FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

SECURE MULTIPARTY COMPUTATION

Segurança e Privacidade

Mestrado de Engenharia e Ciência de Dados
2023/2024

Trabalho realizado por:
Carlos Matos, 2020245868
Mariana Magueijo, 2020246886

Índice

Introdução

Este trabalho foi desenvolvido no âmbito da cadeira de Segurança e Privacidade e tem como objetivo explorar os conceitos de *Secure Multiparty Computation.* Para aplicar este método, usamos os protocolos de *Private Set Intersection* (PSI).

Existem vários protocolos, mas apenas abordamos o *Naive hashing*, *Diffie-Hellman-based* e o *OT-based*, sendo estes explicados como funcionam na teoria, e quais os resultados obtidos na prática ao aplicá-los.

Caracterização do dataset

O *dataset* escolhido para a realização deste trabalho consiste num conjunto de filmes com diferentes classificações dadas pelos utilizadores, em plataformas como a Netflix e a HBO. Existem 6 colunas que constituem este *dataset*: nome do filme, realizador, duração, género, data de lançamento, rating atribuído a cada música. Os dados encontram-se organizados da seguinte maneira:

	Filme	Realizador	Ano lancamento	Rating	Duracao	Genero
0	Dick Johnson Is Dead	Kirsten Johnson	2020	4	90 min	Documentaries
1	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	2021	1	91 min	Children & Family Movies
2	Sankofa	Haile Gerima	1993	3	125 min	Dramas
3	The Starling	Theodore Melfi	2021	1	104 min	Comedies
4	Je Suis Karl	Christian Schwochow	2021	4	127 min	Dramas
...	...	...	...	...	...	...
6126	Zinzana	Majid Al Ansari	2015	5	96 min	Dramas
6127	Zodiac	David Fincher	2007	4	158 min	Cult Movies
6128	Zombieland	Ruben Fleischer	2009	3	88 min	Comedies
6129	Zoom	Peter Hewitt	2006	5	88 min	Children & Family Movies
6130	Zubaan	Mozez Singh	2015	1	111 min	Dramas

Este *dataset* continha mais do que as colunas apresentadas, no entanto foram aproveitados apenas os dados necessários para este trabalho.
Como objetivo principal pretende-se que cada plataforma (Netflix e HBO) consiga trocar entre si os ratings de um utilizador, de forma privada, com o intuito de tornar possível obter a interseção dos dois conjuntos de dados. Através desta interseção, cada plataforma terá conhecimento dos filmes que o utilizador classificou e, com base nisso, poderá efetuar sistemas de recomendação de filmes.

Tratamento de colunas

O *dataset* original continha 12 colunas, mas, como dito anteriormente, este possuía informação irrelevante para o objetivo estabelecido do trabalho.

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
...	...	...	...	...	...	...	...	...	...	...	...	...

Dito isto, foram feitas modificações no mesmo de modo a uma melhor adaptação ao trabalho, como adição, modificação ou remoção de colunas.

Primeiramente, alteramos o nome dos 6 atributos que iríamos utilizar para o nosso *dataset* final para “Filme”, “Realizador”, “Ano lancamento”, “Rating”, “Duracao”, “Genero”.

Na coluna do “type” começamos por remover todas as linhas que tinham a denominação de “TV Show” de modo a ficarmos só com os filmes. A coluna de “Genero”, poderia apresentar mais do que um valor, por exemplo “International TV Shows, TV Dramas, TV Mysteries”. Nestes casos, decidimos simplificar de modo que apenas o primeiro valor da lista dada, no caso do exemplo dado, “International TV Shows”. Por fim, uma vez que o rating apresentava valores que não iam de encontro ao que pretendíamos, substituímos estes valores por valores *random* de 1 a 5.

Por fim, eliminamos todas as outras colunas que não consideramos serem relevantes para este trabalho.

