

Application of Data Masking in Achieving Information Privacy

¹AJAYI, Olusola Olajide

Department of Computer Science, Adekunle Ajasin University
Akungba-Akoko, Ondo State, Nigeria
ajayilastborn@yahoo.co.uk / +2347056433798 / +2348137044500

²ADEBIYI, Temidayo Olarewaju

Department of Computer Science, Adekunle Ajasin University
Akungba-Akoko, Ondo State, Nigeria
puritydayo@yahoo.com / +2347030076819

Abstract

Application of data masking in achieving information privacy is implemented in order to enhance privacy of sensitive data. Data privacy is something that is appreciated by all people in different works of life, not only for self-pride but also for security. Think of a medical institution and his patient: the medical client will appreciate the secrecy of his medical data by the medical personnel; in the case of a bank and his client: the client will appreciate the privacy of his account information; just to mention but few cases. The challenge in data privacy is to share data while protecting personally identifiable information. This research work gives detailed analytical information on application of data masking in achieving information privacy. It covers issues on how data in production environment are being masked to avoid exposure of sensitive of sensitive data to an unauthorized user. The information used for this study was extracted from relevant textbooks, journals and internet. A customer-based application was developed to illustrate this subject. Java (using netbeans as the IDE) and MySQL were used as the programming language and back-end database respectively in the development of the said application.

Introduction

Information privacy or Data privacy is the relationship between collection and dissemination of data, technology, the public expectation of privacy and the legal and political issues surrounding them. Privacy concerns exist wherever personally identifiable information is collected and stored in digital form or otherwise. Improper or non-existent disclosure control can be the root cause for privacy issues. Data privacy issues can arise in response to information from a wide range of sources, such as: healthcare records, criminal justice investigations and proceedings, financial institutions and transactions, biological traits, such as genetic material, residence and geographic records, ethnicity, privacy breach. The challenge in data privacy is to share data while protecting personally identifiable information. The fields of security and information security design and utilize software, hardware and human resources to address this issue.

Information is the facts or details about somebody or something. Information even transcends physicality in some cases. Take the hypothetical and extreme example of identifying someone you are close to, whose appearance has been dramatically, surgically altered. Assume that there were no longer physical attributes that you could certifiably relate to the individual. No finger prints, no voice prints, you could only rely on something that person knows, in order to identify them.

Information can be adequately defined as a collection of data about a specific topic, or less specifically as knowledge about something. This is different from data in that data is just a collection of symbols, signs, and measures. Information implies knowledge about the data.

On the other hand, privacy can be defined simply as "for one's use only". This should not be confused totally with the term security, which means literally the act of securing something from others. Taken together, what we are saying is, an individual's right to keep information that belongs to him out of the reach of others. Granted, data security is often a consideration in the enforcement of information privacy, but it is only a part of the story. Information privacy can be guarded by a variety of mechanism. First, by keeping attackers off the system holding the information, data has some level of information, hardening systems, applying patches, and utilizing host and network based intrusion detection systems, all help to keep hackers away from the information. Another process used in information privacy is regular back-up system applications and stored data. However, these are only baby steps to ensuring information privacy.

To really privatize information, there is need for the introduction of data masking, which is a method of creating a structurally similar but in authentic version of an organization's data that can be used for purposes

such as software testing and user training. The purpose is to protect the actual data while having a functional substitute for occasions when the real data is not required. Although most organization have stringent security controls in place to protect production data in storage or in business use, sometimes that some data has been used for operations that are less secure.

The issue is often compounded if these operations are out sourced and the organization has less control over the environment. In the wake of compliance legislation, most organizations are no longer comfortable exposing real data unnecessarily.

In data masking, the format of data remains the same, only the values are changed. The data may be altered in a number of ways, including encryption, character shuffling and character or words substitution. Whatever method is chosen the values must be changed in some way that makes detection or reverse engineering impossible.

Sensitive data is a part of every organization's normal business practice. Allowing sensitive data from production applications to be copied and used for development and testing environments increases the potential for theft, loss or exposure, thus increasing the organization's risk. Different methods and techniques has been used in order to make information privacy a reality but all these methods and techniques has not given the expected result, therefore result in increased activities of hackers. With this study, we believe that information privacy will be achieved.

