SURVEYING DATA SANITIZATION TECHNIQUE FOR DESINING FILTERING AND ENHANCED MEDICAL SUPPORT SYSTEM USING CLOUD ARCHITECTURE DIAGRAM

P.Saravanan[1], Dr. W. R. Salem Jeyaseelan[2], Dr. M.Ramesh Kumar[3], R. Krishnakumar[4]

[1,] Assistant Professor, Department of Computer Science and Engineering,

[2,] Associate Professor, Department of Information Technology,

[3,] Associate Professor, Department of Computer Science and Engineering

[4,] Assistant Professor, Department of Computer Science and Engineering,

[1,3,4] VSB College of Engineering Technical Campus, Coimbatore, Tamilnadu, India.

[2,] Karpagam College of Engineering (Autonomous), Coimbatore, Tamilnadu, India.

E-mail: psaravananmecse@gmail.com, salemjeyaseelan@kce.ac.in, maestro.ramesh@gmail.com, krishnakumarmecse@gmail.com

Abstract—The emergence of the Cloud has represented a fundamental change in the way information technology services are designed and deployed in business and governments and there is a growing trend of using cloud environments for storage and data processing needs. However, this environment represents a serious threat for data privacy, since document containing confidential information might be made available for unauthorized parties. Although measures to automatically remove or hide sensitive information that may disclose identities of referred entities or reveal their confidential data of publicly available document have been purposed. But there is no big implementation regarding security using sanitization method was implemented for data stored in cloud server. We are developing this Project for Medical Purpose. Here we use the Cloud Server as a main Server, where all the Data from the Users are Stored. We design this system using Registered Doctors, Paid and unpaid users. Here document is sanitizing dynamically so that different users get different view of same document. Data Sanitization is achieved by Three Process. 1. Entity Generalization-Preserving the Privacy data with its semantics. 2. Entity Swapping is used to reduce the Document Size. 3. Noise Addition: an entity substituted by another similar one extracted from another repository.

Index Terms—Data Sanitization, Masking Data, Substitution, Shuffling Records,

1. INTRODUCTION

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.

Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

2. TECHNIQUE: NULL'ING OUT

Simply deleting a column of data by replacing it with NULL values is an effective way of ensuring that it is not inappropriately visible in test environments. Unfortunately it is also one of the least desirable options from a test database standpoint. Usually the test teams need to work on the data or at least a realistic approximation of it. For example, it is very hard to write and test customer account maintenance forms if the customer name, address and contact details are all NULL values.

2.1 TECHNIQUE: MASKING DATA

Masking data means replacing certain fields with a Mask character (such as an X). This effectively disguises the data content while preserving the same formatting on front end screens and reports.

For example, a column of credit card numbers might look like:

4346 6454 0020 5379

4493 9238 7315 5787

4297 8296 7496 8724

And after the masking operation the information would appear as:

4346 XXXX XXXX 5379

4493 XXXX XXXX 5787

4297 XXXX XXXX 8724

The masking characters effectively remove much of the sensitive content from the record while still preserving the look and feel. Take care to ensure that enough of the data is masked to preserve security. It would not be hard to regenerate the original credit card number from a masking operation such as: 4297 8296 7496 87XX since the numbers are generated with a specific and well known checksum algorithm. Also care must be taken not to mask out potentially required information. A masking operation such as XXXX XXXXXXXX 5379 would strip the card issuer details from the credit card number. This may, or may not, be desirable.

2.2 TECHNIQUE: SUBSTITUTION

This technique consists of randomly replacing the contents of a column of data with information that looks similar but is completely unrelated to the real details. For example, the surnames in a customer database could be sanitized by replacing the real

last names with surnames drawn from a largish random list. Substitution is very effective in terms of preserving the look and feel of the existing data. The downside is that a largish store of substitutable information must be maintained for each column to be substituted. For example, to sanitize surnames by Substitution, a list of random last names must be available. Then to sanitize telephone numbers, a list of phone numbers must be available. Frequently, the ability to generate known invalid data (phone numbers that will never work) is a nice-to-have Feature. Substitution data can sometimes be very hard to find in large quantities. For example, if a million random street addresses are required, then just obtaining the substitution data can be a major exercise in itself. Substitution is quite powerful, reasonably fast and preserves the look and feel of the data. Finding the required random data to substitute and developing the procedures to accomplish the substitution can be a major effort.

