

# The Comparison of Pseudonymization and Other Techniques in Data Anonymization

Sidney Sykes  
Data Anonymization  
Department of Computer  
and Information Science  
University of Mississippi  
Email: [swsykes@go.olemiss.edu](mailto:swsykes@go.olemiss.edu)

## Abstract

*Technology has advanced in almost all industries such as education, engineering, government, healthcare, industrial, manufacturing, and real estate in today's climate. Nearly every industry has a website. Most of the time, these websites require our name, residential address, new password, social security numbers, state identification number, and even our passport number (especially for immigrants). The question that arises is, "Are our personal or business identity guaranteed?"*

*I found a dataset that has entries with victims of data breaches. Since the information about the breach are out there, I will anonymize the dataset to attempt to conceal the identity of those entries. They would like for others to respect their privacies. In this project, I will compare anonymization techniques that will evaluate the security and the data protection of the identity of the dataset. I will use at least one method for each technique that produced the result that altered the original data. Options for data anonymization in a dataset are limited; therefore, I am selecting techniques that are database appropriate. I will compare data pseudonymization, data masking, data perturbation, and data generation.*

*Keywords: Database security, pseudonymization, healthcare, encryption, masking, SQL Server, DBM, database management protection*

## 1. Introduction

The healthcare industry depends on patients' data to help meet its need. Patients are easy but forced to trust those services with sensitive information such as social security numbers, addresses, medical information, etc. Moreover, those services perform data anonymization, preserve the data, and send it to third-parties healthcare or research services. The purpose is to avoid readmission, avoid medication errors, improve diagnoses, and decrease duplicate testing. Patients and customers trust healthcare and research services until one unexpected activity can break that. Data Breach is a hacking technique that uses ransomware, phishing,

malware, and denial of service to gain unauthorized access to sensitive information. This dangerous activity can compromise the trust between the patients and those healthcare services. It even happens in education, banking service, finance, etc. The International Business Machines Corporation, or IBM, performs annual data breach report to calculate the average cost of data breaches worldwide, the economic effect on healthcare, and the projected cost in the future. The average price of data breaches is \$4.24 million nationwide. According to reports, healthcare has had the highest cost of data breaches than any other industry for 11 consecutive years. In 2019, it cost the industry \$6.45 Million. As of 2021, the annual cost for the industry is \$9.23 million. To be clear, the United States Department of Health and Human Services Office for Civil Rights has been investigating healthcare services for data breaches for the previous 24 months. In the dataset, the names of those services and the numbers of affected individuals can create many possibilities for another breach by an unauthorized user. The current customers' identity and the reputation of those companies are at stake; therefore, these companies must be protected to prevent current or former patients from being hijacked again.

In this project, I compared anonymization techniques that will help determine which of those can be the most secure in protecting identities. I used data pseudonymization, data perturbation, and synthetic data in this project. Data pseudonymization is a technique in which the data are being altered to produce data that are concealing the dataset's identity. It has methods including tokenization, masking, and encryption. In pseudonymization, data masking and data encryption were the methods for this project. Data perturbation adds noise to the database, allowing confidentiality for the data. Microsoft SQL Server Management Studio was the database management software that I used. I used random functions to represent random number generators to the numerical values to represent data perturbation. I produced

synthetic data (also known as data generation) to help differentiate the actual data in the Mockaroo website.

## 2. Background

Data is a large set of information that contains strings, characters, numerical values, binary code, etc. Data is vital to all industries, such as manufacturing, software development, information technology, healthcare, human resources, etc. Everybody depends on them to understand anything or anybody. With each data, it must be protected, especially if we are sharing our data with health care providers and/or doctors. Patients consciously share their personal information such as name, address, social security number, and birth date to their providers. Research showed that out of 100 individuals, 88 of them neither trusted the government or private sectors with their information. Medical workers must have those data for performing research and/or completing required tasks for their patients. Attackers use public records to catch sensitive information, and patients can easily be compromised in many aspects of their lives to a simple maliciousness. It can cause a breach in the relationships between the patients and healthcare providers. Data anonymization techniques such as tokenization, pseudonymization, generalization, etc., can help protect these data. The data generalization has been used in this paper; however, it did not prove how effective it can be with the quasi-identifier. I was not too fond that data generalization was used, and the author failed to explain how cosine similarity and random forest affect positive changes. Also, I am learning from this paper that there are other ways to protect data. It also became the foundation of my project to compare the effectiveness of anonymization techniques such as pseudonymization and perturbation [2019].

Once upon a time, Static Data Masking was one of the most convenient tools for data masking, especially in database management. Since the security layers in applications rise, Static Data Masking became tolerable for Development and QA environments. Dynamic Data Masking became relevant for database security to match the protection of the application layers. Data masking is a type of anonymization that will enhance data security by almost completely censoring or altering the original data from unauthorized users. The expectation for Dynamic Data Masking is always high as we should expect 100% security from malicious individuals hacking our technologies [2017].

