, onde cada bit corresponde a uma resposta aleatória para algum predicado lógico nas propriedades do cliente de relatório, como seus valores, contexto ou histórico. (Sem perda de generalidade, essa suposição é usada no restante deste artigo.) Por exemplo, um bit em uma resposta RAPPOR pode corresponder a um predicado que indica o gênero declarado, masculino ou feminino, do usuário cliente, ou— tão bem - sua filiação ao Partido Comunista.

A estrutura de uma resposta RAPPOR não precisa ser restringida de outra forma; em particular, (i) os bits de resposta podem ser sequenciais ou não ordenados, (ii) os predicados de resposta podem ser independentes, disjuntos ou correlacionados e (iii) as propriedades do cliente podem ser imutáveis ou mudar com o tempo. No entanto, esses detalhes (por exemplo, qualquer correlação dos bits de resposta) devem ser contabilizados corretamente, pois afetam tanto a utilização quanto as garantias de privacidade do RAPPOR - conforme descrito na próxima seção e detalhado nas seções posteriores.

Em particular, o RAPPOR pode ser usado para coletar estatísticas sobre propriedades categóricas do cliente, fazendo com que cada bit na resposta de um cliente represente se esse cliente pertence ou não a uma categoria. Por exemplo, esses predicados categóricos podem representar se o cliente está ou não utilizando um recurso de software. Nesse caso, se cada cliente puder usar apenas um dos três recursos disjuntos, X, Y e Z, a coleta de uma resposta RAPPOR de três bits dos clientes permitirá medir a frequência relativa com que os recursos são usados pelos clientes.

Quanto à privacidade, cada cliente será protegido pela forma como os três bits são derivados de um único (no máximo) predicado verdadeiro; quanto à utilidade, bastará contar quantas respostas tiveram o bit definido, para cada bit de resposta distinto, para obter uma boa estimativa estatística da distribuição empírica do uso dos recursos.

O RAPPOR também pode ser usado para coletar estatísticas populacionais em valores numéricos e ordinais, por exemplo, associando bits de resposta com predicados para diferentes faixas de valores numéricos ou relatando categorias disjuntas para diferentes magnitudes logarítmicas dos valores. Para tais estatísticas numéricas RAPPOR, a estimativa pode ser melhorada coletando e utilizando informações relevantes sobre os antecedentes e a forma da distribuição empírica, como sua suavização

ness.

Por fim, o RAPPOR permite também recolher estatísticas sobre domínios não categóricos, ou categorias que não podem ser enumeradas antecipadamente, através da utilização de filtros Bloom [5]. Em particular, o RAPPOR permite a coleta de respostas aleatórias compactas baseadas em filtro Bloom em strings, em vez de ter clientes relatando quando eles correspondem a um conjunto de strings escolhidos a dedo, predefinidos pelo operador. Subseqüentemente, essas respostas podem ser comparadas com strings candidatas, à medida que se tornam conhecidas pelo operador, e usadas para estimar strings conhecidas e desconhecidas na população. Técnicas avançadas de decodificação estatística devem ser aplicadas para interpretar com precisão os dados aleatórios e ruidosos nas respostas RAPPOR baseadas no filtro Bloom. No entanto, como no caso das categorias, esta análise precisa apenas considerar as contagens agregadas de bits distintos definidos nas respostas RAPPOR para fornecer bons estimadores para estatísticas populacionais, conforme detalhado na Seção 4.

Sem perda de privacidade, a análise RAPPOR pode ser executada novamente em uma coleção de respostas, por exemplo, para considerar novas strings e casos perdidos em análises anteriores, sem a necessidade de executar novamente a etapa de coleta de dados. As respostas individuais podem ser especialmente úteis para análises de dados exploratórias ou personalizadas. Por exemplo, se a geolocalização dos endereços IP dos clientes for coletada juntamente com os relatórios RAPPOR de seus valores confidenciais, as distribuições observadas desses valores poderão ser comparadas em diferentes geolocalizações, por exemplo, analisando diferentes subconjuntos separadamente. Tal análise é compatível com as garantias de privacidade da RAPPOR, válidas mesmo na presença de dados auxiliares, como a geolocalização. Ao limitar o número de categorias correlacionadas, ou funções de hash do filtro Bloom, relatadas por um único cliente, o RAPPOR pode manter suas garantias de privacidade diferencial mesmo quando as estatísticas são coletadas em vários aspectos dos clientes, conforme descrito a seguir e detalhado nas Seções 3 e 6.

1.3 Ataques RAPPOR e (Longitudinais)

Proteger a privacidade para coletas únicas e múltiplas requer a consideração de vários modelos de ataque distintos. Presume-se que um invasor básico tenha acesso a um único relatório e possa ser interrompido com uma única rodada de resposta aleatória. Um invasor em janela tem acesso a vários relatórios ao longo do tempo do mesmo usuário. Sem modificação cuidadosa das técnicas tradicionais de resposta aleatória, quase certamente aconteceria a divulgação completa de informações privadas. Isso é especialmente verdadeiro se a janela de observação for grande e o valor subjacente não mudar muito. Um invasor com acesso completo aos relatórios de todos os clientes (por exemplo, um insider com direitos de acesso ilimitados) é o mais difícil de parar, mas também é o mais difícil de executar na prática. O RAPPOR fornece compensações explícitas entre diferentes modelos de ataque em termos de proteção de privacidade ajustável para todos os três tipos de invasores.

O RAPPOR baseia-se na ideia básica de memoização e fornece uma estrutura para proteção de privacidade única e longitudinal, jogando o jogo de resposta aleatória duas vezes com uma etapa de memorização intermediária. A primeira etapa, chamada de resposta aleatória permanente, é usada para criar uma resposta “ruidosa” que é memorizada pelo cliente e reutilizada permanentemente no lugar da resposta real. A segunda etapa, chamada de resposta aleatória instantânea, relata a resposta “barulhenta” ao longo do tempo, eventualmente revelando-a completamente. A privacidade longitudinal e de longo prazo é garantida pelo uso da resposta randomizada permanente, enquanto o uso de um Instanta

A resposta aleatória aleatória fornece proteção contra possíveis externalidades de rastreamento.

A ideia de memoização subjacente acaba sendo crucial para a proteção da privacidade no caso em que múltiplas respostas são coletadas do mesmo participante ao longo do tempo.
Por exemplo, no caso da pergunta sobre o partido comunista desde o início do artigo, a memoização pode nos permitir fornecer privacidade diferencial ln(3) mesmo com um número infinito de respostas, desde que a resposta memoizada subjacente tenha esse nível de privacidade diferencial.

Por outro lado, sem memoização ou outra limitação nas respostas, a randomização não é suficiente para manter a negação plausível diante de coletas múltiplas. Por exemplo, se 75 de 100 respostas forem “Sim” para um único cliente no esquema de respostas aleatórias no início deste artigo, a resposta verdadeira terá sido “Não” em um extremamente improvável 1,39 × 10ÿ24 fração de casos.

A memoização é absolutamente eficaz em fornecer privacidade longitudinal apenas nos casos em que o valor real subjacente não muda ou muda de maneira não correlacionada. Quando os relatórios consecutivos dos usuários são correlacionados temporalmente, as garantias diferenciais de privacidade se desviam de seus níveis nominais e se tornam progressivamente mais fracas à medida que as correlações aumentam. Levado ao extremo, ao solicitar aos usuários que relatem diariamente sua idade em dias, são necessárias medidas adicionais para evitar a divulgação completa ao longo do tempo, como interromper a coleta após um determinado número de relatórios ou aumentar exponencialmente os níveis de ruído, conforme discutido mais adiante na Seção 6.

Para um cliente que relata uma propriedade que estritamente alterna entre dois valores verdadeiros, (a, b, a, b, a, b, a, b, . . .), as duas respostas aleatórias permanentes memorizadas para a e b serão ser reutilizado, repetidamente, para gerar dados de relatório RAPPOR. Assim, um invasor que obtém um número grande o suficiente de relatórios pode aprender esses valores “ruidosos” memorizados com certeza arbitrária – por exemplo, analisando separadamente as subsequências pares e ímpares. No entanto, mesmo neste caso, o invasor não pode ter certeza dos valores de a e b por causa da memorização. Dito isso, se a e b estiverem correlacionados, o invasor ainda poderá aprender mais do que aprenderia de outra forma; a manutenção da privacidade diante de qualquer correlação é discutida mais adiante nas Seções 3 e 6 (ver também [19]).

Na próxima seção descreveremos o algoritmo RAPPOR em detalhes. Em seguida, fornecemos intuição e justificativa formal para as razões pelas quais o algoritmo proposto satisfaz as rigorosas garantias de privacidade da privacidade diferencial. Em seguida, dedicamos várias seções à discussão dos aspectos técnicos adicionais do RAPPOR que são cruciais para seus usos potenciais na prática, como seleção de parâmetros, interpretação de resultados por meio de decodificação estatística avançada e experimentos que ilustram o que pode ser aprendido na prática. As seções restantes discutem nossa avaliação experimental, os modelos de ataque que consideramos, as limitações da técnica RAPPOR, bem como trabalhos relacionados.

2 O Algoritmo RAPPOR Fundamental Dado o valor v de um cliente, o algoritmo RAPPOR executado pela máquina do cliente reporta ao servidor um array de bits de tamanho k, que codifica uma representação “ruidosa” de seu verdadeiro valor v. A representação ruidosa de v é escolhido de forma a revelar uma quantidade controlada de informações sobre v, limitando a capacidade do servidor de aprender com confiança o que era v. Isso permanece verdadeiro mesmo para um cliente que envia um número infinito de relatórios sobre um determinado valor v.

