A Survey on Recent Trends, Process and Development in Data Masking for Testing

Ravikumar G K[1] ,Manjunath T N[2], Ravindra S Hegadi[3],Umesh I M[4]

[1]Wipro Technologies, Bangalore, Karnataka, INDIA

[2]Bharathiar University, Coimbatore, Tamilnadu, INDIA

[3]Karnataka University, Dharwad, Karnataka, INDIA

[4]R V College of Engineering,Bangalore, Karnataka,INDIA

Abstract

In today's information age, the data is an important asset of the organization so the security of the data is a vital role in the industry. In order to achieve the above aspect, the data masking is used. The data masking is mainly the data is replaced with realistic but not the original data. The main objective is to make sensitive information is not made available outside of the environment. The data masking is to just provide the copy of the production data in support of the development environment and thus it controls the leakage of the data. Data masking are designed to be repeatable so referential integrity is maintained. We have explored the data masking architecture, techniques with realistic data and order of masking. We hope this paper will help quality analyst in pulling data from production environments into test environment for quality analysis by safe guarding the original information.

Keywords: Data masking, Encryption, Orders of masking.

1. Introduction

Common business applications require constant patch and upgrade cycles and require that 6-8 copies of the application and data be made for testing. While organizations typically have strict controls on production systems, data security in non-production instances is often left up to trusting the employee, with potentially disastrous results [21]. Creating test and development copies in an automated process reduces the exposure of sensitive data. Database layout often changes, it is useful to maintain a list of sensitive columns without rewriting application code. Data masking is an effective strategy in reducing the risk of data exposure from inside and outside of an organization and should be considered a best practice for curing non-production databases [2].Data masking is the process of de-identifying or obscuring specific data within a specific data elements within database table or column. In other words data masking is the replacement of existing sensitive information in test or development databases with information that looks real but is of no use to anyone who might wish to misuse it. In general, the users of the test, development or training databases do not need to see the actual information as long as what they are looking at looks real and is consistent [2].Effective data masking requires data to be altered in a way that the actual values cannot be determined or reengineered, functional appearance is maintained, so effective testing is possible. Data can be encrypted and decrypted, relational integrity is maintained, security polices can be established and separation of duties between security and administration established. Common methods of data masking includes: encryption or decryption, shuffling, masking (i.e. numbers letters), substitution (i.e. All female names = Ragini), nulling (####) or shuffling (zip code12435 = 53421) [11].

2. Data Masking Architectures

Basically, there are two types of architectures which are used in the design of data masking software:

a.On the Fly Server-To-Server Data masking architectures.
b.In-Situ Data Masking Architectures.

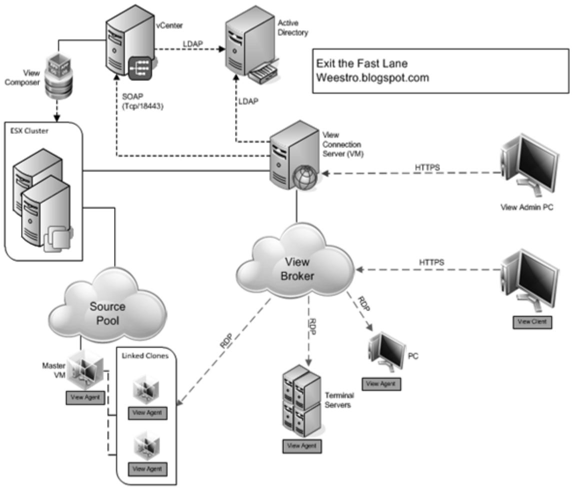


Fig-1: on the Fly, Server-To-Server, data masking architectures

2.1 On the Fly, Server-To-Server, Data Masking Architectures.

Fig-1 shows on the fly, server-to-server data masking architecture, here the data does not exist in the target database prior to masking. The anonymization rules are applied as part of the process of moving the data from the source to the target. Often this type of masking is integrated into the cloning process which creates the target database [19].This architecture is advantage because the data is never present in an unmasked form in the target database. This is not advised for some applications because any errors in the process necessarily interrupt the transfer of the data. The ability to mask data after the transfer has completed can be troublesome. This might happen in cases where the masked target database has been built and it is subsequently decided that a specific column of information really needs to be masked.