Related Literature

Data Masking is the process of obscuring (masking) specific data elements within data stores. It ensures that sensitive data is replaced with realistic but not real data. The goal is that sensitive customer information is not available outside of the authorized environment. Data masking is typically done while provisioning non-production environments so that copies created to support test and development processes are not exposing sensitive information and thus avoiding risks of leaking. Masking algorithms are designed to be repeatable so referential integrity is maintained.

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.

Creating test and development copies in an automated process reduces the exposure of sensitive data. Database layout often change, 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 database. It can be done in a copy THEN mask approach or a mask WHILE copy approach (the latter is branded as dynamic data masking in some products).

Effective data masking requires the alteration of data in such a way that the actual values cannot be determined or re-engineered. Functional appearance is maintained so effective testing is possible. Data can be encrypted and decrypted, relational integrity remains, security policies can be established and separation of duties between security and administration established.

Common method of data masking includes; Encryption/Decryption, Masking (i.e. numbers letters), Substitution (for example, all female names = "JULIE"), Nulling (###), Shuffling (zip code 12345 becomes 53412), Number and data variance.

Five Laws of Data Masking

1. **Masking must not be reversible.** However you mask your data, it should never be possible to use it to retrieve the original sensitive data.
2. **The results must be representative of the source data.** The reason to mask data instead of just generating random data is to provide non-sensitive information that still resembles production data for development and testing purposes. This could include geographic distributions, credit card distributions (perhaps leaving the first 4 numbers unchanged, but scrambling the rest), or maintaining human readability of (fake) names and addresses.
3. **Referential integrity must be maintained.** Masking solutions must not disrupt referential integrity — if a credit card number is a primary key, and scrambled as part of masking, then all instances of that number linked through key pairs must be scrambled identically.
4. **Only mask non-sensitive data if it can be used to recreate sensitive data.** It isn't necessary to mask everything in your database, just those parts that you deem sensitive. For example, if you scramble a medical ID but the treatment codes for a record could only map back to one record, you also need to

scramble those codes. This attack is called inference analysis, and your masking solution should protect against it.

5. **Masking must be a repeatable process.** One-off masking is not only nearly impossible to maintain, but it's fairly ineffective. Development and test data need to represent constantly changing production data as closely as possible. Analytical data may need to be generated daily or even hourly. If masking isn't an automated process it's inefficient, expensive, and ineffective.

Data requirements

- ❖ **Format Preservation:** The mask must produce data with the same structure as the original data. This means that if the original data is 2-30 characters long, the mask should produce data 2-30 characters long. A common example is dates, which must produce numbers in the correct ranges for day, month, and year, likely in a particular text format. This means a masking algorithm must identify the 'meaning' of source data such as "31.03.2012", "March 31, 2012", and "03.31.2012", and generate a suitable date in the same format.
- ❖ **Data Type Preservation:** With relational data storage it is essential to maintain data types when masking data from one database to another. Relational databases require formal definition of data columns and do not tolerate text in number or date fields. In most cases format-preserving masks implicitly preserve data type, but that is not always the case. In certain cases data can be 'cast' from a specific data type into a generic data type (e.g., varchar), but it is essential to verify consistency.
- ❖ **Gender preservation:** When substituting names, male names are only substituted with other male names, and similarly female with only female names. This is of special importance amongst some cultures.
- ❖ **Semantic Integrity:** Databases often place additional constraints on data they contain. In this way you ensure both the format and data type integrity, but the stored value makes sense in a business context as well.
- ❖ **Referential Integrity:** An attribute in one table or file may refer to another element in a different table or file, in which case the reference must be consistently maintained. The reference augments or modifies the meaning of each element, and is in fact part of the value of the data. Relational databases optimize data storage by allowing one set of data elements to 'relate', or refer, to another. Shuffling or substituting key data values can destroy these references (relationships). Masking technologies must maintain referential integrity when data is moved between relational databases, or cascading values from one table to another. This ensures that loading the new data works without errors, and avoids breaking applications which rely on these relationships.
- ❖ **Aggregate Value:** The total and average values of a masked column of data should be retained, either closely or precisely.
- ❖ **Frequency Distribution:** In some cases users require random frequency distribution, while in others logical groupings of values must be maintained or the masked data is not usable. For example, if the original data describes geographic locations of cancer patients by zip code, random zip codes would discard valuable geographical information. The ability to mask data while maintaining certain types of patterns is critical for maintaining the value of masked data for analytics.
- ❖ **Uniqueness:** Masked values must be unique. As an example, duplicate SSNs are not allowed when uniqueness is a required integrity constraint. This is critical for referential integrity, as the columns used to link tables must contain unique values.