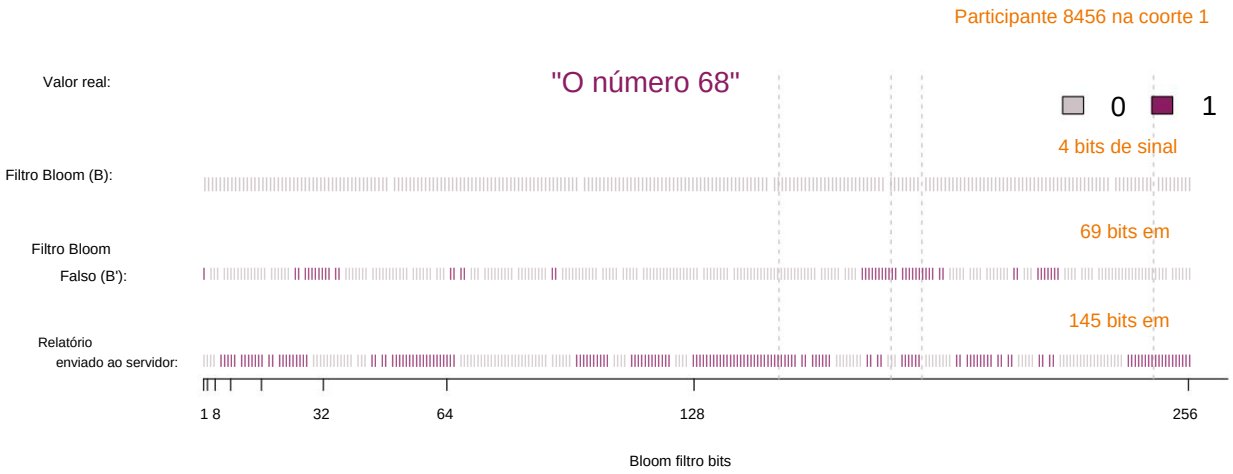

Figura 1: Vida útil de um relatório RAPPOR: O valor do cliente da string “O número 68” é hash no filtro Bloom B usando h (aqui 4) funções de hash. Para esta string, uma resposta aleatória permanente B é produzida e memorizada pelo cliente, e esta B é usada (e reutilizada no futuro) para gerar respostas aleatórias instantâneas S (a linha inferior), que são enviadas para o serviço de coleta.

Para fornecer garantias de privacidade tão fortes, o algoritmo RAPPOR implementa dois mecanismos de defesa separados, ambos baseados na ideia de resposta aleatória e podem ser ajustados separadamente dependendo do nível desejado de proteção de privacidade em cada nível. Além disso, incerteza adicional é adicionada através do uso de filtros Bloom que servem não apenas para tornar os relatórios compactos, mas também para complicar a vida de qualquer invasor (uma vez que qualquer bit no filtro Bloom pode ter vários itens de dados em sua pré-imagem).

O algoritmo RAPPOR recebe o valor verdadeiro v do cliente e os parâmetros de execução k, h, f, p, q, e é executado localmente na máquina do cliente realizando os seguintes passos:

1. Sinal. Hash do valor do cliente v no filtro Bloom B de tamanho k usando funções hash h.

2. Resposta aleatória permanente. Para cada valor de cliente v e bit i, 0 ÿ i < k em B, crie um valor de relatório binário B que é igual a

$$B_{eu} = \begin{cases} 1, \text{ com probabilidade } \frac{1}{2}f \\ 0, \text{ com probabilidade } \frac{1}{2}f \\ B_i, \text{ com probabilidade } 1 - f \end{cases}$$

onde f é um parâmetro ajustável pelo usuário que controla o nível de garantia de privacidade longitudinal.

Posteriormente, este B é memorizado e reutilizado como base para todos os relatórios futuros sobre este valor distinto v.

3. Resposta aleatória instantânea. Aloque um array de bits S de tamanho k e inicialize em 0. Defina cada bit i em S com probabilidades

$$P(S_i = 1) = \begin{cases} q, \text{ se } B_i = 1. \\ p, \text{ se } B_i = 0. \end{cases}$$

4. Relatório. Envie o relatório S gerado para o servidor.

Existem muitas variantes diferentes do mecanismo de resposta aleatória acima. Nosso principal objetivo ao selecionar esses duas versões particulares era tornar o esquema intuitivo e fácil de explicar.

A resposta aleatória permanente (etapa 2) substitui o valor real B por um valor ruidoso aleatório derivado B . B pode ou não conter qualquer informação sobre B, dependendo se os bits de sinal do filtro de Bloom estão sendo substituídos por 0s aleatórios com probabilidade f. A resposta aleatória permanente garante privacidade $\frac{1}{2}f$ devido à limitada do adversário de diferenciar entre bits de sinal verdadeiros e “ruidosos”. É absolutamente crítico que todos os relatórios futuros sobre as informações sobre B usem o mesmo valor aleatório de B para evitar um ataque de “média”, no qual um adversário estima o valor verdadeiro observando várias versões ruidosas dele.

A resposta aleatória instantânea (etapa 3) desempenha várias funções importantes. Em vez de relatar B diretamente a cada solicitação, o cliente relata uma versão aleatória de B . Essa modificação aumenta significativamente a dificuldade de rastrear um cliente com base em B, que poderia ser único em uma população, mesmo com vários relatórios longitudinais. Ele também fornece garantias de privacidade de curto prazo mais fortes (já que estamos adicionando mais ruído ao relatório), que podem ser ajustadas de forma independente para equilibrar riscos de curto prazo versus riscos de longo prazo. Por meio do ajuste dos parâmetros desse mecanismo, podemos equilibrar efetivamente a utilidade contra diferentes modelos de invasores.

A Figura 1 mostra uma execução aleatória do algoritmo RAPPOR. Aqui, o valor de um cliente é v = “68”, o tamanho do filtro Bloom é k = 256, o número de funções hash é h = 4 e os parâmetros de resposta aleatória ajustáveis são: p = 0,5, q = 0,75, ef = 0,5. A matriz de bits relatada enviada ao servidor é mostrada na parte inferior da figura. 145 de 256 bits são definidos no relatório. Dos quatro bits do filtro Bloom em B (segunda linha), dois são propagados para o filtro Bloom ruidoso B . Destes dois bits, ambos são ativados no relatório final.

Os outros dois bits nunca são reportados por este cliente devido à natureza permanente de B . Com várias coletas deste cliente no valor “68”, o atacante mais poderoso acabaria aprendendo B, mas continuaria a ter lim

capacidade limitada de raciocinar sobre o valor de B, medido pela garantia diferencial de privacidade. Na prática, aprender sobre o valor real v do cliente é ainda mais difícil porque vários valores são mapeados para os mesmos bits no filtro Bloom [4].

2.1 Modificações do RAPPOR

O algoritmo RAPPOR pode ser modificado de várias maneiras, dependendo das particularidades do cenário em que a coleta de dados de preservação da privacidade é necessária. Aqui, listamos três cenários comuns em que a omissão de certos elementos do algoritmo RAPPOR leva a um procedimento de aprendizado mais eficiente, especialmente com tamanhos de amostra menores.

- RAPPOR único. A cobrança única, aplicada pelo cliente, não requer proteção de privacidade longitudinal. A etapa de resposta aleatória instantânea pode ser ignorada neste caso e uma randomização direta no valor do cliente real é suficiente para fornecer uma forte proteção de privacidade.

- RAPPOR básico. Se o conjunto de strings sendo coletado for relativamente pequeno e bem definido, de forma que cada string possa ser mapeada de forma determinística para um único bit na matriz de bits, não há necessidade de usar um filtro Bloom com várias funções de hash. Por exemplo, coletar dados sobre o sexo do cliente poderia simplesmente usar um array de dois bits com “masculino” mapeado para o bit 1 e “feminino” mapeado para o bit 2. Essa modificação afetaria a etapa 1, onde um filtro Bloom seria substituído por um mapeamento determinístico de cada string candidata para um e apenas um bit na matriz de bits. Nesse caso, o número efetivo de funções hash, h, seria 1.

- RAPPOR básico único. Esta é a configuração mais simples do mecanismo RAPPOR, combinando as duas primeiras modificações ao mesmo tempo: uma rodada de randomização usando um mapeamento determinístico de strings em seus próprios bits exclusivos.

3 Privacidade diferencial do RAPPOR

A escala e a disponibilidade de dados no mundo atual tornam viáveis ataques cada vez mais sofisticados, e qualquer sistema que espere resistir a tais ataques deve ter como objetivo garantir garantias de privacidade rigorosas, e não meramente intuitivas. Para nossa análise, adotamos a noção rigorosa de privacidade, privacidade diferencial, que foi introduzida por Dwork et al [12] e tem sido amplamente adotada [10]. A definição visa garantir que a saída do algoritmo não dependa significativamente dos dados de nenhum indivíduo em particular. A quantificação do aumento do risco que a participação em um serviço representa para um indivíduo pode, portanto, capacitar os clientes a tomar uma decisão mais informada sobre se desejam que seus dados façam parte da coleta.

Formalmente, um algoritmo aleatório A satisfaz a privacidade -diferencial [12] se para todos os pares de valores do cliente v1 e v2 e para todo R ÿ Range(A),

P(A(v1) ÿ R) ÿ e P(A(v2) ÿ R).

Provamos que o algoritmo RAPPOR satisfaz a definição de privacidade diferencial a seguir. Intuitivamente, a parte da resposta aleatória permanente garante que o valor “ruidoso” derivado do valor verdadeiro proteja a privacidade, e a resposta aleatória instantânea fornece proteção contra o uso dessa resposta por um rastreador longitudinal.

