Design of Data Leakage Prevention Model using Anonymization Technique and Time Restriction

A Thesis submitted to the Faculty of Engineering & Technology, Jadavpur University in partial fulfillment of the requirements for the Degree of Master of Engineering in Software Engineering

By

Piyush Kanti Samanta

Examination Roll Number: **M4SWE19004**
Registration Number: 140980 of 2017-2018
Class Roll Number: 001711002024
Jadavpur University

Under the Guidance & Supervision of

Dr. PARAMA BHAUMIK

Associate Professor
Department of Information Technology
Jadavpur University

Department of Information Technology
Faculty of Engineering and Technology
Jadavpur University (Salt Lake Campus)
Kolkata-700098

2019

Jadavpur University
Department of Information Technology
Faculty of Engineering & Technology

Certificate of Recommendation

We hereby recommend the thesis, entitled "**Design of Data Leakage Prevention Model using Anonymization Technique and Time Restriction**" prepared under the guidance of Associate Professor **Dr. Parama Bhaumik**, Dept. of Information Technology,Jadavpur University, Saltlake Campus, Kolkata submitted by **Piyush Kanti Samanta** (Examination Roll Number: **M4SWE19004**, Registration Number:140980 of 2017-18), may be accepted in partial fulfilment of the requirements for the Degree of Master of Engineering in Software Engineering from the Department of Information Technology of Jadavpur University.

....

Dr. Parama Bhaumik
Associate Professor
Dept. of Information Technology,
Jadavpur University

Counter Signed by:

....

Head of the Department
Department of Information Technology,
Jadavpur University

....

Dean
Faculty of Engineering and Technology,
Jadavpur University

Jadavpur University
Department of Information Technology
Faculty of Engineering & Technology

Certificate of Approval

The foregoing thesis, entitled as **"Design of Data Leakage Prevention Model using Anonymization Technique and Time Restriction"** is hereby approved by the committee of final examination for evaluation of thesis as a creditable study of an engineering subject carried out and presented by **Piyush Kanti Samanta** (Examination Roll Number : M4SWE19004 , Registration Number: 140980 of 2017-18) in a manner satisfactory to warrant its acceptance as a perquisite to the Degree of Master of Software Engineering. It is understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose for which it is submitted.

....
Signature of the Examiner

....
Signature of the Supervisor
Dr. Parama Bhaumik
Associate Professor
Dept. of Information Technology,
Jadavpur University

Jadavpur University
Department of Information Technology
Faculty of Engineering & Technology

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as a part of his Master of Engineering in Software Engineering.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all the material and results that are not original to this work.

Name (in Block Letters): **PIYUSH KANTI SAMANTA**

Exam Roll Number: **M4SWE19004**

Registration Number: 140980 of 2017-18

Class Roll Number: 001711002024

Thesis Title: **Design of Data Leakage Prevention Model using Anonymization Technique and Time Restriction**

……………………………………..

Signature with Date

Acknowledgments

I would like to express my heartfelt gratitude to my supervisor and guide Dr. Parama Bhaumik (Associate Professor, Department of Information Technology, Jadavpur University) for her guidance and support during the entire course of my post-graduate studies and ME thesis work. Her encouraging words and positive reinforcements helped me to remain confident in myself and motivated me to keep working toward my ME Thesis during times when I felt lost.

The constructive and insightful discussions that I have had with her over the past one year, has inspired me to complete this work successfully. I have always been benefited by her constant enthusiasm and zeal for research. It is certain the project would not have achieved the success it did without her help and support. It has been a privilege to have her as my mentor.

My sincere obligation goes to Associate Professor Bhaskar Sardar, Head of the Department, Department of Information Technology, Jadavpur University for his support. My whole hearted thanks to the staffs of the Department of Information Technology, Jadavpur University, and my friends for their help.

I would also like to thanks Jadavpur University, for giving me this opportunity to pursue this ME program including this research.

I also would like to thank all of my friends for offering words of encouragement and support in times when I most needed it.

Finally, heartfelt thanks to my Parents and the Almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Regards

Piyush Kanti Samanta

Abstract

Today’s life everything including digital economy, data enter and leaves cyberspace at record rates. A contemporary business relies on sharing and transferring the information among various stakeholders such as employees, owners (shareholders), creditors and suppliers within or outside the organization. As the shared critical data can be leaked by some malicious entity, the persistency of preventing data misuse is escalated. Severe damage caused by the sharing of sensitive information constitutes a grievous threat to the organization's assets. Protection of confidential data from unauthorized revelation is a matter of concern for any enterprise. Data leakage causes a negative impact on companies. So preventing the data many vendors currently offer data leak prevention. The data stored in any device can be leaked in two ways; if the system is hacked or if the internal resources intentionally or unintentionally make the data public.

Therefore, organizations should take measures to understand the sensitive data they hold, how it’s controlled, and how to prevent it from being leaked or compromised.
So that purpose in this paper, data is preventing by using **Anonymization technique** of data leak prevention. **Anonymization technique** can be used, including data substitution and shuffling specific fields (sensitive data) and data sets with the particular timestamp (Time restriction), after that timestamp the sensitive data will be hidden and again shuffling (for two or more time, sensitive data access).

This report summarizes a 1-year research project in analysis, literature survey and implementation of new technique model for sensitive data leakage prevention.

Table of Contents

Chapter No Topics Page No

1. Introduction

2. Literature Survey

3. Working Principle

Chapter No Topics Page No

List of Figures

SL Title of Figure Page No.

CHAPTER 1

Introduction

1.1. Data Leakage Defined

Data leakage is the unauthorized transmission of data from within an organization to an external destination. The term can be used to describe data that is transferred electronically or physically. Data leakage threats usually occur via the web and email, but can also occur via mobile data storage devices such as optical media, USB keys, and laptops.

Barely a day goes by without a confidential data breach hitting the headlines. Data leakage, also known as low and slow data theft, is a huge problem for data security, and the damage caused to any organization, regardless of size or industry, can be serious. From declining revenue to a tarnished reputation or massive financial penalties to crippling lawsuits, this is a threat that any organization will want to protect them from.

1.2. Overview of Data Leakage Prevention

In information security data leakage threat has become an important issue especially data leakage caused by insider threat, Most of the computer attacks are from authorized users of the system. With the widespread of the internet, the insider threat is more serious.

Sending confidential data to an unauthorized party called as data leakage. To prevent the data leaving from the outside of the organization private network called as data leakage prevention. Data leakage prevention system is a collection of sub-systems which helps to identify the confidential data and also prevents the data leakages.

Data leakage prevention systems consider two parameters for preventing data leakage. One parameter is data states. In general, data is available in three states i.e. rest state, use state and move state another parameter is deployment scheme i.e. where we are deploying our Data Leakage Prevention system Based on these two parameters the Data Leakage Prevention solutions are changed.

1.3. Types of Data Leakage

There are many different types of data leakage and it is important to understand that the problem can be initiated via an external or internal source. Protective measures need to address all areas to ensure that the most common data leakage threats are prevented.

1.3.1. The Accidental Breach

"Unauthorized" data leakage does not necessarily mean intended or malicious. The good news is that the majority of data leakage incidents are accidental. For example, an employee may unintentionally choose the wrong recipient when sending an email containing confidential data. Unfortunately, unintentional data leakage can still result in the same penalties and reputational damage as they do not mitigate legal responsibilities.

1.3.2. The Intentioned or Disgruntled Employee

When we think of data leakages, we think about data held on stolen or misplaced laptops or data that is leaked over email. However, the vast majority of data loss does not occur over an electronic medium; it occurs via printers, cameras, photocopiers, removable USB drives and even dumpster diving for discarded documents. While an employee may have signed an employment contract that effectively signifies trust between employer and employee, there is nothing to stop them from later leaking confidential information out of the building if they are disgruntled or promised a hefty payout by cybercriminals. This type of data leakage is often referred to as data exfiltration.

1.3.3. Electronic Communications with Malicious Intent

Many organizations give employees access to the internet, email, and instant messaging as part of their role. The problem is that all of these mediums are capable of file transfer or accessing external sources over the internet. Malware is often used to target these mediums and with a high success rate. For example, a cybercriminal could quite easily spoof a legitimate business email account and request sensitive information to be sent to them. The user would unwittingly send the information, which could contain financial data or sensitive pricing information.

Phishing attacks are another cyber-attack method with a high data leakage success rate. Simply by clicking on a link and visiting a web page that contains malicious code could allow an attacker to access a computer or network to retrieve the information they need.

1.3.4. An SQL Injection Attack

The Astonishing Furniture mock website, built using the free, open-source software program Drupal, features an online application for a store credit card. Here, consumers would enter sensitive information, including their social security number, date of birth and income, which would be stored in a database that is vulnerable to an SQL injection attack.

An SQL injection attack exploits a vulnerability in the software where the user inputs data. What the vulnerability in Drupal allowed is for the hacker to enter the code in the user name and password field. From there, the hacker could assign an administrative user name and password and execute commands on the server, including downloading sensitive data.

“If we think of Astonishing Furniture as an example of a typical commercial entity, our data shows us they probably do not have a plan in the event of an attack,” says Travelers Cyber Lead Tim Francis, who says that small and mid-sized companies often are the least prepared. “They lack some of the resources and the expertise to adequately prevent against these attacks from occurring in the first place and when these attacks do occur, they are often the least likely to be able to respond.”

1.4. Research Objective

In information security data leakage threat has become an important issue especially data leakage caused by insider threat [1]. Most of the computer attacks are from authorized users of the system. With the widespread of internet, the insider threat is more serious.

Sending confidential data to an unauthorized party called as data leakage. To prevent the data leaving from the outside of the organization private network called as data leakage prevention. Data leakage prevention system is a collection of sub-systems which helps to identify the confidential data and also prevents the data leakages.

Data leakage prevention systems consider two parameters for preventing data leakage. One parameter is data states. In general, data is available in three states i.e. rest state, use state and move state another parameter is deployment scheme i.e. where we are deploying our Data Leakage Prevention system Based on these two parameters the Data Leakage Prevention solutions are changed.

CHAPTER 2

Literature Survey

2.1. The 18 biggest data breaches of the 21st century

Data breaches happen daily, in too many places at once to keep count. But what constitutes a huge breach versus a small one? CSO compiled a list of 18 of the biggest or most significant breaches of the 21st century.

This list is based not necessarily on the number of records compromised, but on how much risk or damage the breach caused for companies, insurers, and users or account holders. In some cases, passwords and other information were well protected by encryption, so a password reset eliminated the bulk of the risk.

2.1.1. Yahoo

Date: 2013-14
Impact: 3 billion user accounts

Details: In September 2016, the once-dominant Internet giant, while in negotiations to sell itself to Verizon, announced it had been the victim of the biggest data breach in history, likely by "a state-sponsored actor," in 2014. The attack compromised the real names, email addresses, dates of birth and telephone numbers of 500 million users. The company said the "vast majority" of the passwords involved had been hashed using the robust bcrypt algorithm.

A couple of months later, in December, it buried that earlier record with the disclosure that a breach in 2013, by a different group of hackers had compromised 1 billion accounts. Besides names, dates of birth, email addresses, and passwords that were not as well protected as those involved in 2014, security questions and answers were also compromised. In October of 2017, Yahoo revised that estimate, saying that, in fact, all 3 billion user accounts had been compromised.

The breaches knocked an estimated $350 million off Yahoo's sale price. Verizon eventually paid $4.48 billion for Yahoo's core Internet business. The agreement called for the two companies to share regulatory and legal liabilities from the breaches. The sale did not include a reported investment in Alibaba Group Holding of $41.3 billion and an ownership interest in Yahoo Japan of $9.3 billion.

Yahoo, founded in 1994, had once been valued at $100 billion. After the sale, the company changed its name to Altaba, Inc.

2.1.2. Marriott International

Date: 2014-18

Impact: 500 million customers

Details: In November 2018, Marriott International announced that cyber thieves had stolen data on approximately 500 million customers. The breach actually occurred on systems supporting Starwood hotel brands starting in 2014. The attackers remained in the system after Marriott acquired Starwood in 2016 and were not discovered until September 2018.

For some of the victims, only name and contact information were compromised. The attackers were able to take some combination of contact info, passport number, Starwood Preferred Guest numbers, travel information, and other personal information. Marriott believes that credit card numbers and expiration dates of more than 100 million customers were stolen, although the company is uncertain whether the attackers were able to decrypt the credit card numbers.

The breach was eventually attributed to a Chinese intelligence group seeking to gather data on US citizens, according to a New York Times article. If true, this would be the largest known breach of personal data conducted by a nation-state.

2.1.3. Adult Friend Finder

Date: October 2016

Impact: More than 412.2 million accounts

Details: The Friend-Finder Network, which included casual hookup and adult content websites like Adult Friend Finder, Penthouse.com, Cams.com, iCams.com, and Stripshow.com, was breached sometime in mid-October 2016. Hackers collected 20 years of data on six databases that included names, email addresses, and passwords.

Most of the passwords were protected only by the weak SHA-1 hashing algorithm, which meant that 99 percent of them had been cracked by the time LeakedSource.com published its analysis of the entire data set on November 14.

CSO online Steve Ragan reported at the time that, "a researcher who goes by 1x0123 on Twitter and by Revolver in other circles posted screenshots were taken on Adult Friend Finder (that) show a Local File Inclusion vulnerability (LFI) being triggered." He said the vulnerability, discovered in a module on the production servers used by Adult Friend Finder, "was being exploited."

