POIS Statement

Team - Risk Rangers
Santhoshini Thota
Abhay Patil
Pranjal Thapliyal
Praddyumn Shukla
Ishank

May 7, 2024

1 Literature Survey

The literature review we conducted for the project proposal helped us to find out more about significant advancements and methodologies in the field of data privacy, particularly in the realms of cryptography, blockchain privacy, differential privacy, and anonymous communication networks. These key ideas lay the groundwork for addressing privacy concerns across various domains, including healthcare, finance, and digital communication:

- **Advancements in Cryptocurrency Privacy:** Studies on Monero and Zcash highlight the pursuit of enhanced privacy in blockchain transactions. The use of ring signatures, stealth addresses, and zero-knowledge proofs (zk-SNARKs) showcases innovative approaches to maintaining transaction privacy while addressing scalability and auditability challenges.
- **Innovations in Anonymous Authenticated Encryption:** The exploration of anonymous AE (anAE) and the NonceWrap method underscores the importance of preserving sender anonymity in encrypted communications. This addresses the need for encryption techniques that protect against traffic-flow analysis and other forms of privacy breaches.
- **Differential Privacy for Enhancing Client Confidentiality:** The concept of round differential privacy within the trusted curator model presents a novel approach to safeguarding client data in both traditional and decentralized financial systems. This includes mechanisms to prevent front-running and leakage, emphasizing the importance of privacy-preserving market mechanisms.
- **Protecting Anonymity in Digital Communications:** Research into securing networks like Tor against tagging attacks illustrates the ongoing

challenges in ensuring user anonymity online. Proposals for enhancing encryption schemes indicate the critical need for constant evolution in technologies that protect user identities and data.

- **Blockchain Privacy-Preserving Transactions:** The development of methods for statistical data collection within blockchain systems, without compromising individual privacy, showcases an innovative approach to maintaining data integrity. The use of verifiable local differential privacy and zk-SNARKs points to sophisticated cryptographic solutions for privacy preservation.

2 Limitations of Current Approaches

1. **Masking**
    - **Reversibility:** While masking replaces sensitive data with pseudonyms or other placeholders, this process can sometimes be reversed. Advanced cross-Refer toencing techniques, data linkage with other sources, or even statistical analysis can potentially uncover the original values, especially if the masking technique is simplistic or well-documented.
    - **Security:** If the method for pseudonymization or the pseudonyms themselves are compromised, the protection can be nullified, leading to exposure of sensitive information.
2. **Generalization**
    - **Loss of Detail:** Generalization involves grouping similar but distinct data points into broader categories. This can significantly alter the granularity of the data, which can obscure or alter important nuances and patterns in the dataset. The more aggressive the generalization, the greater the potential distortion of the data's original structure and content.
    - **Utility vs. Privacy Trade-off:** There's often a delicate balance between generalizing data enough to protect privacy while retaining sufficient detail for analysis. Over-generalization can render the data nearly useless for specific analytical purposes.
3. **Suppression**
    - **Data Loss:** Suppression involves completely omitting or blanking out data points to protect sensitive information. This can lead to significant data loss, particularly in datasets where many attributes are sensitive or where the risk of identification is high.
    - **Analytical Impact:** Removing data can reduce the statistical power and representativeness of a dataset, making any conclusions drawn

from the data less reliable or valid. This is particularly problematic in research or settings where comprehensive data is crucial for accurate analysis.

3 Problem Statement

3.1 Problem

Organizations, such as hospitals, generate valuable datasets that could provide significant insights for statistical analysis, machine learning, and pattern recognition. However, there is a critical privacy concern regarding the identity of the patients whose PII and SPII is contained within the datasets. In fear of violating HIPAA regulations, hospitals are often constrained to processing the data themselves. This approach is far from ideal, considering that these hospitals fail to take advantage of state of the art data processing and machine learning tools provided by external entities. This problem also plagues other domains, such as finance, telecommunications, education etc.

This problem is not only intellectually stimulating but also highly relevant in today's data-driven world, where the balance between data utility and privacy is of paramount importance.

4 Architecture

We propose Ashe - A data anonymization tool that takes advantage of state of the art differential privacy algorithms that allows organizations to anonymize datasets before making them public for statistical analysis. Such algorithms are designed to protect individual identities by tweaking the attributes of each record, in such a way that they remain statistically significant. Properties like average, standard deviation etc. are maintained.