2.2 In-Situ Data Masking Architectures

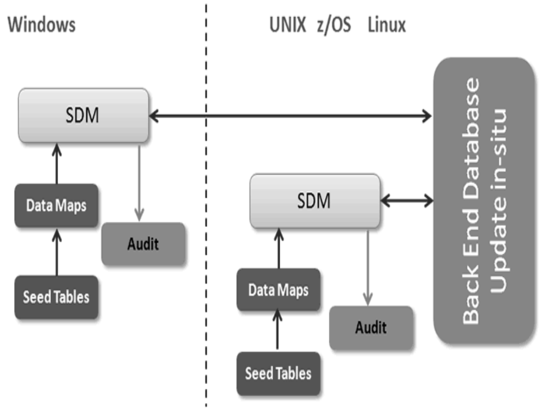


Fig-2: In-Situ data masking architectures

Fig-2 shows in-suite data masking architecture, here the clone of the database to be masked is created by other means and the software simply operates on the cloned database. There are two types of in-situ masking: masking rules which are executed and controlled as a standalone entity on the target and data masking rules which are controlled by a different system which then connects to the target and controls the execution of the rules [21].This architecture is advantage because it is possible to apply additional masking operations at any time. The masking operations are separate from the copy process so existing cloning solutions can be used and the data masking rules are possibly simpler to maintain and not advised for some applications because the data is present in an unmasked state in the target database and hence additional security measures will be required during that time.

3. Different types of Data Masking Techniques

Currently many data masking types are available the following are the important data masking techniques [22].

3.1 Substitution

The Substitution technique replaces the existing data with random values from a pre-prepared dataset i.e 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 available 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 [22]. Frequently, the ability to generate known invalid data (credit card numbers that will pass the checksum tests but never work) is a nice-to-have feature. Substitution data can sometimes be very hard to find in large quantities - however any data masking software should contain datasets of commonly required items. When evaluating data masking software the size, scope and variety of the datasets should be considered. Another useful feature to look for is the ability to build your own custom datasets and add them for use in the masking rules [22].

3.2 Shuffling

The Shuffling technique uses the existing data as its own substitution dataset and moves the values between rows in such a way that the no values are present in their original rows i.e 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 [22].There is a certain danger in the shuffling technique. It does not prevent people from asking questions like “I wonder if so-and-so is on the supplier list?” In other words, the original data is still present and sometimes meaningful questions can still be asked of it. Another consideration is the algorithm used to shuffle the data. If the shuffling. Method can be determined and then the data can be easily “un-shuffled”. For example, if the shuffle algorithm simply ran down the table swapping the column data in between every group of two rows it would not take much work from an interested party to revert things to their un-shuffled state [21].

Shuffling is rarely effective when used on small amounts of data. For example, if there are only 5 rows in a table it probably will not be too difficult to figure out which of the shuffled data really belongs to which row. On the other hand, if a column of numeric data is shuffled, the sum and average of the column still work out to the same amount. This can sometimes be useful. Shuffle rules are best used on large tables and leave the look and feel of the data intact. They are fast, but great care must be taken to use a sophisticated algorithm to randomize the shuffling of the rows.

3.3 Number and Date Variance

The Number and Date Variance technique varies the existing values in a specified range in order to obfuscate them. For example, birth date values could be changed within a range of +/- 60 days. The Number Variance technique is useful on numeric or date data. Simply put, the algorithm involves modifying each number or date 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 data while still keeping the range and distribution of values in the column to within existing limits. For example, a column of salary details 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. Date fields are also a good candidate for variance techniques. Birth dates, for example, could be varied with in an arbitrary range of ± 120 days which effectively disguises the personally identifiable information while still preserving the distribution. The variance technique can prevent attempts to discover true records using known date data or the exposure of sensitive numeric or date data [21].

