|  |  |
| --- | --- |
| **CONTENTS**  Candidate’s declaration |  |
| Acknowledgements |  |
| Certificate |  |
| Contents | **i** |
| List of Figures | **iii** |
| List of Tables | **v** |
| Abstract | 1 |

1. **INTRODUCTION 3**
2. Evolution of cloud computing 3
3. The Could Environment 7
4. Cloud Security Issues 10
5. Cloud Threat Models 12
6. Steganography as a solution of the threats 13
7. Types of Stegnography 16
8. Scope and Objective 16

1.7.1 Study of cloud computing 17

1.7.2 Study of threats and security issues in cloud computing 17

1.7.3 Study of image steganography for security in clouds 17

1.7.4 Encryption of data on cloud using image steganography 18

with LSB technique

1.7.5 Decryption of data on cloud using image steganography 18

with LSB technique

1. Organization of dissertation 18
2. **LITERATURE SURVEY 20**
3. Emergence and growth of cloud computing 20
4. Making the clouds secure: issues and remedies 26
5. Steganography for secure data transmission on clouds 28
6. **EMBEDDING THE DATA IN AN IMAGE 33** 
   1. Introduction 33
   2. Image stegnography 34

3.2.1 Image definition 34

3.2.2. Image Compression 35

3.2.3 Image and Transform Domain 35

* 1. Evaluation of different techniques 41
  2. Least – Significant Bit (LSB) Technique 44

3.4.1 Data Embedding 47

3.4.2 Process of image dividing 48

3.4.3 Encryption process (Cover\_Image, Data\_File) 49

3.4.4 Stegnography (Cover\_Image, Byte\_Array, Num) 51

* 1. The automated system to upload the data on cloud 52
  2. Summary 58

1. **EXTARCTION OF DATA FROM IMAGE 59**
2. Introduction 59
3. Data Extraction 60
   1. Decription (Size\_data\_file, Number\_file) 60
   2. Automated procedure to recover the image 62
   3. Summary 67
4. **ANALYSIS OF IMAGE STEGNOGRAPHIC METHOD FOR DATA 68**

**SECURITY ON CLOUDS**

1. Introduction 68
2. The Taken Example 68
3. Analysis of the designed system 71
4. Graphical analysis of the images 73
5. Summary 77
6. **CONCLUSIONS AND FUTURE SCOPE 78**
7. **REFERANCES vi**
8. **PUBLICATION xi**

**List of Figures**

**Figure Title Page**

1. Types of steganography 14
2. Image steganography process 15
3. Visualization based computing system 21
4. Initial cloud server 23
5. A view of image stegnography 34
6. Categories of image steganography 36
7. Embedding the data using image steganography 47
8. Process of LSB encoding 47
9. The flow chart of data upload over cloud 49
10. Home page of the designed system 53
11. Login page of the designed system 53
12. Login process of the designed system 54
13. Client account window 54
14. Selection of file for upload 55
15. Submission process of file and cover image 56
16. Created copies of the cover image 56
17. Uploading of images to cloud 57
18. Creation of multiple stegao images 57
19. The steganographic process 59
20. Flowchart for data download from cloud 61
21. The download window 62
22. The verification procedure using onetime pass word 63
23. Download options after verification 63
24. Download options after verification 64
25. File options in the window 64
26. Name wise sorting of the files 65
27. Size wise sorting of the files 65
28. Type wise sorting of the files 66
29. Regestration window to be authorized client 66
30. The cover image 68
31. (a) Stegano Image 1 (b) Difference from cover image 70
32. RGB plot of the cover image 73
33. RGB plot of the Stegano image (1) 74
34. RGB plot of the Stegano image (2) 74
35. RGB plot of the Stegano image (3) 75
36. RGB plot of the Stegano image (4) 75
37. RGB plot of the Stegano image (5) 76
38. RGB plot of the Stegano image (6) 76