4.1 Threat Model

4.1.1 Perfect Anonymization

Perfect anonymization would imply that an adversary would not gain any additional information from the anonymized dataset. However, it is impossible to construct a threat model around perfect anonymization for differential privacy. This is implicit due to the constraint that the anonymized values must be somewhat close to the true values if we want to maintain statistical significance.

4.1.2 Adversary Goal

If the adversary, with full access to the anonymized dataset, can guess the true identity of any of the records, we declare that the adversary succeeds.

4.2 Network Assumptions

4.2.1 Data Transfer Security

We assume that the data transferred over the network, specifically when the organization uploads the true dataset to the Ashe application, is secured using encryption protocols such as HTTPS. In other words, the adversary has no access whatsoever to the true dataset.

4.2.2 Client Server Architecture

Ashe will be designed with a client-server architecture, where the anonymization tool runs on the client side, and interacts with a server which provides differential privacy algorithms, ability to share datasets etc.

4.3 Overview

Refer to Fig 1.

5 Ash Demo

6 Implementation Details

Refer to Figs 2, 3, 4, 5 and 6.

7 Results

The lower the Epsilon values, the higher standard deviation of the resulting values will be. Users can tune this Epsilon values according to sensitivity of the data. Refer to Figs. 7 and 8.

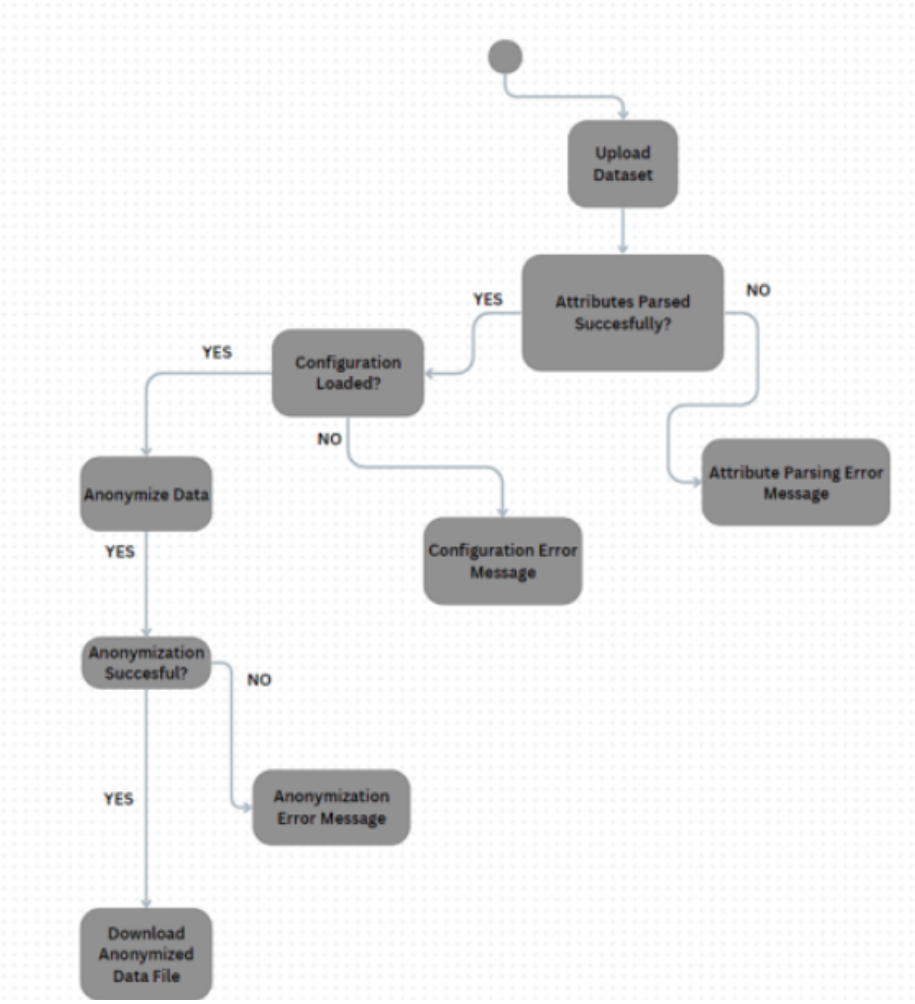

Figure 1: Architecture

```
const handleAnonymize = () => {
    const formData = new FormData();
    formData.append("file", file);

    // Collect epsilon values from the col array
    const epsilonValues = cols.map(column => column.epsilon)
    formData.append("epsilonValues", JSON.stringify(epsilonValues))

    // Collect checked values from the col array
    const checkedValues = cols.map(column => column.checked)
    formData.append("checkedValues", JSON.stringify(checkedValues))

    axios.post('http://localhost:5000/anonymize', formData)
        .then(response => {
            console.log("[ANONYMIZER] Anonymization successful:", response);
```