3.4 Encryption

The Encryption technique algorithmically scrambles the data. This usually does not leave the data looking realistic and can sometimes make the data larger.

This technique offers the option of leaving the data in place and visible to those with the appropriate key while remaining effectively useless to anybody without the key. This would seem to be a very good option yet, for anonymous test databases, it is one of the least useful techniques [22].The advantages of having the real data available to anybody with the key is actually a major disadvantage in a test or development database. The “optional” visibility provides no major advantage in a test system and 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 outsourced, stored or saved copies of the data are all still 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 character set issues when manipulating encrypted varchar fields. Certain types of encryption impose constraints on the data format as well. In effect, this means that the fields must be extended with a suitable padding character which must then be stripped off at decryption time [11].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 largest 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. Never, ever, use a simplistic encryption scheme designed by amateurs. For example, one in which the letter ‘A’ is replaced by ‘X’ and the letter ‘B’ by ‘M’etc. is trivially easy to decrypt based on letter frequency probabilities. In fact, first year computer science students are often asked to write such programs as assignments [21].

3.5 Nulling Out or Deletion

The Nulling Out technique simply removes the sensitive data by deleting it 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. NULL’ing or truncating data is useful in circumstances where the data is simply not required, but is rarely useful as the entire data sanitization strategy [22].

3.6 Masking Out

The Masking Out technique sanitizes the data by replacing certain specified characters with mask characters. For example, a credit card number might be masked as 4929 1344 XXXX XXXX.

Masking data, besides being the generic term for the process of data anonymization, means replacing certain fields with a mask character (such as an X). This effectively disguises the data content while preserving the same omitting on front end screens and reports. For example, a column of credit card numbers might look like below [21].

4346 6454 0020 5379
4493 9238 7315 5787
4297 8296 7496 8724

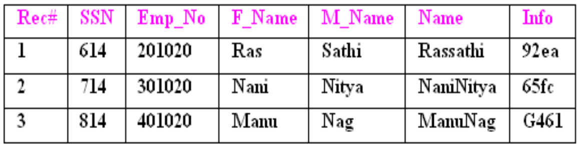
after the masking operation the information would appear as:

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 [21].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 XXXX XXXX 5379 would strip the card issuer details from the credit card number. This may, or may not, be desirable [7]. If the data is in a specific, invariable format, then masking out is a powerful and fast option. If numerous special cases must be dealt with then masking can be slow, extremely complex to administer and can potentially leave some data items in appropriately masked [21].

3.7 Row-Internal Synchronization

If a row in a database table contains denormalized data derived from other columns in that row and those columns are masked then the denormalized data row will need to be rebuilt. This technique is called Row-Internal Synchronization. Data is rarely consistently available in a fully normalized format. Consider the example below in which the NAME field is composed of data from other columns in the same row.



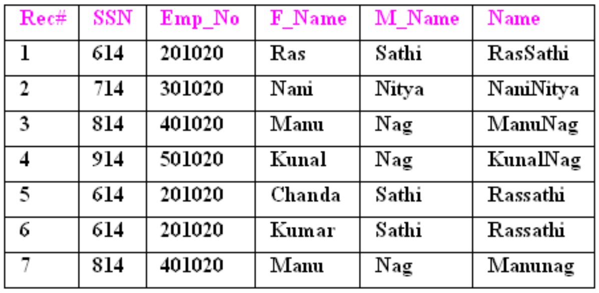
Rec#	SSN	Emp_No	F_Name	M_Name	Name	Info
1	614	201020	Ras	Sathi	Rassathi	92ea
2	714	301020	Nani	Nitya	NaniNitya	65fc
3	814	401020	Manu	Nag	ManuNag	G461