Data Masking Techniques

Substitution

The substitution technique replaces the existing data with random values from a pre-prepared dataset.

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 number values are present in their original rows.

Number and data variance

The number and data 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 +/- 60days.

Encryption

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

Nulling out or Deletion

The nulling out technique simply removes the sensitive data by deleting it.

Masking Out

If two tables contain the columns with the same renormalized data values and those columns are masked in one table then the second table will need to be updated with the changes. Technique is called Table-To-Table synchronization.

Design Techniques In Data Masking Design

Substitution

Substitution is simply replacing one value with another. For example, the mask might substitute a person's first and last names with names from a random phone book entry. The resulting data still constitutes a name, but has no logical relationship with the original real name unless you have access to the original substitution table.

Nulling

This substitution simply replaces sensitive data with a generic value, such as 'X'. For example, we could replace a phone number with "(XXX)XXX-XXXX", or a Social Security Number (SSN) with XXX-XX-XXXX. This is the simplest and fastest form of masking, but the output provides little or no value.

Shuffling

Shuffling is a method of randomizing existing values vertically across a data set. For example, shuffling individual values in a salary column from a table of employee data would make the table useless for learning what any particular employee earns without changing aggregate or average values for the table. Shuffling is a common randomization technique for disassociating sensitive data relationships (e.g., Bob makes \$X per year) while preserving aggregate values.

Blurring

Taking an existing value and alter it so that the value falls randomly within a defined range.

Averaging

Averaging is an obfuscation technique where individual numbers are replaced by a random value, but across the entire field, the average of these values remains consistent. In our salary example above, we could substitute individual salaries with the average across a group or corporate division to hide individual salary values while retaining an aggregate relationship to the real data.

De-identification

De-identification is important for dealing with complex, multi-column data sets that provide sufficient clues to reverse engineer masked data back into individual identities.

Tokenization

Tokenization is substitution of data elements with random placeholder values, although vendors overuse the term 'tokenization' for a variety of other techniques. Tokens are non-reversible because the token bears no logical relationship to the original value.

Masking Out Data

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 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

Format Preserving Encryption

Encryption is the process of transforming data into an unreadable state. Unlike the other methods listed, the original value can be determined from the encrypted value, but can only be reversed with special knowledge (the key). While most encryption algorithms produce strings of arbitrary length, format preserving encryption transforms the data into an unreadable state while retaining the format (overall appearance) of the original values. Technically encryption violates the first law of data masking.

The Algorithm

The private key here is a pair of number's (N,d), and the public key is a pair of numbers (N,e). note that N is common to the private and public keys. The sender uses the followings algorithm to encrypt the message.

$$C = P^e \text{ mod } N$$

In this algorithm, P is the plain text, which represents Bobented as a number; C is the number that represents bobents as the ciphertext. The two numbers e and N are components of the public key. Plaintext P is raised to the power e and divided by N. the mad term indicates that the remainder is sent as the ciphertext.

The receiver uses the following algorithm to decrypt the message:

$$P = C^d \text{ mod } N$$

In this algorithm, P and C are the same as before. The two numbers "d" and "N" are components of the private key.

