Privacy Preserving Optimized Fuzzy like Search over Encrypted Data using Phonology

Saurabh Gupta, PhD
Scientist-G
National Informatics Centre
New Delhi, India

Piyushank Gupta
Scientist-C
National Informatics Centre
Lucknow, Uttar Pradesh, India

Anup Kumar
Scientist-C
National Informatics Centre
New Delhi, India

Mohd. Wasim
Technology Consultant
N.I.C.S.I
New Delhi, India

ABSTRACT
With increasing need of e-Governance for boosting principle of "Minimum Government - Maximum Governance", lots of government software applications were developed to capture citizen's data and deliver various services to them. Sometimes captured data (like KYC data) is highly sensitive in nature and governed by personal information protection laws. Data at rest is of no use unless it is analyzed to generate useful information. Since the ownership of data lies with different government bodies, analysis over this data was possible only in isolation or else data privacy is compromised on data sharing. One approach to maintain data privacy before sharing it is to encrypt it using an encryption technique but strong Encryption techniques use diffusion principle identified by Claude Shannon which makes analysis over Cipher text inefficacious. This is major deterrent in discovery of useful patterns, relationships and insights in contrast to when analysis is done over multiple data sources collectively which can be important input for Government Agencies in thwarting untoward incidents. This paper presents an approach to make use of Double Metaphone algorithm which reduces English words to their phonetic representations by using combination of consonant sounds and plausible pronunciations thereby achieving efficient fuzzy like search using exact match over encrypted phonetic representations. The string variations created by typo mistakes during search by a user was handled by the phonology and consonant substitution, thereby achieving speedy and accurate fuzzy like search over encrypted data. This not only preserves the privacy but enables data owners to share the data to the cloud for running further analytics over cipher text and sharing encrypted results with authorized government agencies on demand for further decryption with pre-shared symmetric key.

Keywords
Data Privacy, Data Analytics, Data Governance, Advanced Encryption Standard, Double Metaphone, Fuzzy Search, Phonetic Matching

1. INTRODUCTION
With the rapid development of technology and decreasing computing costs, 21st century is referred as Information Age which signalizes the rapid shift from traditional industry to knowledge economy in which data is treated as new gold. Recognizing its significance, Government of India has launched Digital India initiative [1]. The objective was to transform India into a digitally empowered society by creation of ICT infrastructure, computerization of government service delivery workflows, digitized healthcare and educational services, cashless economy and digital transactions, promotion of digital literacy among citizen and round-the-clock availability of governance at doorsteps. As a result of which, today government is largest data generator in the country. The ease of data collection and low cost of storing and processing information resulted in the prevalence of long-term storage of information as well as collection of increasingly minute details about an individual which allows an extensive user profile to be created. Government applications are collecting mostly two types of data viz KYC data and Transactional data.

KYC data like Passport, Voter ID, Driving License etc. are very sensitive and personal in nature. Large number of benefits can be gained by analyzing this personal data. For instance, Identification of duplicate records within these datasets helps in cleaning datasets [8] and doing entity profiling. Another type of data is transactional data which is structured but contains many duplicate values about an individual and less trustworthy because of lack of caution was excised during capturing of such data due and is thus less standardized, contains many spelling mistakes and no unique identifiers which uniquely identifies multiple records belonging to same individual. For example, travel data of railways, call records of mobile companies, financial transactions etc. lies in this category. Analysis of this data and its linkage with KYC data helps in identifying mischievous patterns tagged by complete profiles which help various agencies in intelligence gathering and countering terrorism. It also helps government in targeting delivery of social welfare benefits, effective planning and implementation of various schemes.

While insights derived after analyzing data are very useful, the arbitrary and unregulated use of sensitive personal data has raised concerns regarding the privacy and autonomy of an individual. Some of the concerns are related to data sharing with central database, profiling of citizens, increased surveillance and a consequent erosion of individual autonomy because valuable insights can only be generated when all data from different stakeholders is brought to a central location for analytical purposes. This was also highlighted in landmark

judgment of the Supreme Court of Puttaswamy [2], which recognized the right to privacy as a fundamental right. Further, it states that informational privacy is an important aspect of the right to privacy and directed the Union Government to formulate a robust data protection regime for protecting individual's privacy from the dangers caused by state and non-state actors in this information age.

In this context, in order to gain benefits from knowledge economy and mitigate the harms consequent to it, Data Governance [3] is required which is a collection of processes, policies, roles, standards, and metrics that are needed to protect data assets to guarantee correct, understandable, trustworthy, complete, and secure data. It defines access control and authorization policies over data assets, how those data assets are used within the organization and how it can be shared with others maintaining trade-off between privacy concerns and national interest.