Table-1: Row Internal Synchronization

After the data has been masked, the F_NAME and M_NAME columns will have been changed to other values. For the information to be secure, clearly the NAME field must also change. However, it must change to contain values synchronized with the rest of the data in the row so that the masked data reflects the denormalized structure of the row. This type of synchronization is called Row-Internal Synchronization and it is quite distinct from the other two types: Table-Internal and Table-To-Table Synchronization [19].The Row-Internal Synchronization technique updates a field in a row with a combination of values from the same row. This means that if, after masking, the F_NAME and M_NAME change to Rassathi [12].

3.8 Table-Internal Synchronization

If a database table contains multiple rows containing identical columns and those columns are masked then the denormalized data rows will need to be set to the same value. This technique is called Table-Internal Synchronization. Sometimes the same data appears in multiple rows within the same table. In the example below, the name Robert Smith appears in the F_NAME and NAME columns in multiple rows [21].



Rec#	SSN	Emp_No	F_Name	M_Name	Name
1	614	201020	Ras	Sathi	RasSathi
2	714	301020	Nani	Nitya	NaniNitya
3	814	401020	Manu	Nag	ManuNag
4	914	501020	Kunal	Nag	KunalNag
5	614	201020	Chanda	Sathi	Rassathi
6	614	201020	Kumar	Sathi	Rassathi
7	814	401020	Manu	Nag	Manunag

Table-2: Table-Internal Synchronization

In other words, some of the data items are denormalized because of repetitions in multiple rows. If the name Ras Sathi changes to Ram Kumar after masking, then the same Rass Sathi referenced in other rows must also hange to Ram kumar in a consistent manner. This requirement is necessary to preserve the relationships between the data rows and is called Table-Internal Synchronization [18].A Table-Internal Synchronization operation will update columns in groups of rows within a table to contain identical values. This means that every occurrence of Ras Sathi in the table will contain A Ram Kumar. Good data anonymization software should provide support for this requirement [21].

3.9 Table-To-Table Synchronization

If two tables contain the columns with the same denormalized data values and those columns are masked in one table then the second table will need to be updated with the changes. This technique is called Table-To-Table Synchronization. It is often the case in practical information systems that identical information is distributed among multiple tables in a denormalized format. For example, an employee name may be held in several tables. It is desirable (often essential) that if the name is masked in one column then the other tables in which the information is held should also be updated with an identical value. This requirement is called Table-To-Table Synchronization and is the type of synchronization most people think of when considering a data anonymization process. It should be noted that there are also two other types of synchronization: Row-Internal and Table-Internal, Table-1 and Table-2 and all two have quite different purposes and functions [13].Table-To-Table Synchronization operations are designed to correlate any data changes made to a table column with columns in other tables. Thus performing Table-To-Table Synchronization operations requires the knowledge of a join condition between the two tables. The join condition is used to ensure the appropriate rows are updated correctly. Also required, is knowledge of the columns in each of the source and target tables which must be synchronized. As an example, consider the two tables as shown below.

TABLE SOURCE
COLUMN ID_NUM number (10)
COLUMN NAME varchar (40)

TABLE TARGET
COLUMN ID_NUM number (10)
COLUMN NAME varchar (40)

Assume the NAME column must be synchronized between the two tables and the join condition is SOURCE.ID_NUM=TARGET.ID_NUM. If a Substitution operation is performed on the SOURCE.NAME column and a subsequent Table-To-Table Synchronization operation is applied, then each TARGET.NAME column will receive values identical to the ones in the SOURCE table where the join condition matches. It is important to realize that if there are ID_NUM values in the TARGET table that are not present in the SOURCE table, the Table-To-Table Synchronization rule will have no effect on those rows. This issue is called a Table-To-Table Skip and is discussed in detail in the Data Masking Issues section.Table-To-Table Synchronization is probably the most common synchronization task, in fact, it is rare that a data scrambling process does not require it. Every data masking solution should provide support for this operation [21].