Atributos

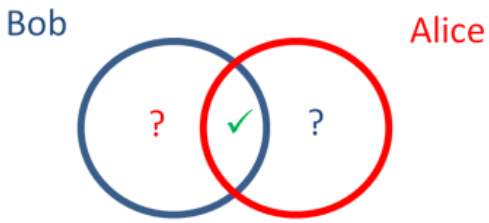
Após uma breve apresentação do *dataset* é necessário explicitar melhor o significado dos mesmos.

- Nome do filme;
- Realizador: se um indivíduo gostar de um filme de um determinado realizador, existe uma maior probabilidade de voltar a ver outros filmes do mesmo;
- Duração: duração, em minutos, do filme;
- Género: indica o tipo de filme. Poderá permitir recomendar aos utilizadores géneros que eles tenham visto anteriormente e que, à partida, sejam do seu agrado. É composto por diversas categorias;
- Data lançamento: ano em que o filme foi lançado;
- Rating atribuído: classificação que o utilizador atribui ao filme, sendo o fator principal a considerar durante a criação de modelos de recomendação. Poderá apresentar valores de 1 a 5.

Estes serão, assim, os atributos que representam os dados em questão e os quais se pretende trocar/intersetar de forma segura e privada, utilizando os protocolos PSI.

Aplicação dos protocolos PSI

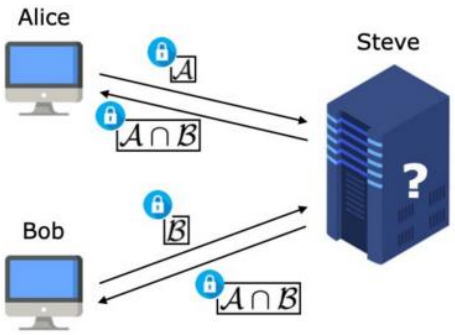
O Private Set Intersection (PSI) é um protocolo que permite que duas partes colaborem e descubram a interseção de seus conjuntos de dados sem revelar informações específicas sobre os elementos individuais desses conjuntos. É útil em situações em que duas partes têm conjuntos de dados privados e querem descobrir se existem elementos em comum entre eles, mas não desejam divulgar os detalhes específicos desses elementos.



- **Naive hashing**: considerando apenas 2 entidades (A e B), ambas chegam a um consenso de usar a função de hash H(x). Posto isto, B envia os seus dados encriptados com a função H(x), A compara os seus hashes com os enviados por B e reenvia a interseção entre estes, mais especificamente os que apresentam valores idênticos.

- **Diffie-Hellman-based**: pode ser utilizado para estabelecer uma chave secreta entre duas partes, que posteriormente é usada para proteger a troca de informações durante o processo de interseção privada de conjuntos. Isso ajuda a garantir a confidencialidade dos dados durante a execução do protocolo PSI. Este protocolo assume claramente um alto grau de confiança entre A e B: A pode falsificar uma correspondência com B enviando de volta a B a mensagem que recebeu; B (ou A) também pode descobrir se tem uma correspondência sem revelá-la a A, enviando lixo na última etapa do protocolo.

- **Server-aided**: tipo de protocolo em que um servidor desempenha um papel ativo na realização da computação ou execução de operações criptográficas em conjunto com os participantes. A e B confiam no servidor, sendo que este não deve guardar informação de A e B. A única informação dada ao servidor é o número de itens que A e B originalmente têm e o tamanho da sua interseção (A interseção B).



- **OT-based**: imaginando que A pretende saber uma das 4 mensagens de B. A gera um par de mensagens público-privadas e 3 chaves públicas. Estas chaves públicas são enviadas para B, encriptando as suas mensagens com as mesmas. B envia as mensagens encriptadas para A. Como A apenas possui uma chave privada que corresponde a uma das chaves públicas enviadas, logo apenas poderá desencriptar uma mensagem e aprender a mesma. Deste modo, como B desconhece a chave privada de A não saberá qual foi a mensagem que esta conseguiu desencriptar.