Today, government agencies look for holistic patterns in data segregated from various sources where the search within data should be approximate (Fuzzy) rather than exact match. For this, Fuzzy match [4] is being used to identify two elements of text, strings, or entries that are approximately similar but not exactly the same. This was done to consider spelling variations of various generic attributes in data like name, father name, or address so that patterns in the data can be exploited to generate more meaningful insights. To apply this approach with encrypted (AES 256 encryption) data, fuzzy search over encrypted phonetic representations (using Double Metaphone [7]) has been done. Phonetic code generation algorithms have their own history and lot of advancements have been done right from the inception of Soundex [9] and to maintain privacy, the proposed approach does not need to maintain a dictionary of keywords to achieve fuzziness over encrypted data which was a tedious task [10][11].

2. METHODOLOGIES USED

Fig 1 shows the block sequence of activities that were performed for preserving privacy while doing fuzzy like search over encrypted phonetic representations of generic entity attributes. An AES 256 secret key was generated by Data Provider organization (data owner) and shared it to the querying user agency. Data Provider organization first generated phonetic representation (Phonetic Code [PC]) of sensitive generic personal attribute values like Name, Father /Spouse name, Address at their own premise using Double Metaphone algorithm. These representations as well as original personal attribute values (Unencrypted Data [PT]) were encrypted using AES 256-bit encryption technique [5] in counter mode using the generated shared secret key. Finally, encrypted personal attribute values E(PT) and corresponding encrypted phonetic codes E(PC) were outsourced to cloud service provider for central processing thereby preserving privacy. Indexes were generated over both E(PT) and E(PC) using B-tree and were uploaded by data provider organization to cloud service thereby improving search speed and maintaining privacy.

At the end of querying user agency, phonetic representation of search values was generated on the fly using Double Metaphone and was encrypted with same secret key previously shared by intended data provider organization for firing search. An exact match of these encrypted values was made over indexes shared by data owner and corresponding encrypted results, E(PT), was sent back to the querying user agency. Double Metaphone handled fuzziness in data. Now querying user agency can use same shared secret key to decrypt results, E(PT), to access plain text, PT. In this whole process, the cloud service provider need not to access plain data to provide a fuzzy like search service.