AFF Vice President Diana Ballou issued a statement saying, “We did identify and fix a vulnerability that was related to the ability to access source code through an injection vulnerability.”

2.1.4. eBay

Date: May 2014

Impact: 145 million users compromised

Details: The online auction giant reported a cyber-attack in May 2014 that it said exposed names, addresses, dates of birth and encrypted passwords of all of its 145 million users. The company said hackers got into the company network using the credentials of three corporate employees, and had complete inside access for 229 days, during which time they were able to make their way to the user database.

It asked its customers to change their passwords but said financial information, such as credit card numbers, was stored separately and was not compromised. The company was criticized at the time for a lack of communication informing its users and poor implementation of the password-renewal process.

CEO John Donahue said the breach resulted in a decline in user activity but had little impact on the bottom line – its Q2 revenue was up 13 percent and earnings up 6 percent, in line with analyst expectations.

2.1.5. Equifax

Date: July 29, 2017

Impact: Personal information (including Social Security Numbers, birth dates, addresses, and in some cases drivers' license numbers) of 143 million consumers; 209,000 consumers also had their credit card data exposed.

Details: Equifax, one of the largest credit bureaus in the U.S., said on Sept. 7, 2017, that an application vulnerability on one of their websites led to a data breach that exposed about 147.9 million consumers. The breach was discovered on July 29, but the company says that it likely started in mid-May.

2.1.6. Heartland Payment Systems

Date: March2008

Impact: 134 million credit cards exposed through SQL injection to install spyware on Heartland's data systems.

Details: At the time of the breach, Heartland was processing 100 million payment card transactions per month for 175,000 merchants – most small- to mid-sized retailers. It wasn't discovered until January 2009, when Visa and MasterCard notified Heartland of suspicious transactions from accounts it had processed.

Among the consequences were that Heartland was deemed out of compliance with the Payment Card Industry Data Security Standard (PCI DSS) and was not allowed to process the payments of major credit card providers until May 2009. The company also paid out an estimated $145 million in compensation for fraudulent payments.

A federal grand jury indicted Albert Gonzalez and two unnamed Russian accomplices in 2009. Gonzalez, a Cuban-American, was alleged to have masterminded the international operation that stole the credit and debit cards. In March 2010 he was sentenced to 20 years in federal prison. The vulnerability to SQL injection was well understood and security analysts had warned retailers about it for several years. Yet, the continuing vulnerability of many Web-facing applications made SQL injection the most common form of attack against Web sites at the time.

2.1.7. Target Stores

Date: December 2013

Impact: Credit/debit card information and/or contact information of up to 110 million people compromised.

Details: The breach actually began before Thanksgiving, but was not discovered until several weeks later. The retail giant initially announced that hackers had gained access through a third-party HVAC vendor to its point-of-sale (POS) payment card readers, and had collected about 40 million credit and debit card numbers.

By January 2014, however, the company upped that estimate, reporting that personally identifiable information (PII) of 70 million of its customers had been compromised. That included full names, addresses, email addresses, and telephone numbers. The final estimate is that the breach affected as many as 110 million customers.

Target's CIO resigned in March 2014, and its CEO resigned in May. The company recently estimated the cost of the breach at $162 million.

The company was credited with making significant security improvements. However, a settlement announced in May 2017 that gave Target 180 days to make specific security improvements was described by Tom Kellermann, CEO of Strategic Cyber Ventures and former CSO of Trend Micro, as a "slap on the wrist." He also said it, "represents yesterday's security paradigm,” since the requirements focus on keeping attackers out and not on improving incident response.

2.1.8. TJX Companies, Inc.

Date: December 2006

Impact: 94 million credit cards exposed.

Details: There are conflicting accounts of how this happened. One supposes that a group of hackers took advantage of a weak data encryption system and stole credit card data during a wireless transfer between two Marshall's stores in Miami, Fla. The other has them breaking into the TJX network through in-store kiosks that allowed people to apply for jobs electronically.

Albert Gonzalez, hacking legend and ringleader of the Heartland breach, was convicted in 2010 of leading the gang of thieves who stole the credit cards, and sentenced to 20 years in prison, while 11 others were arrested. He had been working as a paid informant for the US Secret Service, at a $75,000 salary at the time of the crimes. The government claimed in its sentencing memo that companies, banks, and insurers lost close to $200 million.

2.1.9. Uber

Date: Late 2016

Impact: Personal information of 57 million Uber users and 600,000 drivers exposed.

Details: The scope of the Uber breach alone warrants its inclusion on this list, and it's not the worst part of the hack. The way Uber handled the breach once discovered is one big hot mess, and it's a lesson for other companies on what not to do.

The company learned in late 2016 that two hackers were able to get names, email addresses, and mobile phone numbers of 57 users of the Uber app. They also got the driver license numbers of 600,000 Uber drivers. As far as we know, no other data such as credit card or Social Security numbers were stolen. The hackers were able to access Uber's GitHub account, where they found username and password credentials to Uber's AWS account. Those credentials should never have been on GitHub.

Here's the really bad part: It wasn't until about a year later that Uber made the breach public. What's worse, they paid the hackers $100,000 to destroy the data with no way to verify that they did, claiming it was a "bug bounty" fee. Uber fired its CSO because of the breach, effectively placing the blame on him.

The breach is believed to have cost Uber dearly in both reputation and money. At the time that the breach was announced, the company was in negotiations to sell a stake to Softbank. Initially, Uber's valuation was $68 billion. By the time the deal closed in December, its valuation dropped to $48 billion. Not all of the drop is attributable to the breach, but analysts see it being a significant factor.

2.1.10. JP Morgan Chase

Date: July 2014

Impact: 76 million households and 7 million small businesses

Details: The largest bank in the nation was the victim of a hack during the summer of 2014 that compromised the data of more than half of all US households – 76 million – plus 7 million small businesses. The data included contact information – names, addresses, phone numbers, and email addresses – as well as internal information about the users, according to a filing with the Securities and Exchange Commission.

The bank said no customer money had been stolen and that there was “no evidence that account information for such affected customers – account numbers, passwords, user IDs, dates of birth or Social Security numbers – was compromised during this attack."

Still, the hackers were reportedly able to gain “root" privileges on more than 90 of the bank’s servers, which meant they could take actions including transferring funds and closing accounts. According to the SANS Institute, JP Morgan spends $250 million on security every year.

In November 2015, federal authorities indicted four men, charging them with the JP Morgan hack plus other financial institutions. Gery Shalon, Joshua Samuel Aaron and Ziv Orenstein faced 23 counts, including unauthorized access of computers, identity theft, securities and wire fraud and money laundering that netted them an estimated $100 million. A fourth hacker who helped them breach the networks was not identified.

Shalon and Orenstein, both Israelis, pleaded not guilty in June 2016. Aaron was arrested at JFK Airport in New York last December.

2.1.11. US Office of Personnel Management (OPM)

Date: 2012-14

Impact: Personal information of 22 million current and former federal employees

Details: Hackers, said to be from China, were inside the OPM system starting in 2012, but were not detected until March 20, 2014. A second hacker, or group, gained access to OPM through a third-party contractor in May 2014 but was not discovered until nearly a year later. The intruders exfiltrated personal data – including in many cases detailed security clearance information and fingerprint data.

Last year, former FBI director James Comey spoke of the information contained in the so-called SF-86 form, used for conducting background checks for employee security clearances. “My SF-86 lists every place I’ve ever lived since I was 18, every foreign travel I’ve ever taken, all of my family, their addresses,” he said. “So it’s not just my identity that’s affected. I’ve got siblings. I’ve got five kids. All of that is in there.”

A report, released last fall by the House Committee on Oversight and Government Reform summed up the damage in its title: “The OPM Data Breach: How the Government Jeopardized Our National Security for More than a Generation.”

2.1.12. Sony's PlayStation Network

Date: April 20, 2011

Impact: 77 million PlayStation Network accounts hacked; estimated losses of $171 million while the site was down for a month.

Details: This is viewed as the worst gaming community data breach of all-time. Of more than 77 million accounts affected, 12 million had unencrypted credit card numbers. Hackers gained access to full names, passwords, e-mails, home addresses, purchase history, credit card numbers, and PSN/Qriocity logins and passwords. "It's enough to make every good security person wonder, 'If this is what it's like at Sony, what's it like at every other multi-national company that's sitting on millions of user data records?'" said eIQnetworks' John Linkous. He says it should remind those in IT security to identify and apply security controls consistently across their organizations. For customers, "Be careful whom you give your data to. It may not be worth the price to get access to online games or other virtual assets."

In 2014, Sony agreed to a preliminary $15 million settlement in a class action lawsuit over the breach.

2.1.13. Anthem

Date: February 2015

Impact: Theft of personal information on up to 78.8 million current and former customers.

Details: The second-largest health insurer in the U.S., formerly known as WellPoint, said a cyber-attack had exposed the names, addresses, Social Security numbers, and date of birth and employment histories of current and former customers – everything necessary to steal an identity.

Fortune reported in January that a nationwide investigation concluded that a foreign government likely recruited the hackers who conducted what was said to be the largest data breach in healthcare history. It reportedly began a year before it was announced, when a single user at an Anthem subsidiary clicked on a link in a phishing email. The total cost of the breach is not yet known, but it is expected to exceed $100 million. Anthem said in 2016 that there was no evidence that members' data have been sold, shared or used fraudulently. Credit card and medical information also allegedly has not been taken.

2.1.14. RSA Security

Date: March 2011

Impact: Possibly 40 million employee records stolen.

Details: The impact of the cyber-attack that stole information on the security giant's SecurID authentication tokens is still being debated. RSA, the security division of EMC, said two separate hacker groups worked in collaboration with a foreign government to launch a series of phishing attacks against RSA employees, posing as people the employees trusted, to penetrate the company's network.

EMC reported last July that it had spent at least $66 million on remediation. According to RSA executives, no customers' networks were breached. John Linkous, vice president, chief security and compliance officer of eIQnetworks, Inc. doesn't buy it. "RSA didn't help the matter by initially being vague about both the attack vector and (more importantly) the data that was stolen," he says. "It was only a matter of time before subsequent attacks on Lockheed-Martin, L3, and others occurred, all of which are believed to be partially enabled by the RSA breach." Beyond that was psychological damage. Among the lessons, he said, are that even good security companies like RSA are not immune to being hacked.

Jennifer Bayuk, an independent information security consultant and professor at Stevens Institute of Technology, told Search Security in 2012 that the breach was, “a huge blow to the security product industry because RSA was such an icon. They’re the quintessential security vendor. For them to be a point of vulnerability was a real shocker. I don’t think anyone’s gotten over that,” she said.

2.1.15. Stuxnet

Date: Sometime in 2010, but origins date to 2005

Impact: Meant to attack Iran's nuclear power program, but will also serve as a template for real-world intrusion and service disruption of power grids, water supplies or public transportation systems.

Details: The immediate effects of the malicious Stuxnet worm were minimal – at least in the United States – but numerous experts rank it among the top large-scale breaches because it was a cyber-attack that yielded physical results.

Its malware, designed to target only Siemens SCADA systems, damaged Iran’s nuclear program by destroying an estimated 984 uranium enrichment centrifuges. The attack has been attributed to a joint effort by the US and Israel, although never officially acknowledged as such.

2.1.16. VeriSign

Date: Throughout 2010

Impact: Undisclosed information stolen

Details: Security experts are unanimous in saying that the most troubling thing about the VeriSign breach, or breaches, in which hackers gained access to privileged systems and information, is the way the company handled it – poorly. VeriSign never announced the attacks. The incidents did not become public until 2011, and then only through a new SEC-mandated filing.

As PC World put it, “VeriSign buried the information in a quarterly Securities and Exchange Commission (SEC) filing as if it was just another mundane tidbit.”

VeriSign said no critical systems such as the DNS servers or the certificate servers were compromised but did say that "access was gained to information on a small portion of our computers and servers." It has yet to report what the information stolen was and what impact it could have on the company or its customers.

2.1.17. Home Depot

Date: September 2014

Impact: Theft of credit/debit card information of 56 million customers.

Details: The hardware and building supply retailer announced in September what had been suspected for some weeks – that beginning in April or May, its POS systems had been infected with malware. The company later said an investigation concluded that a “unique, custom-built” malware had been used, which posed as anti-virus software.

In March 2016, the company agreed to pay at least $19.5 million to compensate US consumers through a $13 million fund to reimburse shoppers for out-of-pocket losses and to spend at least $6.5 million to fund 1 1/2 years of cardholder identity protection services.

The settlement covers about 40 million people who had payment card data stolen and more than 52 million people who had email addresses stolen. There was some overlap between the groups. The company estimated $161 million of pre-tax expenses for the breach, including the consumer settlement and expected insurance proceeds.

2.1.18. Adobe

Date: October 2013

Impact: 38 million user records

Details: Originally reported in early October by security blogger Brian Krebs, it took weeks to figure out the scale of the breach and what it included. The company originally reported that hackers had stolen nearly 3 million encrypted customer credit card records, plus login data for an undetermined number of user accounts.

Later in the month, Adobe said the attackers had accessed IDs and encrypted passwords for 38 million "active users." But Krebs reported that a file posted just days earlier, "appears to include more than 150 million usernames and hashed password pairs taken from Adobe." After weeks of research, it eventually turned out, as well as the source code of several Adobe products, the hack had also exposed customer names, IDs, passwords and debit and credit card information.

In August 2015, an agreement called for Adobe to pay a $1.1 million in legal fees and an undisclosed amount to users to settle claims of violating the Customer Records Act and unfair business practices. In November 2016, the amount paid to customers was reported at $1 million.