3.10 Multiple Data Masking Techniques and Algorithms

Non-deterministic randomization replaces a sensitive field with a randomly generated value subject to various constraints Blurring adds a random variance to the original value Repeatable masking maintains referential integrity by generating values that are both repeatable and unique Substitution randomly replaces original values with false but realistic-looking values .Built-in methods maintain repeatability and preserve referential integrity across multiple tables Fine-grained controls produce randomized output while also preserving the original data properties, such as width, format, and range.

3.11 Specialized, Built-In Data Masking Rules and Content

Easily substitute name, address, and company data Leverage prepackaged rules for special fields such as Social Security, credit card, and telephone numbers Use sample mappings for common data masking scenarios. Enable outsourcing and off shoring by ensuring all private data is masked according to security policies before sending to outsourced partners or contractors [21].

4. Orders of Masking

It is noticed that not all data masking operations have the same effectiveness, nor do all data masking operations seek to plug the same weaknesses in the data. For this reason, we will talk about the *order* of a data masking operations [19].

i.First order: This relates exclusively to the content of the data. Masking of fields such as names, addresses, social security numbers, dates of birth, telephone contact details, and so forth fall under this category. Usually we consider first order masks to be the atmost minimum that any masking solution should implement, and these typically require the least amount of investment. [19]

Production Data			Test Data		
Ramu	Neelam	20	Nitin	Suma	23
Prasanna	Vinay	40	Sudha	Ravi	44
Laksh	Kash	60	Mahesh	Manju	36

Table-3: First order masking

The need for these masks is usually the easiest to identify, as direct uses of sensitive data are usually well-known within the project. Projects that use more data of this type (e.g. HCM, CRM, healthcare) will take longer to mask it than projects that do not (e.g. ERP, supply chain management, project management) as there is more work to be done, but in both cases progress happens at a fairly constant rate.

ii.Second order: This involves a combination of the content and the structure of the data. These are fields that are sensitive in their own right, but do not come under first order masking because they are fields which are used multiple times across the dataset (usually in different tables). Account numbers can be considered the canonical example of this, they appear more than once (say, in ‘credit card’ and ‘orders’ tables) and are related via a foreign key constraint. It goes without saying that these require more substantial effort to achieve. As with first order masks, the need for these masks is again fairly easy to identify, but masking them is more time-consuming as the changes are more widespread, making referential integrity harder to maintain. Systems that are more structurally complex take longer to mask as there are more changes to be made.

iii.*Third order:* these are purely structural masks and relate to how the data is related both between tables and within tables. The primary metric here is the invariance of a query with respect to outliers, trends, aggregate queries, and so forth. Fields that usually fall under this category are currency amounts (be they order amounts, salaries etc.), product order quantities, and any other field that can give rise to any trend or can be subjected to an aggregate query. It is also important to note that these kinds of masks are hardly straightforward and do incur a steep computational penalty, but are required to maximize the security of the data and minimize liability [21].

This third order of masking is most commonly overlooked by users of data masking techniques, partly because it is difficult to identify which correlations can yield valid data, and partly because they fail to acknowledge that data masking is about more than regulatory compliance: many of the trends and statistics available through correlation are of serious commercial value, and could be very damaging if obtained by one’s competitors.

iv.*Fourth order:* These masks are also purely structural, but rather than dealing with interrelationships between individual rows of data, they address information that can be obtained from the database schema itself. This entails using things like the data types of fields, foreign-key relationships between tables, or the very fact that a given field exists, to answer queries about the business’s operating conditions and expectations. For example, the fact that the dataset includes a table for recording which products a customer browsed before making a purchase demonstrates that the company is gathering such data, most likely because they have a strategy of using such data to improve customer targeting [20].