2.3 TECHNIQUE: SHUFFLING RECORDS

Shuffling is similar to substitution except that the substitution data is derived from the column itself. Essentially the data in a column is randomly moved between rows until there is no longer any reasonable correlation with the remaining information in the row.

2.4 TECHNIQUE: NUMBER VARIANCE

The Number Variance technique is useful on numeric data. Simply put, the algorithm involves modifying each number value in a column by some random percentage of its real value. This technique has the nice advantage of providing a reasonable disguise for the numeric data while still keeping the range and distribution of values in the column within viable limits.

2.5 TECHNIQUE: GIBBERISH GENERATION

In general, when sanitizing data, one must take great care to remove all imbedded references to the real data. For example, it is pointless to carefully remove real customer names and addresses while still leaving intact in stored copies of correspondence in another table. This is especially true if the original record can be determined via a simple join on a unique key.

2.6 TECHNIQUE: ENCRYPTION/DECRYPTION

This would seem to be a very good option – yet, as with all techniques, it has its strengths and weaknesses. The big plus is that the real data is available to anybody with the key – for example administration personnel might be able to see the personal details on their front end screens but no one else would have this capability. This “optional” visibility is also this techniques biggest weakness. The encryption password only needs to escape once and all of the data is compromised. Of course, you can change the key and regenerate the test instances – but stored or saved copies of the data are immediately available under the old password. Encryption also destroys the formatting and look and feel of the data. Encrypted data rarely looks meaningful, in fact, it usually looks like binary data. This sometimes leads to NLS character set issues when manipulating encrypted varchar fields. Certain types of encryption impose

constraints on the data format as well. For example, the Oracle Obfuscation toolkit requires that all data to be encrypted should have a length which is a multiple of 8 characters. In effect, this means that the fields must be extended with a suitable padding character which must then be stripped off at decryption time. The strength of the encryption is also an issue. Some encryption is more secure than others. According to the experts, most encryption systems can be broken – it is just a matter of time and effort. In other words, not very much will keep the national security agencies of largish countries from reading your files should they choose to do so. This may not be a big worry if the requirement is to protect proprietary business information. The security is dependent on the strength of the encryption used. It may not be suitable for high security requirements or where the encryption key cannot be secured. Encryption also destroys the look and feel of the sanitized data. The big plus is the selective access it presents.

2.7 TECHNIQUE: NUMBER VARIANCE

The Number Variance technique is useful on numeric data. Simply put, the algorithm involves modifying each number value in a column by some random percentage of its real value. This technique has the nice advantage of providing a reasonable disguise for the numeric data while still keeping the range and distribution of values in the column within viable limits. For example, a column of sales data might have a random variance of 10% placed on it. Some values would be higher, some lower but all would be not too far from their original range. Verdict: The number variance technique is occasionally useful and can prevent attempts to correlate true records using known numeric data. This type of Data Sanitization really does need to be used in conjunction with other options though.

3. ATTACK

3.1.SQL Injection Attack

SQL Injection Attackers include user data in the SQLquery and arbitrary code is added in such a way that apart of the input is understood as SQL code.

There are two techniques of SQLIA i.e. access through input fields and access through URL. In first technique attacker always bypass the authentication of user and password. Attacker can perform this technique through multiple queries, extended stored procedure and ‘or ‘condition etc. In second technique attacker manipulates the query string in URL. This vulnerability can be represented as:

SELECT * FROM admin WHERE username ='admin123' AND password = 'secret'

3.2. Tautology Attacks.

The tautology attack always uses conditional statements to inject the code. This attack bypasses the authentication of the web page and extracts the important data.

SELECT accounts FROM users WHERE

Login ='nil' OR 1=1---AND password = 'nil'

3.3. UNION Attacks

This technique is combine two queries using UNION keyword. First query is original and second is injected query.

SELECT accounts From user

WHERE login=' 'UNIONSELECT credit card

WHERE accno=02220 – – AND Password=' '

3.4. Piggybacked Query

In this type of attack additional query will be added along with original query. The additional query is a injected query which is also called piggy-back onto the original query

SELECT accounts FROM user WHERE login='

smit' AND pass=''; drop table user – – 'AND pin=231

3. 5. Logical Incorrect Query Attack.