2.2. Related Works

Data Leakage Prevention (DLP), or Information Leakage Prevention (ILP), has been subjected to several types of research and commercial products, where major Information Security vendors struggle for developing innovative technologies. Often regarded as an 'insider threat', data leakage can be treated by employing honeypots or honey-tokens. Where sensitive data is distributed, data can be traced either by 'watermarking' or by unobtrusive techniques. Previous mainstream works were focused on encountering data leakage at three stages. Endpoint solutions enforce access policies in desktop machines and thus can prevent confidential documents from leaving the organization boundaries. Network traffic monitors continuously analyze network communication to identify whether a sensitive file was sent while violating security policies. File-level systems embed security-related information as metadata in the sensitive files [1] [2] [3].

In [3] authors focus on privacy-preserving detection of sensitive data exposure. They presented a data-leak detection solution which can be outsourced and deployed in a semi-honest detection environment. The advantage of their method is that it enables the data owner to safely delegate the detection operation to a semi-honest provider without revealing the private data to the provider.

They used a fuzzy fingerprint technique that enhances data privacy during data-leak detection operations. The data owner preprocesses and prepares fuzzy fingerprints and release the fingerprints to DLD provider. The DLD provider computes fingerprints from the network traffic and identifies potential leaks in them. To prevent the DLD provider from gathering exact knowledge about the sensitive data, the collection of potential leaks is composed of real leaks and noises. He reports all data leak alerts to the data owner. Data owner then post-processes the potential leaks sent back by the DLD provider and decides whether there is any real data leak.

In [4] authors focus on inadvertent leak detection. Detecting the exposure of sensitive information is challenging due to data transformation in the content. Transformations (such as insertion and deletion) result in highly unpredictable leak patterns. In the data leak detection model, they analyze two types of sequences: sensitive data sequence and content sequence. The content sequence is the sequence to be examined for leaks. The content may be data extracted from file systems on personal computers, workstations or payloads extracted from supervised network channels. Sensitive data sequence contains the information (e.g., customer's records, proprietary documents) that need to be protected and cannot be exposed to unauthorized parties. The sensitive data sequences are known to the analysis system. Here they utilized sequence alignment techniques for detecting complex data-leak patterns.

In [5] authors formalize the problem of provably associating the guilty party to the leakages, and work on the data lineage methodologies to solve the problem of information leakage in various leakage scenarios. They define LIME, a generic data lineage framework for data flow across multiple entities in the malicious environment. Three characters are involved- owner, consumer, and auditor. The auditor determines a guilty party for any data leak and defines the exact properties for communication between these roles.
The key advantage of the model is that it enforces accountability by design. This helps to overcome the existing situation where most lineage mechanisms are applied only after the leakage has happened. They present an accountable data transfer protocol to transfer data between two entities. To deal with an untrusted sender and an untrusted receiver scenario associated with data transfer between two consumers, the protocols employ an interesting combination of the robust watermarking, oblivious transfer, and signature primitives. Cox algorithm is used for watermarking.

In [6] the authors study unobtrusive techniques for detecting leakage of a set of objects or records. They developed a model for finding the guilty of agents. They also present algorithms for sharing objects to agents, in a way that enhances the chances of identifying a leaker. Finally, they also considered the choice of adding fake objects to the distributed set. Such objects do not match to real entities but come into sight realistic to the agents. In a sense, here the fake objects act as a type of watermark for the entire set, without modifying any separate members. If an agent was given one or more fake objects that were leaked, then the distributor can be more assured that the agent was guilty.

In [7] authors focus to data leakage prevention system with a time-stamp. In Data Leakage Prevention, the time stamp is very important for giving permission to access a particular data, as in a particular period of time the data is confidential after the time stamp the same data could be non-confidential. In time stamped based DLP two phases are there, Learning Phase and Detection Phase.

In the learning, phase collect confidential and non-confidential documents of an organization. Then create clusters using K-means with cosine similarity function. For each cluster identify the key terms based on their frequency. For each key term calculate the score and assign time stamp for a document based on deadlines of organization schedule. In the detection phase, the tested document is compared with the confidential score and time stamp, if the time stamp of the tested document is greater than or equal to the time stamp then that document is treated as a confidential and it is blocked.

In [8] a new context-based model for accidental and intentional data leakage prevention is proposed. The context-based approach they proposed leverages the advantages of preventing data leakage by either looking for specific keywords and phrases or by using various statistical methods. Their new model consists of two phases: training and detection. During the training phase, they created clusters of documents. Then a graph representation of the confidential content of each cluster is generated. This representation consists of key terms and the context in which they need to appear in order to be considered confidential.

During the detection phase, the document tested is assigned to several clusters. Its contents are then matched to each cluster’s respective graph in an attempt to determine the

confidentiality of the document. One of the main advantages of their method is It detects small sections of confidential information embedded in non-confidential documents. It generates a well-understood model that can be reviewed and even modified by its users.

In [9] authors aims to prevent the data leakage stemming from corporate email. When, an employee sends an email, which contains an attachment, from his corporate account to a recipient, the generated email is forwarded to the SMTP port which accepts outbound emails, on his system. SMTP proxy server can pick up the email and trigger the steganography scanner. Attachments are scanned and if they are clean the email is sent to main corporate server and finally send to the intended recipient. If the attachment is not clean, ie a steganography payload is detected, alert for data leak can be triggered and that email will not be sent.

In [10] authors present a trustworthiness-based distribution model that aims at data leakage prevention. They study the application where there is a distributor, as a trusted party, managing and distributing files that contain sensitive information to authorized users when they require. In their model, first, the distributor calculates the user's trustworthiness based on his historical behaviors. Then according to the user's trustworthiness and his obtained file set overlapping leaked file set, the distributor accesses the probability of the user's intentional leak behavior as the subjective risk assessment. Then the distributor evaluates the user's platform vulnerability as an objective element. Finally, the distributor makes decisions about whether to distribute the file based on the integrated risk assessment.

Another common solution is to encrypt sensitive files, preventing them from being opened in a readable form in a non-authorized environment. First, this paper contributes by presenting novel and a completely different approach to encounter the DLP problem. Second, encryption-based solutions do not prevent the file from being *spread* over the external network once it is leaked. The sensitive document can be sent freely, shared and accessed over the Internet, not only harming the organization's reputation but allowing a motivated adversary to decrypt it. Our method, on the other hand, may halt the initial leakage, and limit the propagation of the leaked document, by preventing access to it. Furthermore, our method may provide forensic evidence concerning the source and route of the tagged file, by analyzing related AV and security systems reports and logs, even outside the organization perimeter. In short, the presented solution can be used along with encryption-based methods to limit the harmful effects of unintentional data leakage.

One commercial solution is Microsoft Information Rights Management (IRM), for Office [4]. This solution prevents unauthorized users, within or outside the organization's boundaries, from reading a Microsoft Office document, which was unintentionally sent to them. However, this kind of solution is limited to Microsoft Office documents and it does not prevent the file from spreading further on the external network once it is leaked. It also does not prevent a skilled hacker from attempting to decipher the contents of the file.

Here we consider another goal: assuming that a sensitive digital document has already leaked somehow, we shall consider a method of containing the leakage to minimize its scope and its damage. Our concept is aimed at a wide range of documents (preferably any kind of file), it aims at impeding and limiting the spreading of the leaked file beyond the organization's boundaries, and it aims at preventing access to the leaked file, preferably by having it deleted before unauthorized users access it.

The term 'unintentional leakage' [10] denoted cases where the leakage of sensitive data is not caused by intentionally malicious actions. This term encloses cases where an insider is unaware of the sensitivity of the data which he deals with or is unaware that some classified material got mixed with the unclassified message or media which he prepares. Consequently, and despite common security measures, sensitive data may escape the organization's boundaries through an insider's unintentional or at least without deliberate malicious intention. We assume that the data may initially escape the organization's boundaries either online (e.g. by email, file sharing services), or offline (e.g. by removable media). We further assume that an ordinary person, who tries to open the leaked document outside the boundaries of the organization, will not mess with it if it appears to be contaminated. Such 'ordinary person' may be either the initial unintentional leaker or someone else.

Note that, despite the rising popularity of "Bring Your Own Device" (BYOD), enterprise organizations are still greatly concerned about their sensitive data being accessed by an unauthorized device or a partially authorized BYOD [11].

CHAPTER 3

Working Principle

3.1. Confidential Data replacement based Data leakage prevention

In Data Leakage Prevention identification of confidential data is very important, along with the data one more parameter i.e. confidential data replacement also considered as an important aspect in the Data Leakage Prevention.

In our confidential data replacement based DLP, two phases are there

3.1.1. Detection Phase

3.1.2. Replacement Phase

In Detection Phase, the documents are identified as confidential documents with predefine confidential data pattern. Fig 1 represents the pictorial representation of the detection phase.

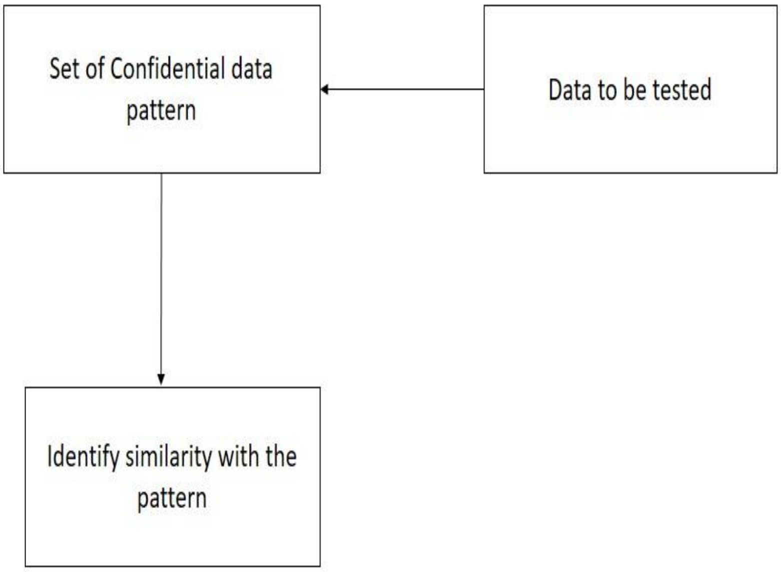

Set of Confidential data pattern
Data to be tested
Identify similarity with the pattern

Fig. 1. Detection phase.

In Replacement Phase, the identified confidential documents are replaced by the **Anonymization technique** where confidential data has been shuffled randomly without seed. Fig 2 represents the pictorial representation of the replacement phase.

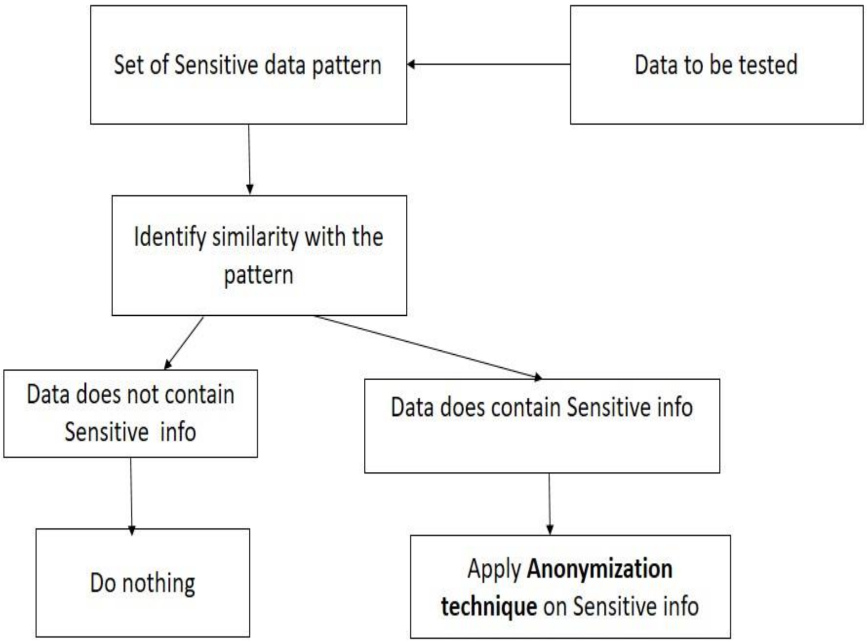

Set of Sensitive data pattern
Data to be tested
Identify similarity with the pattern
Data does not contain Sensitive info
Data does contain Sensitive info
Do nothing
Apply **Anonymization technique** on Sensitive info

Fig. 1. Replacement phase.

Algorithm for confidential information identification and replacement of a document.

Input:

Document corpus

Output:

1. Collection of Confidential and non-confidential documents of an organization
2. Applied **Anonymization technique** on the confidential document.
3. Store both previous & anonymized document in a different place.

3.2. Time Restriction based Data leakage prevention

For each pattern identify the key terms based on their frequency.

This is one of the important steps for our method, for each pattern we should identify the key terms using the concept called language modeling technique. In this method, we used formula for a language model i.e. the term frequency of a particular term divided by a total number of terms in that document. For each pattern language model is created for both confidential and non-confidential documents. The tested document’s keys are compared with confidential data pattern, if the pattern has been matched with the key of the tested document then that document is treated as a confidential and it is assigned with a timestamp for another data user. Fig 3 represents the pictorial representation of Time restriction phase.