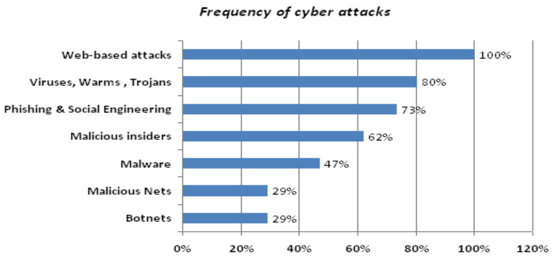
The need for these masks is often difficult to identify, as it is difficult to predict which aspects of the business’s operating conditions are sensitive; most of the operating conditions for a business are common to any business within that field, or can be easily determined through other means, so masking would be pointless. Implementing masks at this level also has severe implications for the verisimilitude of testing; while the actual data content of the dataset will be highly variable and can thus be masked or synthesized without invalidating the coverage given by the tests, the schema usually does not vary at all, such that it is impossible to create a ‘test schema’ that masks sensitive information whilst still being a potential image of a production database.

v.*Fifth order:* These masks deal with meta-information, such as the RDBMS used to store data within the company, or the testing methodology being employed. For instance, knowing which RDBMS is used to store the test data is usually a good indicator of which RDBMS is used to store production data, as the tests would commonly seek to replicate as much of the RDBMS configuration as possible. This could be useful information to anyone seeking to illegally access the production data, by exploiting security flaws in the RDBMS [14].

In practice, masking at this level is never performed because the information is not sufficiently sensitive; the majority of companies have policies in place that mitigate the risk posed, such as a rigorous maintenance policy to keep the RDBMS updated with the latest security patches.

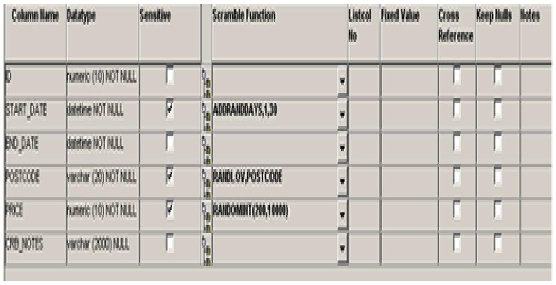
5. Results

Below graph-1 shows the benchmark sample of 45 organizations experiences 50 discernible and successful cyber attacks per week, which translate to more than one successful attack per company per week.



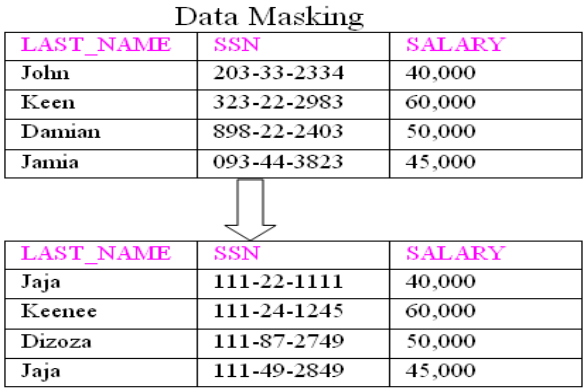
Graph-1: Frequency of cyber attacks experienced by benchmark sample

Below table-4 shows sample snapshot for data masking for sql server data.Table-5 shows the sample data masking for employee data in oracle database.



Column Name	Datatype	Sensitive	Scramble Function	Listcol No	Fixed Value	Cross Reference	Keep Nulls	Notes
ID	numeric (10) NOT NULL	□				□	□	
START_DATE	datetime NOT NULL	☑	ADDRANDDAYS,1,30			□	□	
END_DATE	datetime NOT NULL	□				□	□	
POSTCODE	varchar (20) NOT NULL	☑	RANDLOV,POSTCODE			□	□	
PRICE	numeric (10) NOT NULL	☑	RANDOMINT(200,1000)			□	□	
CRM_NOTES	varchar (2000) NULL	□				□	□	