This type of attack gathers important information about the type and structure of the back end database in the web application. When the attacker uses this logical incorrect query, the application server displays error page, which can serve to expose sensitive information about the databases to the attacker.

SELECT accounts FROM users WHERE login=''

AND pass='' AND pin= convert (int,(select

top 1 name from sys objects where xtype='u'))

3.6. Blind Injection. In this technique, attacker asks true and false question to the server through the web page. If the injected statement evaluates to the true, the web site works normal.

3.7. Logical Incorrect Query Attack. This type of attack gathers important information about the type and structure of the back end database in the web application

SELECT accounts FROM users WHERE login=''

AND pass='' AND pin= convert (int,(select

top 1 name from sys objects where xtype='u'))

The select query extracts the user table and then converts this table name into an integer.

4. SANITIZATION METHODS

Data sanitization is context sensitive which means that the data can be generalized or perturbed in several different ways, the choice of which depends on the context. Two primary techniques are used. Generalization replaces a value with a range of possible values that the attribute may assume. For example, replacing a birth date with the birth year

replaces the actual value (a date) with a range of values (365 or 366 possible dates). Deletion is a form of generalization, because then the attribute could be any legal value. Perturbation retains a single value, but transforms it in some way. For example, adding a random value to a datum perturbs it. When this is done, the sanitizer must be sure that the results of the analysis of the perturbed data match those of the raw data. K-anonymity, a widely-used method of sanitizing data, generalizes information so that the generalization is valid for at least k entities. Several variants of k-anonymity have been proposed to overcome specific problems. For example, one study extended the model to limit the confidence of inferring a sensitive value . Another proposed a technique to achieve k-anonymity not just in one dataset, but over many datasets, by applying k-anonymity to the record owner level rather than the record level over the join of all the datasets. l-diversity is a variant of k-anonymity in which every group of QIDs must have some number of distinct values for the sensitive attribute [60]. Other variants abound [54, 56, 59, 70, 96], as do other generalization techniques [10, 12, 41, 47, 91]. Perturbation techniques change the data to achieve anonymity. One such method of achieving this is by masking, which if done appropriately can enable analysis to achieve results similar to those of the analysis on the raw data. An
Example is adding noise [44, 50, 66].

Define two levels of security for a sanitization process: (i) the *clearing* level; (ii) the *sanitization* or *purging* level. The clearing level states that a single overwrite of the affected areas is enough to protect against casual attacks and robust keyboard attacks. The purging level states that the devices have to be either Degaussed or destroyed to protect against laboratory attacks.

There are some customers (or data) that only require the clearing level and some that require the purging level. The sanitization process we came up with complies with both levels with one common mechanism. The basic idea is to overwrite data to handle the clearing level. The pat- tern used for the overwrites can be zeros, random pattern or any user-specified pattern. We have chosen to use zeros. If the purging level is required, we first perform the clearing level which compacts the clean data and allows the clean data to be efficiently migrated to another box by replicating clean post-reduplication data rather than pre-reduplication data.

Algorithm 1*Sanitization for a read-only file system*

(1) Merge phase

Setup a marker for the last container to be processed; Create a consistency point, say CP0, of the file system. A consistency point is an in-memory snapshot;

Flush the in-memory fingerprint index buffer and merge it with the on-disk index;

(1) Analysis phase

Traverse the on-disk index for all fingerprints; Build *PH* vec for all fingerprints found;

Record the range of containers covered by *PH* vec;

(2) Enumeration phase

Traverse all the files in CP0 (i.e., entire file system); Mark all fingerprints found as live in *PH* vec;

(3) Copy phase

Select containers with at least one dead chunk; Copy all live chunks from the selected containers into new containers; Delete the selected containers;

(4) Zero phase

Zero out the free blocks;

Zero out contaminated areas (NVRAM, Swap, etc.);