Simple masking, numeric manipulation, and data substitutions are three applied techniques for data masking. In this paper, I agree that they show the

results of data makings and pointed out some of the types of data masking. If they can touch on the modern-day approaches to set up Dynamic Data Making and its functions, this paper will be more interesting to read [2015]. I am reinventing the wheels by using this masking method, and I will touch on some of the functions that I have used in data masking.

When a database encounters encryption, it will be challenging but possible for unauthorized users to access it. Encrypting a database requires us to restrict access through authorization and authentication. The authorized user can limit who can use that data through authorization and authentication with a username and a password. When plaintext is saved in the database, the user can use whatever skills that he or she has and access the database, which will manipulate the data. Once the plaintext is being encrypted with a key, it will be converted to a ciphertext and can be decrypted back to plaintext. However, when plaintext is being converted to a hash value, the change is permanent. In this paper, I am interested in the fact that database security is being emphasized, and they compared encryption and hash value conversion. I am using encryption as one of my pseudonymization techniques to convert my dataset to ciphertext with two functions in the SQL Server Management Studio feature.

In my project, I am using something similar to the random number generator. I read a paper that will relate to this technique that I am using to represent the data perturbation technique. Pritanka, Imran Hussain, and Aqeel Khalique, all of Jamia Handard University, issued the following paper: Random Number Generators and their Applications: A Review [2019]. Random numbers are sequences of numbers that are independently being calculated in uniform distribution. They pointed out that random number generator is enormous in our daily lives, especially when we visit our financial institution, participate in gaming, read data from analytics, etc. Random number generators could be parts of the encryption; however, it is tough to overcome algorithms such as AES, RSA, and ECC. Randomness and unpredictability are required for a random number to have the sequence.

True Random Numbers Generators and Pseudorandom Numbers Generators are two categories of random number generators. True Random Numbers Generators have four major generators: random.com, Hotbits, Lasers, and Oscillators. Linear Congruential Generators, Blum BlumShub Generator, and Linear Feedback shift Register are the types of Pseudorandom Numbers Generators. This paper successfully touched on the background information of Random number

generations; however, they failed to address how it can contribute to the cybersecurity and protection-privacy world. Also, they did not do an excellent job explaining their results thoroughly [2019].

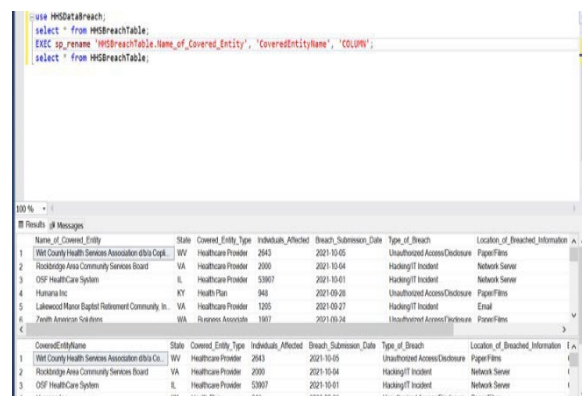
I am using my own random number generator version to produce artificial numerical values when demonstrating data perturbation. I am using functions to demonstrate my understanding of SQL programming.

Synthetic data is a type of data that contains constraints to prevent the revealing of the original data. It is one of the most vital topics in data protection. There are two types of synthetic data: fully synthetic data and hybrid synthetic data. The fully synthetic data is unique and artificial, while partially synthetic data only produces selective, genetic data. Various authors studied one of the types of synthetic data, heavily fully synthetic, and they explained their proposed models and their algorithms behind them. This paper should have displayed some results in this paper. Therefore, I generated a hybrid synthetic data (mostly fully synthetic data), except that a column would have values with matching values (randomly), and I am displaying my results and explaining them.

### 3. Approach of Techniques and Methods

In this project, my overall goal is to test a few anonymization techniques to determine which techniques can conceal the entries' information from the United States Department of Health and Human Services Office for Civil Rights' s dataset. I renamed the columns to make them easier to code.

For most of my project, I used the Microsoft SQL Server for the following pseudonymization techniques: masking and encryption.



```
use HHSDataBreach;
select * from HHSBreachTable;
EXEC sp_rename 'HHSBreachTable.Name_of_Covered_Entity', 'CoveredEntityName', 'COLUMN';
select * from HHSBreachTable;
```