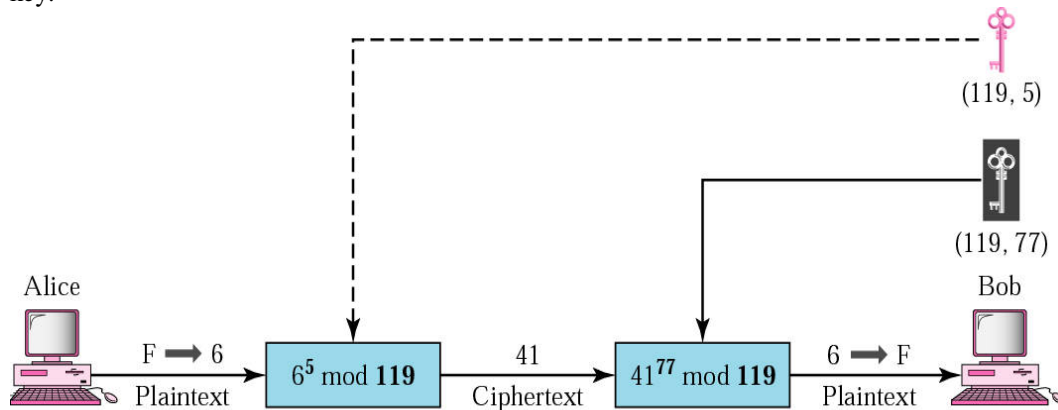


Fig. 1: Message Encryption

Program flow charts

In digital signature, the private key is used for encryption and the public key is used for decryption. This is possible because the encryption and decryption algorithms used today, such as RSA, are mathematical formulas and their structures are familiar. This is illustrated below.

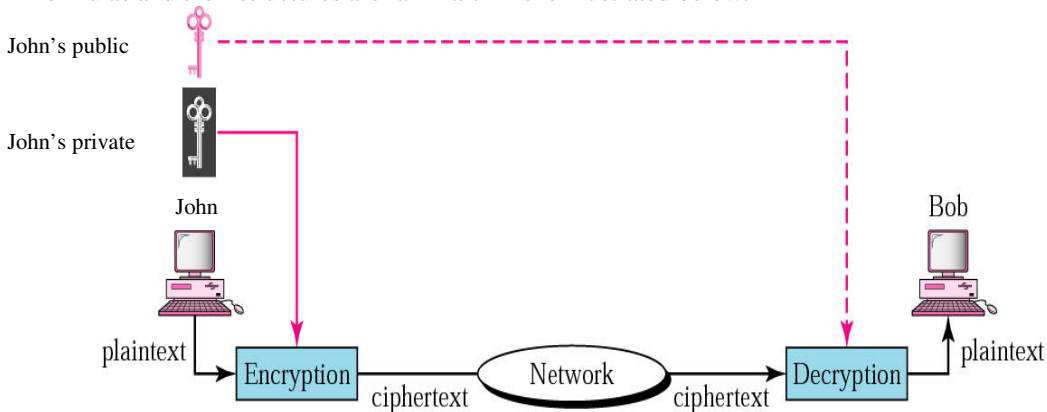


Fig. 2: Digital signatures using public and private key cryptography

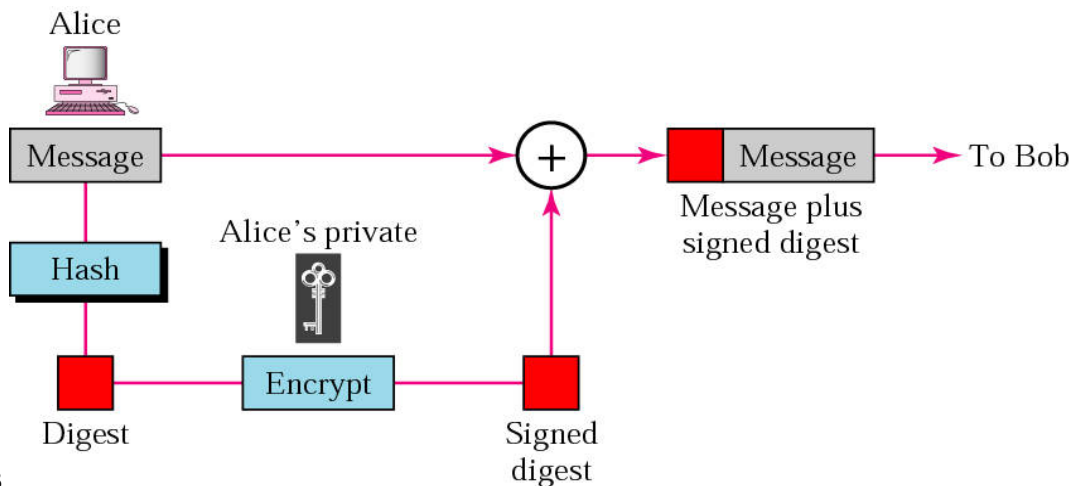


Fig. 3.

Modelling For Data Masking
Data Modelling For Data Masking

We select data from a file or database, mask the sensitive data, and then either update the original file/database, or store the masked data to a new location.