3.1 Privacidade Diferencial da Resposta Randomizada Permanente

Teorema 1. A resposta randomizada Permanente (Passos 1 e 2 do RAPPOR) satisfaz privacidade ÿ-diferencial onde

$$\ddot{y} = 2h \ln \frac{1 - \frac{1}{2}f}{\frac{1}{2}f}.$$

Prova. Seja S = s1, . . . , sk seja um relatório randomizado gerado pelo algoritmo RAPPOR. Então a probabilidade de observar qualquer relatório dado S dado o verdadeiro valor do cliente v e assumindo que B é conhecido é

$$\begin{aligned} P(S = s|V = v) &= P(S = s|B, B, v) \cdot P(B|B, v) \cdot P(B|v) \\ &= P(S = s|B) \cdot P(B|B) \cdot P(B|v) \\ &= P(S = s|B) \cdot P(B|B). \end{aligned}$$

Como S é condicionalmente independente de B, dado que a , a primeira probabilidade de B não fornece nenhuma informação adicional sobre B. P(B|B) é, no entanto, crítico para a proteção longitudinal da privacidade. As probabilidades relevantes são

$$P(bi = 1|bi = 1) = \frac{1}{2}f + 1 \ddot{y} f = 1 \ddot{y} \frac{1}{2}f \text{ e}$$

$$P(bi = 1|bi = 0) = \frac{1}{2}f.$$

Sem perda de generalidade, deixe o Bloom filtrar os bits 1, . . . , ser definido, ou seja, $h = \{b1 = 1, \ldots, bk = 0\}$. $bh = 1, bh+1 = 0, \ldots,$ b Então,

$$\begin{aligned} P(B = b|B = b\ddot{y}) = {} & \left(\frac{1}{2}f\right)^{b_1} \left(1 - \frac{1}{2}f\right)^{1-b_1} \times \ldots \\ & \times \left(\frac{1}{2}f\right)^{b_h} \left(1 \ddot{y} \frac{1}{2}f\right)^{1-b_h} \times \ldots \\ & 1 \times 1 \ddot{y} f 2^{b_{h+1}} \left(\frac{1}{2}f\right)^{1\ddot{y}b_{h+1}} \times \ldots \\ & 1 \times 1 \ddot{y} f 2^{b_k} \left(\frac{1}{2}f\right)^{1-b_k}. \end{aligned}$$

Seja RRÿ a razão de duas dessas probabilidades condicionais com valores distintos de B, B1 e B2, ou seja, RRÿ = P (BÿRÿ|B=B1) P (BÿRÿ|B=B2) . Para que a condição de privacidade diferencial seja mantida, RRÿ precisa ser limitado por exp(ÿ).

$$\begin{aligned} RR\ddot{y} &= \frac{P(B\ddot{y}R\ddot{y}|B = B1)}{P(B\ddot{y}R\ddot{y}|B = B2)} \\ &= \frac{\sum_{B i\ddot{y}R\ddot{y}} P(B = B i|B = B1)}{\sum_{P(B = B B i\ddot{y}R\ddot{y}} |B = B2)} \\ &\leq \frac{P(B = B B i\ddot{y}R\ddot{y}\ 2(b\ 2+\ldots+b\ h\ddot{y}b\ h+1\ddot{y}b\ h+2\ddot{y}..\ddot{y}b\ 2h)}{|B = B2)} \quad \text{(pela Observação 8)} \\ &= \left(\frac{1}{2}f\right) \times 1\ddot{y}2^{\left(\frac{1}{2}f\right)^{2(b\ h+1+b\ h+2+\ldots+b\ 2h\ddot{y}b\ 1\ddot{y}b\ 2\ddot{y}\ldots\ddot{y}b\ h)}}. \end{aligned}$$

A sensibilidade é maximizada quando $b\ 1$ e $b = b_{h+1} = bh+2 = \ldots = b_{2h} = 1\ 2\ 1\ 1\ddot{y}f\ 2\ f = \ldots = b_h = 0$. Então,

$$RR\ddot{y} = \frac{\frac{1}{2}f}{\frac{1}{2}f}^{2h} \text{ e } \ddot{y} = 2h \ln \frac{1\ 1\ddot{y}f\ 2}{\frac{1}{2}f}. \quad \square$$

Observe que $_{ý}$ não é uma função de k. É verdade que um k menor, ou uma taxa mais alta de colisão de bits do filtro de Bloom, algumas vezes melhora a proteção de privacidade, mas, por si só, não é suficiente nem necessário para fornecer privacidade diferencial.

3.2 Privacidade Diferencial da Resposta Instantânea Randomizada

Com uma única coleta de dados de cada cliente, o conhecimento do invasor sobre B deve vir diretamente de um único relatório S gerado pela aplicação da randomização duas vezes, fornecendo assim um nível mais alto de proteção de privacidade do que sob a suposição de conhecimento completo de B .

Por causa de uma randomização em duas etapas, a probabilidade de observar um 1 em um relatório é uma função tanto de q como de p, bem como de f.

Lema 1. A probabilidade de observar 1 dado que o bit do filtro de Bloom subjacente foi definido é dada por

$$q^{ý} = P(Si = 1|bi = 1) = \frac{1}{2}f(p + q) + (1 ÿ f)q.$$

A probabilidade de observar 1 dado que o bit do filtro Bloom subjacente não foi definido é dada por

$$p^{ý} = P(Si = 1|bi = 0) = \frac{1}{2}f(p + q) + (1 ÿ f)p.$$

Omitimos a prova porque o raciocínio é direto de que as probabilidades em ambos os casos são misturas de respostas aleatórias e verdadeiras com a proporção de mistura f.

Teorema 2. A resposta aleatória instantânea (Etapa 3 de RAPPOR) satisfaz a privacidade 1-diferencial, onde $q ÿ(1ÿp ÿ) h \log e p pÿ(1ÿqÿ)$ $_1 = \frac{\quad}{\quad}$ e $q^{ý}$ ^{ý} como definido no Lema 1.

Prova. A prova é análoga ao Teorema 1. Seja RR1 a razão de duas probabilidades condicionais, ou seja, RR1 = $\frac{P (SÿR|B=B1)}{P (SÿR|B=B2)}$. Para satisfazer a condição de privacidade diferencial, essa proporção deve ser limitada por exp(1).

$$RR1 = \frac{P(S ÿ R|B = B1)}{P(S ÿ R|B = B2)}$$

$$= \frac{\prod_{sjÿR} P(S = sj|B = B1)}{\prod_{P(S = sj|B = B2)} sjÿR}$$

$$\max_{ÿ (1 ÿ p)(1 ÿ q) + pq} \frac{P(S = sj|B = B1) ÿ}{P(S = sj|B = B2) sjÿR}$$

$$= \frac{q}{h}$$

e

$$_1 = \log h \frac{qÿ (1 ÿ p}{ÿ) p ÿ(1 ÿ q ÿ)}.$$

□

A prova acima se estende naturalmente a N relatórios, pois cada relatório que não é alterado contribui com um valor fixo para a probabilidade total de observar todos os relatórios e entra tanto no denominador quanto no denominador de forma multiplicativa (por causa da independência). Como nossa estrutura de privacidade diferencial considera entradas que diferem apenas em um único registro, j, (o conjunto de relatórios D1 torna -se D2 , diferindo em um único relatório, Sj = sj), todos os dos termos do produto acabam se anulando na razão

$$\frac{P(S1 = s1, S2 = s2, \ldots, Sj = sj, \ldots, SN = sN | B1)}{P(S1 = s1, S2 = s2, \ldots, Sj = sj, \ldots, SN = sN | B2)} =$$

$$\frac{\prod_{i=1}^{N} P(Si = si|B1)}{\prod_{i=1}^{N} P(Si = si|B2)} = \frac{P(Sj = sj|B1)}{P(Sj = sj|B2)}.$$

O cálculo de n para a enésima coleção não pode ser feito com suposições adicionais sobre a eficácia com que o atacante pode aprender B a partir dos relatórios coletados. Continuamos trabalhando para fornecer esses limites em várias estratégias de aprendizado. No entanto, à medida que N se torna grande, o limite se aproxima de ÿ, mas sempre permanece estritamente menor.

4 Decodificação de Relatórios de Alta Utilidade Na maioria dos casos, o objetivo da coleta de dados usando RAPPOR é aprender quais strings estão presentes na população amostrada e quais são suas frequências correspondentes. Como fazemos uso do filtro Bloom (perda de informação) e propositalmente adicionamos ruído para proteção da privacidade, a decodificação requer técnicas estatísticas sofisticadas.

Para facilitar o aprendizado, antes do início de qualquer coleta de dados, cada cliente é designado aleatoriamente e se torna um membro permanente de uma das m coortes. Coortes implementam diferentes conjuntos de funções h hash para seus filtros Bloom, reduzindo assim a chance de colisões acidentais de duas strings em todos eles. A redundância introduzida pela execução simultânea de m coortes melhora muito a taxa de falsos positivos. A escolha de m deve ser considerada com cuidado, no entanto. Quando m é muito pequeno, as colisões ainda são bastante prováveis, enquanto quando m é muito grande, cada coorte individual fornece sinal insuficiente devido ao tamanho pequeno da amostra (aproximadamente N/m, onde N é o número de relatórios).
Cada cliente deve relatar seu número de coorte a cada relatório enviado, ou seja, não é privado, mas tornado privado.

Propomos a seguinte abordagem para aprender com os relatórios coletados:

- Estime o número de vezes que cada bit i dentro da coorte j, tij , é realmente definido para cada coorte, [illegible] um conjunto de relatórios Nj , a estimativa é dada por

$$= \frac{cij ÿ (p + tij \frac{1}{2}fq ÿ \frac{1}{2}fp)Nj}{(1 ÿ f)(q ÿ p)}.$$

  Seja Y um vetor de tij 's, i ÿ [1, k], j ÿ [1, m].

- Crie uma matriz de projeto X de tamanho km × M onde M é o número de strings candidatas em consideração.
  X é principalmente 0 (esparso) com 1 nos bits do filtro Bloom para cada string para cada coorte. Portanto, cada coluna de X contém hm 1's nas posições em que uma string candidata específica foi mapeada pelos filtros Bloom em todas as m coortes. Use a regressão Lasso [26] para ajustar um modelo Y ÿ X e selecione strings candidatas correspondentes a coeficientes diferentes de zero.