Divisão do dataset

De forma a ser possível aplicar os protocolos PSI e analisar os resultados, foi necessário criar dois *datasets*, um para a Netflix e outro para a HBO. Ambos os *datasets* devem ter um conjunto de dados semelhantes para tornar possível a sua interseção. Assim sendo, utilizamos os dados de *dataset* original e preenchemos o resto das linhas com dados *random*, criando assim dois *datasets* cada uma com 50000 linhas.

```python
def new_dataset():
    df_aux1 = None
    df_aux2 = None
    count = 1
    for i in range(2):
        array_filme = []
        array_realizador = []
        array_duracao = []
        array_genero = []
        array_lancamento = []
        array_rating = []
        array_filme.extend(df["Filme"])
        array_realizador.extend(df["Realizador"])
        array_duracao.extend(df["Duracao"])
        array_genero.extend(df["Genero"])
        array_lancamento.extend(df["Ano lancamento"])
        array_rating.extend(df["Rating"])
```

```python
for j in range(n_lines):
    array_filme.append("filme" + str(count))
    array_realizador.append(random.choice(nome_realizador))
    array_duracao.append(str(random.randint(30, 240)) + " min")
    array_genero.append(random.choice(tipo_genero))
    array_lancamento.append(random.randint(1960, 2023))
    array_rating.append(random.randint(1, 5))
    count += 1

d = {"Filme": array_filme, "Realizador": array_realizador,
     "Ano lançamento": array_lancamento,
     "Rating": array_rating, "Duração": array_duracao,
     "Género": array_genero}

df_d = pd.DataFrame(d)

if i == 0:
    df_aux1 = df_d
    df_d.to_csv("dataset/PartA/dataset" + str(tam)+ ".csv", index=False)
    shuffle_df("dataset/PartA/dataset" + str(tam)+ ".csv")
elif i == 1:
    df_aux2 = df_d
    df_d.to_csv("dataset/PartB/dataset" + str(tam)+ ".csv", index=False)
    shuffle_df("dataset/PartB/dataset" + str(tam)+ ".csv")
```

Por fim é aplicado um *shuffle* aos filmes de cada *dataset*, de modo a evitar que os dados em comum estejam todos seguidos.

Como é solicitado 5 *subdatasets*, cada uma com um tamanho diferente, dividimos o nosso *dataset* nos seguintes tamanhos: 10000, 20000, 30000, 40000 e o original 50000. A função usada para fazer isto foi a seguinte:

```python
def subdivide_dataset(path1):
    path = path1 + "dataset50000.csv"
    file = pd.read_csv(path)
    df = pd.DataFrame(file)
    sd = 10000
    for i in range(5):
        df2 = pd.DataFrame(df.head(sd))
        df2.to_csv(path1 + "dataset" + str(sd) + ".csv", index=False)
        sd += 10000

subdivide_dataset("dataset/PartA/")
subdivide_dataset("dataset/PartB/")
```

Acreditamos que esta divisão seria a ideal para este trabalho e para a aplicação dos protocolos, pois o tamanho dos *datasets* aumenta uniformemente, assim como as interseções de dados entre *datasets* com o mesmo tamanho.

Naive Hashing

	10000	20000	30000	40000	50000
Tempo	0.141	0.236	0.343	0.465	0.569
Pacotes	39	48	58	66	78
Bytes	182657	362930	544052	724448	905222

Diffie-Hellman-based

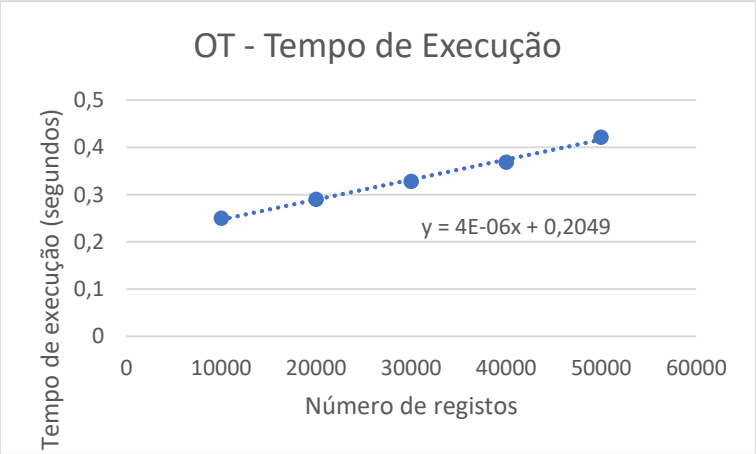
	10000	20000	30000	40000	50000
Tempo	5.597	10.027	15.027	20.302	25.262
Pacotes	83	125	160	189	231
Bytes	1065677	2128782	3191224	4092162	5118812