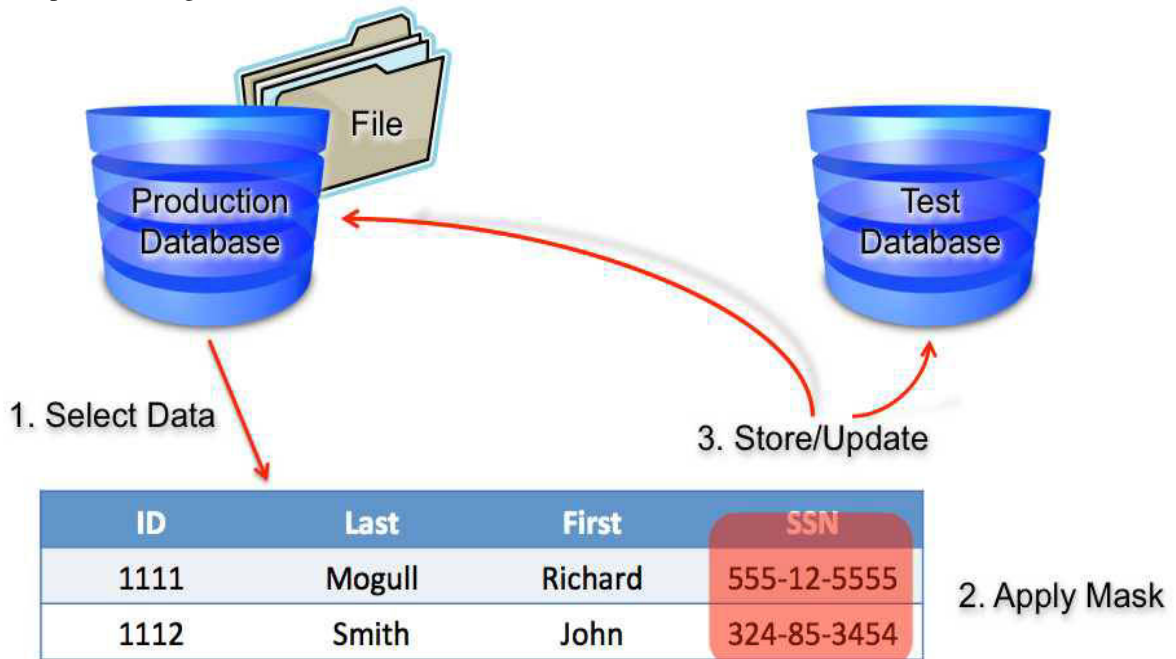


Fig. 4. Understanding and Selecting Data Masking Solutions

Data Masking Life Cycle

The diagram below outlines the four phases of the typical Data Masking Lifecycle and highlights the steps involve in evaluating and implementing the Data Masking solution.



Analyze – Discover And Analyze Sensitive Data

The goal of the Analyze Phase is to identify data that needs to be masked in order to provide sufficient protection without compromising data utility. This stage involves documentation of requirements and education on the implications of masking necessary for the creation of configurations to be used in masking data during the Execute stage of the Data Masking Lifecycle. This phase is typically 20% of the data masking project. Automated discovery of sensitive data is a key factor in minimizing analysis effort and maximizing accuracy.

Model - Establish Context For Discovered Data

The Model Phase is intended to establish criteria that will be used in determining how to mask the data. It is during this phase that context around the information discovered during the Analyze Phase is created. The Model Phase is typically 40% of the entire data masking project. Data classification plays an important role in determining the sensitivity of various data, its intended use(s) and how it will ultimately be masked.




