5. UNDOING THE SANITIZATION

An analysis of the literature in desensitization reveals four properties that adversaries depend on. First, external information, when correlated with the sanitized data, enables the adversary to determine the sensitive data. This is by far the most widely-publicized technique of de sanitization. It gained prominence in the AOL release of data in 2006,5 in which New York Times reporters were able to correlate contents of search queries to public records, and from that determine the identity of the anonymized queries with pseudonym 4417749 . An interesting aspect of this result was the analysis of the search queries. They were often about medical conditions such as hand tremors, bipolar disorders, and nicotine effects on the body none of which were true of the user; i an interview, she said that she often helped friends research their medical questions and conditions on line. This is an example of the need to prevent unwanted inferences from being drawn; these inferences could result in correct or incorrect deductions. More recently, Narayanan and Shmatikov attacked the Netix Prize Dataset containing data for anonymized users. The data associated with each user is a set of pairs of movie titles and ratings, and of ratings and rating times. The researchers correlated these pairs with data from the Internet Movie Database6 (IMDB), a public database in which viewers can rate movies. They were able to match Netix data with the IMDB data, and associate IMDB identities with Netix sanitized identities. Netix claims that the associations are invalid because they perturbed the data [80]; however, that correlations could be made illustrates how effective the use of external data can be. Other papers discuss techniques for performing these correlations [24, 34, 35, 46].Second, patterns in raw data often reject similar patterns in sanitized data, so if the adversary knows those patterns, she can infer the sensitive data. Desanitization social networks uses this type of relationship.

6.SYSTEM ARCHITECTURE

The paid users and the doctor are register to access the cloud server where the information are stored. When the paid user request some information to the doctor, the sanitized information will be available to the users. The sanitization is performed by the following three sanitization techniques. These techniques hides the sensitive terms intends of removing them. So, it preserve the utility of the information. Moreover, it allows the user to configure the level of sanitization applied to the document being more flexible than methods based on a fixed sanitization policies

(1) **Entity Generalization:** entities can be generalized to achieve some degree of privacy while preserving some of their semantics.

(2) **Entity swapping:** entities of different documents of the same set or within the same document can be swapped depending on the concrete case

(3) **Entity noise addition:** an entity can be substituted by another similar one extracted from another repository.

The sanitization process consists of two tasks: (i) detection of identifiable information within text; and (ii) information hiding, in a way that the disclosure risk is minimized and, ideally, the utility of the sanitized text is maximized.

Here a single user can connect to multiple doctors. Paid users are only allowed to access the Doctor's Opinion/Suggestion/ Prescriptions. Registered Doctors can only Reply to the User's/ Patients. The unpaid users can view the document but cannot post any request to the doctors.

This system architecture provides more security to the data stored in the cloud server, by allowing only authorized entity to access the server.

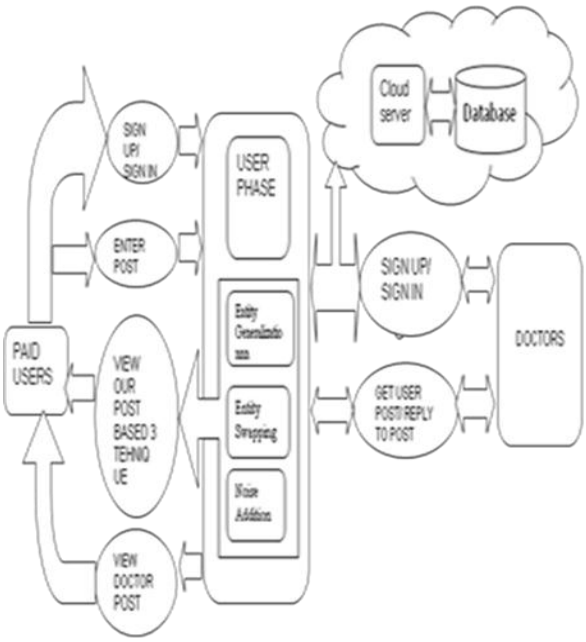

Fig 1.1 SYSTEM ARCHITECTURE

7.IMPLEMENTATION

The system implements new prototype version of re- verse proxy server which prevents the SQL Injection Attacks (SQLIA) and Cross Site Scripting (XSS) At- tacks. Java is selected for implementation as it is a plat- form independent language, also during the literature survey we have observed that all types of SQLIA and XSS attacks were successes on the web application which are built using Java. This system uses simple web

technologies like HTML, JSP (Java Server Pages), etc. This technique is fully automated. For user inputs we require a web application. Here we are using a banking application. As shown in Fig. 3, following three mod- ules are implemented.