**List of Tables**

**Table Title Page**

TABLE 1 COMPARISION OF IMAGE STEGNOGRAPHY ALGORITHMS 9

TABLE 5.1 CORRELATION COEFFICIENTS OF ALL IMAGE COMPONENTS 72

TABLE 5.2 ENTROPY OF DIFFERENT IMAGES 73

**ABSTARCT**

The term cloud computing is now and then used to represent a new standard it can be understood as a new technology that offers IT resources and services over the Internet. The technology analysts see cloud computing as a so-called “emerging technology”on its way to the publicity. According to one more definition “Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds.” Standardizing a common protocol for the Internet leads straightforwardly to quick enlargement in the number of users online. This has motivated technologists to create even more changes in present protocols and to make new ones. Using technologies such as server virtualization, parallel processing, vector processing, symmetric multiprocessing, and massively parallel processing has fueled radical change. The major aspects to think about are security and privacy concerns which are considered the primary threats to a wide utilization of clouds. The new concepts that are included in clouds are multi-occupancy, resource sharing and outsourcing; demand a very high security level. The remedies to these issues demand the system to be contained with ability to provide and tune the security measures developed for traditional computing systems, containing new security policies, models, and protocols to address the unique cloud security concerns. In this work, we provide a comprehensive study of cloud computing security and privacy concerns. The cloud computing has a wast possibilities and exciting features but also comes with the threats of the data security as the data is available on the cloud so some secure mechanism should be introduced to make sure that the data which belongs to the specific client must be explored to the respective client only, data security and resource sharing responsibility between cloud provider and its customer for security and privacy and Providing secure multi-client environment (safe and sound and well-organized partitioning of virtualized and shared transportation among various clients). Steganographic techniques can be used to provide a perfect tool for data security and encryption, to permit network attacks or secret communication among secret parties. The aim of these techniques is to hide secret data (stegano grams) in the in some carrier e.g. in normal transmissions of clients. In ideal situation hidden data exchange cannot be detected by third parties. The best carrier for steganograms must possess two features: it should be popular i.e. usage such carrier should not be considered as an anomaly itself and modification of the carrier related to inserting the steganogram should not be “visible” to third party not aware of the steganographic procedure.

The work focuses on characterization of information hiding possibilities in Cloud Computing via image steganographic process. After general introduction to cloud computing and its security we move to brief description of steganography. In particular we introduce classification of steganographic communication scenarios in cloud computing which is based LSB hiding scheme and a perfect reconstruction of the data at the receiving end. These scenarios as well as the threats that steganographic methods can cause must be taken into account when designing secure cloud computing services. An analysis is performed over the stegano objects formed between the processes and reconstructed data to ensure a good encryption as well as good recovery.

**Chapter - 1**

**INTRODUCTION**

**1.1 Introduction**

Cloud Computing is a definition associated to a latest fashion in computing service terms. This drift has seen the technological and cultural shift of computing service provision from being provided locally to being provided remotely and en masse, by third-party service providers. These third-parties offer consumers an affordable and flexible computing service that consumers would otherwise not have been accessible, let alone afford. This new means of service provision has evolved from and is the culmination of research stemming from (among others) distributed and networked systems, utility computing, the web and software services research. This paradigm shift has led to computing being seen as another household utility, aka “fifth utility", and has prompted many a business and individual to migrate parts of their IT infrastructure to the cloud and for this data to become managed and hosted by Cloud Service Providers (CSPs). However, Cloud Computing is the cause celebre among tech pundits and has led to the term `Cloud Computing' as an umbrella term being applied to differing situations and their solutions. As such a broad range of definitions for Cloud Computing exists, each of which differ depending on the originating authors' leaning. This chapter seeks to provide a coherent and general introduction to Cloud Computing.

**1.2** **Computing as a Service**

One of the main tenets of Cloud Computing is the `as-a-Service' paradigm in which `some' service is offered by a Service Provider (also known as a Cloud Service Provider) to a User (consumer) for use. This service can also be categorised according to the application domain of its deployment. Examples of application domains that offer services are: Financial e.g. Mint.com, Managerial e.g. Ever Note and Analytical e.g. Google Analytics. The agreed terms of use, indicating the actions that must be taken by both the provider and consumer, are described in a contract that is agreed upon before service provision. Failure to honour this agreement can lead to denial of service for the consumer or legal liability for the service provider. This contract is often described as a Terms of Service or Service Level Agreement. Moreover, as part of this agreement the service provider will provide a Privacy Policy which outlines how the users data will be stored, managed, used and protected.

**1.2.1 Service Levels**

The services offered are often categorised using the SPI Service Model. This model represents the different layers/levels of service that can be offered to users by service providers over the different application domains and types of cloud available. Clouds can be used to provide as-a-Service: software to use, a platform to develop on, or an infrastructure to be utilized. Figure below summarizes the SPI Service Model.