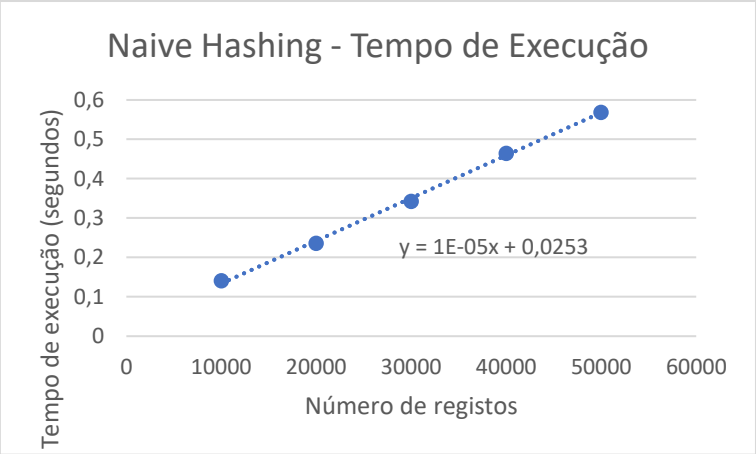
OT-based

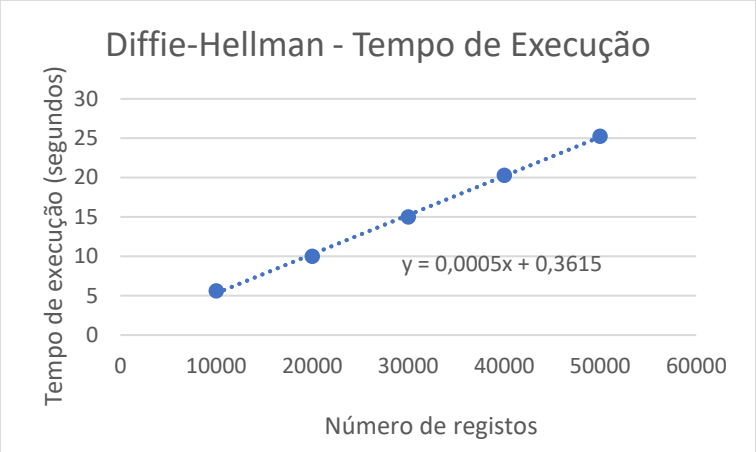
	10000	20000	30000	40000	50000
Tempo	0.25	0.29	0.328	0.369	0.422
Pacotes	96	137	170	196	229
Bytes	1081928	2108298	3167172	4061978	5151201

Tempo de execução

O tempo de execução é importante para avaliar a performance dos protocolos PSI, pois se o tempo de execução for excessivamente longo, o protocolo pode não ser prático para cenários do mundo real, quando trabalhando com elevada quantidade de dados. Em comparação com isto, é importante ver a escalabilidade do protocolo à medida que o tamanho dos conjuntos de dados aumenta.