Figure 2: PyDP - Pass data to app.py

```
@app.route('/anonymize', methods=['POST'])
def anonymize():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})

    file = request.files['file']
```

```
# Apply differential privacy (bounded sum) to the dataset
modified_dataset = private_sum(dataset.copy(), epsilonValues, checkedValues)
```

Figure 3: PyDP - Calls private_sum

```
def private_sum(dataset, epsilonValues, checkedValues):
    if len(dataset.columns) == 0:
        return dataset

    print("Epsilon values: " + str(epsilonValues))
    print("Dataset cols = ", str(dataset.columns))

    for column, epsilon, anonymize in zip(dataset.columns[1:], epsilonValues, checkedValues):
        x = BoundedSum(epsilon=epsilon, delta=0, lower_bound=0, upper_bound=100, dtype="float")
        if not anonymize: continue
        dataset[column] = dataset[column].apply(lambda value: x.quick_result([value]))

    return dataset
```

Figure 4: PyDP – Anonymizes the dataset

```python
# Return the filename, content, and epsilonValues in the response
return jsonify({
    'filename': file.filename,
    'content_anon': csv_content_anon,
    'epsilons': epsilonValues,
    'plot_image': 'average_age_with_std.png'
})
```

Figure 5: PyDP - Return the anonymized dataset

```javascript
axios.post('http://localhost:5000/anonymize', formData)
    .then(response => {
        console.log("[ANONYMIZER] Anonymization successful:", response);

        // Parse the file content using Papa
        Papa.parse(response.data.content_anon, {
            complete: function (results) {
                let headers = results.data[0]
```

Figure 6: PyDP- Return the anonymized dataset

7.1 Mean and standard deviation

- DP conserves aggregate statistics such as sums, counts, and means
- DP doesn't attempt to preserve standard deviation
- Standard deviation varies with epsilon

7.2 Distance Metrics

Refer to Fig. 9.

8 Challenges Faced

1. **Dockerizing PyDP:** Dockerizing React and Flask was straightforward. However, since PyDP was not as standard, it presented challenges. Refer to Fig. 10.
2. **CORS ERROR:** CORS allows our Flask application to take requests from React - localhost:5173. For reasons unknown to us, the traditional method of disabling CORS did not work. Workaround - we used a chrome extension that did this for us. Refer to Fig. 11.