* *Software as a Service:* The first and highest layer is known as: Software as a Service (SaaS). It represents the applications that are deployed/ enabled over a cloud by CSPs. These are mature applications that often offer an API to allow for greater application extensibility. For instance, Google Docs can be seen as the archetypal SaaS application, it has been deployed solely within the Cloud and offers several APIs to promote use of the application.

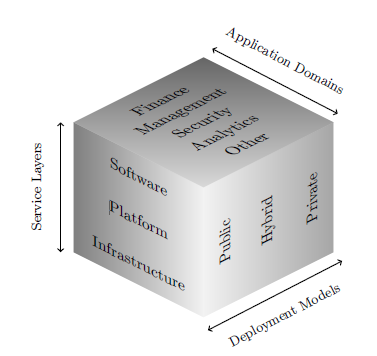


Fig.1 Cloud computing schema

* *Platform as a Service:* The next layer is known as: Platform as a Service (PaaS). This represents a development platform that developers can utilise to write, deploy and manage applications that run on the cloud. This can include aspects such as development, administration and management tools, run-time and data management engines, and security and user management services. For instance, Force.com and Amazon Web Services offers a suite of services that allows developers to construct an application that is deployed using web-based tooling.
* *Infrastructure as a Service*: The final and lowest layer is known as: Infrastructure as a Service (IaaS). CSP offer developers, a highly scaled and elastic computing infrastructure that are used to run applications. This infrastructure can be comprised of virtualised servers, storage, databases and other items. Two well known examples are the Amazon Elastic Compute Cloud, a commercial platform offered as part of Amazon.com's Web Service platform and Eucalyptus, an open source platform that offers the same functionality.

**1.2.2 Entities Involved**

Cloud actors/entities can be divided into two main categories: A) CSP or Service Provider those who provide a service; and B) Cloud Service User (Users) those who use a service. Within Cloud Computing the differences between the role played by a service provider and a user can be blurred. The service provider could also be the user of another service e.g. when infrastructure is the service. The exact definition whether an entity is a provider or user is dependent on the context of the interaction and the service being offered. Some service providers will offer services at all three service levels, or offer just one particular level of service and have their own internal IaaS infrastructure. A possible refinement could be that CSP providers are either: a) Infrastructure Service Providers those that offer IaaS and own and run the data centers that physically house the servers and software; or b) Service Providers those that offer PaaS or SaaS services. And that Cloud Service Users are either: A) Platform Users| are users who buy into a service providers platform e.g. Facebook; and B) Consumers are service users who use either SaaS or IaaS services.

**1.2.3 Defining the Cloud**

The term `cloud' has been used traditionally as a metaphor for networks and helps abstract over their inherent complexity. This term, however, has evolved to encompass the transparency between the technological infrastructure of the CSP and the consumers point of view. A cloud can be one of the following types:

* *Public*: Constituting publicly accessible services that are accessed over the Internet and are often described using the term The Cloud".
* *Private*: These are private services deployed on private networks. Such clouds may also bemanaged by third parties.
* *Hybrid*: A combination of services offered both privately and publicly. For example core-services may be offered on a private cloud; other services originate from public clouds.

“a large pool of easily usable and accessible virtualized resources (such as hard- ware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs."

The provision of virtualised resources, as a service, allows for an elastic and on demand computing environment that can be expanded and shrunk as needed, depending on the needs of the consumer. Mather, Kumaraswamy and Latif provides a slightly alternate definition. This definition is based on five attributes that can be used to describe a cloud-based system. They are:

* *Multi-tenancy*: The sharing of resources being offered by the service providers to the consumers at the network, host and application level.
* *Massive Scalability:* The ability to scale the storage, bandwidth and systems available to proportions unattainable if performed by the organization itself.
* *Elasticity:* Consumers can rapidly and dynamically increase and decrease the resources required as needed.
* *Pay-as-you-go*: Consumers only pay for the resources they consume such as processing cycles and disk space.
* *Self-Provisioning of Resources:* Consumers have the ability for the self-provisioning of resources.

This is a better definition; it does not pertain to a particular technology. However, one attribute Pay-as-you-go is in itself too constrictive as it pertains to a particular style of payment and prohibits subscription based models. A better attribute should be:

Flexible Payment Payment for resource usage is flexible and consumers can be offered different payment models dependent on their intended use of the resources.