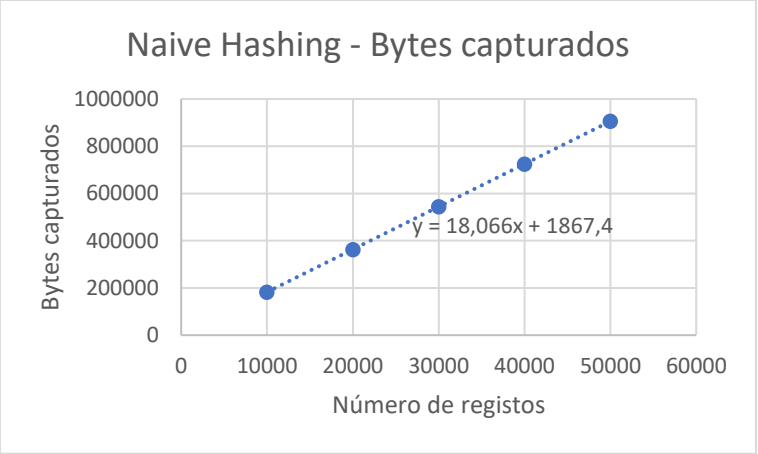
	10000	20000	30000	40000	50000
Naive Hashing	0.141	0.236	0.343	0.465	0.569
Diffie-Hellman	5.597	10.027	15.027	20.302	25.262
OT-based	0.25	0.29	0.328	0.369	0.422

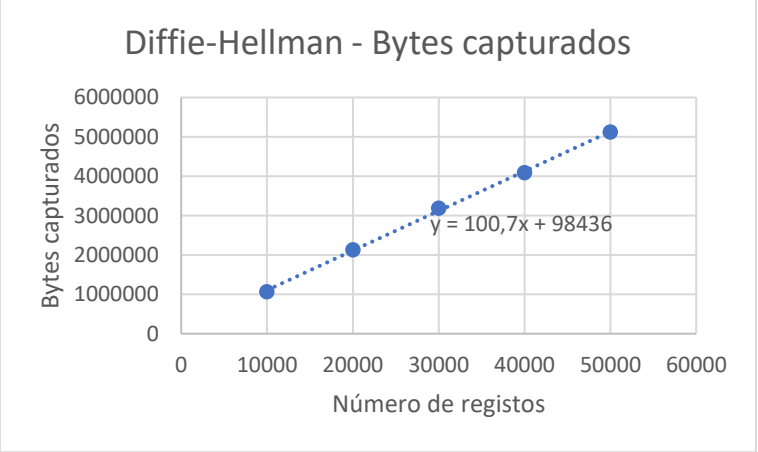
Observando as imagens anteriores, é possível verificar que o tempo de execução evolui de forma linear, à medida que o tamanho dos nossos *datasets* aumenta.

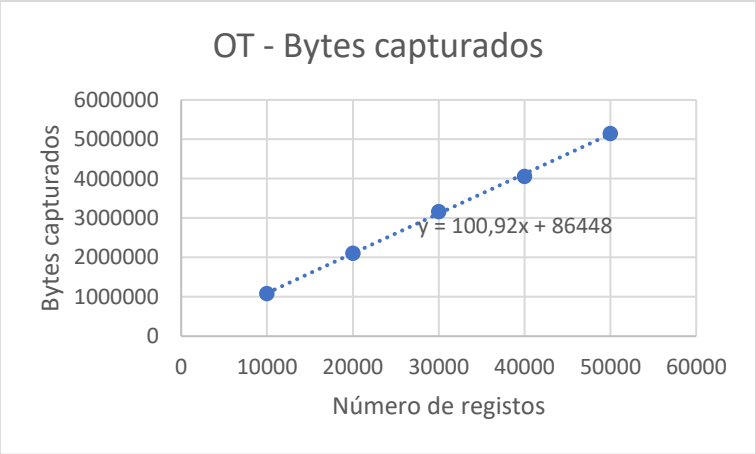
Analisando individualmente é percetível que a abordagem que apresenta melhores resultados é a *Naive Hashing*, devido à sua complexidade comparando com as outras. Apesar de o protocolo *Diffie-Hellman* ter uma troca de pacotes menor do que o protocolo *OT-based*, este acaba por ter um tempo de execução significativamente maior, devido ao tipo de operações que ambas utilizam.