Name_of_Covered_Entity	State	Covered_Entity_Type	Individuals_Affected	Breach_Submission_Date	Type_of_Breach	Location_of_Breached_Information
1 West County Health Services Association dba Co.	WV	Healthcare Provider	2643	2021-10-05	Unauthorized Access/Disclosure	Paper/Film
2 Rockbridge Area Community Services Board	VA	Healthcare Provider	2000	2021-10-04	Hacking/IT Incident	Network Server
3 OSH HealthCare System	IL	Healthcare Provider	53807	2021-10-01	Hacking/IT Incident	Network Server
4 Humana Inc	NY	Health Plan	940	2021-09-28	Unauthorized Access/Disclosure	Paper/Film
5 Lakewood Manor Baptist Retirement Community, Inc.	VA	Healthcare Provider	1205	2021-09-27	Hacking/IT Incident	Email
6 Jewish American Soldiers	WA	Physician Associate	1807	2021-09-24	Unauthorized Access/Disclosure	Paper/Film

CoveredEntityName	State	Covered_Entity_Type	Individuals_Affected	Breach_Submission_Date	Type_of_Breach	Location_of_Breached_Information
1 West County Health Services Association dba Co.	WV	Healthcare Provider	2643	2021-10-05	Unauthorized Access/Disclosure	Paper/Film
2 Rockbridge Area Community Services Board	VA	Healthcare Provider	2000	2021-10-04	Hacking/IT Incident	Network Server
3 OSH HealthCare System	IL	Healthcare Provider	53807	2021-10-01	Hacking/IT Incident	Network Server
4 Humana Inc	NY	Health Plan	940	2021-09-28	Unauthorized Access/Disclosure	Paper/Film

### 3.1. Dynamic Data Masking

For the masking, I applied the Dynamic Data Masking in the dataset to help determine how effective the functions are. I ran two tests to ensure that it can be protected 100 percent.

#### 3.1a. Dynamic Data Masking Test 1

For the first test of the Dynamic Data Masking, I expected this dataset to mask at least half of the data. On this test, I applied the default function in the CoveredEntityName, which represents the names of the Covered Entity, State, IndividualsAffected, and Date columns; and the default function will completely mask the information with multiple Xs in string. The Date column lists the date of the breach that those companies encountered. When I masked the date format in this column, every value in the row became altered to the default value 01-01-1900. The IndividualsAffected column listed the numbers of individuals that the breach affected in each company; therefore, the numerical values became zero by default.

For the BreachType column, which lists the type of breaches (Unauthorized Access, Disclosure, IT Incident, Hacking, etc.) that those companies encountered, and the BreachLocation column, which indicates the locations (emails, internet, laptop, desktops, etc.) of the breaches happened; I used the partial function to partially put Xs after at least the first letter and left the final letter in all of the string values. The purpose of using the partial function is to determine unauthorized users can recognize those values in these two columns.

#### 3.1b. Dynamic Data Masking Test 2

```
/*About To Apply Dynamic Data Masking*/

/*Alter the CoveredEntityName Column*/
alter table HHSActualDataBreachMasking2
alter column CoveredEntityName add masked with (function = 'default()'); /*using the default function*/

/*Not Altering the State Column*/

/*Alter the Individuals_Affected Column*/
alter table HHSActualDataBreachMasking2
alter column Individuals_Affected add masked with (function = 'random(0, 5000000)'); /*using the random function*/

/*Alter the Date Column*/
alter table HHSActualDataBreachMasking2
alter column [Date] add masked with (function = 'default()'); /*using the default function*/

/*Alter the BreachType Column*/
alter table HHSActualDataBreachMasking2
alter column BreachType add masked with (function = 'partial(1,"XXXXXXXX",1)'); /*using the partial function*/

/*Alter the BreachLocation Column*/
alter table HHSActualDataBreachMasking2
alter column BreachLocation add masked with (function = 'partial(1,"XXXX",2)'); /*using the partial function*/
```

I performed a second test for the Dynamic Data Masking technique. This time I applied functions

differently. The default function altered the CoveredEntityName and the Date columns, and the CoveredEntityName column has values censored with X's, and the Date column values have altered to 01-01-1900.

I used the random functions to alter the IndividualsAffected columns by randomizing values up to 5,000,000 affected individuals for each company. Once again, I used the partial function for the BreachType and the BreachLocation columns. I left the State column Unaltered because I am testing to see if there is any possibility that any unauthorized user can use any randomized number, Covered\_Entity\_Type column values, and the Business\_Associate\_Present column values to attempt to link to the companies from the original data.

### 3.2. Encryption

Encryption is the second pseudonymization technique that I used.

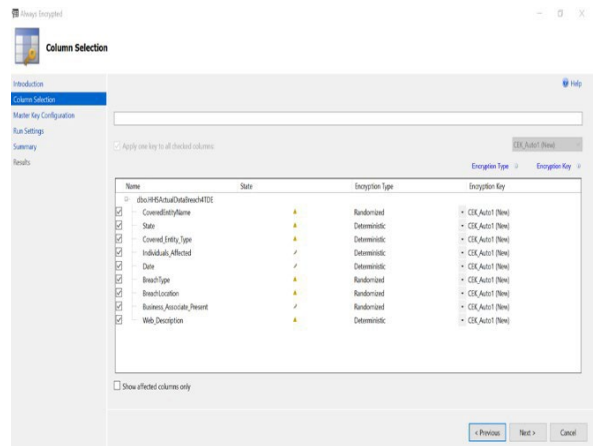
Encryption is the second pseudonymization technique that I used. Transparent data encryption is the type of encryption that database management companies such as Microsoft, Oracle, and IBM.