For instance in Amazon EC2 pricing for image use is dependent on three factors: 1. Image Type On Demand, Reserved, Spot 2. OS Used UNIX/LINUX or Windows; and 3. Data Center Location West Coast America, East Coast America or Ireland. Consumers are billed differently per hour based upon these factors and other services used [AWS-Pricing]. On the other-hand Google Apps Premium Edition will cost companies $50 per user per year for their IT infrastructure solution [Goo Apps] and users will have a fixed set of resources. The use of such models often ensures for a lower operational cost than when compared with an in-house solution.

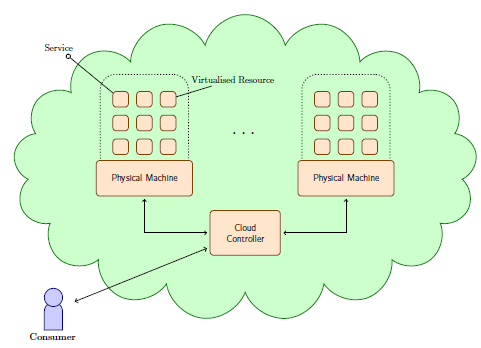


Fig.2 Cloud system overview

When discussing the physical constructions of Clouds the Data Center Model is a popular choice. This model stipulates clusters of machines running specialist, dedicated hardware and software to provide the infrastructure for the large scale provision of virtualised resources. Though a consumer can access the virtualised resources directly, when managing their resources, consumers interact with a Cloud Controller that mediates the interaction between the consumer, and: a) the physical machines; and b) the virtualised resources owned by the consumer. Figure above illustrates a rather simplified view of this data center model. A more interesting realisation of clouds is the Adhoc model in which the existing (idle) computing power within an enterprise is tapped to provide the required infrastructure. The more interested reader is asked to read Kirby, Dearle et al. for more information concerning the adhoc cloud model.



Fig.3 Cloud system Model

**1.3** **Cloud system architecture**

Cloud computing systems fall into one of the following five layers: applications, software environment, software infrastructure, software kernel, and hardware.

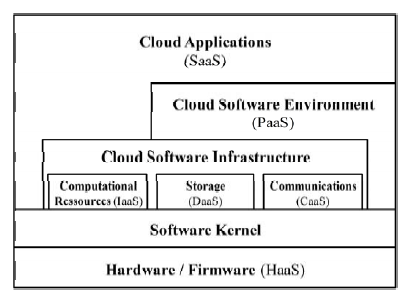


Fig.4 Cloud system architecture

Each layer represents a level of abstraction, hiding the user from all underlying components and thus providing simplified access to the resources or functionality. In the following section we are going to describe each layer of Youseff’s, Butrico’s and Da Silva’s (2008) model.

