Data Science and Engineering (DSE) Record, Volume 1, issue 2.

# Privacy Preservation for Data Ingression in Data Pipeline

Thanawat Kaewwiroon [1,2] and Juggapong Natwichai [3]

[1] Data Science Program, Chiang Mai University, Chiang Mai, Thailand
[2] Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand
[3] Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand

thanawat_kaewwiroon@cmu.ac.th

**Abstract.** This independent study aims to develop a data pipeline system that is able to preserve the privacy for data ingression in the data pipeline. The system developed using the k-anonymity method with generalization and suppression. The precision of information lossy is concerned and minimized process of Preferred Minimal Generalization Algorithm (MinGen). The system will first calculate the data precision for all possible of the domain generalization hierarchies. Then, process the satisfied k value for the data set for which pattern of generalization level. Finally, the data in database system will be transform to the privacy preservation. The demographic synthesized dataset is generated, and domain categorizes, and its level of quasi-attributes are created which prepared for evaluate the data pipeline system. The indicator of success in this independent study are processing time with the different amount of data records which satisfied the k value. For the results, the more data records spend less processing time for the k value satisfied. Because of the more data records increasing the possible of k records that is similar to others and satisfied the k value. Thus, the Privacy Preservation for Data Ingression in Data Pipeline system which developed in the independent study can process data to satisfied k-anonymity technique which also minimized loss of data precision for demographic data in data pipeline.

**Keywords:** privacy preservation, k-anonymity.

## 1 Introduction

A lot of data was created and generated by human activities including sensors, mobile devices, and business services. The data could contain personal privacy which needed in processing for help and improve our living and use for creating new technology. Many of data was published to public for statistic and public research. Those public data especially health area and demographic data was attacked by hacker using re-identification technic to reveal person identity. Thus, there are many research that are about to prevent the privacy in public data to be attacked such as k-anonymity: a model for protecting privacy from Sweeney [1] which make the data to result the same value

at lease k records. The randomization technic in Randomization-based Privacy-preserving Frameworks [2] who put the extra records randomized into the data but still maintain the overall data statistic as same as before. And, using the differential privacy technic [3] by adding the noise data based on data's statistic and mathematics algorithm to the dataset. These technics can help preventing the personal privacy to be re-identified. The data pipeline is the one data management and big data platform are using widely. To implement the privacy protection in the data pipeline might improve the data management process. This study was conducted to demonstrate designing the data pipeline and implementing the privacy protection with k-anonymity technics.

# 2 Method

## 2.1 Data synthesis

This study uses the synthesized demographic dataset which is health category. The idea of choosing this type is the health dataset is risk to be re-identified the privacy by the attacker.[4] There are 10 columns as shown in Figure 1. There are 4 quasi-identifiers which are state, occupation, age, and weight. Then, construct the value generalization domain categorizes for all 4 quasi-identifiers.

| No. | Name | Type | Description | Primary Key | Nullable | Remark |
|---|---|---|---|---|---|---|
| 1 | id | int | Identifier | Y | N | |
| 2 | first_name | varchar(255) | First name | | N | |
| 3 | last_name | varchar(255) | Last name | | N | |
| 4 | email | varchar(255) | Email | | N | |
| 5 | gender | varchar(1) | Gender | | N | |
| 6 | birthdate | date | Birthdate | | N | |
| 7 | state | varchar(255) | State | | N | |
| 8 | weight | decimal(7,2) | Weight | | N | |
| 9 | occupation | varchar(255) | Occupation | | N | |
| 10 | bloodtype | varchar(2) | Blood type | | N | |

**Fig. 1.** The synthesized demographic schema

## 2.2 Database design

The database schema designing for target table of the data pipeline is same to the sources file structure. The extra column is age which is calculated by birthdate column. And the table for store processed generalization hierarchies which are added columns of each generalization level for all quasi-identifier along all records.

## 2.3 Data pipeline design

Data pipeline was developed in the Talend Open Studio for Big Data application. There are 3 sections of data pipeline as shown in Figure 2, and each section labelled which

are: 1) The JSON configuration files generator which prepared the target database connection information and configuration of quasi-identifier columns such as column name and data type. 2) The data file reading which format is CSV file and the ETL processing, then the output is target database of pipeline. 3) The batch processing of k-anonymity using command line and python script. There are the reasons that uses the batch processing script which are 1) Talend component is complicated to develop and the need specialization of SQL syntax when do query processing the data and need specific database driver library. 2) The processing of k-anonymity in Talend need the resources depends on the combination of level of hierarchy and number of quasi-identifiers. It is possible to reach insufficient of processor resources in Talend which run on desktop computer. Thus, the solution is using the python script do the batch processing for k-anonymity.