- Ajustar uma regressão regular de mínimos quadrados usando as variáveis selecionadas para estimar contagens, seus erros padrão e valores-p.

- Compare os valores-p com um nível corrigido de Bonferroni de ÿ/ M = 0,05/M para determinar quais frequências são

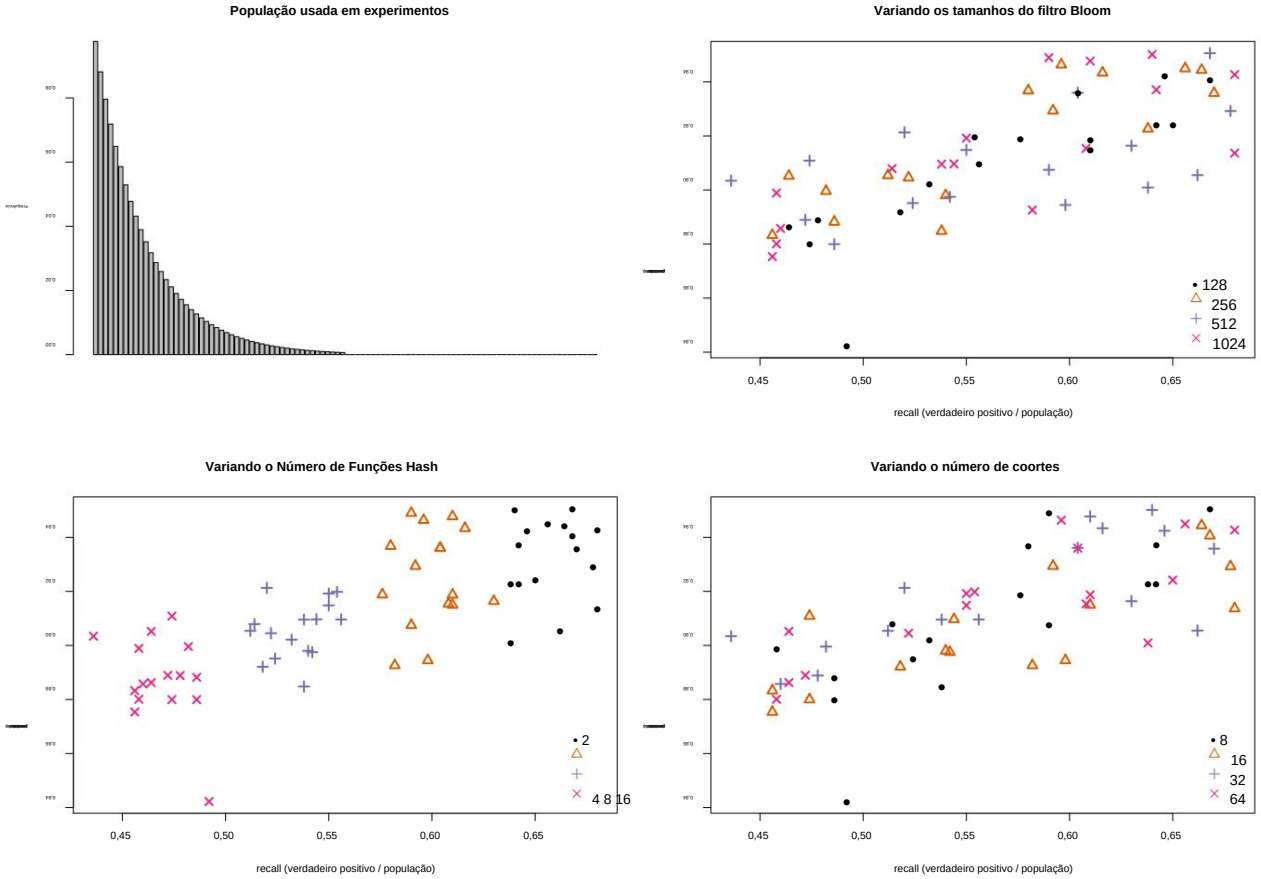

Figura 2: Recall versus precisão dependendo da escolha dos parâmetros k, h e m. O primeiro painel mostra a verdadeira distribuição da população a partir da qual os relatórios RAPPOR foram amostrados. Os outros três painéis variam um dos parâmetros enquanto mantêm os outros dois fixos. A melhor precisão e recuperação são alcançadas com o uso de 2 funções de hash, enquanto as opções de k e m não mostram preferências claras.

estatisticamente significante a partir de 0. Alternativamente, o controle da Taxa de Descoberta Falsa (FDR) no nível ÿ usando o procedimento de Benjamini-Hochberg [3], por exemplo, poderia ser usado.

4.1 Seleção de Parâmetros

A implementação prática do algoritmo RAPPOR requer a especificação de vários parâmetros. p, q, f e o número de funções de hash h controlam o nível de privacidade para coleções únicas e longitudinais. Claramente, se nenhum dado longitudinal estiver sendo coletado, então podemos usar a modificação RAPPOR única. Com exceção de h, a escolha dos valores para esses parâmetros deve ser orientada exclusivamente pelo nível de privacidade desejado. em si pode ser escolhido dependendo das circunstâncias do processo de coleta de dados; os valores na literatura variam de [1, 10] a 10 (veja Tabela 1 de [15]).

O tamanho do filtro Bloom, k, o número de coortes, m e h também devem ser especificados a priori. Além de h, nem k nem m estão relacionados às considerações de privacidade de pior caso e devem ser selecionados com base nas propriedades de eficiência do algoritmo em reconstruir o sinal dos relatórios ruidosos.

Fizemos várias simulações (média de 10 réplicas) para entender como esses três parâmetros afetam a decodificação; veja a Figura 2. Todos os cenários assumidos = ln(3) garantia de privacidade. Como foi simulado apenas um único relato de cada usuário, foi utilizado o One-time RAPPOR. A população amostrada é mostrada no primeiro painel e contém 100 sequências diferentes de zero com 100 sequências com probabilidade zero de ocorrência. Freqüências de strings diferentes de zero seguiram uma distribuição exponencial conforme mostrado na figura.

Nos outros três painéis, o eixo x mostra a taxa de rechamada e o eixo y mostra a taxa de precisão. Em todos os três painéis, o mesmo conjunto de pontos é plotado e apenas rotulado de forma diferente, dependendo de qual parâmetro muda em um painel específico. Cada ponto representa uma recuperação média e precisão para uma combinação única de k, h e m. Por exemplo, o segundo painel mostra o efeito do tamanho do filtro Bloom tanto na precisão quanto na rechamada, mantendo ambos h e m fixos. É difícil tirar conclusões definitivas sobre o tamanho ideal do filtro de Bloom, pois tamanhos diferentes funcionam de maneira semelhante, dependendo dos valores de h e m. O terceiro painel, no entanto, mostra uma clara preferência por usar apenas duas funções hash do ponto de vista da utilidade, pois a diminuição do número de funções hash usadas aumenta o esperado

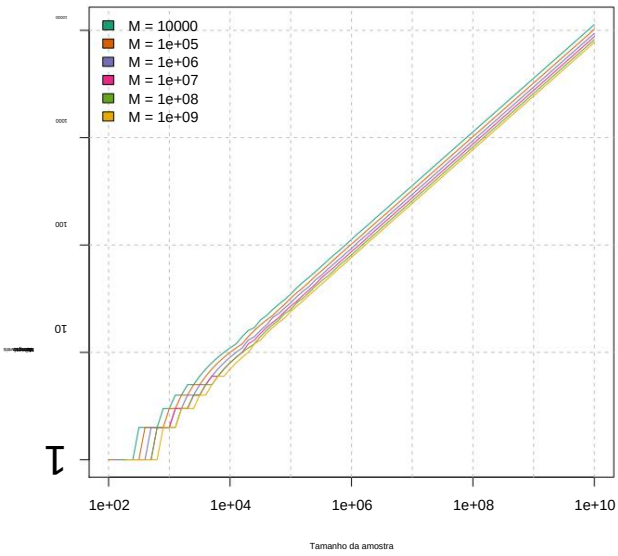

Figura 3: Tamanho da amostra versus o limite superior nas cordas cuja frequência pode ser aprendida. Sete linhas coloridas representam diferentes cardinalidades do conjunto de strings candidato. Aqui, p = 0,5, q = 0,75 e f = 0.

lembrar. O quarto painel, assim como o segundo, não indica definitivamente a direção ótima para a escolha do número de coortes.

4.2 O que podemos aprender?

Na prática, é comum usar limites no número de envios únicos para garantir alguma privacidade. No entanto, abundam os argumentos sobre como esses limites devem ser estabelecidos e, na maioria das vezes, eles são baseados em uma 'sensação' do que é aceito e carecem de qualquer justificativa objetiva. O RAP POR também requer , um parâmetro ajustável pelo usuário, que impõe um limite inferior da frequência, ou seja, coloca um limite inferior no número de vezes que uma string precisa ser observada em uma amostra antes que possa ser reli habilmente identificados e sua frequência estimada. A Figura 3 mostra a relação entre o tamanho da amostra (eixo x) e o limite superior teórico (eixo y) de quantas strings podem ser detectadas naquele tamanho de amostra para uma escolha específica de p = 0,5 e q = 0,75 (com f = 0) em um determinado nível de confiança ÿ = 0,05.

Talvez seja surpreendente que não aprendamos mais em tamanhos de amostra muito grandes (por exemplo, um bilhão). A principal razão é que, à medida que o número de cordas na população aumenta, suas frequências diminuem proporcionalmente e tornam-se difíceis de detectar nessas frequências baixas.

Só podemos detectar com segurança cerca de 10.000 cordas em uma amostra de dez bilhões e cerca de 1.000 com uma amostra de cem milhões. Uma regra geral é ÿ N /10, onde N é o tamanho da amostra. Esses cálculos teóricos são baseados no algoritmo Basic One-time RAPPOR (a terceira modificação) e são o limite superior do que pode ser aprendido, pois não há incerteza adicional introduzida pelo uso do filtro Bloom. Detalhes dos cálculos são mostrados no Apêndice.