3.2.1. Algorithm for detection & assigning time stamp phase

```
1. D- document to be a test
2. Identify similar keys using cosine similarity
3. For each pattern identify documents
4. For each document check the key-pattern matching
5. Calculate confidential score of a document
6. If key-pattern matching score $\geq$ one
7.      D is a confidential document
8.      Assigning a time stamp
9. Else D is a non-confidential document
```

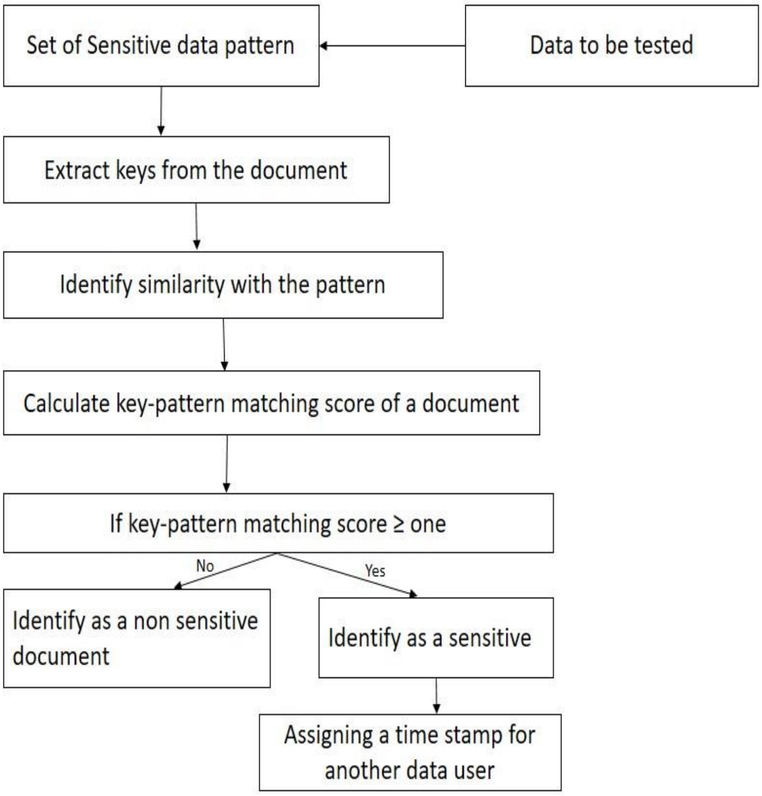

Set of Sensitive data pattern
Data to be tested
Extract keys from the document
Identify similarity with the pattern
Calculate key-pattern matching score of a document
If key-pattern matching score ≥ one
No
Yes
Identify as a non sensitive document
Identify as a sensitive
Assigning a time stamp for another data user

Fig 3 represents the pictorial representation of Time restriction phase.

1. In our method the documents are in text format, so we perform data preprocessing for the tested document. As a part of data preprocessing apply stop words (unnecessary words like is, the, numbers…) and stemming algorithm.

2. With the help of Cosine, similarity function identify the keys for the tested document.

3. Extract the keys from the document. Now the document contains confidential key-pattern matching score.

5. If the key-pattern matching score greater or equal to one then assigning a particular time stamp.

6. The tested document considered as a confidential and after the particular timestamp, this document could be hidden for another data user.

3.3. Two or more time data use restriction

This is one of important steps for our method, after Time restriction phase the tested document's keys are compared with sensitive data pattern, if the pattern has been matched with the key of the tested document then that document is treated as a sensitive then our algorithm checked another data user's opened data file is accessed by him/her two times or more. If the data user opened it the very first time then it shows the data file depends upon time stamp after that the data file must be hidden. If the data user trying to open a file then the data file will be a call to Anonymization technique to change the sensitive information of the opened file data to make valueless. When another data user trying to visit the same data file again and again then the Anonymization technique will be applied on this file after two times open strategy. Fig.4. represents the pictorial representation of two or more time data uses restriction.

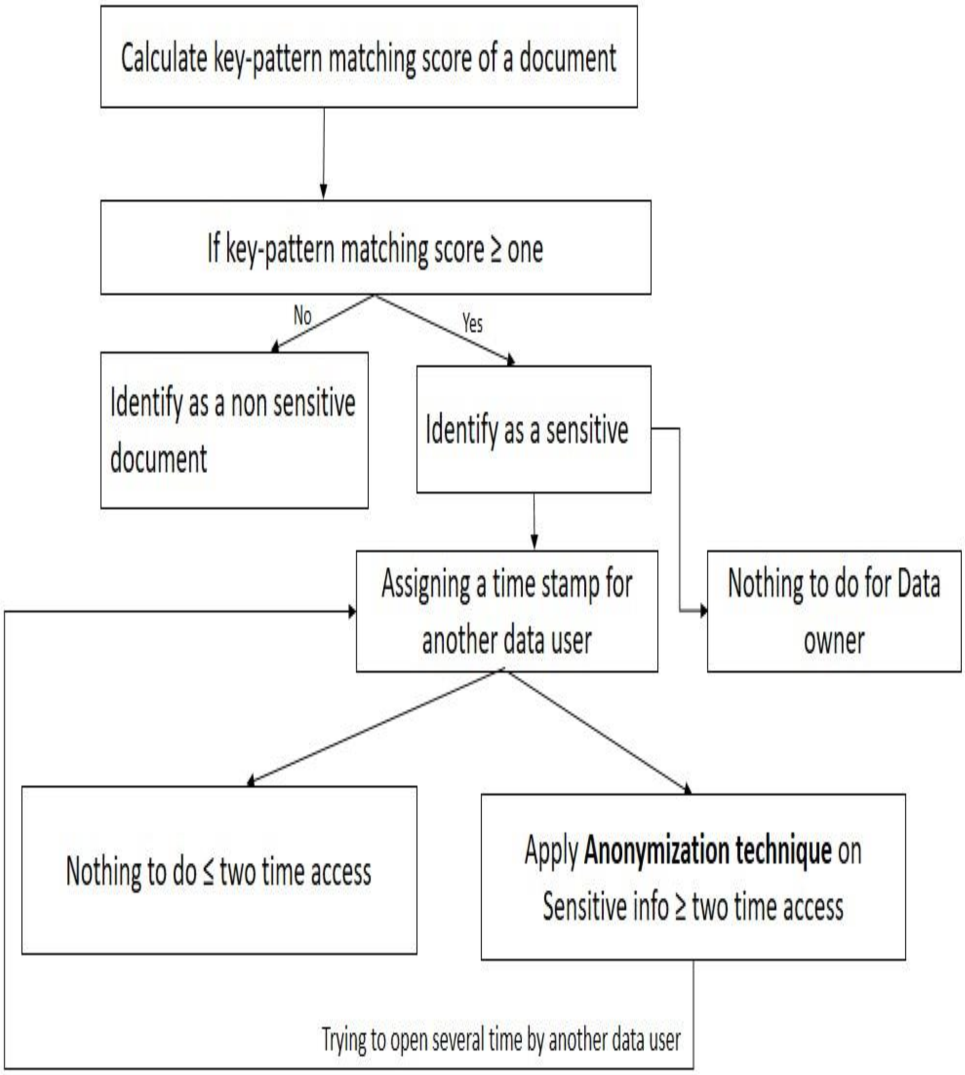

Calculate key-pattern matching score of a document
If key-pattern matching score ≥ one
No
Yes
Identify as a non sensitive document
Identify as a sensitive
Assigning a time stamp for another data user
Nothing to do for Data owner
Nothing to do ≤ two time access
Apply **Anonymization technique** on Sensitive info ≥ two time access
Trying to open several time by another data user

Fig. 4.Two or more time data uses restriction.

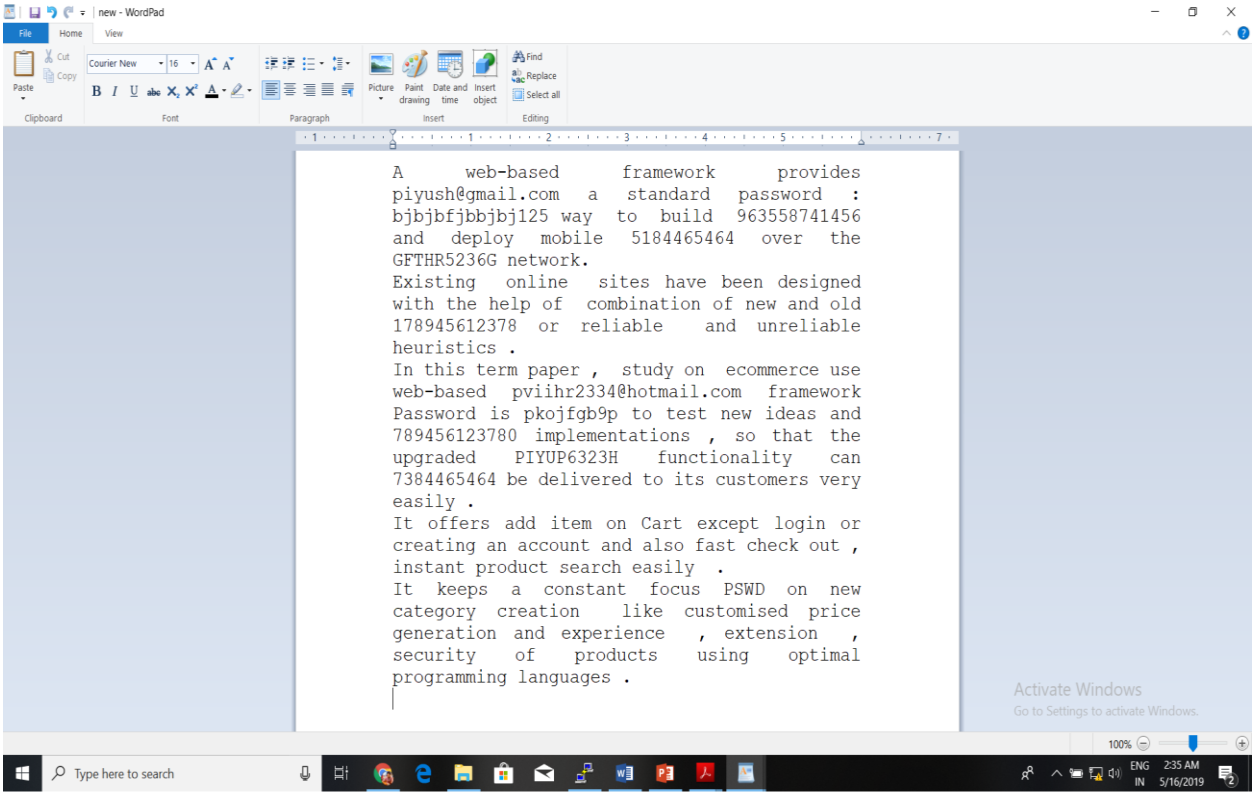
CHAPTER 4

Simulation and Results

4.1. Preconditions

We simulate our method on two way i.e. File data and Database records.

At first we access a file data i.e text file which contains sensitive data and our algorithm which is on detection phase where the file data will be compared with the sensitive data pattern if the data contain sensitive information then the sensitive portion of the text file will be changed through the Anonymization technique, then the sensitive portion of the information will be meaningless. For differentiation, we store the anonymized data file on another data file as well as a text file.



A web-based framework provides piyush@gmail.com a standard password : bjbjbfjbbjbj125 way to build 963558741456 and deploy mobile 5184465464 over the GFTHR5236G network.

Existing online sites have been designed with the help of combination of new and old 178945612378 or reliable and unreliable heuristics .

In this term paper , study on ecommerce use web-based pviihr2334@hotmail.com framework Password is pkojfgb9p to test new ideas and 789456123780 implementations , so that the upgraded PIYUP6323H functionality can 7384465464 be delivered to its customers very easily .

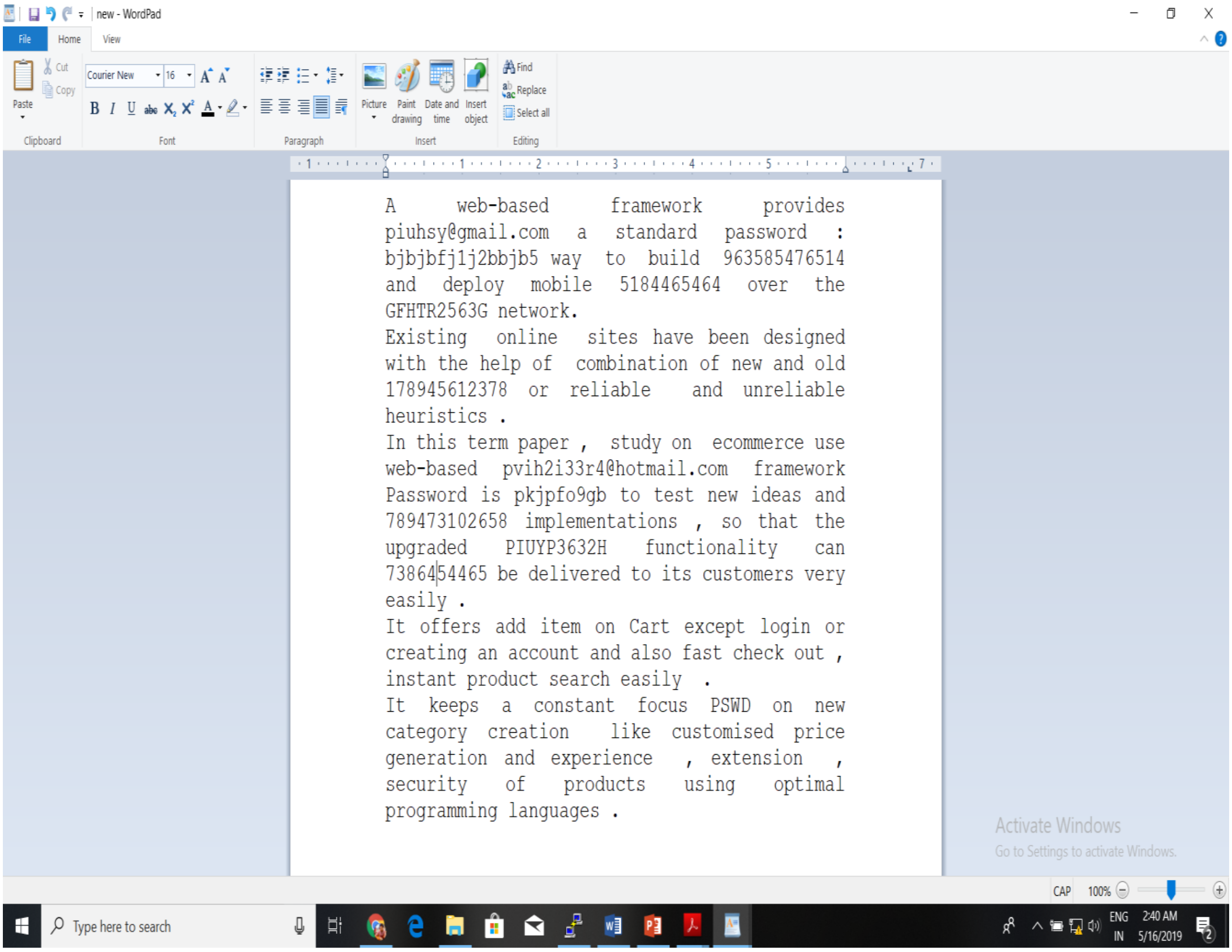
It offers add item on Cart except login or creating an account and also fast check out , instant product search easily .