Develop - Create Data Masking Configurations

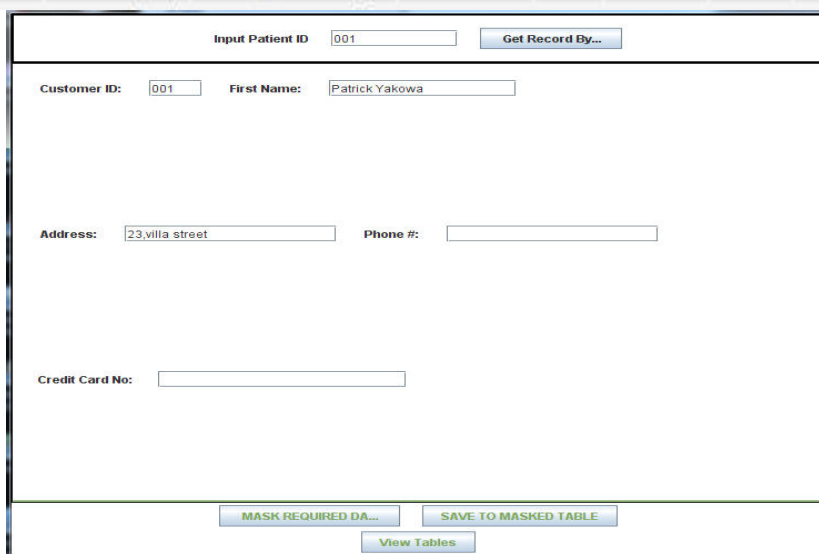
The goal of the Develop Phase is to create data masking configurations based upon customer-specific Functional Masking Requirements defined in prior phases. Consideration is given as to how data masking configurations will be integrated into the overall refresh process for non-production environment. This phase also provides an opportunity to develop data masking schedules and establish appropriate change management process. The develop phase is typically data masking project. Data masking software that is easy-to-use, flexible and scalable is critical for accommodating varying and often complex requirement.

Execute - Integrate Data Masking In The Business Process

The Execute phase is intended to transition data masking into the refresh process for non-production environments taking the overall business process(es) into account. This phase entails executing configurations constructed during the Develop Phase – such as creating database subsets and executing data masking job scheduling scripts. The Execute phase is typically 10% of the data masking project. Appropriate knowledge transfer and hand-off are key to successful completion of the roll-out.

Design Specifications Input Design

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	customerid	varchar(4)	latin1_swedish_ci		No			     
	fname	varchar(50)	latin1_swedish_ci		No			     
	address	varchar(50)	latin1_swedish_ci		No			     
	phone_no	varchar(20)	latin1_swedish_ci		No			     
	card_no	varchar(50)	latin1_swedish_ci		No			     



Input Patient ID:

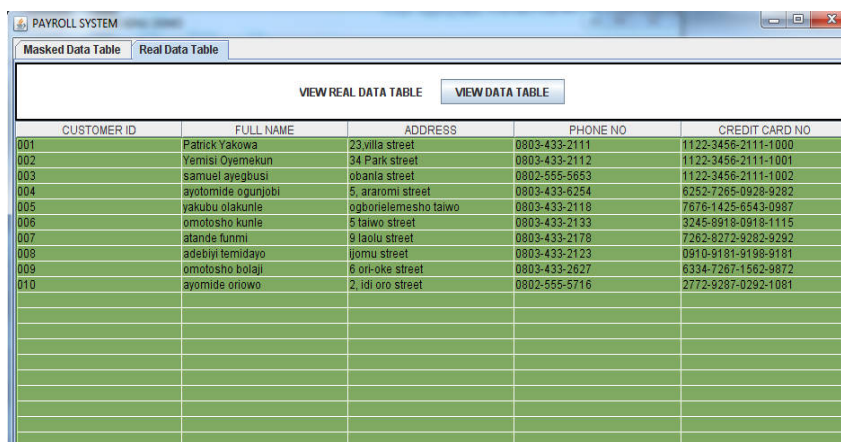
Customer ID: First Name:

Address: Phone #:

Credit Card No:

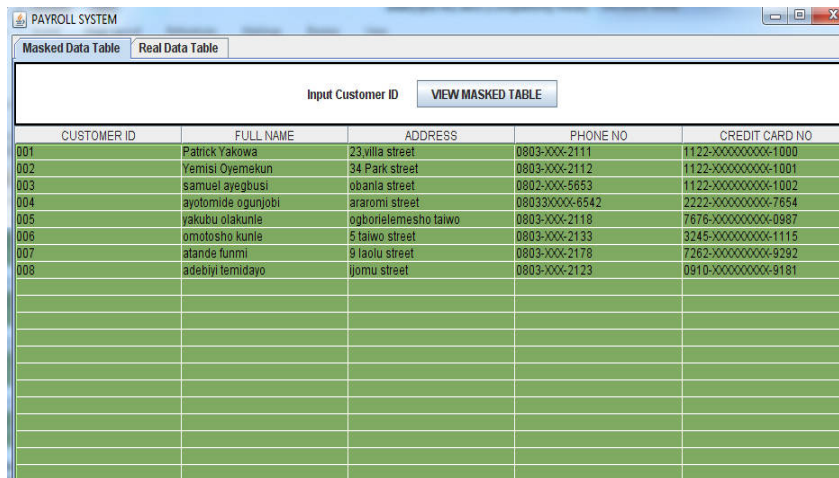
Output Design

This section demonstrates the outputs design of the proposed system
Output of Real Data