In the SQL Server Management Studio, I used the master database and created my master key. Afterward, I created a certificate and switched to the database I used to perform the encryption. I made my database encryption key that the Server Certificate protected in the master database. Then, I turned on the encryption on the database that I used. Afterward, I created a backup certificate.

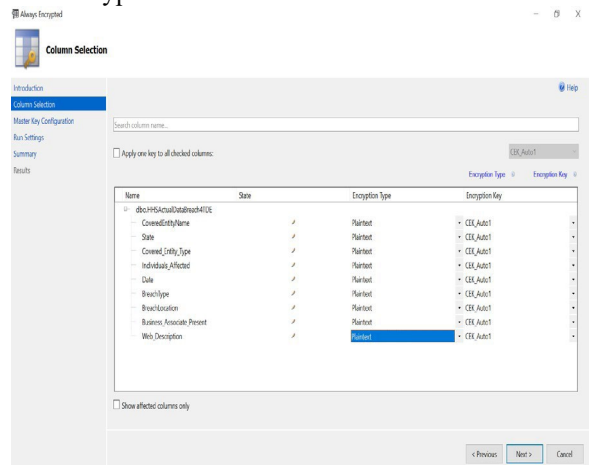
Now I had to restore the database to ensure that TDE became existed. When I opened the location of the backup file, I realized that I did not have permission to open it.

Now I used an encryption feature in SQL Server Management Studio called Always Encrypted. The Always Encrypted feature contains two functions, deterministic and randomized.

The deterministic function will produce the same outputs based on the inputs, and the randomized function will produce random results.



Afterwards, I converted the dataset back to plaintext and decrypted the database.



### 3.3 Other Anonymization Techniques

Once again, the options for anonymizing data with datasets and scripts are limited; therefore, I decided to use data perturbation and synthetic data. For data perturbation, I realized that I could use functions to alter the integer values of the dataset. Therefore, I used the absolute value function, abs(), to generate any number between 1 and 1000000000 in the Individuals\_Affected column. Afterward, I used the DateAdd function, DATEADD(), which altered the actual dates displayed and use dates between 11-01-2019 and 11-01-2021. For the Business\_Associate\_Present column, I randomized the 0s and the 1s with the absolute value function.

```

--use H5ActualDataBreachRMG;

select * from H5ActualDataBreachRMG;

update H5ActualDataBreachRMG set Individuals_Affected = ABS(checksum(NewId())) % 100000000 where Individuals_Affected is not null;

select * from H5ActualDataBreachRMG;

/*Declaring Start and End Date*/
DECLARE @start DATE = '2019-11-01'
DECLARE @end DATE = '2021-11-01'

update H5ActualDataBreachRMG set [Date] = DATETIME(DAY,ABS(CHECKSUM(NewId())) % DATETIMEOFF(DAY,@start,@end) @start) where [Date] is not null;

select * from H5ActualDataBreachRMG;

update H5ActualDataBreachRMG set Business_Associate_Present = ABS(checksum(NewId())) % 2 where Business_Associate_Present is not null;

select * from H5ActualDataBreachRMG;

```

For the Synthetic Data, it is a technique that represents Data Generation. It produced artificial data that is unique to the original data. I used the website called Mockaroo.

The Mockaroo website interface shows a schema configuration for a database table. The schema includes the following fields and options:

- CoveredEntityName**: Fake Company Name, blank, 0%, 0%.
- State**: State (abbrev), restrict states..., Only US, blank, 0%, 0%.
- Covered\_Entity\_Type**: Custom List, Healthcare Provider, Business Associate, Health Plan, Healthcare Oes, random, blank, 0%, 0%.
- Individuals\_Affects**: Number, min: 1, max: 100000000, decimals: 0, blank, 0%, 0%.
- Date**: Datetime, 11/01/2019 to 11/15/2021, format: yyyy-mm-dd, blank, 0%, 0%.
- BreachType**: Custom List, Unauthorized Access/Disclosure/Hacking/IT Incident/Theft/Loss, random, blank, 0%, 0%.
- BreachLocation**: Custom List, Paper/Films, Network Server, Email, Laptop, Other, Desktop Computer/, random, blank, 0%, 0%.
- Business\_Associate**: Number, min: 0, max: 1, decimals: 0, blank, 0%, 0%.
- Web\_Description**: Buzzword, blank, 0%, 0%.

Buttons at the bottom include: DOWNLOAD DATA, PREVIEW, CHANGES SAVED, CREATE API, and MORE.