* *Cloud Application Layer:* When it comes to user interaction, the cloud application layer is the most visible layer to the end customer. It is usually accessed through web-portals and thus builds the front-end, the user interacts with when using cloud services. A Service in the application layer may consist of a mesh of various other cloud services, but appears as a single service to the end-customer. This model of software provision, normally also referred to as *Software-as-a-Service (SaaS)*, appears to be attractive for many users. Reasons for this are the reduction of software and system maintenance costs, the shift of computational work from local systems into the cloud, or a reduction of upfront investments into hardware and software licenses. Also the service provider has advantages over traditional software licensing models. The effort for software upgrades is reduced, since patches and features can be deployed centrally in shorter cycles. Depending on the pricing model a continuous revenue stream can be obtained. However, security and availability aspects are issues that still need to be addressed. Also the migration of user data is a task that should not be underestimated. Examples for applications in this layer are numerous, but the most prominent might be Salesforce’s Customer Relationships Management (CRM) system2 or Google’s Apps, which include word-processing, spreadsheet and calendaring3.
* *Cloud Software Environment Layer:* The cloud software environment layer (also called software platform layer) provides a programming language environment for developers of cloud applications. The software environment also offers a set of well-defined application programming interfaces (API) to utilize cloud services and interact with other cloud applications. Thus developers benefit from features like automatic scaling and load balancing, authentication services, communication services or graphical user interface (GUI) components. However, as long as there is no common standard for cloud application development, lock-in effects arise, making the developer dependent on the proprietary software environment of the cloud platform provider. This service, provided in the software environment layer is also referred to as *Platform-as-a-Service (PaaS)*. A known example of a cloud software platform is Google’s App Engine4, which provides developers a Phyton runtime environment and specified APIs to develop applications for Google’s cloud environment. Another example is Salesforce’s App Exchange platform5 that allows developers to extend the Salesforce CRM solution or even develop entire new applications that runs on their cloud environment. . In that sense, the cloud platform can act as a market place for applications.
* *Cloud Software Infrastructure Layer:* The cloud software infrastructure layer provides resources to other higher-level layers, which are utilized by cloud applications and cloud software platforms. The services offered in this layer are commonly differentiated into computational resources, data storage, and communication. Computational resources in this context are usually referred to as *Infrastructure-as-a-Service (IaaS)*. Virtual Machines are the common form of providing computational resources to users, which they can fully administrate and configure to fit their specific needs. Virtualization technologies can be seen as the enabling technology for IaaS, allowing data center providers to adjust resources on demand, thus utilizing their hardware more efficiently. The downside of the medal is the lack of a strict performance allocation on shared hardware resources. Due to this, infrastructure providers can’t give strong performance guarantees which results in unsatisfactory service level agreements (SLA). These weak SLAs propagate upwards in the cloud stack, leading to possible availability problems of cloud applications. The most prominent examples of IaaS are Amazon’s Elastic Compute Cloud6 and Enomalism’s Elastic Computing Infrastructure7. There are also some academic open source projects like Eucalyptus8 and Nimbus9. In analogy to computational resources data storage within the cloud computing model is offered as *Data- Storage-as-a-Service (DaaS)*. DaaS allows users to obtain demand-flexible storage on remote disks which they can access from everywhere. Like for other storage systems, trade-offs must be made between the partly conflicting requirements: high availability, reliability, performance, replication and data consistency, which in turn are manifested in the service providers SLAs. Examples of DaaS are Amazon’s Elastic Block Storage (EBS) 10 or its Simple Storage Service (S3)11 and Rackspace’s Cloud Files12. In addition, to simple storage space, data can be offered as service as well. Amazon for example offers the human genome or the US census as public data sets to use for other services or analytics13. A fairly new idea is *Communication-as-a-Service (CaaS)*, which shall provide quality of service ensured communication capabilities such as network security, dedicated bandwidth or network monitoring. Audio and video conferencing is just one example of cloud applications that would benefit from CaaS. So far this service is more of a research interest than in commercial use. However, Microsoft’s Connected Service Framework (CSF)14 can be counted into this class of services. As Figure 2 shows, cloud applications must not necessarily be developed upon a cloud software platform, but can also run directly on the cloud software infrastructure layer or even the software kernel, thus bypassing the aforementioned layers. Although this approach might offer some performance advantages, it is directly dependent on lower level components and does not make use of development aids such as the automatic scaling provided by the cloud software platform.

* *Software Kernel Layer:* The software kernel layer represents the software management environment for the physical servers in the datacenters. These software kernels are usually implemented as OS kernel, hypervisor, virtual machine monitor or clustering middleware. Typically this layer is also the level where grid computing applications are deployed. Globus15 is an example of a successful grid middleware. At this layer, cloud computing can benefit from the research already undertaken in the grid computing research community.
* *Hardware / Firmware Layer:* At the bottom end of the layered model of cloud computing is the actual physical hardware, which forms the backbone of any cloud computing service offering. Hardware can also be subleased from datacenter providers to, normally, large enterprises. This is typically offered in traditional outsourcing plans, but in an as-a-service context also referred to as *Hardware-as-a-Service (HaaS)*. One example of this is IBM’s Managed Hosting service16. With regard to the layered model of Youseff, Butrico and Da Silva (2008), described above, cloud computing can be perceived as a collection of pre-existing technologies and components. Therefore we see cloud computing as an evolutionary development and re-conceptualization, rather than a disruptive technological innovation. In our opinion cloud computing is rather an innovation in the delivery model of IT services. Therefore we are showing the evolution of cloud computing in the context of IT provisioning in the following chapter.

**1.4 Pros and cons of cloud computing**

Many of the benefits to be had when using Cloud Computing are the lower costs associated. At the infrastructure level, virtual images can be scaled and contracted with complete disregard for any associated hardware costs such as equipment procurement, storage, maintenance and use. This is all taken care of by the service provider and will be factored into the payment for the service: capital expenditure has been converted into operational expenditure. Resources within the cloud can be treated as a commodity, an `unlimited' medium. At both the platform and software level similar benefits are seen. Aspects such as software installation, deployment and maintenance is virtually non-existent. This is taken care of by the provider within their own infrastructure. The service user only pays technical support.