It keeps a constant focus PSWD on new category creation like customised price generation and experience , extension , security of products using optimal programming languages .

Fig. 5 Text file which contains sensitive information.

4.1.2. Data replacement Phase

On Fig. 5. screenshot denoted that in a text file data where the sensitive information is available. In this text file where email address, password, mobile number, Aadher number, PAN card number which are very sensitive information and also confidential. It is very interesting where the sentences are less meaning full also but sensitive information is very confidential and also it need not leak. Our algorithm pattern matching technique helps to identify the actual sensitive information on the text file.



A web-based framework provides piuhsy@gmail.com a standard password : bjbjbfj1j2bbjb5 way to build 963585476514 and deploy mobile 5184465464 over the GFHTR2563G network.

Existing online sites have been designed with the help of combination of new and old 178945612378 or reliable and unreliable heuristics .

In this term paper , study on ecommerce use web-based pvih2i33r4@hotmail.com framework Password is pkjpfo9gb to test new ideas and 789473102658 implementations , so that the upgraded PIUYP3632H functionality can 7386454465 be delivered to its customers very easily .

It offers add item on Cart except login or creating an account and also fast check out , instant product search easily .

It keeps a constant focus PSWD on new category creation like customised price generation and experience , extension , security of products using optimal programming languages .

Fig. 6 Text file after applying the Anonymization technique.

On Fig. 6. screenshot denoted that the original data file has been changed, where the portion of data mean to sensitive information. Here, email addresses have changed due to Anonymization technique and also password, Aadher card number, mobile number, PAN card number has changed, and made this data to meaningless sensitive data.

4.1.3. Time Restriction phase

This is one kind of interesting and important phase where the whole information or data store in a database, when the data distributor or owner store date that time the algorithm has been applied as to divide into the same type of data one is original data another is after applying Anonymization technique data. When data owner accesses the data then this data fetched by the original database without time restriction. If another data user wants to access another data distributer’s data then the data fetched by anonymized database column, if this data contain sensitive information inside the data then time restriction has been allowed (data will be live for only a few times) unless full data will be live till the end.

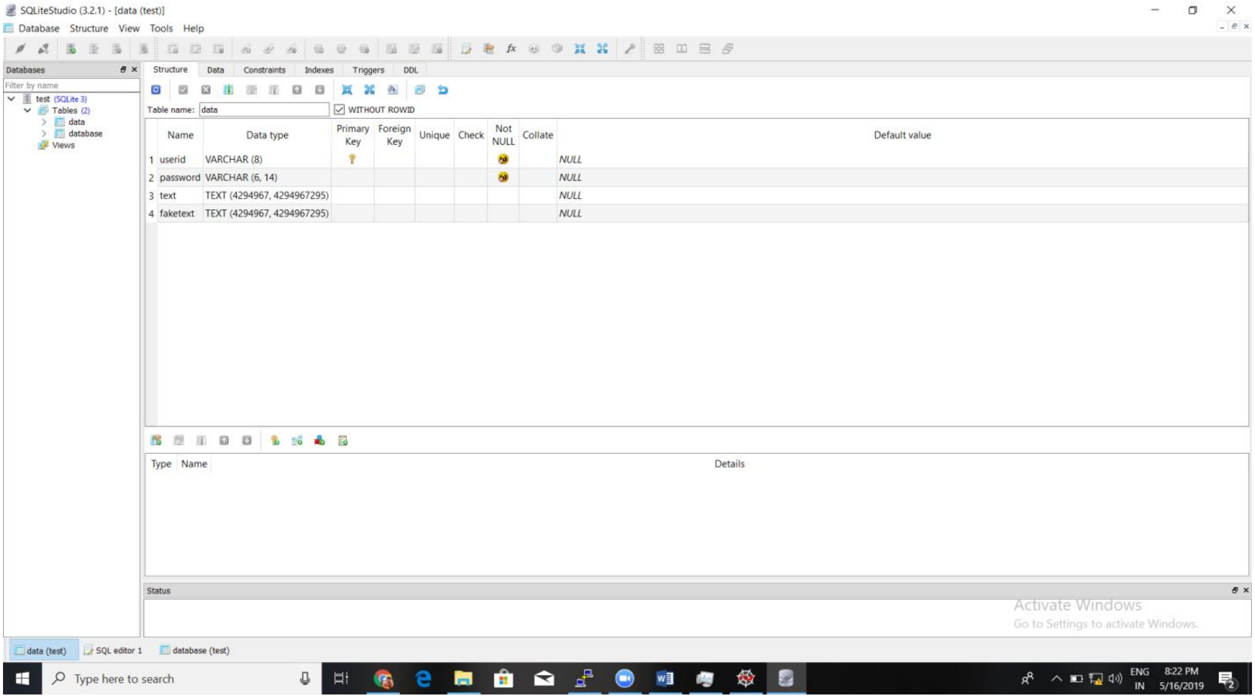
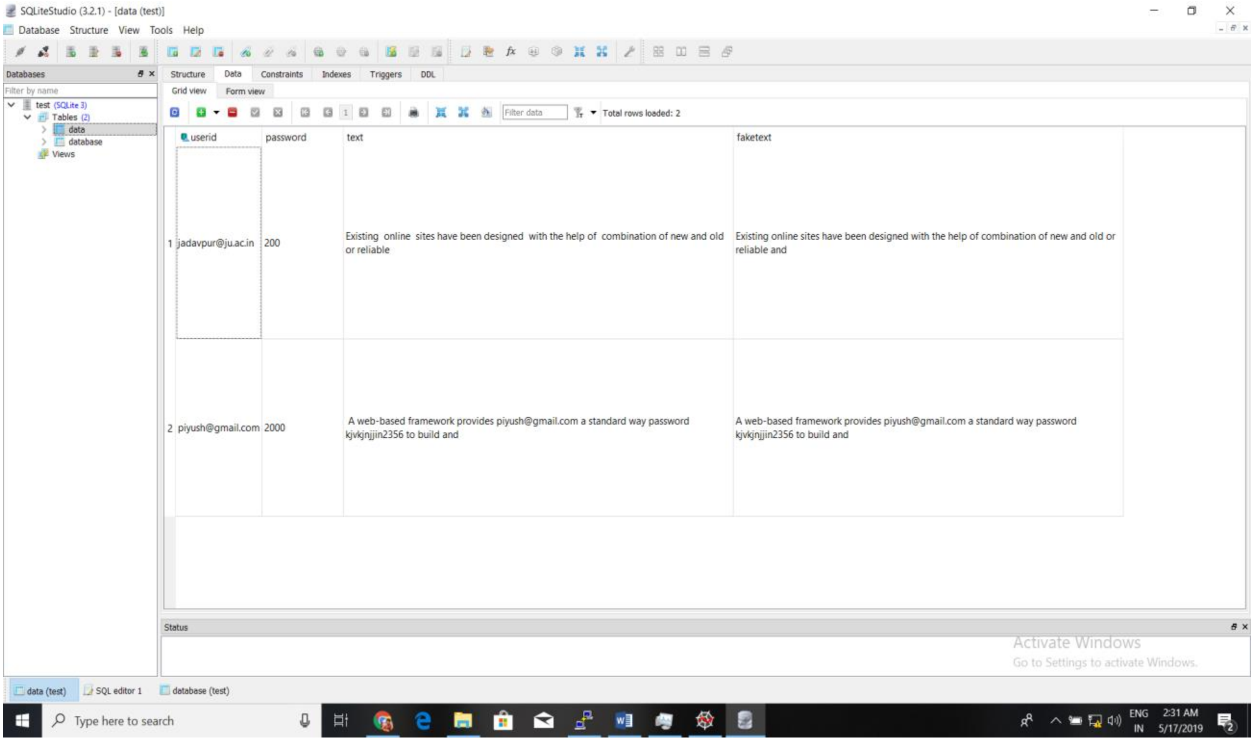

SQLiteStudio (3.2.1) - [data (test)]
Database Structure View Tools Help
Databases
Filter by name
test (SQLite 3)
Tables (2)
data
database
Views
Structure Data Constraints Indexes Triggers DDL
Type Name Details
Status
Activate Windows
Go to Settings to activate Windows.
data (test) SQL editor 1 database (test)
Type here to search
ENG IN
8:22 PM 5/16/2019

Table name: data ☑ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	userid	VARCHAR (8)	🔑				●		NULL
2	password	VARCHAR (6, 14)					●		NULL
3	text	TEXT (4294967, 4294967295)							NULL
4	faketext	TEXT (4294967, 4294967295)							NULL

Fig. 7.1 Structure of Database Table

In Fig. 7.1 The database table denoted where the data distributor store the data and the anonymized data which will be accessed by another data user.



SQLiteStudio (3.2.1) - [data (test)]
Database Structure View Tools Help
Databases
Filter by name
test (SQLite 3)
Tables (2)
data
database
Views
Structure Data Constraints Indexes Triggers DDL
Grid view Form view
Filter data
Total rows loaded: 2
Status
Activate Windows
Go to Settings to activate Windows.
data (test) SQL editor 1 database (test)
Type here to search
ENG IN 2:31 AM 5/17/2019

	userid	password	text	faketext
1	jadavpur@ju.ac.in	200	Existing online sites have been designed with the help of combination of new and old or reliable	Existing online sites have been designed with the help of combination of new and old or reliable and
2	piyush@gmail.com	2000	A web-based framework provides piyush@gmail.com a standard way password kjvkjnjjin2356 to build and	A web-based framework provides piyush@gmail.com a standard way password kjvkjnjjin2356 to build and