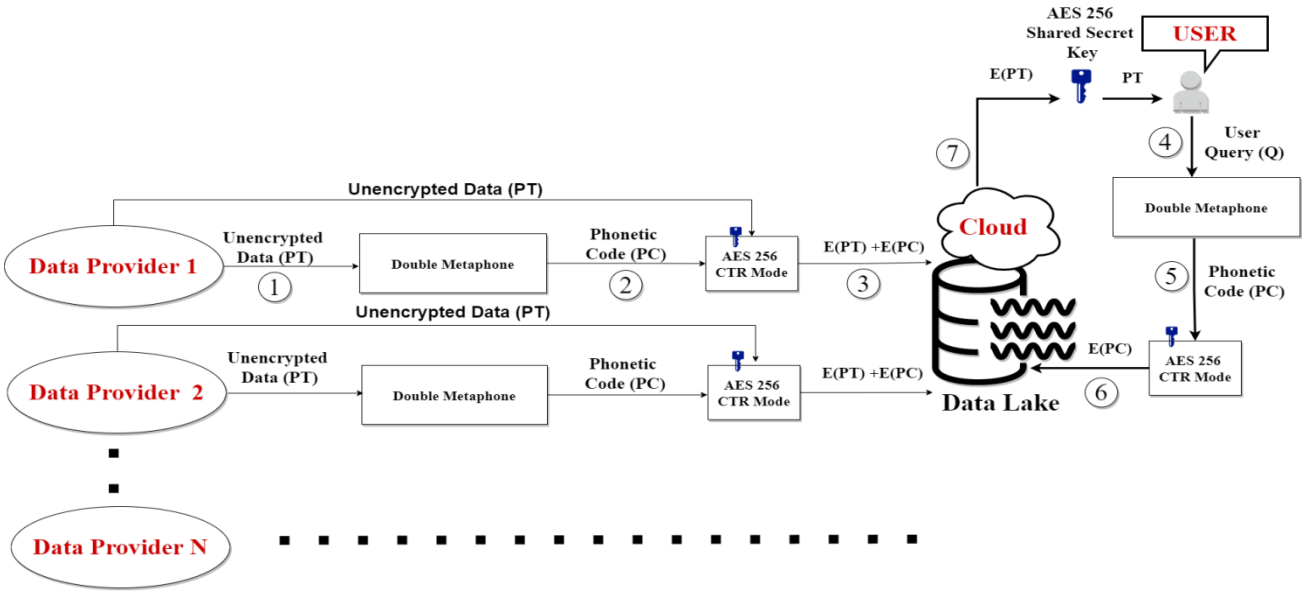

Fig 1: Block Diagram of Privacy Preserved Optimized Fuzzy like Search

3. DETAILED IMPLEMENTATION AND IMPROVEMENTS

In Computer Science and Data Analytics, 'approximate string matching' plays a significant role in finding strings that match a pattern approximately. The era of AI using Natural Language Processing has made approximate matching (also known as Fuzzy Matching) really powerful so that useful insights from data sources can be generated. The discussed approach deals with efficient fuzzy like search maintaining privacy of sensitive data. Since a Fuzzy match makes use of edit distance algorithm which is a way to quantify how similar or dissimilar two strings are by doing multiple insertion, deletion and substitution operations, it is compute intensive. This paper presents an efficient fuzzy like search technique

over encrypted strings that can work in real time. Following methods were followed:

3.1 Data Pre-processing

Appropriate data pre-processing was needed before running Double Metaphone over attribute values. This includes:

3.1.1 Making attribute values in a uniform case and removing characters like .(dot), / , - etc.

3.1.2 Remove salutations like Mr., Mrs., Dr., Professor, Shri, sri, smt, etc.

3.1.3 Remove other non-alphabetic characters.

3.2 Metaphone and Double-Metaphone

Metaphone algorithm is based on the idea of constructing a phonetic representation [6] for an input string. Two strings are then deemed similar if they have same phonetic code (based on similar pronunciation). Since regional language names written in English can't handle variations in pronunciations, the Double Metaphone system was used which computes two "sounds like" strings for a given input string — a "primary" and an "alternate", allowing for greater ambiguity. Table 1 shows some sample inputs and their outputs from Double Metaphone algorithm. As shown in table, entity names which can be a person, place etc. were converted to their corresponding phonetic representations that omit use of vowels but consonants had been preserved. Only those consonants that contributes to the accent of a particular word, is mapped to certain keyword(s) in Double Metaphone algorithm. All encrypted phonetic codes of generic attribute values stored in the database were generated in advance whereas phonetic codes of user query values were generated on the fly.

Table 1. Sample inputs and corresponding phonetic representations using Double Metaphone Algorithm

Input String	Entity Type	Phonetic Code
Saurabh, Sourabh	Name	SRP
Anup, Anoop	Name	ANP
Piyushank, Peeyushank	Name	PXNK
Wasim	Name	ASM, FSM
Waseem	Name	ASM, FSM
Kathyayini	Name	K0N, KTN
Smith	Name	SM0, XMT
Sachin	Name	SXN, SKN
Lucknow	Place	LKN, LKNF
Mussoorie	Place	MSR
Trivandrum	Place	TRFNTRM

3.3 Phonological Consonant Substitution in User Queries

As discussed in the previous section, only consonant(s) were considered to generate phonetic representation, but some of these consonants sound similar which may confuse user while making search and thereby increase the chances of misspellings in query values. Consonants J and Z have same sound, Jeenat can be misspelled as Zeenat and vice versa. But phonetic code of Jeenat is JNT, ANT and Zeenat is SNT which is different from JNT and SNT. Since an exact match between encrypted phonetic codes of user query and database stored values had been done, these misspellings were handled by phonological consonant substitution method. In this method, one member of a group was replaced by another member and database was being searched over both string variations in a single query.

Table 2. Pair of consonants having similar phonetic sounds

Group	Members	Cost of Substitution
1	J, Z	1
2	W, V	1
3	C, K, Q	Not required to be substituted in case of Double Metaphone

As shown in table 2, Group 1 and 2 are having cost of substitution as 1 and don't need any insertion and deletion operation for pattern matching. While Group 3 was already handled by Double Metaphone algorithm (no change in the phonetic codes), cost of insertion, substitution and deletion is zero. This eradicated the cost of insertion and deletion which was a considerable factor in the speed of edit distance based fuzzy match like Levenshtein distance [12].

3.4 Encryption using Advanced Encryption Standard

Advanced Encryption Standard (AES) in counter mode was used to encrypt data with 256 bit as key size. Data Provider Organization generates secret key and shares it with government agency (querying user) in advance. It also encrypts phonetic codes of generic attribute values along with raw data using shared secret key and forward this encrypted data for storage over cloud service. Agencies who want to perform fuzzy search over this uses shared secret key of respective provider organization for encrypting (AES 256) phonetic code of search values. All those encrypted data records were returned to user agency whose encrypted phonetic code matches exactly with encrypted phonetic code of search values. This match is being searched over a B-tree index already supplied by data owner organization to the cloud. Hence, the index is made over the same encrypted column, the AES-256 cipher text of codes generated by Double Metaphone. Finally, user agency decrypts results using same key to get real records. In this way, cloud service provider who processes user data, does not need to decrypt the data to search patterns in data.