Service providers at the SaaS level, often tout features that allow users to collaborate and interact with each other, in real-time, within the scope of the service being offered. For example, Google Docs allows users to edit documents simultaneously and for users to see each others edits in real time. Moreover, the provision of platform and software `as a service' allows cloud service users the ability to aggregate services together either for their own use or to promote as another service i.e. Mashups. The aggregation could imply the combination of functionality from several services, or the change/combination of output from the services involved.Here are some of the benefits and challenges faced today in the shift from centralized, to distributed, and back to centralized compute (or at least a hybrid).

**Benefits**

* Cloud services are available at a very less cost and without any risk to organization and services are provided by cloud providers.
* In the cloud services the organization has to pay only as peruse while the payment amount in the IT services was not certain and clear.
* The computer resources being used are being shared among so many consumers so no individual have to bear the cost of the set up, as the cost of the service the organization have to pay some amount like rent which is very less. So the costing to the individual organization reduced in an exponential nature along with the threat of out dating of resources in not now a risk to organization.
* Costumer organizations is not required an professional IT staff expanding a large amount, because it is very hard to search the talented technical person a lot to invest on their training a separate IT unit management is a far more cumbersome task so the easier solution is to concentrate on its core business and leave the work for cloud service provider. This enhances superiority, ability to maintain the work, safety, and reduces cost to the customers of cloud.

**Challenges**

* In the present market the number of cloud providers is very less so the costumer is not having a option to choose among the several.  Several bequest IT integrators claim to supply cloud remedies but are fundamentally unmoving bequest managed service providers.
* Organizations have important inherited computing equipments (servers, data centers, and IT personnel) that will need to be shifted or removed in order to attain factual price savings and suppleness provided by cloud lenders.
* Very important IT services are needed to shift over the cloud which are core and very important to the business.  The task is very typical and very time consuming also.  Businesses require to assess that the new type of service application is actually required and value the re-investment, or if an option previously cloud-enabled service is a better fit in the long-term.
* Initial funding and budgeting for cloud services very high even to some money-making and government organizations.
* The whole environment is to be replaced which needs a typical mindset of staff as well as the managing authorities of the organization to make the transition smooth.

**1.5 Cloud Security Issues**

The greater than before use of cloud environment such as Gmail and Google Docs has pushed the subject of solitude concern of such type of service to the greatest significance. The cloud service providers are in a condition that with an enhanced use of such services of cloud computing services has specified right to use to a excess of data. This right to use has the huge danger of data being exposed to any other client either by chance or on purpose. The persons concerned to privacy have opposed the cloud model for giving hosting companies' greater ease to control and thus, to monitor at will communication between host company and end client, and access client data (with or without permission). If some research program is going in the organization then all the data is available on the cloud similarly all the costumer is a big threat of security like if the data is disclosed data and information is also available on the cloud so there is a big privacy concern which arises here . A cloud computing company can cause difficulties in interpretation of data privacy due to the degree of things which are available on the machines which does not exist physically because a virtual space is used for the data storage in the cloud computing services. CSP operations, customer or tenant data may not remain on the same system, or in the same data center or even within the same provider's cloud; this can lead to legal concerns over jurisdiction. While there have been efforts (such as [US-EU Safe Harbor](http://en.wikipedia.org/wiki/US-EU_Safe_Harbor)) to "resonate" the legal atmosphere, providers such as Amazon motionless provide to main markets by using local transportation and permiting clients to choose "areas and ease of use zones". Cloud computing contains privacy issues due to cloud computing company has right to use the data that is on the cloud at any time. The provider may knowingly or unknowingly access the data and even delete or modify it any time. This becomes a major concern as these service providers, who employ administrators which can leave room for potential unwanted disclosure of information on the cloud.

As cloud computing is being popular day by day, the question marks are being intensify for the security concerns and other problems still available in this computing. The usefulness and good organization of conventional security mechanisms are taken back into consideration as the distinctiveness of this inventive consumption representation can be different extensively from those of conventional infrastructure. An substitute viewpoint on the subject of cloud protection is that this is but another, even if fairly large, case of "practical defense" and that alike protection philosophy that be relevant in communal multi-client traditional computing protection models be relevant with cloud security.