- SQL Injection preventer
- Cross Site Scripting preventer
- Analysis Module

The user sends the input through the login page of web application. The data redirector program, which is installed on the server gateway, gets the user input and redirects the request to the reverse proxy server. At the same time the data redirector encrypts the request into XML format. The reverse proxy server logs the IP addresses of the computers from where the request has originated. The reverse proxy server has the SQL injection preventer module and cross site scripting preventer module installed in it. The SQL injection preventer module validates the request against SQLIA and tokenizes the request. Various signature checks are carried out on the user request, like comments, white spaces, Meta characters, etc. If the special characters are not found in the user request then it is passed on to the cross site scripting preventer module. Cross site scripting pre- venter module validates the request against XSS attacks by carrying out signature checks through the regular expression. This module prevents the forbidden tags and removes all unwanted and malicious code.

The analysis module checks attacker's activities. If the attacker attacks more than three times consecutively the IP address of the attacker gets blocked for three hour. Also the account holder gets an email notification and the account gets blocked for three hours.

The analysis module can prepare the following reports

(a) Attack's List,
(b) Blocked IP List,
(c) IP Based Analysis and
(d) Web Based Analysis.

In the analysis of Attack's List includes types of attacks, description of attacks, IP address of attackers, browser, URL, and timestamp. The blocked IP analysis includes IP address of attackers and number of attacks from a particular IP. In IP based analysis we observe User-ID, IP address, number of requests and time- stamps. The web based analysis displays the browser name and count of attack. Administrator's Console: This interface creates a table on the basis of analysis done by the analysis module. The table contains following attributes.

- Attacker ID.
- Attackers IP address which linked to view attack and IP base analysis module.
- The login name or password issued to perform the attack.

- Browser details that uses in attack.
- Timestamp when the attack detects.

8. RUNTIME ANALYSIS

We have evaluated effectiveness of our approach and compared with various other approaches.

Methodology	Change in source code	Detection /Mitigation of Attacks
SQLCheck [8]	Necessary	Partially Auto-mated
SQLRand [3]	Necessary	Fully Automated
AMNESIA [7]	Not Necessary	Fully Automated
SQLProb [6]	Not Necessary	Fully Automated
R_Proxy (Pro-posed)	Not Necessary	Fully Automated

Table 1Analysis of Methodologies Curbing SQLIA

Table 1 indicates comparison between various methodologies with respect to whether changes are require to the source code for intrusion prevention. We also show in the table that our system is fully automated as compared to other partially automated systems for detection of attacks. We also evaluated execution time of proposed system for varying number of user requests. Table 2 shows that the execution time required by the proposed. Fig.2 illustrates comparison between the execution time required by existing technique and the proposed technique.

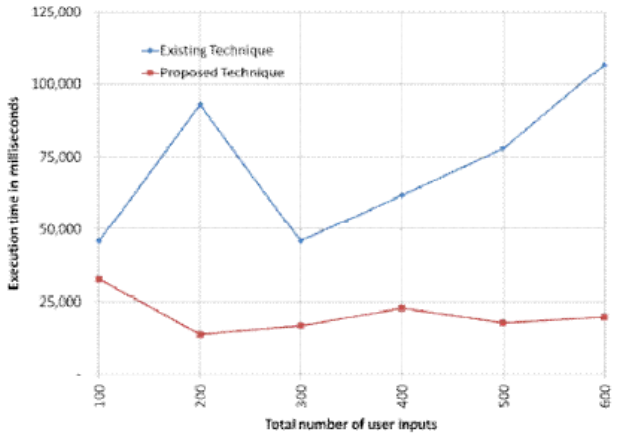

Fig 2Comparison of Time required for Execution

The time required for execution is plotted on 'X' axis in millisecond and total number of user request on 'Y' axis. It can be clearly seen that the proposed technique requires much less time for execution as compared to existing techniques. Proposed technique requires much less time for executions compared to existing techniques.

The time required for execution is plotted on 'X' axis in millisecond and total number of user request on 'Y' axis. It can be clearly seen that the proposed technique requires much less time for execution
as compared to existing techniques.

Total User Requests	Execution time in millisecond	
	Existing Technique	Proposed Technique
100	46000	33000
200	93000	14000
300	46000	17000
400	62000	23000
500	78000	18000
600	107000	20000

Table 2 Execution Time Comparison

9. CONCLUSION

Since cloud computing is the vast developing technology, security is the major in the cloud environment. To overcome this drawback many existing approaches has been introduced but they have not fulfilled the security issue. At this situation storing medical records in the cloud environment is the major issue. Because the data will be hacked (corrupted) modified by the unauthorized person in the network. So we need new mechanism need to be implemented in the cloud environment for storing and access the medical records stored in the cloud servers. By implementing this project we can allow the authorized doctors to enter into the network and respond to the queries submitted by the registered patient in the cloud network using sanitation mechanism so that we get more result about the query that the user is enter. In future we allow the registered doctors, nurse and pharmacist based on the attributed based encryption scheme to view the records so that we can increase the security level and we can also try to improve the way of detection of sensitive information.