Fig. 7.2 Database Table with Data

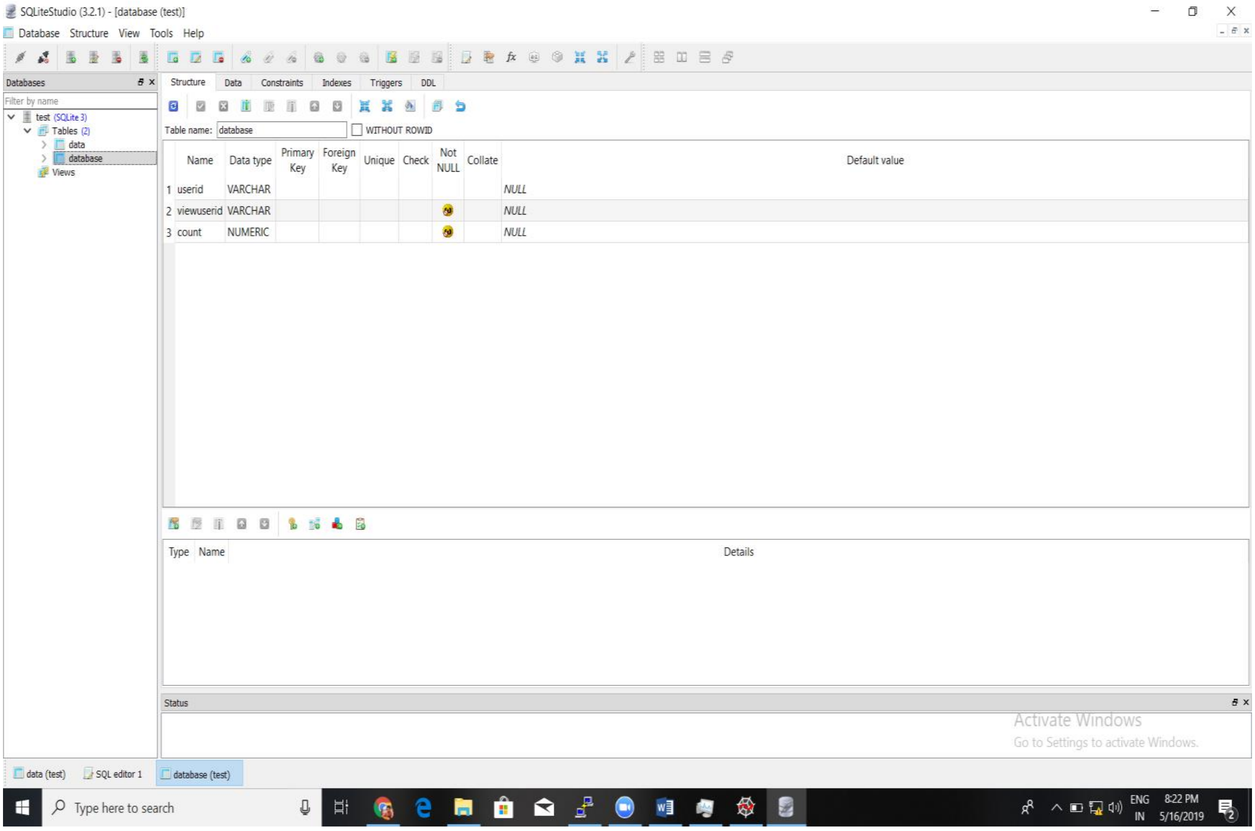
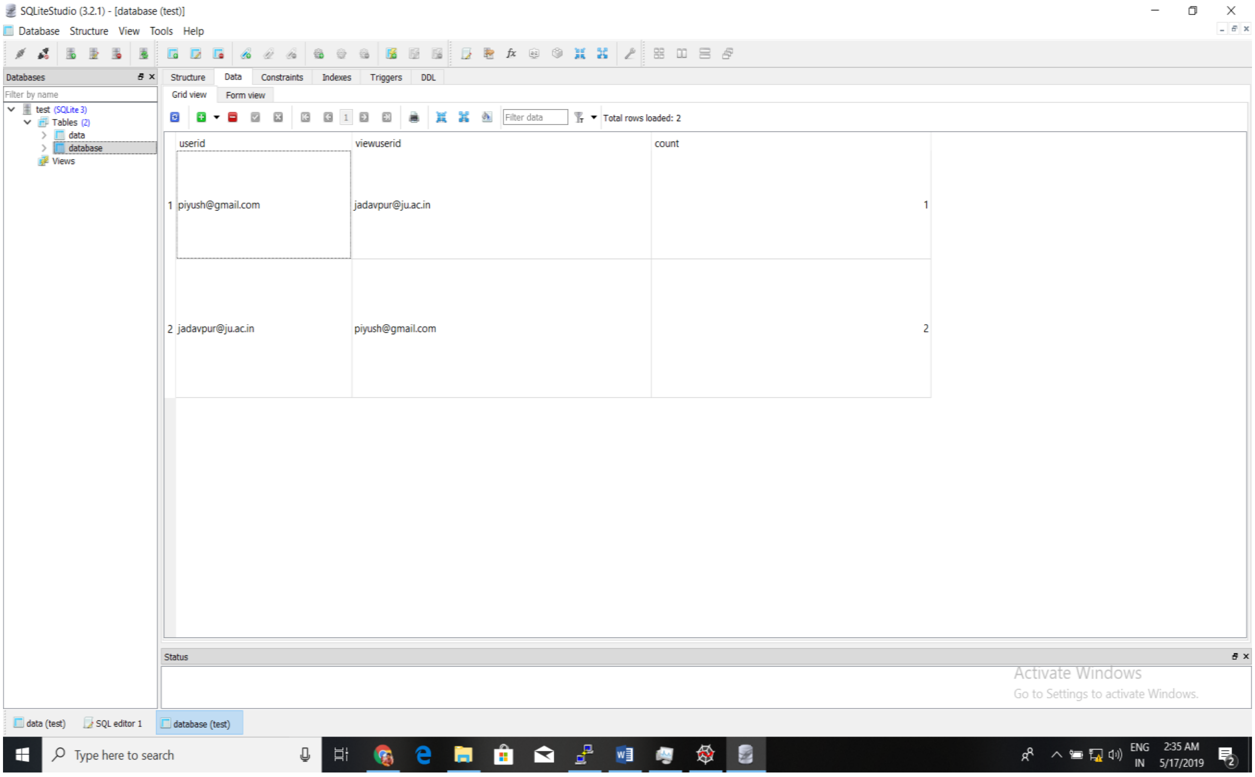

SQLiteStudio (3.2.1) - [database (test)]
Database Structure View Tools Help
Databases
Filter by name
test (SQLite 3)
Tables (2)
data
database
Views
Structure Data Constraints Indexes Triggers DDL
Type Name Details
Status
Activate Windows
Go to Settings to activate Windows.
data (test) SQL editor 1 database (test)
Type here to search
ENG IN 8:22 PM 5/16/2019

Table name: database □ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	userid	VARCHAR							NULL
2	viewuserid	VARCHAR					●		NULL
3	count	NUMERIC					●		NULL

Fig. 8.1 Structure of Time Database Table

In Fig. 8.1 The database table denoted which contain another data user’s user id and to be accessed user id and also a time of access.



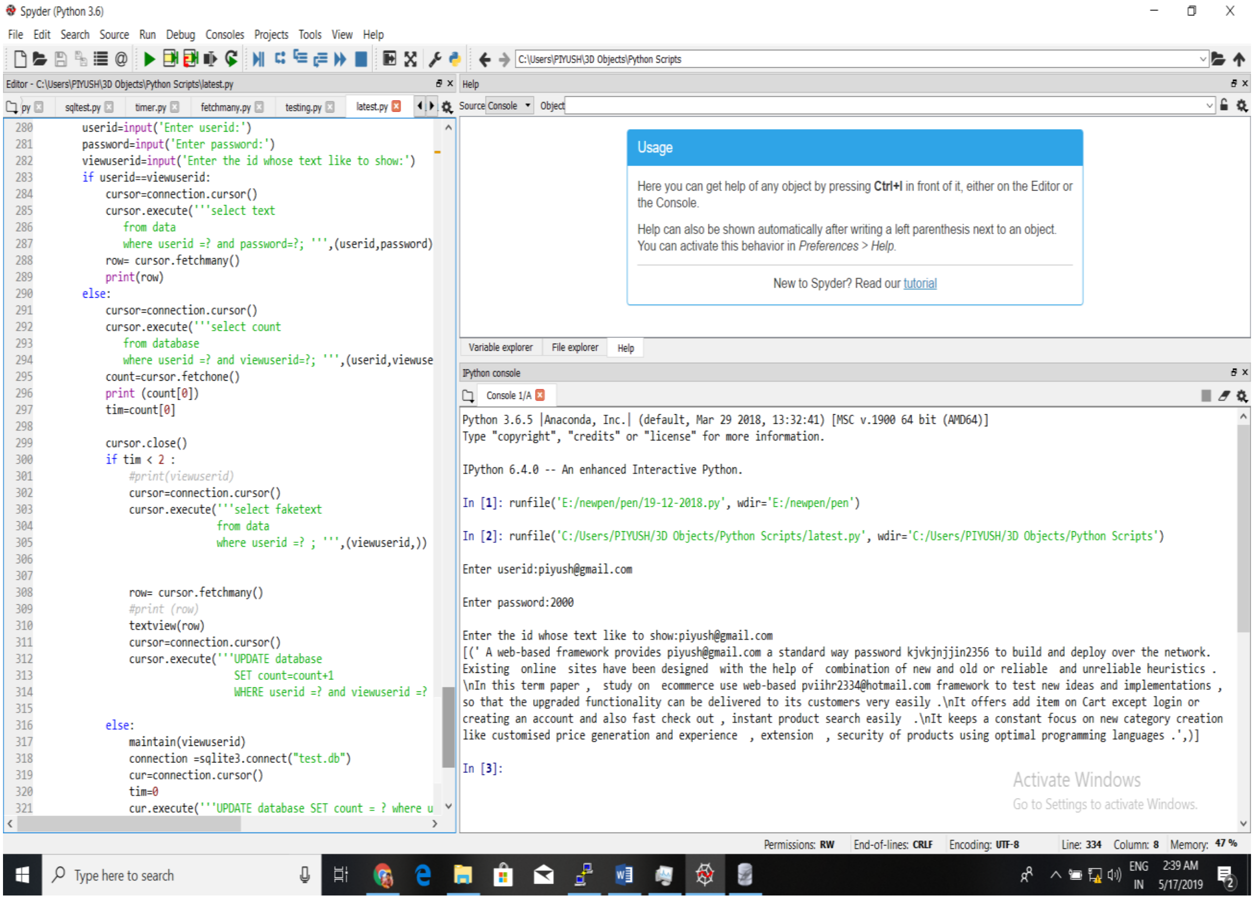
SQLiteStudio (3.2.1) - [database (test)]
Database Structure View Tools Help
Databases
Filter by name
test (SQLite 3)
Tables (2)
data
database
Views
Structure Data Constraints Indexes Triggers DDL
Grid view Form view
Filter data
Status
data (test) SQL editor 1 database (test)
Type here to search

Total rows loaded: 2

	userid	viewuserid	count
1	piyush@gmail.com	jadavpur@ju.ac.in	1
2	jadavpur@ju.ac.in	piyush@gmail.com	2

Fig. 8.2 Time Database Table with data

When Data owner wants to see his /her own data, where data is sensitive or non-sensitive doesn’t matter. He or she will see his /her own data without applying the Anonymization technique and also time restriction. Fig. 9 showing the data when the data distributor showing his/her own data.



```
        userid=input('Enter userid:')
        password=input('Enter password:')
        viewuserid=input('Enter the id whose text like to show:')
        if userid==viewuserid:
            cursor=connection.cursor()
            cursor.execute('''select text
               from data
               where userid =? and password=?; ''',(userid,password)
            row= cursor.fetchmany()
            print(row)
        else:
            cursor=connection.cursor()
            cursor.execute('''select count
               from database
               where userid =? and viewuserid=?; ''',(userid,viewuse
            count=cursor.fetchone()
            print (count[0])
            tim=count[0]

            cursor.close()
            if tim < 2 :
                #print(viewuserid)
                cursor=connection.cursor()
                cursor.execute('''select faketext
                               from data
                               where userid =? ; ''',(viewuserid,))


                row= cursor.fetchmany()
                #print (row)
                textview(row)
                cursor=connection.cursor()
                cursor.execute('''UPDATE database
                                  SET count=count+1
                                  WHERE userid =? and viewuserid =?

            else:
                maintain(viewuserid)
                connection =sqlite3.connect("test.db")
                cur=connection.cursor()
                tim=0
                cur.execute('''UPDATE database SET count = ? where u
```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

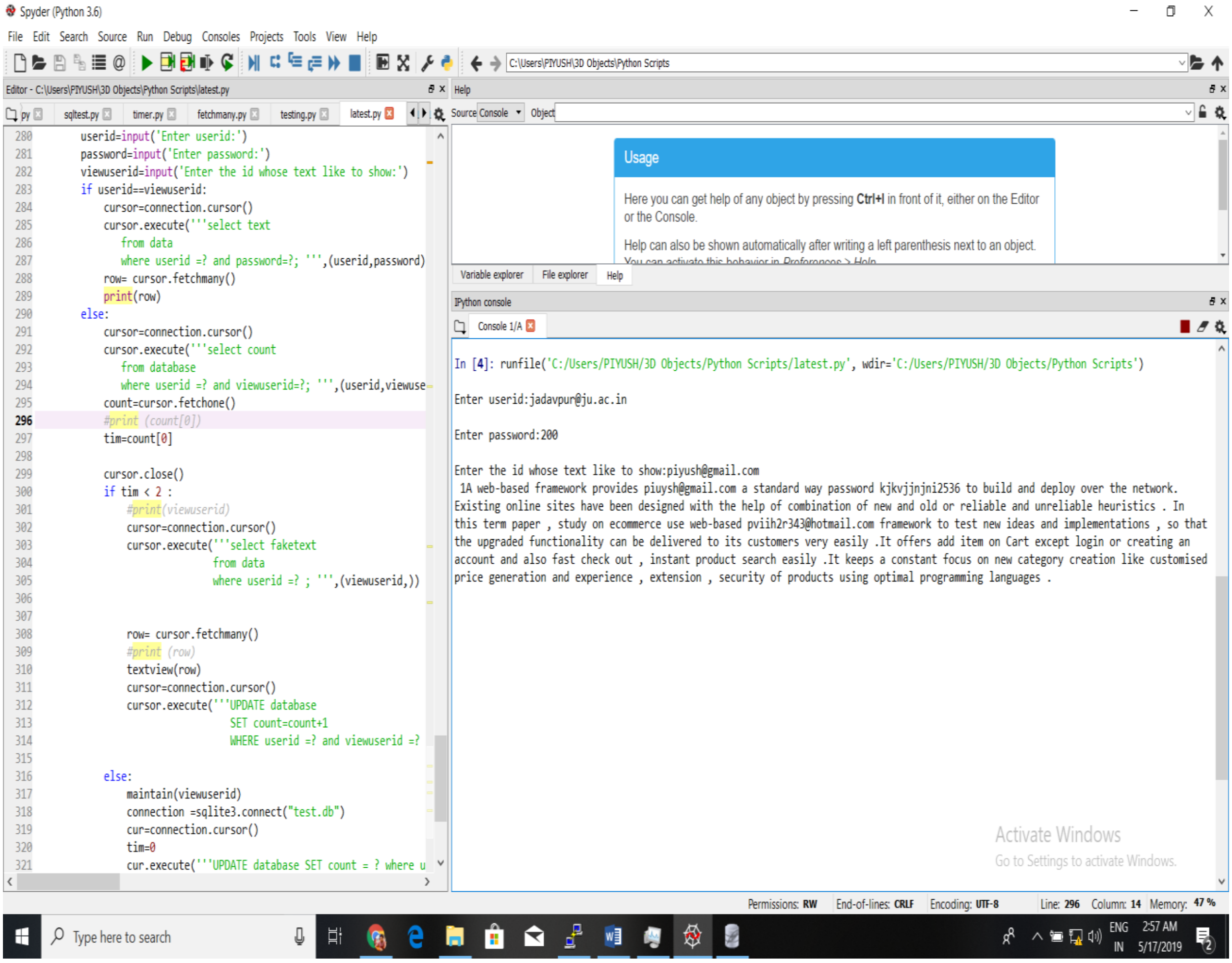
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('E:/newpen/pen/19-12-2018.py', wdir='E:/newpen/pen')