Table 4-Data masking for sql server



Data Masking

LAST_NAME	SSN	SALARY
John	203-33-2334	40,000
Keen	323-22-2983	60,000
Damian	898-22-2403	50,000
Jamia	093-44-3823	45,000

LAST_NAME	SSN	SALARY
Jaja	111-22-1111	40,000
Keenee	111-24-1245	60,000
Dizoza	111-87-2749	50,000
Jaja	111-49-2849	45,000

Table 5-Sample data masking for employee data (Oracle DB)

6. Conclusions

This Survey explores the necessity of data masking in present information age, no researcher has consolidated the survey as data masking architecture and techniques for realistic situations, data masking will enable to accomplish the following:(a). Increases protection against data theft. (b). Enforces 'need to know access'. (c). Researchers in 2006 found that almost 80 to 90 percent of Fortune 500 companies and government agencies have experienced data theft. (d). Reduces restrictions on data use. (e). Provides realistic data for testing, development, training, outsourcing, data mining/research, etc. (f). Enables off-site and cross-border software development and data sharing. (g). Supports compliance with privacy legislation & policies. (h). Data masking demonstrates corporate due diligence regarding compliance with data privacy legislation. (i).Improves client confidence. (j). Provides a heightened sense of security to clients, employees, and suppliers.This paper will help in analyzing the level of security needed for real time applications when publishing it in QA environment.

7. Future scope

Security, privacy, and identity management will remain at the top of information security spending priorities the incidence of data breaches will continue to rise unless organizations enforce additional measures to protect sensitive data, both in production and non-production environments.

Acknowledgements

This paper is prepared through exhaustive discussions with Subject Matter Experts (SME), data masking groups, Data Quality Experts of various organizations in India and abroad through discussions. The authors gratefully acknowledge the time spend in this discussions provided by Mr.Shahzad, SME, CSC USA, Mr. Parswanath Project Manager (Data Warehouse Wing). Wipro

Technologies, India. Mr. Govardhan (Architect) IBM India Pvt Ltd, Mr. Arun Kumar Data Architect KPIT Cummins India,Mr.Raghavendra SME, iGATE Global solutions,India,Mrs.Radha Sarvana,Analyst,Wipro Technologies,India.

References

[1].Adam N. R., and Wortmann, J. C. 1989. “Security-Control Methods for Statistical Databases: A Comparative study,” ACM Computing Surveys (21:4), pp. 515 –556.

[2]. Manjunath T.N, Ravindra S Hegadi, Ravikumar G K."A Survey on Multimedia Data Mining and Its Relevance Today" IJCSNS. Vol. 10 No. 11 pp. 165-170.

[3]. Manjunath T.N, Ravindra S Hegadi, Ravikumar G K."Analysis of Data Quality Aspects in Datawarehouse Systems", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (1), 2010, 477-485

[4]. Dalenius, T., and Reiss, S. P. 1982. “Data Swapping: A Technique for Disclosure Control,” Journal of Statistical Planning and Inference (6:1), pp. 73-85.

[5]. Domingo-Ferrer J., and Mateo-Sanz, J. M. 2002. “Practical Data-Oriented Microaggregation for Statistical Disclosure Control,” IEEE Transactions on Knowledge and Data Engineering (14:1), pp. 189-201.

[6]. Epstein, R. A. 2002. “HIPAA on Privacy: Its Unintended and Intended Consequences,” Cato Journal (22:1), Spring/Summer, pp. 13-19.

[7]. Greengard, S.1996. “Privacy: Entitlement or Illusion?” Personnel Journal (75:5), pp. 74-88.

[8]. Kaelber, D., and Jha, A. 2008. “A Research Agenda for Personal Health Records (PHRs),” Journal of the American Medical Informatics Association (15:6), November / December.

[9]. KDnuggets, “Google Subpoena: Child Protection vs. Privacy,” Accessed July 2006, from Http://www.kdnuggets.com/polls/2006/google_subpoena.htm.