REFERENCES

[1] L. Sweeney, "Replacing personally-identifying information in medical records, the scrub system," in Proc. 1996 American Medical Informatics Association Ann. Symp., 1996, pp. 333–337.

[2] L. Sweeney, Computational Disclosure Control: A primer on data privacy protection. Ph.D. Thesis, Massachusetts Institute of Technology, 2001.

[3] L. Sweeney, "K-anonymity: A model for protecting privacy," Int. J. Uncertainty, Fuzziness and Knowledge-Based Syst., vol. 10, no. 5, pp. 557–570, 2002.

[4] A. Tveit, O. Edsberg, T. B. Rost, A. Faxvaag, O. Nytro, M. T. Nordgard, M. T. Ranang, and A. Grimsmo, "Anonymization of general practioner medical records," in Proc. Second HelsIT Conf., Trondheim, Norway, 2004.

[5] Nat. Security Agency, RedactingWith Confidence: How to Safely Publish Sanitized Reports Converted From Word to pdf, Tech. Rep. I333-015R-2005, 2005.

[6] M. M. Douglass, G. D. Cliffford, A. Reisner, W. J. Long, G. B.Moody, and R. G. Mark, "De-identification algorithm for free-text nursing notes," Proc. Computers in Cardiology'05, pp. 331–334, 2005.

[7] D. A. Dorr, W. F. Phillips, S. Phansalkar, S. A. Sims, and J. F. Hurdle, "Assessing the difficulty and time cost of de-identification in clinical narratives," Methods Inf. Medicine, vol. 45, no. 3, pp. 246–252, 2006.

[8] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. Mohania, "Efficient techniques for document sanitization," in Proc. ACM Conf. Information and Knowledge Management'08, 2008, pp. 843–852.

[9] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore, "Automatic de-identification of textual documents in the electronic health record: A review of recent research," BMC Med. Res. Methodol., vol. 10, pp. 70–86, 2010.

[10] D. Sánchez, M. Batet, A. Valls, and K. Gibert, "Ontology-driven web-based semantic similarity," J. Intell. Inf. Syst., vol. 35, no. 3, pp.383–413, 2010.

[11] S. K. Dash, R. Mishra, D. P. Mishra, and A. Tripathy, "A privacy preserving repository for securing data across the cloud," in Proc. 3rd

[12] S. Marston, Z. Li, S. Bandyopadhyay, A. Ghalsasi, and J. Zhang, "Cloud computing the business perspective," Decision Support Syst., vol. 51, no. 1, pp. 176–189, 2011.

[13] D. Abril, G. Navarro-Arribas, and V. Torra, "On the declassification of confidential documents,"

[14] C. Cumby and R. Ghan, "A machine learning based system for semiautomatically redacting documents," in Proc. 23rd Innovative Applications of Artificial Intelligence Conf., 2011, pp. 1628–1635.

[15] National Security Agency, Redaction of pdf Files Using Adobe Acrobat Professional X 2011 [Online].Available: http://www.nsa.gov/ia/files/vtechrep/I73 025R_2011.pdf

[16] B. Anandan and C. Clifton, "Significance of term relationships on anonymization," in Proc. Web

[17] U.S. Department of Justice, U.S. Freedom of Information Act (FOIA) 2012 [Online]. Available: http://www.foia.gov/

[18] S. Martínez, D. Sánchez, A. Valls, and M. Batet, "Privacy protection of textual attributes through a semantic-based masking method," Inf. Fusion, vol. 13, no. 4, pp. 304–314, 2012.

[19] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in Proc. 2012 Int. Conf. Computer Science and Electronics Engineering, 2012, pp. 647–651

[20]B. Anandan, C. Clifton, W. Jiang, M. Murugesan, P. Pastrana-Camacho, and L. Si, "t-plausibility: Generalizing words to desensitize text," Trans. Data Privacy, vol. 5, pp. 505–534, 2012.