Dados trocados entre entidades

A quantidade de dados trocados entre as entidades pode impactar diretamente o desempenho do protocolo PSI. Protocolos que exigem a transmissão de grandes volumes de dados podem enfrentar problemas de latência, consumo excessivo de largura de banda e, consequentemente, maior tempo de execução. Como, mais uma vez a escalabilidade é um ponto importante em qualquer projeto, é necessário garantir que protocolos eficientes sejam capazes de lidar com grandes volumes de dados sem comprometer significativamente o desempenho. A análise dos dados trocados em diferentes tamanhos de conjuntos ajuda a entender a capacidade do protocolo de escalar e maneira eficiente.







	10000	20000	30000	40000	50000
Naive Hashing	182657	362930	544052	724448	905222
Diffie-Hellman	1065677	2128782	3191224	4092162	5118812
OT-based	1081928	2108298	3167172	4061978	5151201

Mais uma vez, é visível uma evolução linear presente nos gráficos, assim como, para o mesmo tamanho do *dataset*, o *Diffie-Hellman* e *OT-based* são parecidos

Após a análise dos gráficos é possível verificar que o declive do *Diffie-Hellman* e *OT-based* é, aproximadamente, 5 vezes superior ao do *Naive Hashing*. Tendo em conta esta análise podemos concluir que a quantidade de bytes necessária para os diferentes tamanhos dos nossos *datasets* aumenta significativamente mais rápido nestas abordagens, devido à sua complexidade.

Registos intersetados entre entidades

Para obter estes resultados, a quantidade de dados intersetados é crucial, pois é desta forma que se consegue perceber a fiabilidade dos dados. Também é de referir que cada protocolo usa processos diferentes, que influenciam os resultados de diferentes formas. Na tabela apresentada a baixo encontram-se os dados existentes em comum entre cada ficheiro, não havendo diferença de valores para diferentes protocolos:

Interseção	10000	20000	30000	40000	50000
Naive Hashing	231	967	2149	3851	6132
Diffie-Hellman	231	967	2149	3851	6132
OT-based	231	967	2149	3851	6132