3.5 Indexing over the encrypted phonetic representations of an entity attribute

There are many RDMS and NoSQL databases available that support indexing over columns of a table. An index makes search faster over the columns involved in the search. PostgreSQL [12] was used to store data which supports indexes like Hash, GiST, B-tree, GIN and SP-GiST. In this work, B-tree index [13] was used because it can be updated quickly with new data. The insertion, deletion and search using a B-tree based index takes logarithmic time and is useful for wide variety of situations [14]. To make search over encrypted data, B-tree indexes were made over encrypted phonetic representations generated by running Double Metaphone over attributes in the table to be searched. This has considerably improved the performance of search.

Table 3. Time required for encryption, phonetic code generation and index creation

S.No.	Activity	Time taken over 1 million records (In seconds)
1	AES-256 encryption with Double Metaphone Generation	10
2	B-tree index creation on S.No. 1	1

For simulation, six database tables were created with 10K, 50K, 100K, 300K, 500K and 1M records. Table 3, represent time taken for index creation over 1M encrypted phonetic code representations.

4. Results and Improvements

The four fuzzy matching techniques were simulated over 1 million records stored in PostgreSQL and compared for the run time. Fuzzy matching using edit distance approach compares two strings to quantify how similar or dissimilar two strings are. Levenshtein distance technique was selected to achieve the fuzzy search which is most widely used edit distance algorithm. Similarity function of PostgreSQL was also compared with PostgreSQL's '%' operator which used a GIN based index to speedup search and outperforms 'Similarity' function in terms of matching speed. GIN is a type of inverted index known as generalized inverted index. GIN was made over a column of type 'tsvector' which stored index entry of each word, with a compressed list of matching locations. Table 4 shows the approach comparisons which were simulated to do search in database (n being the number of records in database).

Table 4. Comparison between various algorithms performance calculating string similarities

Algorithm	Approach	Index Applied	Privacy Support
Levenshtein	Uses insertion, deletion and substitution operations for calculating the distance between two strings with run time complexity as O(m*n) where m & n are the lengths of two words to be matched.	Not Supported	No
Similarity	Splits words into trigrams and compares trigrams of one word to other word with exact match	GIN	No
% Operator	Similar to 'Similarity' function but is little faster as it does not calculate similarity score like 'Similarity'	GIN	No
Double Metaphone	Look up using tree like structure with O(log(n)) as time complexity.	B-tree	Yes

The third column of Table 4 shows the usage of index and their support for privacy preserving fuzzy like search. Since Double Metaphone generates a phonetic code that can be encrypted at the premise of data owner itself, an exact match can be done by the cloud-based server. This way cloud doesn't have to see real data to perform fuzzy like search on behalf of users.

Records	Fuzzy Using Double Metaphone (ms)	Fuzzy Using Edit distance (ms)	Similarity (ms)	% operator (ms)
10k	0.034	199.546	20.486	0.319
50k	0.027	997.965	107.937	1.757
100k	0.035	2002.382	206.470	2.604
300k	0.039	6005.808	630.688	7.433
500k	0.039	10002.361	1046.964	11.162
1M	0.030	19072.975	1984.686	21.520

Fig 2: Snapshot of run time of various algorithms

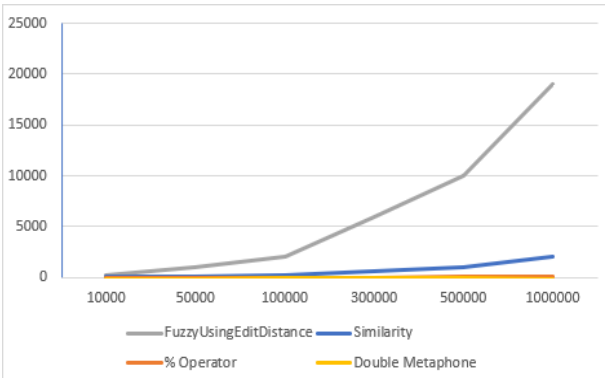

Fig 3: Plot of run time of various algorithms

Fig 2 shows the run time comparisons observed during the simulation process over 1 million records (X axis) and fig 3 is the plotted graph against those results. Unit of run time is in milliseconds (Y axis).