[10]. Liew, C. K., Choi, U. J., and Liew, C. J. 1985. “A Data Distortion by Probability Distribution,” ACM Transactions on Database Systems (10:3), pp. 395-411.

[11]. Xiao-Bai Li, Luvai Motiwalla BY “Protecting Patient Privacy with Data Masking” WISP 2009

[12]. Oracle White Paper—Data Masking Best Practices JULY 2010

[13]. Sachin Lodha BY Data Privacy - TRDDC Silver Jubilee Commemoration Publication - SL Comments.doc

[14]. Sachin Lodha, Ellora Praharaj and Ravishankar Rajamony. Selecting Quasi-Identifiers. Submitted to International Conference on Database Technology (ICDT),2007.

[15]. Sachin Lodha and Sharada Sundaram. Data Privacy. In Proceedings of 2nd World TCS Technical Architects' Conference (TACTiCS), 2005.

[16]. Krish Muralidhar AND Rathindra Sarathy: “Numerical Data Masking Techniques for Maintaining Sub-omainCharacteristics”

[17]. Burridge, J. 2003. Information preserving statistical obfuscation. Statistics and Computing, 13 321-327. [20]. Fuller, W.A. (1993). Masking procedures for microdata disclosure limitation. Journal of Official Statistics. 9, 383-406.

[18]. Muralidhar K., R. Parsa, R. Sarathy. 1999. A general additive data perturbation method for database security.Management Science, 45 1399-1415.

[19]. Data Masking: What You Need to Know What You Really Need To Know Before You Begin *A Net* 2000 Ltd. White Paper

[20]. Data Scrambling Issues *A Net 2000 Ltd. White Paper*.

[21]. Richard Fine and Llyr Jones, Grid-Tools Ltd BY the Mathematics of Data Masking.

[22]. AMBROSIA, V. G., BUECHEL, S. W., BRASS, J. A., and PETERSON, J. R., 1998, An integration of remote sensing, GIS, and information distribution.

Authors Profile

Ravikumar G.K. received his Bachelor's degree from SIT, Tumkur (Bangalore University) during the year 1996 and M. Tech in Systems Analysis and Computer Application from Karnataka Regional Engineering College Surthakal (NITK) during the year 2000. He is working as a Project Manager in Wipro Technologies, Bangalore for Data warehousing projects. He had worked with IGATE Global solutions Bangalore and also has worked with SJBIT as Prof and HOD of Dept of CSE and ISE having around 14 years of Professional experienced which includes Software Industry and teaching experience. His area of interests are Data Warehouse & Business Intelligence, multimedia and Databases. He has published and presented papers in journals, international and national level conferences.

Manjunath T.N. received his Bachelor's degree in computer Science and Engineering from SJCIT, Karnataka, India during the year 2001 and M. Tech in computer Science and Engineering from Jawaharlal Nehru National College of Engineering, Shimoga, Karnataka, India during the year 2004. Currently pursing Ph.D degree in Bharathiar University, Coimbatore, Tamilnadu. He is having total 10 years of experience which includes Industry and academics. His areas of interests are Data Warehouse & Business Intelligence, multimedia and Databases. He has published and presented papers in journals, international and national level conferences.

Dr.Ravindra S Hegadi received his Master of Computer Applications (MCA) & M.Phil and Doctorate of Philosophy Ph.D. in year 2007 in computer science from Gurbarga University, Karnataka; He is having 15 years of Experience. He has visited overseas to various universities as SME.His area of interests are Image Mining, Image Processing and Databases and business intelligence. He has published and presented papers in journals, international and national level conferences.

Umesh.I.M. received his Master of Science (MSc) & M.Phil in year 2007 in computer science from Bharathidasan University, Tamilnadu,He is working in RV College of Engineering,Bangalore,Karnataka,India.He is having 10 years of Experience.His area of interests are Image Mining, Image Processing and Databases and business intelligence. He has published and presented papers in journals, international and national level conferences.