Epsilon – 5

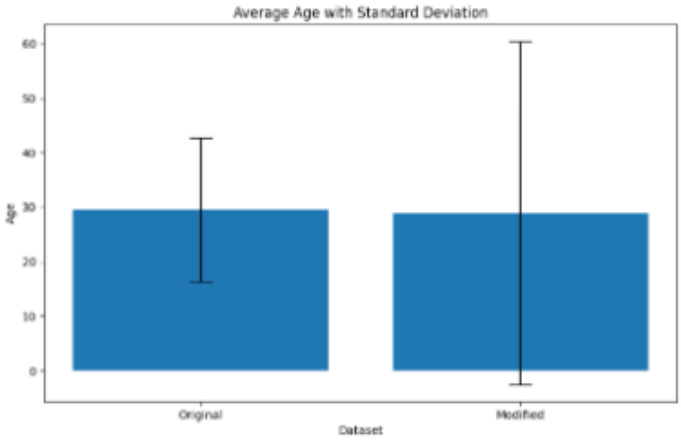

Figure 7: Epsilon - 5

Epsilon – 31

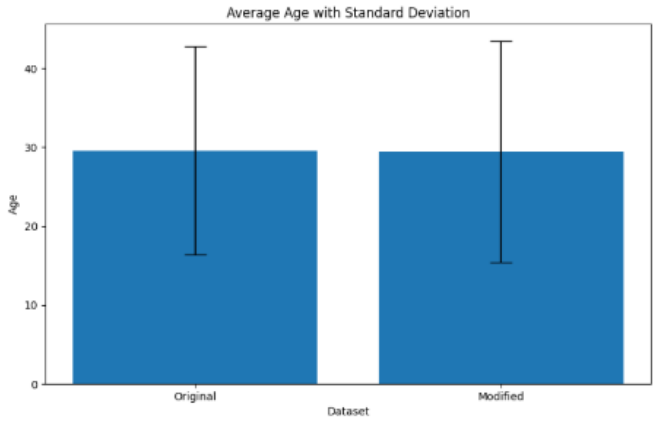

Figure 8: Epsilon - 31

	A	B	C
1	Epsilon	Manhattan	Euclidean
2	10	9.444	0.367
3	20	4.896	0.193
4	30	3.402	0.132
5	40	2.541	0.099
6	50	1.997	0.078
7	60	1.658	0.064
8	70	1.365	0.053
9	80	1.274	0.049
10	90	1.137	0.043
11	100	0.962	0.037

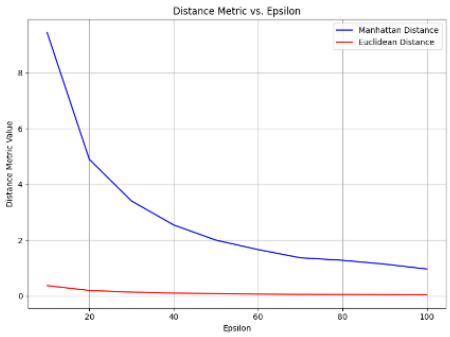

Figure 9: Distance Metrics

Ashe > Dockerfile > ...

```
1   FROM node
2   WORKDIR /app
3
4   COPY package.json .
5   RUN npm i
6
7   COPY . .
8   EXPOSE 5173
9
10  CMD ["npm", "run", "dev"]
```

backend > Dockerfile > ...

```
1   FROM python:3.8-slim
2   WORKDIR /backend
3
4   COPY . /backend
5   RUN pip install --no-cache-dir -r requirements.txt
6   EXPOSE 5000
7
8   ENV NAME World
9   CMD ["python", "app.py"]
10
```

backend > requirements.txt

```
1   Flask==3.0.0
2   python-dp==1.1.4
3   pandas
```

Figure 10: Dockerizing PyDP

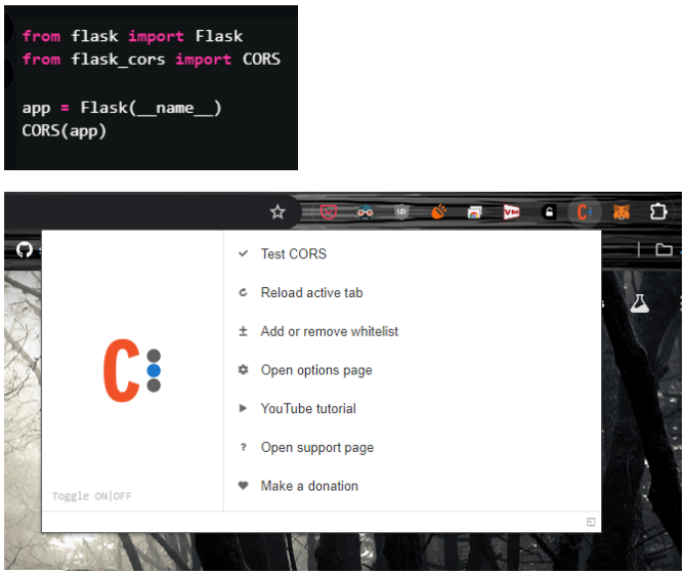

Figure 11: CORS ERROR

Figure 12: Configure Epsilon Values

9 Future Directions

1. Adding Support for other noise mechanisms not just Laplace. Refer to Fig. 12.
2. Support for different Data Types: Extending the application to support different types of data beyond numerical data, such as categorical or text data. This could involve developing specialized mechanisms for adding noise to non-numeric data while preserving privacy and utility.
3. Scalability and Efficiency: Work on enhancing the scalability and efficiency of the anonymization process, especially for large-scale datasets. This could involve optimizations such as parallelization, distributed computing, or leveraging specialized hardware (e.g., GPUs) to speed up computation.
4. Optimizing Noise Parameter: Explore ways to optimize the parameters of the noise distribution (e.g., scale parameter for Laplace noise) to achieve

better utility privacy trade-offs. This could involve research into adaptive mechanisms that adjust noise levels dynamically based on sensitivity of the data or desired level of privacy.