In the Mockaroo website, it allowed me to choose the type of data and the number of rows I desire for my dataset. They offer Fake Company Names, State, A Custom List of Values that the schema of the website can randomize for me, number with ranges, DateTime, and those who are looking for random characters.

## 4. Evaluation

For most of the evaluation, I use Anaconda Jupyter Software to compare the values of the original data and the altered data with the Python programming language.

```

In [1]: #Comparing the Plaintext CSV file to the Original File For Similarities After DDE

import pandas as pd
import numpy as np

originalFile = pd.read_csv('H5ActualDataBreachRMG.csv') #Original Data With Changed Column Names
ddoneFile = pd.read_csv('PlaintextResultsAfterAlwaysEncrypted.csv') #DDM1 Masked Data

#List Method: Check if One DataFrame is Different From Another, Returns Boolean Values Only
array1 = np.array(originalFile)
array2 = np.array(ddoneFile)

originalFile_csv = pd.DataFrame(array1, columns=['CoveredEntityName', 'State', 'Covered_Entity_Type', 'Individuals_Affected', 'Date',
ddoneFile_csv = pd.DataFrame(array2, columns=['CoveredEntityName', 'State', 'Covered_Entity_Type', 'Individuals_Affected', 'Date', 'I

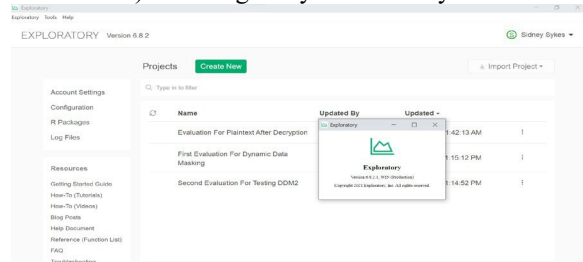
originalFile_csv.index += 1
ddoneFile_csv.index += 1

compareValues = originalFile_csv.eq(ddoneFile_csv).to_string(index=True)
print(compareValues)
print("\n")

compareValuesFile = (originalFile_csv.eq(ddoneFile_csv)).to_csv('compareValuesOriginal2Plaintext.csv') #Creating a File With The

```

Afterward, I uploaded the boolean .csv file created in Jupyter, and I ran a DataFrame with the software called Exploratory. The purpose of using Exploratory is to ensure that the Boolean values of the file that represents two compared data (the original data and the altered data) are being analyzed correctly.



### 4.1a. Evaluation for Dynamic Data Masking 1

I set a goal that I expected the dynamic data masking to hide at least 60 percent of the data in the data set. The results displayed the usage of the default and the partial functions.

	A	B	C	D	E	F	G	H	I
1	CoveredEntityName	Date	Covered_Entity_Type	Individuals_Affected	Date	BreachType	BreachLocation	Business_Associate	Web_Description
2	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	PO000000	PO000000	0 NULL	0 NULL
3	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	0 NULL	0 NULL
4	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	0 NULL	0 NULL
5	xxxx	xxxx	Health Plan	0	1/1/2020 10:00:00:0000000	PO000000	PO000000	1 NULL	1 NULL
6	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	PO000000	PO000000	0 NULL	0 NULL
7	xxxx	xxxx	Business Associate	0	1/1/2020 10:00:00:0000000	PO000000	PO000000	1 NULL	1 NULL
8	xxxx	xxxx	Health Plan	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	0 NULL	0 NULL
9	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
10	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
11	xxxx	xxxx	Health Plan	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	0 NULL	0 NULL
12	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
13	xxxx	xxxx	Health Plan	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
14	xxxx	xxxx	Health Plan	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	1 NULL	1 NULL
15	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
16	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
17	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
18	xxxx	xxxx	Healthcare Clearing Hst	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	1 NULL	1 NULL
19	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	0 NULL	0 NULL
20	xxxx	xxxx	Healthcare Provider	0	1/1/2020 10:00:00:0000000	EX000000	EX000000	0 NULL	0 NULL
21	xxxx	xxxx	Health Plan	0	1/1/2020 10:00:00:0000000	NO000000	NO000000	0 NULL	0 NULL

When I compared the original data with the first masked data with Boolean values, it noted that the Covered\_Entity\_Type and Business\_Associate\_Present columns retain 100 percent of their values respectively. Web\_Description column is null; therefore, it returned as false. Null values in the database world will always be false because it cannot be determined as true or false.



Indeed, the functions that I applied for the first test of Dynamic Data Masking resulted in a 100 percent protection rate. Therefore, this test successfully protected the identity of those companies.