In [2]: runfile('C:/Users/PIYUSH/3D Objects/Python Scripts/latest.py', wdir='C:/Users/PIYUSH/3D Objects/Python Scripts')

Enter userid:piyush@gmail.com

Enter password:2000

Enter the id whose text like to show:piyush@gmail.com
[(' A web-based framework provides piyush@gmail.com a standard way password kjvkjnjjin2356 to build and deploy over the network.
Existing  online  sites have been designed  with the help of  combination of new and old or reliable  and unreliable heuristics .
\nIn this term paper ,  study on  ecommerce use web-based pviihr2334@hotmail.com framework to test new ideas and implementations ,
so that the upgraded functionality can be delivered to its customers very easily .\nIt offers add item on Cart except login or
creating an account and also fast check out , instant product search easily  .\nIt keeps a constant focus on new category creation
like customised price generation and experience  , extension  , security of products using optimal programming languages .',)]

In [3]:
```

Fig. 9 Screenshot of the data owner’s data access

One data user wants to access another data user’s data then he/she shows the anonymized data for a period of time which has been given by time restricted allowing time. Time restriction or anonymization technique will be applied when data must contain sensitive information unless another data user will be shown the same data as the data owner.



Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\PIYUSH\3D Objects\Python Scripts
Editor - C:\Users\PIYUSH\3D Objects\Python Scripts\latest.py
py sqltest.py timer.py fetchmany.py testing.py latest.py
Help
Source Console Object
Usage
Variable explorer File explorer Help
IPython console
Console 1/A
Activate Windows
Go to Settings to activate Windows.
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 296 Column: 14 Memory: 47 %
Type here to search
ENG IN 2:57 AM 5/17/2019

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

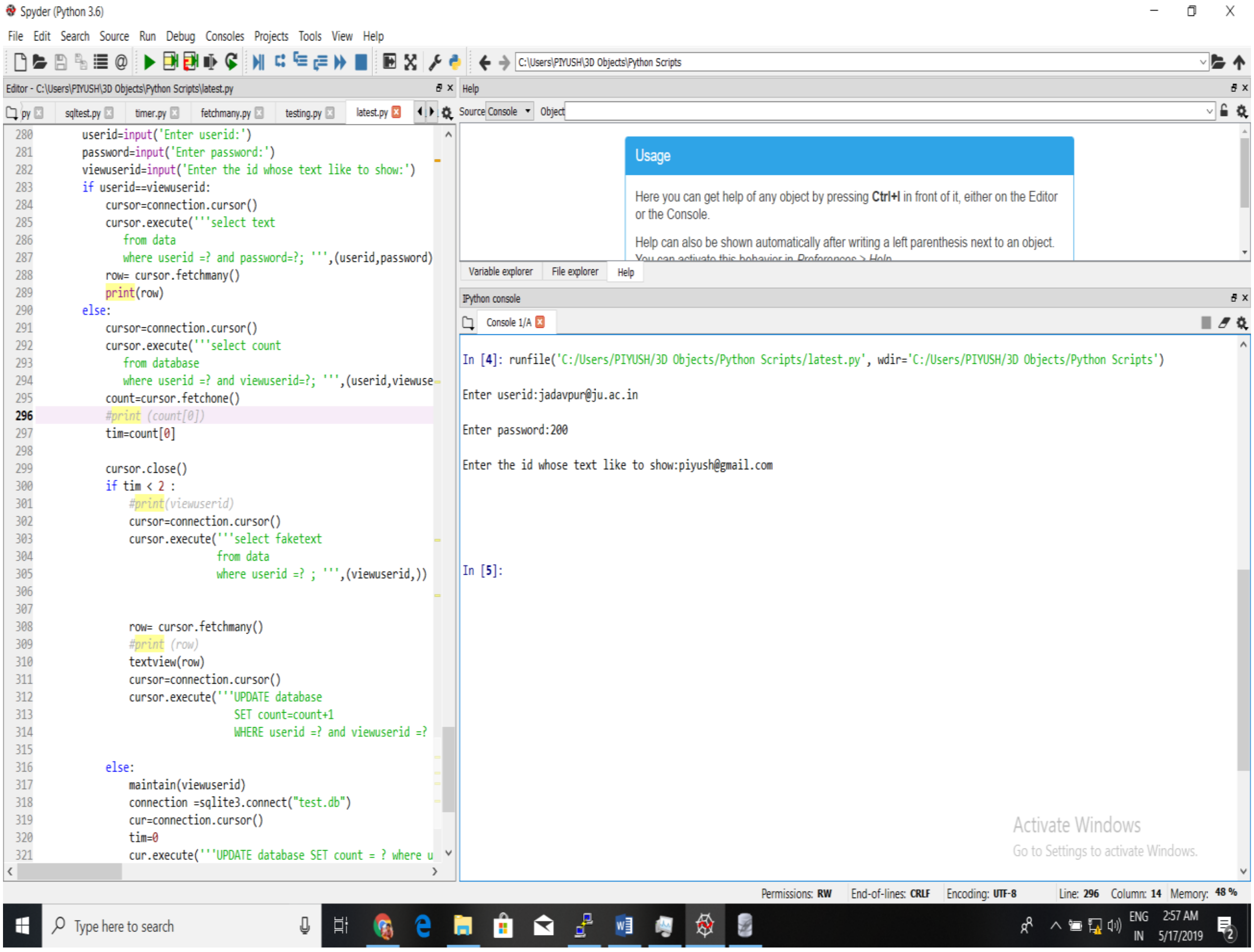
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

```python
        userid=input('Enter userid:')
        password=input('Enter password:')
        viewuserid=input('Enter the id whose text like to show:')
        if userid==viewuserid:
            cursor=connection.cursor()
            cursor.execute('''select text
               from data
               where userid =? and password=?; ''',(userid,password)
            row= cursor.fetchmany()
            print(row)
        else:
            cursor=connection.cursor()
            cursor.execute('''select count
               from database
               where userid =? and viewuserid=?; ''',(userid,viewuse
            count=cursor.fetchone()
            #print (count[0])
            tim=count[0]

            cursor.close()
            if tim < 2 :
                #print(viewuserid)
                cursor=connection.cursor()
                cursor.execute('''select faketext
                               from data
                               where userid =? ; ''',(viewuserid,))


                row= cursor.fetchmany()
                #print (row)
                textview(row)
                cursor=connection.cursor()
                cursor.execute('''UPDATE database
                                  SET count=count+1
                                  WHERE userid =? and viewuserid =?

            else:
                maintain(viewuserid)
                connection =sqlite3.connect("test.db")
                cur=connection.cursor()
                tim=0
                cur.execute('''UPDATE database SET count = ? where u
```

```
In [4]: runfile('C:/Users/PIYUSH/3D Objects/Python Scripts/latest.py', wdir='C:/Users/PIYUSH/3D Objects/Python Scripts')

Enter userid:jadavpur@ju.ac.in

Enter password:200

Enter the id whose text like to show:piyush@gmail.com
 1A web-based framework provides piuysh@gmail.com a standard way password kjkvjjnjni2536 to build and deploy over the network.
Existing online sites have been designed with the help of combination of new and old or reliable and unreliable heuristics . In
this term paper , study on ecommerce use web-based pviih2r343@hotmail.com framework to test new ideas and implementations , so that
the upgraded functionality can be delivered to its customers very easily .It offers add item on Cart except login or creating an
account and also fast check out , instant product search easily .It keeps a constant focus on new category creation like customised
price generation and experience , extension , security of products using optimal programming languages .
```

Fig. 10.1 Screenshot of data where data showing by another data user.

Let’s see if data contain sensitive information data must be applied time restriction and anonymization technique.



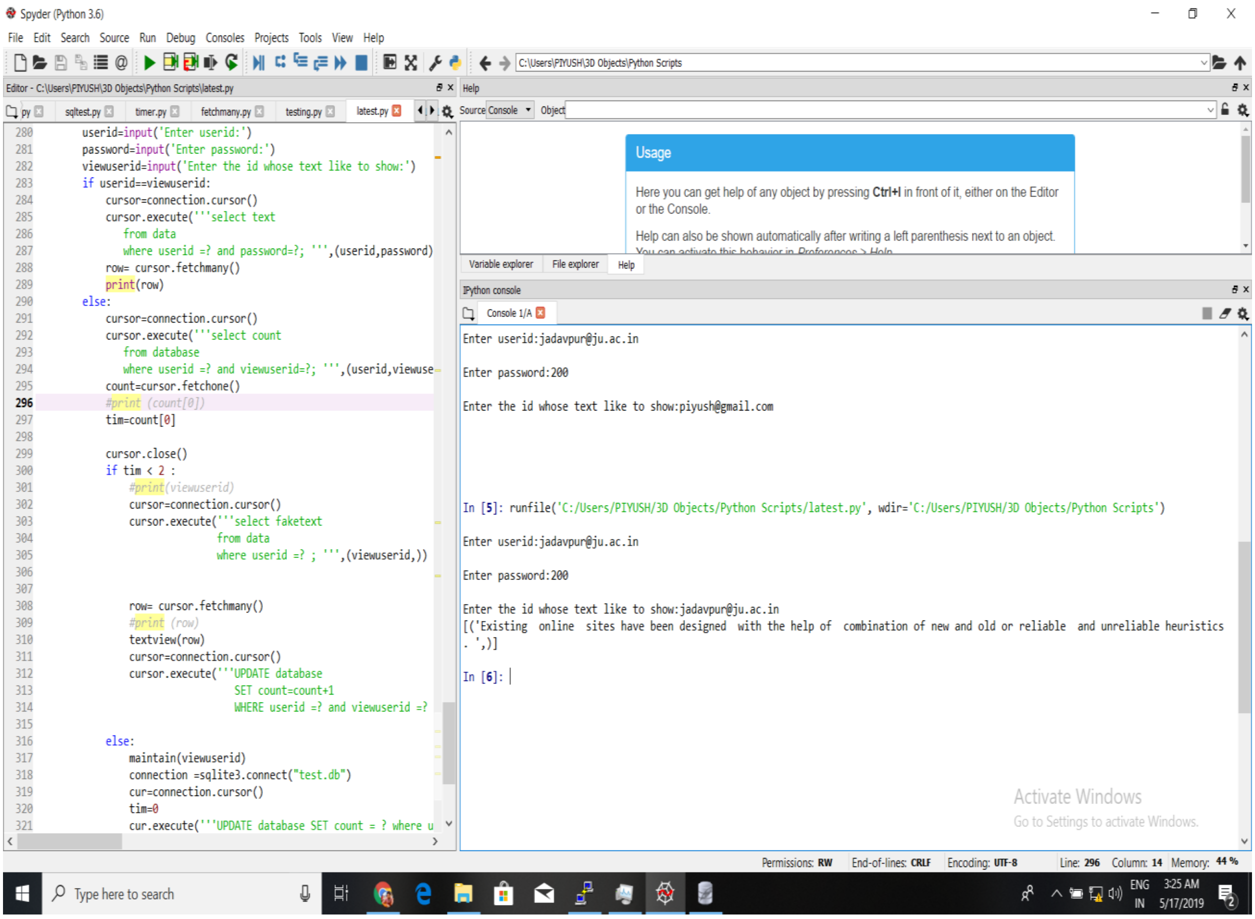
```
        userid=input('Enter userid:')
        password=input('Enter password:')
        viewuserid=input('Enter the id whose text like to show:')
        if userid==viewuserid:
            cursor=connection.cursor()
            cursor.execute('''select text
               from data
               where userid =? and password=?; ''',(userid,password)
            row= cursor.fetchmany()
            print(row)
        else:
            cursor=connection.cursor()
            cursor.execute('''select count
               from database
               where userid =? and viewuserid=?; ''',(userid,viewuse
            count=cursor.fetchone()
            #print (count[0])
            tim=count[0]

            cursor.close()
            if tim < 2 :
                #print(viewuserid)
                cursor=connection.cursor()
                cursor.execute('''select faketext
                               from data
                               where userid =? ; ''',(viewuserid,))


                row= cursor.fetchmany()
                #print (row)
                textview(row)
                cursor=connection.cursor()
                cursor.execute('''UPDATE database
                                  SET count=count+1
                                  WHERE userid =? and viewuserid =?

            else:
                maintain(viewuserid)
                connection =sqlite3.connect("test.db")
                cur=connection.cursor()
                tim=0
                cur.execute('''UPDATE database SET count = ? where u
```

```
In [4]: runfile('C:/Users/PIYUSH/3D Objects/Python Scripts/latest.py', wdir='C:/Users/PIYUSH/3D Objects/Python Scripts')