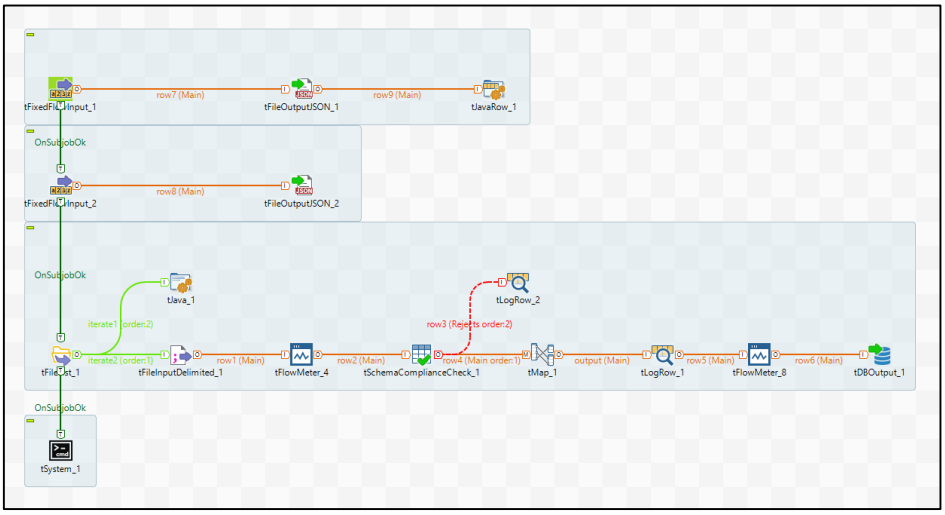


**Fig. 2.** The designed data pipeline in Talend Open Studio for Big Data

### 2.4 K-Anonymity implementation

The k-anonymity processing use python script. The implementation is divided into 3 sections. 1.Configuration reader; the target database connection information and the quasi-identifier columns setting. 2.Check the input; the code processes the input data table and generates all extra column for all levels of generalization hierarchies. 3.Process Minimal Generalization algorithm as show in Figure 3. There is the first validate the k-anonymity satisfied. If not, then generate all possible combination of generalization level with all quasi-identifiers. Next, calculates the Precision metric score and sort descending order. Then, for all possible of sorted score, validate k-anonymity by query data count with group by quasi-identifier until found the valid solution. 4.Display solutions and logs of state processing in console output.

**Input:** Private Table **PT**; quasi-identifier QI = ($A_1$, …, $A_n$),
$k$ constraint; domain generalization hierarchies
$DGH_{Ai}$, where $i$=1,…,$n$, and *preferred()* specifications.
**Output:** MGT, a minimal distortion of PT[QI] with respect to $k$
chosen according to the preference specifications
**Assumes:** |PT| ≥ $k$
**Method:**
1. **if PT**[QI] satisfies $k$-anonymity requirement with respect to $k$ **then do**
    1.1. MGT ← { **PT** } // **PT** is the solution
2. **else do**
    2.1. *allgen* ← {$T_i$ : $T_i$ is a generalization of **PT** over QI}
    2.2. *protected* ← {$T_i$ : $T_i$ ∈ *allgen* ∧ $T_i$ satisfies $k$-anonymity of $k$}
    2.3. MGT ← {$T_i$ : $T_i$ ∈ *protected* ∧ there does not exist
        $T_z$ ∈ *protected* such that *Prec*($T_z$) > *Prec*($T_i$) }
    2.4. MGT ← **preferred**(MGT) // select the preferred solution
3. **return** MGT

**Fig. 3.** Minimal Generalization Algorithm (MinGen)

### 2.5 Evaluation

The evaluation of this study is considered by verify the satisfied k-anonymity of the result data after process choosing the solution of generalization level in each quasi-identifier. The dataset was divided into 3 groups, by selection data 80000,120000 and 160000 records. The selected of k value is reference from the study of health re-identification scenario [5] that the possible of success rate of re-identified personal privacy from dataset is calculated by 1/k. Then, let possible success value is 0.2, then k value is 5 and possible success value is 0.1, then the k value is 10. This study tested in the 2 virtual machines environment. First the PostgreSQL server with 4GB of memory. Second the HDP sandbox with 12GB of memory. Both use the same CPU 4 x Intel Core i7-9750H 2.6GHz as hosts.

## 3 Result

### 3.1 Result from PostgreSQL

There are 1440 precision matric values for this synthesis dataset, and average time spent for calculation is 0.0232 seconds. The result of time spent in k-anonymity process using PostgreSQL shown in Figure 4 (left hand chart). For k =5, the time spent are 36.8, 8.97 and 15.78 seconds for 80000, 120000 and 160000 records. And for k=10 time spent are 37.15, 56.77 and 40.79 seconds respectively. From these results, the k value increasing while time spent trend is decreasing, except the second dataset the result might be an outlier. And the records increase the time spent was not in the same trends. The precision value is increasing when the amount for records is increase. There are 0.49, 0.50 and 0.53 accordingly to amount of records each dataset.

### 3.2 Result from Hive

The precision metric are 1440 values, and time spent to calculate is the same to PostgreSQL result stated in 3.1. The result of time spent in process k-anonymity using Hive shown in Figure 4 (right hand chart). For k =5, the time spent are 389, 261.and 309 mins. for 80000, 120000 and 160000 records. And for k=10 time spent are 402, 237 and 273 mins. respectively. From these results, both k values result trends are considered be the same. The k value increasing while time spent trend is decreasing, like the PostgreSQL database. And the time spent trend is decrease while the number of records is increasing.