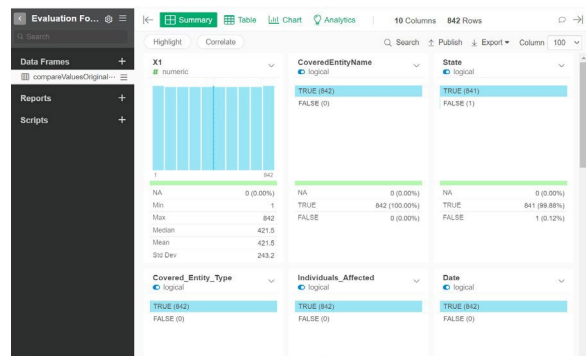
## 4.1 b. Evaluation for Dynamic Data Masking 2

	A	B	C	D	E	F	G	H	I	J
1	CoveredEntity	State	Covered_Individual	Date	BreachType	BreachLoc	Business_Web	Description		
2	xxxx	WV	Healthcar	2696126	1/1/1900	UX000000X	PXXXXms	0	NULL	
3	xxxx	VA	Healthcar	3662188	1/1/1900	HX000000X	NXXXXer	0	NULL	
4	xxxx	IL	Healthcar	3976786	1/1/1900	HX000000X	NXXXXer	0	NULL	
5	xxxx	KY	Health Pla	4409841	1/1/1900	UX000000X	PXXXXms	1	NULL	
6	xxxx	VA	Healthcar	703465	1/1/1900	HX000000X	EXXXXil	0	NULL	
7	xxxx	WA	Business F	308134	1/1/1900	UX000000X	PXXXXms	1	NULL	
8	xxxx	GA	Health Pla	4275335	1/1/1900	HX000000X	NXXXXer	0	NULL	
9	xxxx	TX	Healthcar	802457	1/1/1900	HX000000X	EXXXXil	0	NULL	
10	xxxx	TX	Healthcar	2383373	1/1/1900	HX000000X	EXXXXil	0	NULL	
11	xxxx	IL	Health Pla	1836727	1/1/1900	HX000000X	NXXXXer	0	NULL	
12	xxxx	NY	Healthcar	4710067	1/1/1900	TX000000X	LXXXXop	0	NULL	
13	xxxx	AK	Health Pla	1199560	1/1/1900	HX000000X	DXXXXil	0	NULL	
14	xxxx	CT	Health Pla	4438923	1/1/1900	HX000000X	EXXXXil	1	NULL	
15	xxxx	FL	Healthcar	794371	1/1/1900	UX000000X	OXXXXer	0	NULL	
16	xxxx	TX	Healthcar	883977	1/1/1900	HX000000X	DXXXXer	0	NULL	
17	xxxx	IL	Healthcar	3679519	1/1/1900	HX000000X	NXXXXer	0	NULL	
18	xxxx	GA	Healthcar	1103051	1/1/1900	HX000000X	NXXXXer	1	NULL	
19	xxxx	PA	Healthcar	1414901	1/1/1900	HX000000X	NXXXXer	0	NULL	
20	xxxx	FL	Healthcar	4537107	1/1/1900	TX000000X	DXXXXce	0	NULL	
21	xxxx	WA	Health Pla	887050	1/1/1900	HX000000X	NXXXXer	0	NULL	

As I evaluated the difference between the original data and this test results, there were only false value in the State column. After searching manually in Microsoft Excel, it was only a null value. Despite using the random function for the IndividualsAffected column, it did not play a huge role in attempting to link it with other data information to identify any of the affected companies. Consequently, these functions once again successfully protected the identity of the companies. Despite the null value which summarized as False in the State column, this test earned a perfect score of 100 percent.

Overall, Dynamic Data Masking has done a perfect job concealing the identity of these companies.