Ao fornecer privacidade diferencial ln(3) para uma coleta de tempo, se alguém quiser detectar itens com frequência 1%, então um milhão de amostras são necessárias, 0,1% exigiria um tamanho de amostra de 100 milhões e 0,01% itens seriam identificados apenas em uma amostra de 10 bilhões.

A eficiência do algoritmo RAPPOR não modificado é significativamente inferior quando comparada ao RAPPOR básico de uso único (o preço da compactação). Mesmo para o RAPPOR básico único, o limite fornecido pode ser teoricamente alcançado apenas se a distribuição subjacente das frequências das cordas for uniforme (uma condição sob a qual a menor frequência é maximizada). Com a presença de várias cordas de alta frequência, há menos probabilidade de sobrar massa para a cauda e, com a queda de suas frequências, sua detectabilidade é prejudicada.

5 Experimentos e avaliações Demonstramos nossa abordagem usando dois exemplos de coleção simulados e dois do mundo real. O primeiro simulado usa o Basic One-time RAPPOR onde aprendemos a forma da distribuição Normal subjacente. O segundo exemplo simulado usa RAPPOR não modificado para coletar strings cujas frequências exibem decaimento exponencial. O terceiro exemplo é extraído de um conjunto de dados do mundo real em processos executados em um conjunto de máquinas Windows. O último exemplo é baseado nas coleções de configurações do navegador Chrome.

5.1 Relatórios sobre a distribuição normal Para ter uma noção de quão eficazmente podemos aprender a distribuição subjacente de valores relatados por meio do Basic One-time RAPPOR, simulamos o aprendizado da forma da distribuição Normal (arredondada para números inteiros) com média 50 e padrão desvio 10. As restrições de privacidade foram: q = 0,75 ep = 0,5 fornecendo = ln(3) privacidade diferencial (f = 0).

Os resultados são mostrados na Figura 4 para três tamanhos de amostra diferentes. Com 10.000 relatórios, os resultados são muito ruidosos para obter uma boa estimativa da forma. A curva de sino normal começa a surgir já com 100.000 relatórios e em um milhão de relatórios é traçada muito de perto. Observe o ruído nas caudas esquerda e direita, onde basicamente não há sinal. É exigido pela condição de privacidade diferencial e também dá uma noção de quão incertas são nossas contagens estimadas.

5.2 Relatórios sobre um conjunto de strings distribuído exponencialmente A verdadeira distribuição subjacente de strings das quais amostramos é mostrada na Figura 5. Ela mostra o declínio exponencial comumente encontrado na frequência de strings com vários “rebatedores pesados” e a cauda longa. Depois de amostrar 1 milhão de valores (um evento de coleta por usuário) dessa população aleatoriamente, aplicamos o RAPPOR para gerar 1 milhão de relatórios com p = 0,5, q = 0,75, f = 0,5, duas funções hash, tamanho do filtro Bloom de 128 bits e 16 coortes.

Após a análise estatística usando a correção de Bonferroni discutida acima, 47 strings foram estimadas como tendo contagens significativamente diferentes de 0. Apenas 2 das 47 strings eram falsos positivos, significando que suas contagens verdadeiras eram realmente 0, mas estimadas como significativamente diferentes. As 20 principais cadeias detectadas com suas estimativas de contagem, erros padrão, valores-p e pontuações-z (SNR) são mostradas na Tabela 1. Pequeno

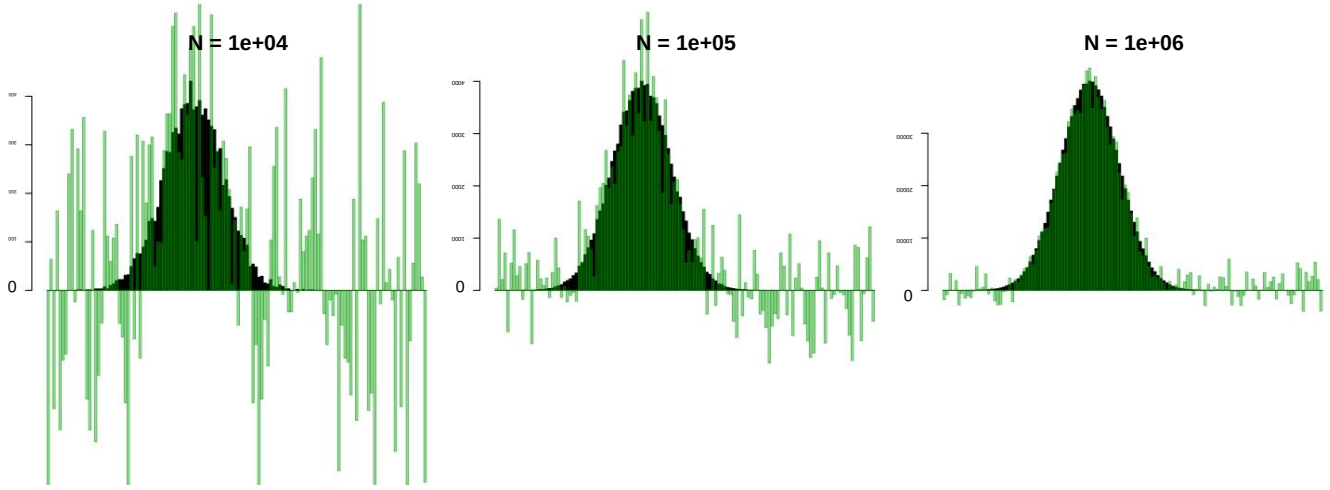

Figura 4: Simulações de aprendizado da distribuição normal com média 50 e desvio padrão 10. Os parâmetros de privacidade do RAPPOR são q = 0,75 ep = 0,5, correspondendo a = ln(3). A verdadeira distribuição da amostra é mostrada em preto; verde claro mostra a distribuição estimada com base nos relatórios RAPPOR decodificados. Não assumimos conhecimento a priori da distribuição Normal na aprendizagem. Se tais informações prévias estivessem disponíveis, poderíamos melhorar significativamente ao aprender a forma da distribuição via suavização.

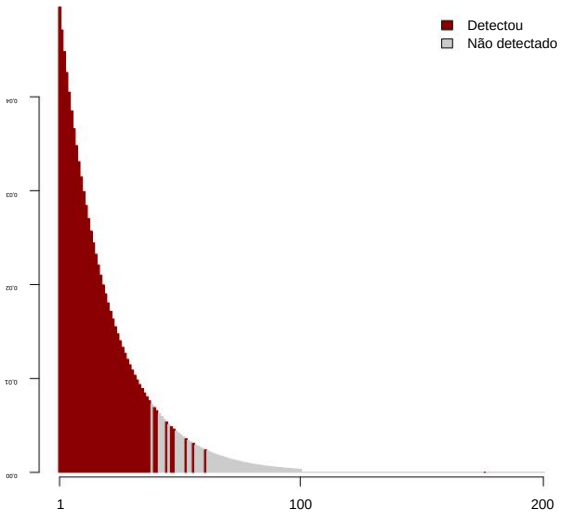

Figura 5: População de cordas com suas frequências verdadeiras no eixo vertical (0,01 é 1%). As strings detectadas pelo RAPPOR são mostradas em vermelho escuro.

Cadeia Est. STDEV P.VALUE VERDADE DE SNR V 1 48803 2808
5.65E-63 [illegible]
4.30e-47 [illegible]
1,31E-44 [illegible] 0,04 13,71
–
–
–
–
–
V 10 34196 2828 1,72E-32 31234 0,03 12,09 V 9 32207 2805 1,45E-29
33106 0,03 11,48
V 12 30688 2822 9.07E-27 28295 0,03 10,87 V 11 29630 2831 5.62E-25
29908 0,03 10,47 V 14 27366 2850 2,33E-21 25984 0,03 V 19 2360
2803 3,41e -7 2005702 0,02020202 2360 2803 3803 3,41e -7 2005702 0,02
0. 26913 0,03 V 15 21752 2825 2,15E-4 8,51
– 7,90
– 7,70
– 7,15
– 6,89
– 6,54
V 21 18267 2828 1.33E-10 17878 0.02 6.46

Tabela 1: As 20 principais sequências com suas frequências estimadas, desvios padrão, valores p, contagens verdadeiras e relações sinal/ruído (SNR ou pontuações z).

Os valores de p mostram alta confiança em nossa avaliação de que as contagens verdadeiras são muito maiores que 0 e, de fato, a comparação das colunas 2 e 5 confirma isso. A Figura 5 mostra todas as 47 strings detectadas em vermelho escuro. Todas as strings comuns acima da frequência de aproximadamente 1% foram detectadas e a cauda longa permaneceu protegida pelo mecanismo de privacidade.

5.3 Relatórios sobre nomes de processos do Windows Coletamos 186.792 relatórios de 10.133 computadores Windows diferentes, amostrando processos em execução ativa em cada máquina. Em média, pouco mais de 18 nomes de processos foram coletados de cada máquina com o objetivo de recuperar os mais comuns e estimar a frequência de um binário particularmente malicioso chamado “BADAPPLE.COM”.

Esta coleção usou 128 filtros Bloom com 2 funções hash e 8 coortes. Os parâmetros de privacidade foram escolhidos de modo que = 1,0743 com q = 0,75, p = 0,5 ef = 0,5.
Dada essa configuração, esperávamos com otimismo descobrir processos com frequência de pelo menos 1,5%.

Identificamos 10 processos mostrados na Tabela 2 variando em frequência entre 2,5% e 4,5%. Eles foram identificados controlando a taxa de falsa descoberta em 5%. O processo “BADAP PLE.COM” foi estimado com frequência de 2,6%.
Os outros 9 processos eram tarefas comuns do Windows que esperaríamos executar em quase todas as máquinas Windows.