Both the tabular representation and graphical representation depict following advantages over each other as simulated over a i7 processor with 6 cores with 16GB of memory:

- Fuzzy using edit distance algorithm (Levenshtein) shows an almost exponentially increasing run time with increasing size of dataset. This is reflected by gray line in the graph plot. For 1 million records, it took 19.072 seconds. This clearly shows the approach is not very scalable as the run time complexity of string matching is O(m*n) where m and n are the lengths of two strings being matched.
- Similarity function of PostgreSQL was useful in speeding up the matching operation with the support of GIN (Generalized Inverted Index) based index and took 1.984 seconds for 1 million records which was almost 10 times faster than traditional string matching. This is depicted by blue line in the graph.
- % Operator of PostgreSQL was also able to use GIN index and works the same way Similarity function works with an exception that it does not spend extra time on calculating similarity scores and was thus significantly faster than Similarity. On a dataset size of 1 million records, it outperforms Similarity significantly. This is depicted by orange line in the graph.
- Fuzzy using Double-Metaphone approach supported B-Tree based index and was the fastest among all

which achieved an astonishing speed and made search over 1 million records in 0.03 milliseconds. yellow line of Double Metaphone almost kept a constant path with increasing data size. B-Tree indexed search has made the search time logarithmic. A B-Tree index-based search has search time complexity of O(log(n)).

Overall simulation shows that double metaphone and % Operator search time complexity is better than Fuzzy Levenshtein and Similarity algorithm. Simulation results may vary over hardware and data size.

5. CONCLUSION

An edit distance based string-matching algorithm cannot be applied over encrypted data and has a quadratic run time: O(m*n) where m, n is length of two strings being compared. The use of Double Metaphone has omitted the cost for insertion, deletion and substitution operations involved in matching two strings using fuzzy search, thereby enabling an exact match between two encrypted phonetic representations. This made match time linear; O(k) (k being length of shorter string) and search time logarithmic; O(log(n)) (n being total number of records) and maintains ambiguity in spellings. The logarithmic search time was made possible by introduction of B-tree based index. The benefits realized in terms of insights generated along with preserving privacy outweigh cost of increased storage requirements for storing phonetic representations as well as their cipher texts.

6. REFERENCES

[1] Vision areas of Digital India Initiative available at https://www.digitalindia.gov.in/content/vision-and-vision-areas

[2] Judgment of Justice K. S. Puttaswamy (Retd.). and ANR by Hon'ble Supreme Court available at https://main.sci.gov.in/supremecourt/2012/35071/35071_2012_Judgement_24-Aug-2017.pdf .

[3] Neha Mishra; Data Governance and Digital Trade in India: Losing Sight of the Forest for the Trees retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3835497

[4] Krishna Prakash Kalyanathaya, Dr.D. Akila, Dr.G. Suseendren "A Fuzzy Approach to Approximate String Matching for Text Retrieval in NLP" in Journal of Computational Information Systems 15:3(2019) Page 26-32

[5] Ako Muhamad Abdullah "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data" published on June 16, 2017 in Cryptography and Network Security

[6] Justin Zobel, Philip Dart, Phonetic String Matching: Lessons from Information Retrieval

[7] Philips, Lawrence. (2000). The Double Metaphone Search Algorithm. C/C++ Users Journal. 18. 38-43.

[8] E. Manogar and S. Abirami, "A study on data deduplication techniques for optimized storage," 2014 Sixth International Conference on Advanced Computing (ICoAC), 2014, pp. 161-166, doi: 10.1109/ICoAC.2014.7229702.

[9] Ankita Pilani and G. Mayil Muthu Kumaran, "Comparative Study of Name Matching Algorithms," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)

[10] Sneha A. Mittal and Dr. C. Rama Krishna, "Privacy Preserving Synonym Based Fuzzy Multi-keyword Search Over Encrypted Cloud Data," 2016 International Conference on Computing, Communication and Automation (ICCCA2016)

[11] GUOXIU LIU et al., "FSSE: An Effective Fuzzy Semantic Searchable Encryption Scheme Over Encrypted Cloud Data," 2020 IEEE Access, Digital Object Identifier 10.1109/ACCESS.2020.2966367

[12] Ristad, Eric & Yianilos, Peter. (1998). Learning String Edit Distance. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 20. 522 - 532. 10.1109/34.682181.

[13] Seema Sultana and Sunanda Dixit, "Indexes in PostgreSQL," 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2017)

[14] Wei Lu et al., "Efficiently Supporting Edit Distance Based String Similarity Search Using B+-Trees," 2014 IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 12