## 4.2 Evaluation for Encryption

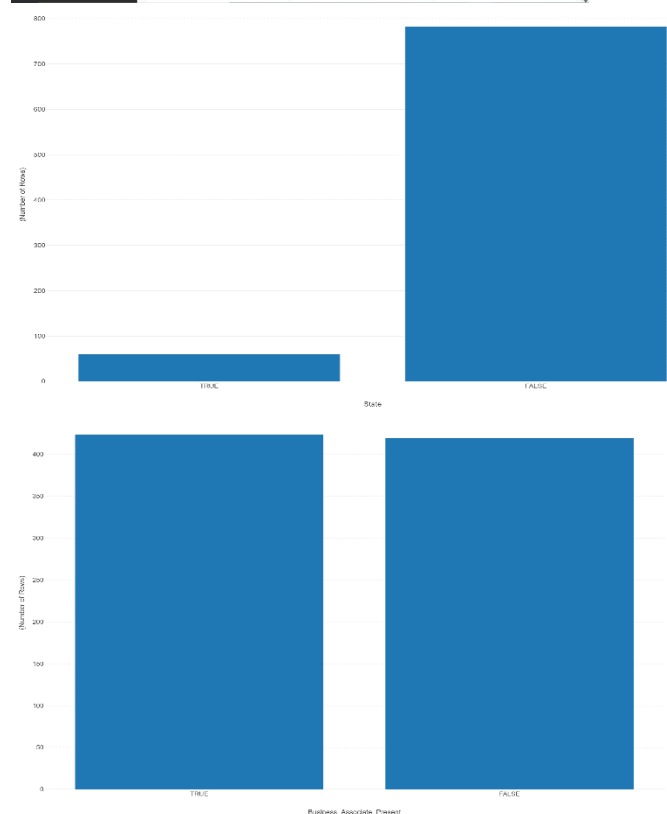
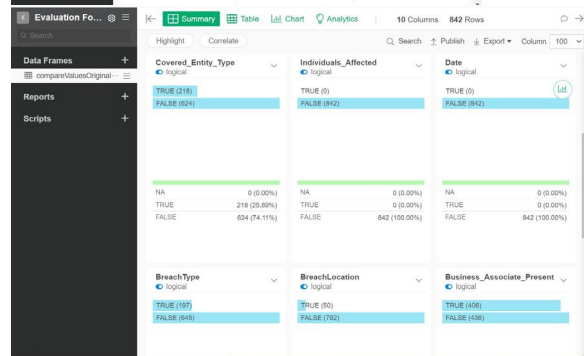
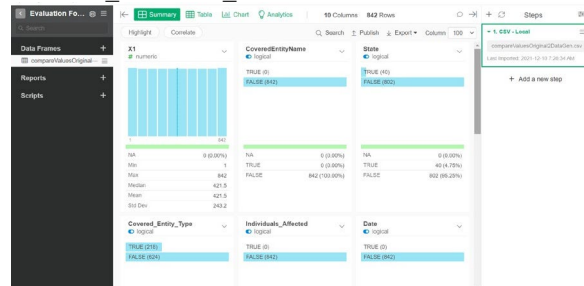


For the encryption part of the project, the transparent data encryption have done an excellent job of encrypting into ciphertext and decrypting back into plaintext. When I compared the original file and the plaintext file, I received a 100 percent turnout as the decryption was successful.