Machine Translated by Google

Tabela 2: Processos do Windows detectados.

Nome do processo	Husa. Stdev P.value Prop.
RASERVER.EXE 8054 1212 1.56E-11 0.04 RUNDLL32.EXE 7488 1212 3.32E-10 0.04 CONHOSTSPPSVC.EXE... 0.03 AITAGENT.EXE 2.037E-1215 1.51E-05 0.03 MSIEXEC.EXE... 1215 1212 2.53E-05 0.03 BADAPPLE.COM 4760 1212 ... 0.03 3.05E-05 0.03 DEFRAG.EXE 4760 1212 4.34E-05 0.03	

5.4 Relatórios sobre as páginas iniciais do Chrome O

navegador Chrome implementou e implantou o RAPPOR para coletar dados sobre clientes Chrome [9]. A coleta de dados foi limitada a alguns dos usuários do Chrome que optaram por enviar estatísticas de uso ao Google e a determinadas configurações do Chrome, com coleta diária de aproximadamente 14 milhões de entrevistados.

As configurações do Chrome, como página inicial, mecanismo de pesquisa e outros, costumam ser alvo de software malicioso e alteradas sem o consentimento dos usuários. Para entender quem são os principais players, é fundamental conhecer a distribuição dessas configurações em um grande número de instalações do Chrome. Aqui, nos concentramos em aprender a distribuição de homepages e demonstramos o que pode ser aprendido com uma dúzia de milhões de relatórios com fortes garantias de privacidade.

Esta coleção usou 128 filtros Bloom com 2 funções hash e 32 coortes. Os parâmetros de privacidade foram escolhidos de modo que = 0,5343 1 com q = 0,75, p = 0,5 ef = 0,75. Dada esta configuração, com otimismo, a análise RAPPOR pode descobrir domínios de URL de homepage, com confiança estatística, se sua frequência exceder 0,1% da população respondente. Na prática, isso significa que mais de aproximadamente 14 mil clientes devem reportar sobre o mesmo domínio URL, antes que ele possa ser identificado na população pela análise RAPPOR.

A Figura 6 mostra as frequências relativas de 31 domínios de página inicial inesperados descobertos pela análise RAPPOR. (Como nem todos são necessariamente maliciosos, a figura não inclui as strings de domínio de URL reais que foram identificadas.)

Como era de se esperar, existem várias páginas iniciais populares, provavelmente definidas intencionalmente pelos usuários, juntamente com uma longa cauda de URLs relativamente raros. Embora menos de 0,5% de 8.616 URLs candidatos forneçam evidências estatísticas suficientes para sua presença (após a correção FDR), eles respondem coletivamente por cerca de 85% da massa total de probabilidade.

6 Modelos de Ataque e Limitações

Consideramos três tipos de invasores com diferentes capacidades de coleta de relatórios RAPPOR.

O invasor menos poderoso tem acesso a um único relatório de cada usuário e é limitado por um nível único de privacidade diferencial 1 sobre quanto ganho de conhecimento é possível. Este invasor corresponde a um espião que possui acesso temporário de escuta aos relatórios dos usuários.

Presume-se que um invasor em janela tenha acesso aos dados de um cliente durante um período de tempo bem definido. Este invasor, dependendo da sofisticação de seu modelo de aprendizado, pode obter mais informações sobre um usuário do que o invasor

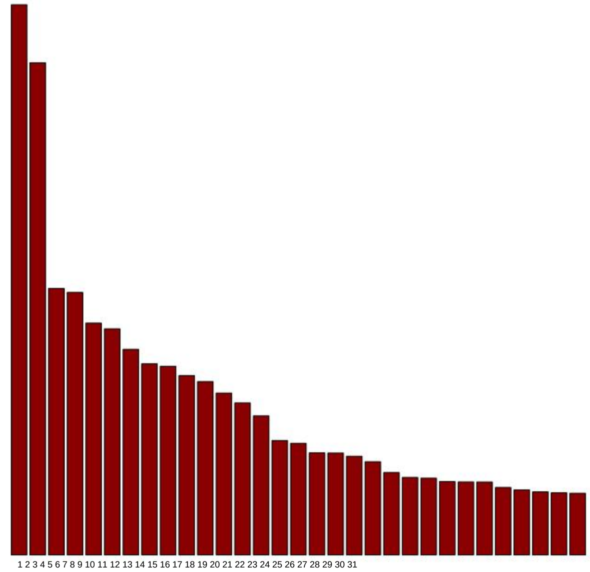

Figura 6: Frequências relativas dos 31 principais domínios inesperados da página inicial do Chrome encontrados pela análise de aproximadamente 14 milhões de relatórios RAPPOR, excluindo os domínios esperados (a página inicial “google.com”, etc.).

do primeiro tipo. No entanto, a melhoria em sua capacidade de violar a privacidade é estritamente limitada pela garantia de privacidade diferencial longitudinal de Esse invasor mais poderoso pode corresponder a um adversário como um funcionário malicioso do serviço de nuvem, que pode ter acesso temporário a relatórios ou acesso a um log de relatórios com limite de tempo.

Supõe-se que o terceiro tipo de invasor tenha recursos de coleta ilimitados e possa aprender a resposta aleatória permanente B com certeza absoluta. Por causa da randomização realizada para obter B de B, ela também está vinculada à garantia de privacidade de e não pode melhorar esse vínculo com mais coleta de dados. Isso corresponde a um adversário de pior caso, mas ainda não tem acesso direto aos valores de dados reais no cliente.

Apesar de imaginar um modelo de privacidade completamente local, onde os próprios usuários liberam dados de forma a preservar a privacidade, os operadores das coleções RAPPOR, no entanto, podem facilmente manipular o processo para aprender mais informações do que ÿ. Solicitar pelo ipate nominal mais de uma vez em uma determinada coleção resulta que usuários participem... garantidos em várias respostas permanentes aleatórias para cada usuário e anula parcialmente os benefícios da memoização. No mundo centrado na web, os usuários usam várias contas e vários dispositivos e podem participar várias vezes sem saber, liberando mais informações do que esperavam. Esse problema pode ser mitigado até certo ponto executando coletas por conta e compartilhando uma resposta aleatória permanente comum. Observe a função do operador para garantir que tais processos estejam em vigor e a confiança exigida ou assumida por parte do usuário.

É provável que alguns invasores tenham como objetivo atingir usuários específicos, isolando e analisando relatórios desse usuário ou de um pequeno grupo de usuários que os inclua. Mesmo assim, alguns

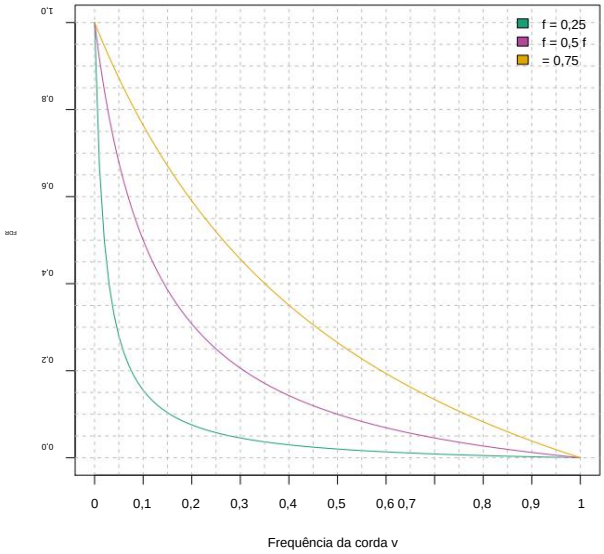

Figura 7: Taxa de detecção falsa (FDR) em função da frequência do string e f. Identificar sequências raras em uma população sem introduzir um grande número de descobertas falsas é inviável. Além disso, FDR é proporcional a f.

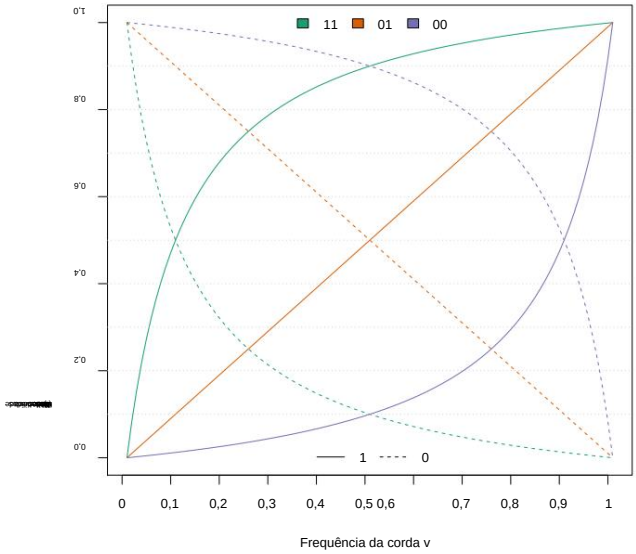

Figura 8: Probabilidades exatas para inferir o valor verdadeiro v dados os dois bits observados em um relatório RAPPOR S correspondente aos dois bits definidos pela string v. Para strings raras, mesmo quando ambos os bits são definidos como 1 (linhas verdes), é ainda muito mais provável que o cliente não tenha relatado v, mas algum outro valor.

os usuários escolhidos aleatoriamente não precisam temer tais ataques: com probabilidade $1 - \frac{h}{2}$, os clientes geram uma resposta aleatória posições dos bits do filtro Bloom definidos. Como esses clientes não estão contribuindo com nenhuma informação útil para o processo de coleta, direcioná-los individualmente por um invasor é contraproducente. Um invasor não tem nada a aprender sobre esse usuário em particular. Além disso, para todos os usuários, em todos os momentos, há negação plausível proporcional à fração de clientes que não fornecem informações.