CUSTOMER ID	FULL NAME	ADDRESS	PHONE NO	CREDIT CARD NO
001	Patrick Yakowa	23, villa street	0803-433-2111	1122-3456-2111-1000
002	Yemisi Ojemekun	34 Park street	0803-433-2112	1122-3456-2111-1001
003	Samuel Ayejobi	Obanla street	0802-555-5653	1122-3456-2111-1002
004	Ayotomide Ogundipe	5, Araromi street	0803-433-6254	6252-7265-0928-9282
005	Yakubu Olakunle	Ogborielemeshe Taiwo	0803-433-2118	7676-1425-6543-0987
006	Omotosho Kunle	5 Iaiwo street	0803-433-2133	3245-8918-0918-1115
007	Atanda Funmi	9 Ialolu street	0803-433-2178	7262-8272-9282-9292
008	Adesbiyi Temidayo	Ijumu street	0803-433-2123	0910-9181-9198-9181
009	Omotosho Bolaji	6 Ori-Oke street	0803-433-2627	6334-7267-1562-9872
010	Ayomide Oriowo	2, Idi Oro street	0802-555-5716	2772-9287-0282-1081

Output Masked Data



CUSTOMER ID	FULL NAME	ADDRESS	PHONE NO	CREDIT CARD NO
001	Patrick Yakowa	22, villa street	0803-XXX-2111	1122-XXXXXXXXXX-1000
002	Yemisi Oyemekun	34 Park street	0803-XXX-2112	1122-XXXXXXXXXX-1001
003	Samuel Ayegbusi	Obania street	0802-XXX-5653	1122-XXXXXXXXXX-1002
004	Ayotomide Ogunjobi	Araromi street	0803-XXXX-8542	2222-XXXXXXXXXX-7654
005	Yakubu Olakunle	Ogbonialemesho Taiwo	0803-XXX-2118	7676-XXXXXXXXXX-0987
006	Omotosho Kunie	5 Taiwo street	0803-XXX-2133	3245-XXXXXXXXXX-1115
007	Atande Funmi	9 Iaolu street	0803-XXX-2178	7262-XXXXXXXXXX-9292
008	Adebiri Temidayo	Ijomu street	0803-XXX-2123	0910-XXXXXXXXXX-9181

Conclusion

There is a growing need to protect sensitive employee, customer, and business data across the enterprise wherever such data may reside. Until recently, most data theft occurred from malicious individuals hacking into production databases that were made available across the Internet. With a number of well-publicized and costly thefts creating both tremendous legal liability and bad publicity for the effected organizations, business has quickly grown more and more sophisticated in protecting against such schemes.

While the industry deals with the most egregious aspects of data theft, many computer systems still remain vulnerable to attack at some level. An important tier of computer data remains practically untouched and unprotected by today's new data security procedures: non-production systems used for in-house development, testing, and training purposes are generally open systems and leave a large hole in the data privacy practices at companies of all sizes. These environments leverage real data to test applications, housing some of the most confidential or sensitive information in an organization, such as Social Security numbers, bank records, and financial documents.

This study discusses one of the best practices for creating data privacy procedures in non-production environments. These procedures include creating a comprehensive set of policies to classify datatype that needs to be protected, integrating these policies into day-to-day business processes, providing ongoing compliance reviews, using a proven commercial solution for masking sensitive data in all nonproduction environments, and integrating these privacy processes and technology across the enterprise.

References

- Becket, B, (1998), introduction to cryptography. London Blackwell Scientific Publication.
- Behrouz, A, and Frozen, P. (2004) Data communication and Networking. New York Nash vice, T. N: Vanderbilt University Press.
- Camouflage data masking specialist, latest edition.
- Data masking best practices, latest edition.
- Data privacy best practices for data protection in nonproduction environments. Published June, 2009.
- R. Agawam, E. Term, (2006). "On Honesty in Sovereign Information Sharing", Proceedings of the 10th International Conference on Extending Database Technology, Munich, Germany.
- asku@infosys.com
- www.axistechnologyllc.com
- www.datamasking.com