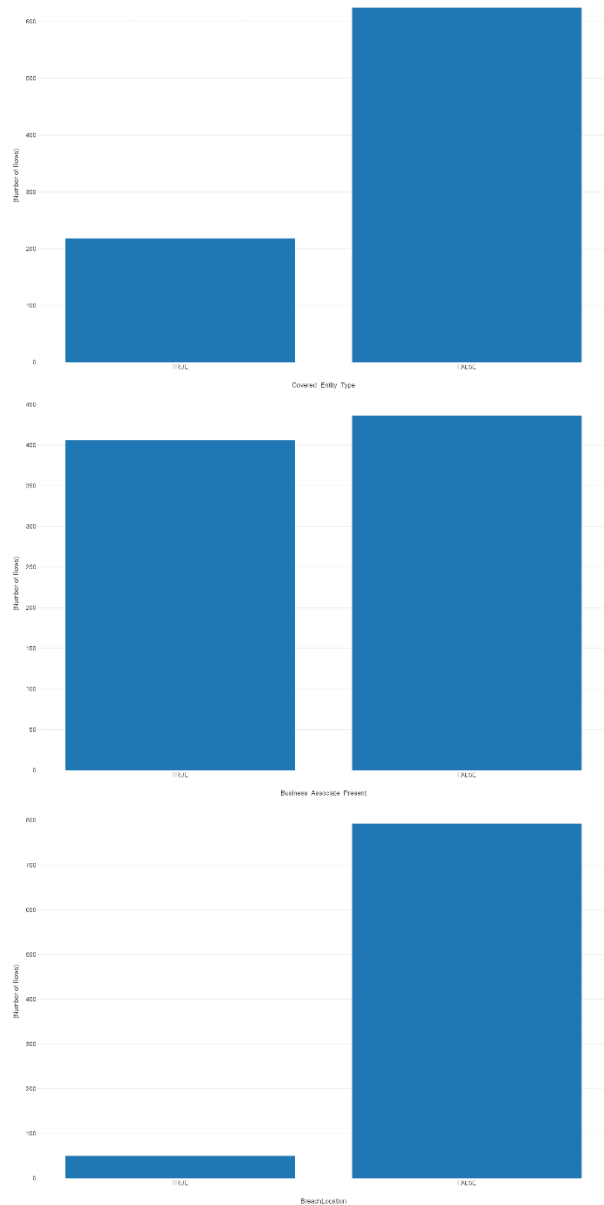
## 4.3 Evaluation for Other Techniques

	A	B	C	D	E	F	G	H	I	J
1	CoveredEntityName	State	Covered_Entity_Type	Individuals_Affected	Date	BreachType	BreachLocation	Business_Associate_Present	Web_Description	
2	West County Health Services, WV		Healthcare Provider	7129636	6/25/2003	Unauthorized Access/DiscPaper/Films	Network Server	0	NULL	
3	Rockledge Area Community Hlth		Healthcare Provider	8327813	2/24/2012	Hacking/Tf	Incident	Network Server	0	NULL
4	CDP HealthCare System, IL		Healthcare Provider	87763237	6/29/2003	Hacking/Tf	Incident	Network Server	0	NULL
5	Hannum Inc		Health Plan	68156545	6/19/2012	Unauthorized Access/DiscPaper/Films	Email	1	NULL	
6	Salmonstad Moore Baptist Church		Health Plan	68236761	8/28/2012	Hacking/Tf	Incident	Desktop Computer, Lapt	0	NULL
7	Janith American Solutions, WI		Business Associate	36955260	11/18/2009	Unauthorized Access/DiscPaper/Films	Network Server	1	NULL	
8	Digital Insurance, LLC doing USA		Health Plan	90777163	4/26/2009	Hacking/Tf	Incident	Network Server	1	NULL
9	Orthopaedic Foundation of Texas TX		Healthcare Provider	47522932	10/5/2012	Hacking/Tf	Incident	Email	0	NULL
10	The Messenger Clinic, TX		Healthcare Provider	60771272	10/5/2010	Hacking/Tf	Incident	Network Server	1	NULL
11	Westat, Inc. Health Plan arch.		Health Plan	63446764	8/7/2003	Hacking/Tf	Incident	Network Server	0	NULL
12	Labrador Animal HSP, NY		Healthcare Provider	875594001	10/10/2007	Theft	Laptop	0	NULL	
13	State of Alaska Department of JC		Health Plan	12100002	7/30/2012	Hacking/Tf	Incident	Desktop Computer, Lapt	0	NULL
14	Antea ACE, CT		Health Plan	41704128	4/3/2002	Hacking/Tf	Incident	Email	0	NULL
15	Central Medical Group, IL		Healthcare Provider	62963036	6/29/2003	Unauthorized Access/DiscPaper/Films	Desktop Computer, Etc	0	NULL	
16	Mulliken Surgical Specialty Ce TX		Healthcare Provider	1784134	3/3/2002	Hacking/Tf	Incident	Network Server	0	NULL
17	Illinois Department of Human S		Healthcare Provider	99438421	12/19/2009	Hacking/Tf	Incident	Network Server	0	NULL
18	Georgia Department of Human CA		Healthcare Provider	14492002	3/7/2002	Hacking/Tf	Incident	Network Server	1	NULL
19	Warner House, WI, & F.A. PL		Healthcare Provider	51207197	7/26/2009	Hacking/Tf	Incident	Network Server	1	NULL
20	Chick & Leaser, MI & F.A. PL		Healthcare Provider	51200082	6/12/2002	Theft	Desktop Computer, Etc	1	NULL	

As expected, I did not expect the results with the random number generator to be as protective as the pseudonymization. Surely enough, there are some values that matched in the State and the Business\_Associate\_Present columns.



I expected data generation to have a turnout rate of 100 percent. However, the Data generation have a turnout of 50 percent because some of the values in the State columns matches as well as the breach locations, number of business associates, and the covered entity.



Those values can be linked together and find the actual companies' names through those values.

## 5. Summary and Conclusion

Overall, I used a dataset from the United States Department of Health and Human Services Office for Civil Rights, which represented the companies who reportedly encountered a data breach within the previous 24 month as of October 2021. Throughout this whole project, I used Microsoft SQL Server Service Management

for all of the techniques except the Synthetic Data. For data pseudonymization, I used Dynamic Data Masking and Transparent Data Encryption with the Always Encrypted feature. The results came out successful and hideable for the pseudonymization. I used functions to represent the random number generator, which I considered as the part of data perturbation. It resulted in less protection because I can only edit the integers. For the synthetic data, it did not fare well enough to prevent users to be able to identify which company can be protected.

If I can continue to expand this project, I will continue to research for techniques that is compatible for database. I would put more effort in the evaluation side of the project as well. I would attempt to add some data visualization in this project to display an overall summarization for each result. I would even try to compare the result of the techniques against one another to test their differences instead of just the original data. I would create multiple datasets with similar and different values on each row, and I would have scatterplots, bar graphs, etc.

## 6. References

- [1] Kumar G.K., R. & Rabi, B. J., and TN, M. (2012). A Study on Dynamic Data Masking with its Trends and Implications. *International Journal of Computer Applications*. 38. 19-24. 10.5120/4612-6828.
- [2] Majeed, A. (2019). Attribute-centric anonymization scheme for improving user privacy and utility of publishing e-health data. *Journal of King Saud University - Computer and Information Sciences*. 31-4. 426-435. 1319-1578. <https://doi.org/10.1016/j.jksuci.2018.03.014>.
- [3] Priyanka, Hussain, I., and Khaliq, A. 2019. Random Number Generators and their Applications: A Review. 7. 1777-1781.
- [4] Singh, P. and Kaur, K. (2015). Database security using encryption. *2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE 2015*. 353-358.
- [5] Surendra H., Mohan H.S. 2017. A Review of Synthetic Data Generation Methods For Privacy Preserving Data Publishing. 6. 2277-8616.