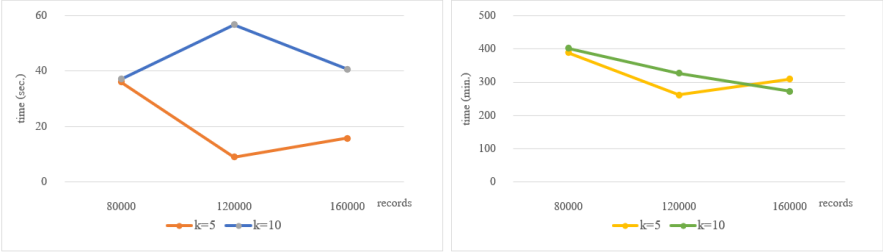


**Fig. 4.** The average time spent results of PostgreSQL (left) and Hive (right)

### 3.3 Result from Data Pipeline

The time spent in processing data in pipeline are 8.8, 12.6 and 16.6 seconds for data set 80000, 120000 and 160000 records. The trends of time spent is linear, while number of records is increasing , time spent also increases. The result is same to both k values. Considering the ratio of time spent processing data in pipeline to the processing k-anonymity, on PostgreSQL ratio is 24.39 for k=5 and ratio is 23.69 for k=10. The ratio in Hive for k=5 is 0.0003767 and for k=10 is 0.0003458. So, the time spent in data pipeline is not changed much with the k value, but it depends on number of data records.

## 4 Discussion

Time spent trend when processing the k-anonymity with Minimal Generalization algorithm is decrease while amount of data records is increasing. That because there is possibility that more records can generate the number of same data value in each group. So, applied the algorithm, it could reach the satisfied k-anonymity properties. The trends occur same to both PostgreSQL and Hive database. Except the k value is 5 with the 120000 records dataset, which time is drop significantly. It may cause by the characteristic of synthesized data that make the processing reach the solution with small amount of time. But the k value is 10 in PostgreSQL, the average time spent is increasing. The possible of cause could be the amount of data records reach the limits of resources of virtual machine. Then, the OS might use swap memory which consume

overhead time in processing data. The time-based result of Hive is minutes unit because the Hive database system is different from PostgreSQL. This study was not comparing the speed of these 2 systems. But to study the data pipeline design and observe the trends of time spent in data process. Furthermore, the k value increase effect to the information loss is increase because the solution needs a greater number of records in same group which have same value. Our result was consistent with the previous study A General Algorithm for k-anonymity on Dynamic Databases[6]. So, when the numbers of records are higher, it is easier to determine the group of records with the same quasi-identifier. Considering the data pipeline processing, the ratio of time spent in processing data in the pipeline to the time spent only processing k-anonymity has stable trends for all 3 sizes of data records. When the number of records increases, the time spent in data pipeline is increased linearly. Thus, the time spent in data pipeline is not much effected to the overall time spend in this study.

## 5 Conclusion

Our study was demonstrated that we can apply privacy protection in data pipeline using k-anonymity method and implement the Minimal Generalization algorithm along with less information loss by Precision Metric score. The implementation of data pipeline was done with desktop application, Talend Open Studio for Big Data, and works with python script as batch process with the target data in both the relational database management system and Hive database in big data platform. The result shows the performance of time spent in processing the k-anonymity which is when amount of data records increasing trends of time spent is decrease. When the k value is increased, the processing time is also increased. The time overhead in the data pipeline grows linearly to the amount of data records. Last, our result concurs with the literature that when the number of data records is increased, the information loss is decreased since the groups of records can be easier to determine.

## References

1. Latanya Sweeney. "k-anonymity: a model for protecting privacy." International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5) 2002; 557–570
2. Zeynep Batmaza, Huseyin Polat. "Randomization-based Privacy-preserving Frameworks for Collaborative Filtering." 20th International Conference on Knowledge Based and Intelligent Information and Engineering. Systems. Procedia Computer Science Vol. 96. 2016 p. 33 – 42. ISSN 1877-0509
3. Nguyen, Hiep & Kim, Jong & Kim, Yoonho. "Differential Privacy in Practice. Journal of Computing Science and Engineering." Journal of Computing Science and Engineering 7(3) September 2013. DOI:10.5626/JCSE.2013.7.3.177
4. El Emam, Khaled, and Fida Kamal Dankar. "Protecting privacy using k-anonymity". Journal of the American Medical Informatics Association : JAMIA vol. 15,5 (2008): 627-37. doi:10.1197/jamia.M2716

5. Ouazzania, Zakariae El and Bakkalia, Hanan El. "A new technique ensuring privacy in big data: K -anonymity without prior value of the threshold k". Procedia Computer Science. Vol. 127. 2018, 52-59. 10.1016/j.procs.2018.01.097
6. Salas J., Torra V. "A General Algorithm for k-anonymity on Dynamic Databases". In: Garcia-Alfaro J., Herrera-Joancomartí J., Livraga G., Rios R. (eds) Data Privacy Management, Cryptocurrencies and Blockchain Technology. DPM 2018, CBT 2018. Lecture Notes in Computer Science, vol 11025 2018. Springer, Cham