Em um cenário de ataque específico, imagine um invasor interessado em saber se um determinado cliente tem um determinado valor v, cuja frequência populacional é conhecida como fv. A evidência mais forte em apoio de v vem na forma de ambos os bits do filtro Bloom para v serem definidos no relatório do cliente (se duas funções de hash forem usadas). O invasor pode formular seu conjunto de alvos selecionando todos os relatórios com esses dois bits definidos. No entanto, esse conjunto perderá alguns clientes com v e incluirá outros clientes que não relataram v. A taxa de descoberta falsa (FDR) é a proporção de clientes no conjunto de destino que relataram um valor diferente de v. A Figura 7 mostra o FDR como um função de fv, a frequência da string v. Sem dúvida, para valores relativamente raros, a maioria dos clientes no conjunto de destino terá, de fato, um valor diferente de v, o que provavelmente impedirá qualquer possível invasor.

A principal razão para a alta taxa de FDR em baixas frequências fv decorre da evidência limitada fornecida pelos bits observados em apoio a v. Isso é claramente ilustrado pela Figura 8, onde a probabilidade de v ter sido relatado (1) ou não relatado (0) pelo cliente é plotado como uma função de fv. Para strings relativamente raras (aquelas com menos de 10% de frequência), mesmo quando ambos os bits correspondentes a v são definidos no relatório, a probabilidade de v ser relatado é muito menor do que de não sendo informado. Como a probabilidade anterior fv é tão pequena, os relatórios de um único cliente não podem fornecer evidências suficientes em favor de v.

6.1 Cuidados e Correlações

Apesar de ser um avanço no estado da arte, o RAPPOR não é uma panacéia, mas simplesmente uma ferramenta que pode trazer benefícios significativos quando utilizada de forma criteriosa e correta, utilizando parâmetros adequados ao seu contexto de aplicação. Mesmo assim, o RAPPOR deve ser usado apenas como parte de uma estratégia abrangente de proteção de privacidade, que deve incluir retenção limitada de dados e outros processos pragmáticos mencionados na Seção 1.1, e já em uso pelos operadores de Cloud.

Como em trabalhos anteriores sobre privacidade diferencial para registros de banco de dados, o RAPPOR fornece garantias de privacidade para as respostas de clientes individuais. Uma das limitações de nossa abordagem diz respeito ao “vazamento” de informações adicionais quando os respondentes utilizam vários clientes que participam do mesmo evento de cobrança. No mundo real, esse problema é atenuado até certo ponto pela dificuldade intrínseca de vincular diferentes clientes ao mesmo participante. Problemas semelhantes ocorrem quando predicados altamente correlacionados, ou mesmo exatamente iguais, são coletados ao mesmo tempo. Esse problema, no entanto, pode ser tratado principalmente com um design de coleção cuida

Tais correlações inadvertidas podem surgir de muitas maneiras diferentes em aplicativos RAPPOR, em cada caso possivelmente levando à coleta de muitas informações correlacionadas de um único cliente ou usuário e uma correspondente degradação das garantias de privacidade. Obviamente, é mais provável que isso aconteça se os relatórios RAPPOR forem coletados, de cada cliente, em muitas propriedades de clientes diferentes. No entanto, pode

também acontecem de maneiras mais sutis. Por exemplo, o número de coortes usado no desenho da coleção deve ser cuidadosamente selecionado e alterado ao longo do tempo, para evitar implicações de privacidade; caso contrário, as coortes podem ser tão pequenas que facilitam o rastreamento dos clientes, ou os clientes podem relatar como parte de diferentes coortes ao longo do tempo, o que reduzirá sua privacidade.
As respostas RAPPOR podem até afetar o anonimato do cliente, quando são coletadas em valores de cliente imutáveis que são os mesmos em todos os clientes: se as respostas contiverem muitos bits (por exemplo, os filtros Bloom são muito grandes), isso pode facilitar o rastreamento de clientes, pois o bits das respostas aleatórias permanentes são correlacionados. Algumas dessas preocupações podem não se aplicar na prática (por exemplo, rastrear respostas pode ser inviável devido à criptografia), mas todas devem ser consideradas no design da coleção RAPPOR.

Em particular, a proteção longitudinal da privacidade garantida pela resposta aleatória permanente assume que o valor do cliente não muda com o tempo. É apenas ligeiramente violado se o valor mudar muito lentamente. No caso de um fluxo de valores correlacionados e em rápida mudança de um único usuário, medidas adicionais devem ser tomadas para garantir a privacidade longitudinal.
A maneira prática de implementar isso seria fazer um orçamento ÿ ao longo do tempo, gastando uma pequena parcela em cada relatório. No algoritmo RAPPOR, isso seria equivalente a deixar q se aproximar cada vez mais de p a cada evento de coleta.

Como a privacidade diferencial lida com o pior cenário, a incerteza introduzida pelo filtro de Bloom não desempenha nenhum papel no cálculo de seus limites. Dependendo do sorteio aleatório, pode ou não haver várias strings candidatas mapeadas para os mesmos h bits no filtro Bloom. Para a análise de privacidade de caso médio, no entanto, o filtro de Bloom fornece proteção de privacidade adicional (uma espécie de k-anonimato) devido à dificuldade em inferir de forma confiável o valor v de um cliente a partir de sua representação de filtro de Bloom B [4].

7 Trabalho Relacionado

A coleta de dados de clientes de forma a preservar sua privacidade e ao mesmo tempo permitir inferências agregadas significativas é uma área ativa de pesquisa tanto na academia quanto
indústria. Nosso trabalho se encaixa em uma categoria de problemas recentemente explorados em que um agregador não confiável deseja aprender os “rebatedores pesados” nos dados dos clientes - ou executar certos tipos de algoritmos de aprendizado nos dados agregados - enquanto garante a privacidade de cada cliente contribuinte , e, em alguns casos, restringindo a quantidade de comunicação do cliente ao agregador não confiável [7, 16, 18, 20]. Nossa contribuição é sugerir uma alternativa às já exploradas que seja intuitiva, de fácil implementação e potencialmente mais adequada a determinados problemas de aprendizagem e fornecer uma metodologia de decodificação estatística detalhada para nossa abordagem, bem como dados experimentais em seu desempenho. Além disso, além de garantir a privacidade diferenciada, damos passos algorítmicos explícitos para a proteção contra vinculação entre relatórios do mesmo usuário.

É natural perguntar por que construímos nossos mecanismos sobre a resposta aleatória, em vez de dois primitivos mais comumente usados para obter privacidade diferencial: os mecanismos Laplace e Exponencial [12, 21]. O mecanismo de Laplace não é adequado porque os valores relatados pelo cliente podem ser categóricos, em vez de numéricos, caso em que a adição direta de ruído não faz sentido semântico. O ex

mecanismo potencial não é aplicável devido ao nosso desejo de implementar o sistema num modelo local, onde a privacidade é assegurada por cada cliente individualmente sem necessidade de um terceiro de confiança. Nesse caso, o cliente não possui informações suficientes sobre o espaço de dados para fazer a amostragem tendenciosa necessária exigida pelo mecanismo Exponencial. Finalmente, a resposta aleatória tem o benefício adicional de ser relativamente fácil de explicar ao usuário final, tornando o raciocínio sobre o algoritmo usado para garantir a privacidade mais acessível do que outros mecanismos que implementam a privacidade diferencial.

O uso de várias técnicas de redução de dimensionalidade para melhorar as propriedades de privacidade dos algoritmos, mantendo a utilidade, também é bastante comum [1, 17, 20, 22]. Embora nossa confiança nos filtros Bloom seja motivada pelo desejo de obter uma representação compacta dos dados para reduzir os custos potenciais de transmissão de cada cliente e pelo desejo de usar tecnologias já amplamente adotadas na prática [6], o trabalho relacionado neste espaço no que diz respeito à privacidade pode ser uma fonte de otimismo também [4]. É concebível que, por meio de uma seleção cuidadosa de funções de hash ou da escolha de outros parâmetros do filtro de Bloom, seja possível aumentar ainda mais as defesas de privacidade contra invasores, embora não tenhamos explorado essa direção com muitos detalhes.

A obra mais parecida com a nossa é de Mishra e Sandler [24]. Uma das principais contribuições adicionais de nosso trabalho é a etapa de decodificação mais extensa, que fornece análises experimentais e estatísticas dos dados coletados para consultas mais complexas do que as consideradas em seu trabalho. A segunda distinção é o uso da segunda etapa de randomização, a resposta aleatória instantânea, para dificultar a tarefa de vincular relatórios de um único usuário, juntamente com modelos mais detalhados das capacidades dos invasores.

O desafio de eliminar a necessidade de um agregador confiável também foi abordado com soluções distribuídas, que depositam confiança em outros clientes [11]. Desta forma, diferentes protocolos privados podem ser implementados, sobre dados de usuários distribuídos, confiando em proxies ou agregadores honestos, mas curiosos, vinculados a certos compromissos [2, 8].

Diversas linhas de trabalho visam abordar a questão da coleta de dados longitudinais com privacidade. Alguns trabalhos recentes consideram cenários em que muitas consultas predicadas são feitas no mesmo conjunto de dados e usam uma abordagem que, em vez de fornecer randomização para cada resposta separadamente, tenta reconstruir a resposta para algumas consultas com base nas respostas dadas anteriormente a outras. consultas [25].
A ideia de alto nível do RAPPOR tem alguma semelhança com esta técnica – a resposta aleatória instantânea está reutilizando o resultado da etapa de resposta aleatória permanente. No entanto, o objetivo geral é diferente - em vez de responder a um número diversificado de consultas, o RAPPOR coleta relatórios para a mesma consulta sobre dados que podem mudar com o tempo. Embora não opere sob o mesmo modelo local do RAPPOR, trabalhos recentes sobre streaming pan-privado e privacidade sob observação contínua introduzem ideias adicionais relevantes para a coleta de dados longitudinais com privacidade [13, 14].