Enter userid:jadavpur@ju.ac.in

Enter password:200

Enter the id whose text like to show:piyush@gmail.com



In [5]:
```

Fig. 10.2 Screenshot of data where data showing by another data user.

When data contain sensitive information then time restriction technique has been applied on it and after a predefine timestamp then whole data will be hidden as well as flash out.

After that when data does not contain sensitive information then data does not affect by the anonymization technique and also time restriction phase.


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\PIYUSH\3D Objects\Python Scripts
Editor - C:\Users\PIYUSH\3D Objects\Python Scripts\latest.py
py sqltest.py timer.py fetchmany.py testing.py latest.py
Help
Source Console Object
Variable explorer File explorer Help
IPython console
Console 1/A
Activate Windows
Go to Settings to activate Windows.
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 296 Column: 14 Memory: 44 %
Type here to search
ENG IN 3:25 AM 5/17/2019

```python
        userid=input('Enter userid:')
        password=input('Enter password:')
        viewuserid=input('Enter the id whose text like to show:')
        if userid==viewuserid:
            cursor=connection.cursor()
            cursor.execute('''select text
               from data
               where userid =? and password=?; ''',(userid,password)
            row= cursor.fetchmany()
            print(row)
        else:
            cursor=connection.cursor()
            cursor.execute('''select count
               from database
               where userid =? and viewuserid=?; ''',(userid,viewuse
            count=cursor.fetchone()
            #print (count[0])
            tim=count[0]

            cursor.close()
            if tim < 2 :
                #print(viewuserid)
                cursor=connection.cursor()
                cursor.execute('''select faketext
                               from data
                               where userid =? ; ''',(viewuserid,))


                row= cursor.fetchmany()
                #print (row)
                textview(row)
                cursor=connection.cursor()
                cursor.execute('''UPDATE database
                                  SET count=count+1
                                  WHERE userid =? and viewuserid =?

            else:
                maintain(viewuserid)
                connection =sqlite3.connect("test.db")
                cur=connection.cursor()
                tim=0
                cur.execute('''UPDATE database SET count = ? where u
```

Usage

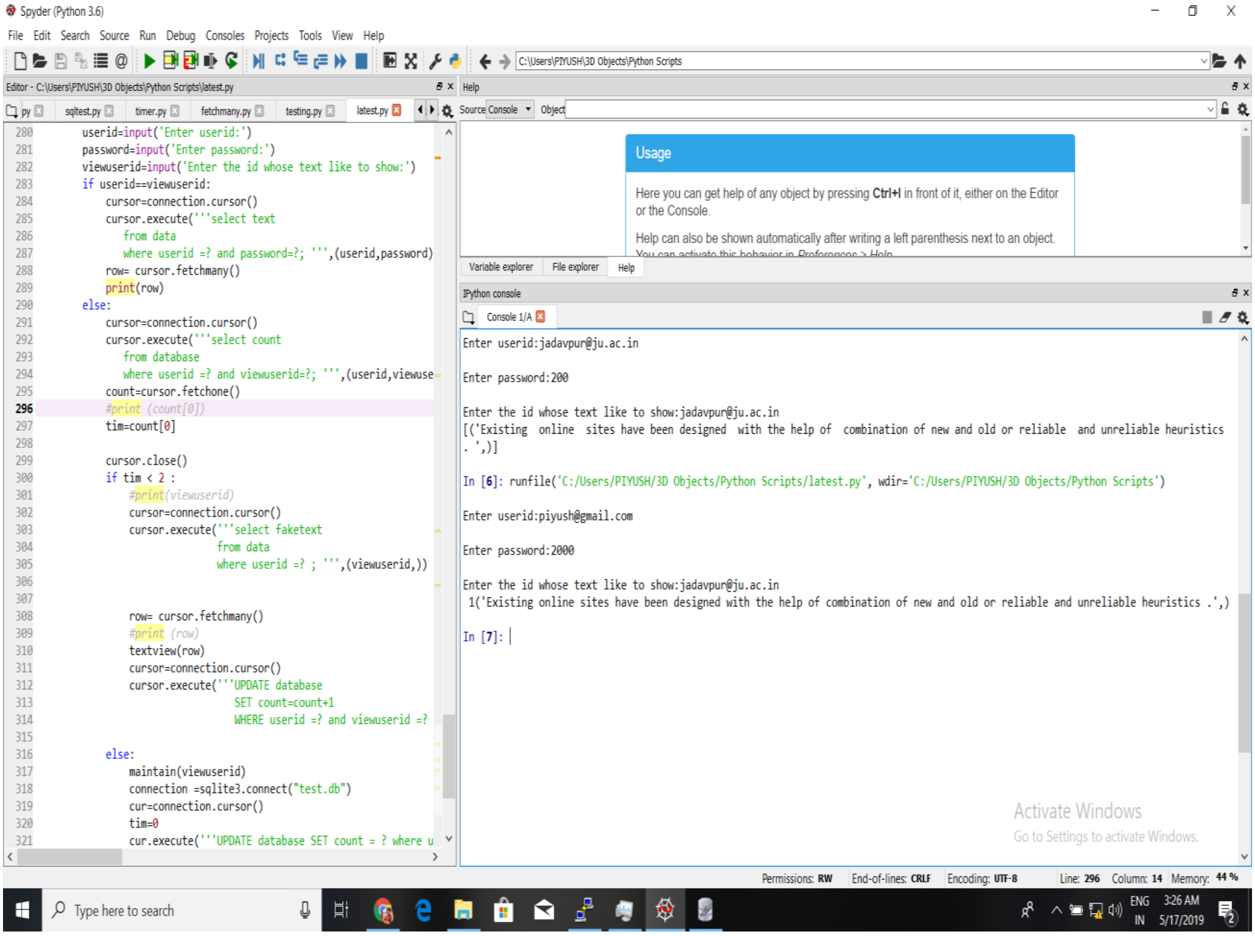
Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

```
Enter userid:jadavpur@ju.ac.in

Enter password:200

Enter the id whose text like to show:piyush@gmail.com




In [5]: runfile('C:/Users/PIYUSH/3D Objects/Python Scripts/latest.py', wdir='C:/Users/PIYUSH/3D Objects/Python Scripts')

Enter userid:jadavpur@ju.ac.in

Enter password:200

Enter the id whose text like to show:jadavpur@ju.ac.in
[('Existing  online  sites have been designed  with the help of  combination of new and old or reliable  and unreliable heuristics
. ',)]

In [6]:
```

Fig. 11.1. Screenshot of data owner data access

In Fig. 11.1, where data does not contain any sensitive information as well as it’s a non-confidential data and it’s shown by the data owner.


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\PIYUSH\3D Objects\Python Scripts
Editor - C:\Users\PIYUSH\3D Objects\Python Scripts\latest.py
py sqltest.py timer.py fetchmany.py testing.py latest.py
Help
Source Console Object
Usage
Variable explorer File explorer Help
IPython console
Console 1/A
Activate Windows
Go to Settings to activate Windows.
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 296 Column: 14 Memory: 44 %
Type here to search
ENG IN 3:26 AM 5/17/2019

```
        userid=input('Enter userid:')
        password=input('Enter password:')
        viewuserid=input('Enter the id whose text like to show:')
        if userid==viewuserid:
            cursor=connection.cursor()
            cursor.execute('''select text
              from data
              where userid =? and password=?; ''',(userid,password)
            row= cursor.fetchmany()
            print(row)
        else:
            cursor=connection.cursor()
            cursor.execute('''select count
              from database
              where userid =? and viewuserid=?; ''',(userid,viewuse
            count=cursor.fetchone()
            #print (count[0])
            tim=count[0]

            cursor.close()
            if tim < 2 :
                #print(viewuserid)
                cursor=connection.cursor()
                cursor.execute('''select faketext
                                from data
                                where userid =? ; ''',(viewuserid,))


                row= cursor.fetchmany()
                #print (row)
                textview(row)
                cursor=connection.cursor()
                cursor.execute('''UPDATE database
                                  SET count=count+1
                                  WHERE userid =? and viewuserid =?

            else:
                maintain(viewuserid)
                connection =sqlite3.connect("test.db")
                cur=connection.cursor()
                tim=0
                cur.execute('''UPDATE database SET count = ? where u
```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

```
Enter userid:jadavpur@ju.ac.in

Enter password:200

Enter the id whose text like to show:jadavpur@ju.ac.in
[('Existing  online  sites have been designed  with the help of  combination of new and old or reliable  and unreliable heuristics
. ',)]

In [6]: runfile('C:/Users/PIYUSH/3D Objects/Python Scripts/latest.py', wdir='C:/Users/PIYUSH/3D Objects/Python Scripts')

Enter userid:piyush@gmail.com

Enter password:2000

Enter the id whose text like to show:jadavpur@ju.ac.in
 1('Existing online sites have been designed with the help of combination of new and old or reliable and unreliable heuristics .',)

In [7]:
```

Fig. 11.2 Screenshot of data where data showing by another data user.

In Fig. 11. 2 where data does not contain sensitive information then data does not effect by the anonymization technique and also time restriction phase when data is shown by another data user.

CHAPTER 5

Conclusion & Future Work

The use of the internet for communication purpose has rapidly increased and it magnified the attacks to users. Protecting the data is a big challenge for computer users. The leak of sensitive data on computer systems poses a serious threat to organizational security. Statistics show that the lack of proper encryption on files and communications due to human errors is one of the leading causes of data loss. Data Leakage Prevention with Anonymization technique best suited for both large and small text dataset. Where Anonymization technique deals with sensitive information to make sensitive information to meaningless sensitive information in a data which is accessed by another data user, and Time restriction policy help to not to give any scope to identify the actual sensitive or confidential information of the data to another data user. Our method used by different application where we need to match the content of the documents with predefined sensitive data pattern. Documents with sensitive or non-sensitive content are almost detected by our method and also prevented by Anonymization technique and Time restriction policy.

In the future, we extend our method for detecting sensitive information or non-sensitive information in any kind of documents and prevent them from the leak.

CHAPTER 6

Bibliography

1. A. Shabtai, Y. Elovici and L. Rokach, "A Survey of Data Leakage Detection and Prevention Solutions," Springer, 2012

2. Z. Xiaosong, L. Fei, C. Ting and L. Hua, "Research and Application of the Transparent Data Encpryption in Intranet Data Leakage Prevention," Computational Intelligence and Security, 2009. CIS '09. , vol. II, pp. 376-379, 2009.

3. XiaokuiShu, Danfeng Yao and Elisa Bertino, “Privacy- Preserving Detection of Sensitive Data Exposure”, IEEE Transactions on Information Forensics and Security, 1092- 1103.

4. XiaokuiShu and Jing Zhang, Danfeng Daphne Yao and Wu Chun Feng, “ Fast Detection of Transformed Data Leaks, IEEE Transactions on Information Forensics and Security”, 528-542.

5. Michael Backes ,Niklas Grimm and Aniket Kate, “Data Lineage In Malicious Enviornments, IEEE Transactions on Dependable and Secure Computing”, 178-191.

6. P. Papadimitriou, H. Garcia-Molina, “Data Leakage Detection”, IEEE Transactions On Knowledge And Data Engineering, 51-63.

7. SubhashiniPeneti and B. Padmaja Rani, “Data Leakage Prevention System WithTimeStamp”, International Conference on Information Communication and Embedded Systems, 1-6.

8. Gilad Katz, Yuval Elovici, and BrachaShapira, “CoBAn: A context based model for data leakage prevention”, Information science on Springer.

9. VeronikiStamatiKoromina and Christos Ilioudis, “Insider Threats in Corporate Environments: A Case Study for DLP”, in Proc. ACM.

10. Limiting Access to Unintentionally Leaked Sensitive Documents Using Malware Signatures.

11. Scarfo, A. 2012. "New security perspectives around BYOD," In Proceedings of the 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications (pp. 446-451). IEEE Computer Society.

12. Raschke, T. "The Forrester Wave ™: Data Leak Prevention, Q2 2008," Technical report, Forrester Research, Inc. 2008.

6. Lawton, G. "New technology prevents data leakage," Computer 41.9 (2008): 14-17.

13. Spitzner, L. "Honeypots: Catching the insider threat," Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, 2003.

14. Storey, D. "Catching flies with honey tokens," Network Security 2009.11 (2009): 15-18.

15. Papadimitriou, P, and Garcia-Molina, H. "Data leakage detection," Knowledge and Data Engineering, IEEE Transactions on 23.1 (2011): 51-63.

16. C. Phua, "Protecting organisations from personal data breaches," Computer Fraud & Security, vol. 2009, no. 1, p. 13–18, 2009.

17. Microsoft, "About Information Rights Management," Microsoft Office Website, 2013. [Online]. Available:http://office.microsoft.com/en-us/help/about-information-rights-management-HP006220859.aspx.

18. Simon Liu, Rick Kuhn , " Data Loss Prevention " , IEEE,USA ,2010.

19. Preeti Raman, Hilmi GUne□ Kayaclk, and Anil Somayaji ,"Understanding Data Leak Prevention",ANNUAL SYMPOSIUM ON INFORMATION ASSURANCE (ASIA) Journal , JUNE 7-
8,2011.

20. ISACA,"Data Leak Prevention", vailable, http://www.isaca.org/KnowledgeCenteriResearch/ ResearchDeliverables/Pages/DataLeak-Preventi on. aspx.

21. Tomoyoshi Takebayshi ,Hiroshi Tsuda , takayuki Hasebe ," Data Loss Prevention Technologies " , FUJITSU SCI,2010.