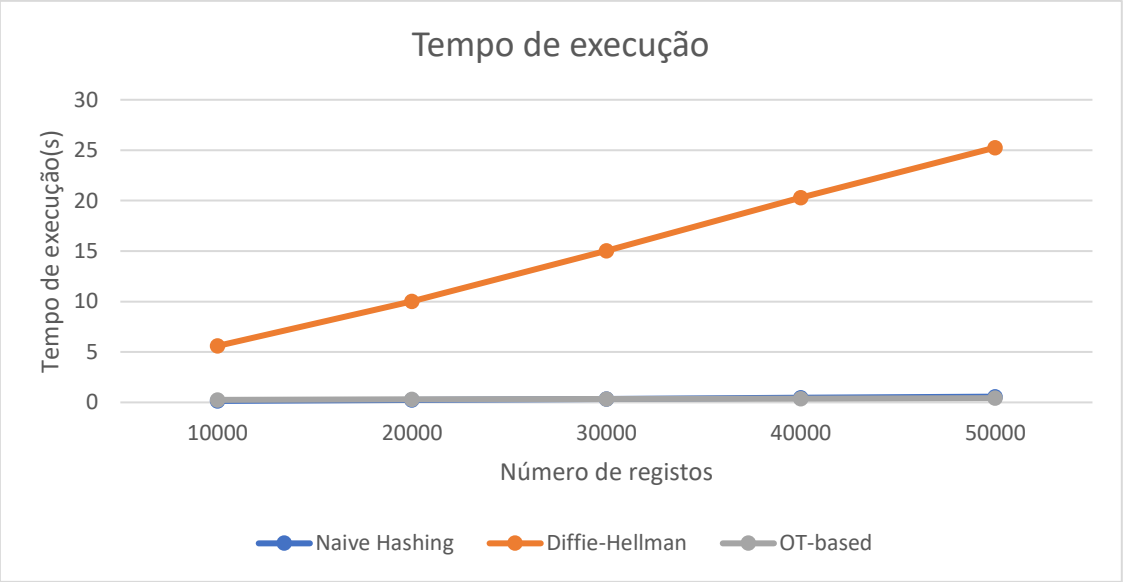
Nível de segurança e privacidade

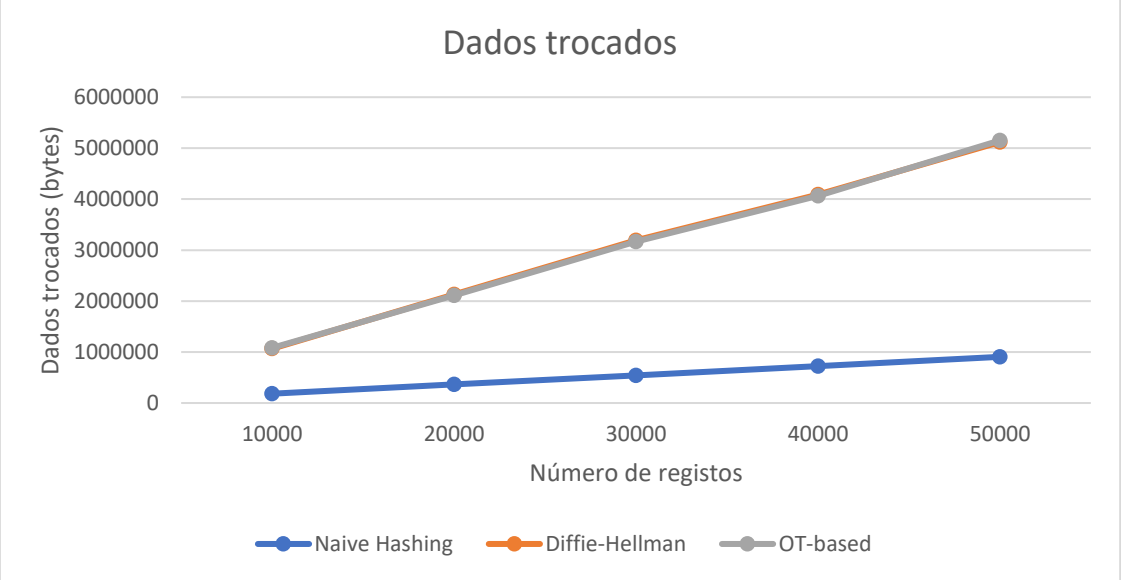
- **Naive hashing**: apresenta limitações na privacidade, já que utiliza apenas *hashes* para interseção de conjuntos. Se um atacante tiver acesso a um conjunto suficientemente grande de *hashes*, pode potencialmente inferir quais são os registos originais, comprometendo assim a privacidade dos dados.
- **Diffie-Hellman**: oferece uma abordagem assimétrica robusta para troca de chaves, garantindo privacidade nas comunicações. Cada parte envolvida não

conhece diretamente a chave da outra parte. No entanto, um ataque comum ao *Diffie-Hellman* é o ataque "*men in the middle*", no qual um adversário intercepta e altera as comunicações entre as partes. Isso pode comprometer a confidencialidade da chave compartilhada e permitir que o atacante decifre ou manipule as mensagens, comprometendo a segurança

- **OT-based**: Assegura privacidade em ambas as partes, pois cada usuário apenas tem conhecimento de alguns registos do outro, impossibilitando o conhecimento total de ambas as partes, revelando apenas a interseção.

Para fazer uma comparação com tudo o que foi dito acima, apresentamos os seguintes gráficos:





Conclusão

Neste trabalho, exploramos a importância e funcionamento dos protocolos de Conjunto Privado de Interseção (PSI) na troca segura de informações entre entidades, desempenhando um papel fundamental na preservação da privacidade e segurança dos dados durante a interseção dos mesmos.

Começamos por definir um *dataset* para mais tarde o testar com diferentes tamanhos (10000, 20000, 30000, 40000, 50000 linhas) com diferentes protocolos. De seguida avaliámos métricas como tempo de execução, quantidade de dados transmitida e quantidade de interseções encontradas.

Referências