8 Resumo

RAPPOR é uma plataforma flexível, matematicamente rigorosa e prática para coleta de dados anônimos para fins de crowdsourcing de preservação da privacidade de estatísticas populacionais

em dados do lado do cliente. O RAPPOR lida graciosamente com várias coletas de dados do mesmo cliente, fornecendo garantias de privacidade diferenciais longitudinais bem definidas. Parâmetros altamente ajustáveis permitem equilibrar risco versus utilidade ao longo do tempo, dependendo das necessidades e avaliação da probabilidade de diferentes modelos de ataque. O RAPPOR é puramente uma solução de privacidade baseada no cliente. Ele elimina a necessidade de um servidor confiável de terceiros e coloca o controle sobre os dados do cliente de volta em suas próprias mãos.

Agradecimentos Os autores

gostariam de agradecer aos muitos colegas do Google e sua equipe do Chrome que ajudaram neste trabalho, com agradecimentos especiais a Steve Holte e Moti Yung.
Agradecemos também aos revisores do CCS e a muitos outros que forneceram feedback perspicaz sobre as ideias e este artigo, em particular, Frank McSherry, Arvind Narayanan, Elaine Shi e Adam D. Smith.

9 Referências

[1] CC Aggarwal e PS Yu. Na preservação da privacidade de texto e dados binários esparsos com esboços. Em Proceedings of the 2007 SIAM International Conference on Data Mining (SDM), páginas 57–67, 2007.

[2] IE Akkus, R. Chen, M. Hardt, P. Francis e J. Gehrke. Análise da web sem rastreamento. Em Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS), páginas 687–698, 2012.

[3] Y. Benjamini e Y. Hochberg. Controlando a taxa de descoberta falsa: uma abordagem prática e poderosa para vários testes. Journal of the Royal Statistical Society Series B (Methodological), 57(1):289–300, 1995.

[4] G. Bianchi, L. Bracciale e P. Loreti. 'Melhor que Nada de privacidade com os filtros Bloom: até que ponto? Em Proceedings of the 2012 International Conference on Privacy in Statistical Databases (PSD), páginas 348–363, 2012.

[5] BH Bloom. Compensações de espaço/tempo na codificação de hash com erros permitidos. Comum. ACM, 13(7):422–426, julho de 1970.

[6] AZ Broder e M. Mitzenmacher. Rede aplicações dos filtros Bloom: Uma Pesquisa. Internet Mathematics, 1(4):485–509, 2003.

[7] T.-HH Chan, M. Li, E. Shi e W. Xu. Monitoramento contínuo diferencialmente privado de rebatedores pesados de fluxos distribuídos. Em Proceedings of the 12th International Conference on Privacy Enhancing Technologies (PETS), páginas 140–159, 2012.

[8] R. Chen, A. Reznichenko, P. Francis e J. Gehrke. Para consultas estatísticas sobre dados de usuários privados distribuídos. Em Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (NSDI), páginas 169–182, 2012.

[9] Chromium.org. Documentos de Projeto: RAPPOR (Respostas Ordinais de Preservação de Privacidade Agregável Randomizada). http://www.chromium.org/developers/design-documents/rappor .

[10] C. Dwork. Uma base sólida para análise de dados privados. Comum. ACM, 54(1):86–95, janeiro de 2011.

[11] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov e M. Naor. Nossos dados, nós mesmos: Privacidade via geração de ruído distribuído. Em Proceedings of 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), páginas 486–503, 2006.

[12] C. Dwork, F. McSherry, K. Nissim e A. Smith. Calibrando o ruído para a sensibilidade na análise de dados privados. Em Proceedings of the 3rd Theory of Cryptography Conference (TCC), páginas 265–284, 2006.

[13] C. Dwork, M. Naor, T. Pitassi e GN Rothblum. Privacidade diferencial sob observação contínua. Em Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC), páginas 715–724, 2010.

[14] C. Dwork, M. Naor, T. Pitassi, GN Rothblum e S. Yekhanin. Algoritmos de streaming pan-privados. In Proceedings of The 1st Symposium on Innovations in Computer Science (ICS), páginas 66–80, 2010.

[15] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, BC Pierce e A. Roth. Privacidade diferencial: um método econômico para escolher o epsilon. Nos Anais do 27º IEEE Computer Security Foundations Symposium (CSF), 2014.

[16] J. Hsu, S. Khanna e A. Roth. Rebatedores pesados privados distribuídos. Em Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming (ICALP) - Volume Part I, pages 461–472, 2012.

[17] K. Kenthapadi, A. Korolova, I. Mironov e N. Mishra. Privacidade por meio da transformada de Johnson-Lindenstrauss. Journal of Privacy and Confidentiality, 5(1):39–71, 2013.

[18] D. Keren, G. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman e A. Deligiannakis. Monitorando fluxos de dados heterogêneos e distribuídos: o surgimento de zonas seguras. Em Proceedings of the 1st International Conference on Applied Algorithms (ICAA), páginas 17–28, 2014.

[19] D. Kifer e A. Machanavajjhala. Não há almoço grátis em privacidade de dados. Em Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), páginas 193–204, 2011.

[20] B. Liu, Y. Jiang, F. Sha e R. Govindan. Aprendizagem colaborativa de preservação de privacidade habilitada para nuvem para detecção móvel. Em Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys), páginas 57–70, 2012.

[21] F. McSherry e K. Talwar. Projeto de mecanismo via privacidade diferencial. Em Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS), páginas 94–103, 2007.

[22] DJ Mir, S. Muthukrishnan, A. Nikolov e RN
Wright. Algoritmos pan-privados via estatísticas em esboços.
In Proceedings of Symposium on Principles of Database Systems
(PODS), páginas 37–48, 2011.

[23] I. Mironov. Sobre o significado dos bits menos significativos para
privacidade diferencial. Em Proceedings of ACM Conference on
Computer and Communications Security (CCS), páginas 650–661,
2012.

[24] N. Mishra e M. Sandler. Privacidade através de esboços pseudo-
aleatórios. Em Proceedings of Symposium on Principles of
Database Systems (PODS), páginas 143–152, 2006.

[25] A. Roth e T. Roughgarden. Privacidade interativa através do
mecanismo mediano. Em Proceedings of the 42nd ACM Symposium
on Theory of Computing (STOC), páginas 765–774, 2010.

[26] R. Tibshirani. Encolhimento de regressão e seleção via
o laço. Journal of the Royal Statistical Society, Série B, 58:267–288,
1994.

[27] SL Warner. Resposta aleatória: uma pesquisa
técnica para eliminar o viés de resposta evasiva. Jornal da
Associação Estatística Americana, 60(309):pp. 63–69, 1965.

[28] Wikipédia. Resposta randomizada.
http://en.wikipedia.org/wiki/Randomized_response.

APÊNDICE

Observação 1

Para a, b ÿ 0 e c, d > 0: $\frac{a+b}{c+d}$ ÿ máx(a $\frac{}{c}$, $\frac{}{bd}$).

Prova. Assuma que wlog é $\frac{}{c}$ ÿ $\frac{}{bd}$, e suponha que o estado falso, ou seja, . Então $\frac{a+b}{c+d}$ < $\frac{}{c}$ ou
bc > ad, uma contradição com a suposição de que $\frac{}{c}$ ÿ $\frac{}{bd}$. □

Derivando Limites de Aprendizagem

Consideramos um algoritmo Basic One-time RAPPOR para estabelecer limites teóricos sobre o que pode ser aprendido usando uma configuração de parâmetro particular e uma série de relatórios coletados N. Como o Basic One-time RAPPOR é mais eficiente (sem perdas) do que o RAPPOR original, o seguinte fornece um limite superior estrito para todas as modificações do RAPPOR.

A decodificação para o Basic RAPPOR é bastante simples. Aqui, assumimos que f = 0. O número esperado que o bit i é definido em um conjunto de relatórios, Ci, é dado por

E(Ci) = qTi + p(N ÿ Ti), onde

Ti é o número de vezes que o bit i foi realmente definido (era o bit de sinal). Isso fornece imediatamente o estimador

$$\hat{T}i = \frac{Ci \ddot{y} pN}{q \ddot{y} p}.$$

Pode-se mostrar que a variância do nosso estimador sob a suposição de que Ti = 0 é dada por

$$Var(\hat{T}i) = \frac{p(1 \ddot{y} p)N}{(q \ddot{y} p)^2}.$$

Determinar se Ti é maior que 0 se resume ao teste de hipótese estatística com H0 : Ti = 0 vs H1 : Ti > 0. Sob a hipótese nula H0 e deixando p = 0,5, o desvio padrão de Ti é igual

$$sd(\hat{T}i) = \frac{2q \ddot{y} 1}{\ddot{y} N}.$$

Rejeitamos H0 quando

$$\hat{T}i > Q \times sd(\hat{T}i)$$
$$> \frac{Q \ddot{y} \sqrt{N}}{2q \ddot{y} 1},$$

onde Q é o valor crítico da distribuição normal padrão Q = ÿÿ1 (1ÿ $\frac{0,05}{M}$) (ÿÿ1 é o inverso da CDF normal padrão). Aqui, M é igual ao comprimento da matriz de bits. Dividindo por M, a correção de Bonferroni, é necessário ajustar para testes múltiplos para evitar um grande número de achados falsos positivos.

Seja x o maior número de bits para os quais esta condição é verdadeira (ou seja, rejeitando a hipótese nula). x é maximizado quando x de M itens tem uma distribuição uniforme e uma massa de probabilidade combinada de quase 1. Os outros M ÿ x bits têm probabilidade essencialmente 0. Neste caso, cada bit diferente de zero terá frequência 1/x e sua contagem esperada será E($\hat{T}$i) = N/x ÿi.

Assim nós exigimos

$$\frac{N}{x} > \frac{Q \ddot{y} \sqrt{N}}{2q \ddot{y} 1},$$

onde resolver para x dá

$$\ddot{y} \frac{(2q \ddot{y} 1)\ddot{y} N}{Q \sqrt{x}}.$$