The relative protection of cloud computing services is a controversial subject that may be time taking in its acceptance. Substantial power of the confidential Cloud apparatus is more safe and sound than having the apparatus off site and beneath somebody another's power. Substantial power and the capability to visually scrutinize data relations and contact port is mandatory in order to make sure data relations are not compromised. Concerns with the exception of the acceptance of cloud computing are outstanding in great fraction to the classified and public sectors' discomfort neighboring the peripheral organization of security-based services. It is the extremely scenery of such type of services, private or public that encourage outer administration of landed facilities. This provides immense motivation to cloud computing facilities Lander to prioritize construction and maintaining strong supervision of protected services. Privacy concerns are classified into sensitive data access, data separation, privacy, bug exploitation, recovery, accountability, malicious insiders, management console security, account control, and multi-tenancy issues. Remedies to various cloud security concerns differ, from cryptography, mainly public key architecture (PKA), to use of numerous cloud computing firms, standardization of APIs, and improving virtual machine support and legal support.

Cloud computing offers a lot of payback, at the same time it is also very susceptible to threats. In the growing scenario of the cloud computing, it is the hackers or he professional hackers try to find the ways to take advantage of system’s pits and falls. There are so many small things and problems which are to be overcome by the system to reduce the security issues and risks in cloud computing so as to decrease the danger of data negotiation. To moderate the problem, cloud service providers should spend a great deal in danger evaluation to make sure that the arrangement encrypts to defend data, forms a reliable establishment to safe and sound the display place and architecture, and creates advanced guarantee into auditing to make stronger fulfillment. Protection concerns required to be introduced to preserve belief in cloud computing technology.

Data contravene is a big distress in such type of computing. A shared architecture of server could considerably damage the clients as well as service provider firms. A range of data may be theft or misused. These take account of credit card and social security information, addresses, and individual communication. The countries now necessitate cloud providers to give notice clients of breaches. Once notified, clients at the present have to be concerned about recognize robbery and racket. While the cloud computing firms, have to contract with centralized investigation, lawsuits, and dreadful standing.

Protection issues are categorized in both technological and socio-technical in starting point. To envelop all the sanctuary issues probable inside the cloud, and thoroughly would be phenomenal a mission not well-matched still for Heracles himself. Obtainable labors give the impression of being to make available taxonomy above the issues visualized. The Cloud Security Alliance1 is a non-profit association that intends to endorse the most excellent practices for assuring safety measures inside the cloud computing countryside. In the literature of cloud computing the Cloud Security Alliance recognize seven intimidations to cloud computing that can be interpreted as a categorization of protection concerns establish contained by the cloud. They are:

1. Abuse and Nefarious Use of Cloud Computing
2. Insecure Application Programming Interfaces
3. Malicious Insiders
4. Shared Technology Vulnerabilities
5. Data Loss/Leakage
6. Account, Service and Trafic Hijacking
7. Unknown Risk Profile

For an alternate categorization of intimidation to Cloud Computing, one can seek advice from. This section presents a universal outlook of the safety measurement issues contained by the compass of the intimidation obtainable by the Cloud Security Alliance.

* *Abuse and Nefarious Use of Cloud Computing***:** Justifiable CSPs can be badly treated for despicable purpose, sustaining against the law or other troublesome behavior towards clients. For case in point, services can be used to congregation malevolent regulations or used to smooth the progress of communication between inaccessible entities i.e. bonnets. The prominence is that justifiable services are used with malevolent purpose in intelligence. Other concern seen take account of the stipulation of with determination self-doubting services used for data capture. The cloud computing firms may attract prospective clients with offers as well superior to be accurate. For illustration the guarantee of unconstrained possessions or a `30-day Free Trial'. For the duration of the registration procedure the purchaser will be asked to be available with more information than what would usually be obligatory beneath the pretence of given that service personalization e.g. position or age based announcement. Frequently asked data include the customers name, email/postal address, D.O.B or even credit card particulars. Clients are fundamentally being motivated to fraction with additional information than necessary as a qualification for service use. Hateful individuals can then use this data for reprehensible purposes, most extraordinarily identity stealing. Even if the entities are not malevolent the revelation of information to the CSP can in addition be said to be an ill-treatment of service by the CSP themselves. CSPs may accumulate this information, or several other information given at afterward stage, and advertise this data to third parties for data withdrawal purposes. This is of the same kind to accumulate cards such as credit or debit cards in which the apparent reimbursement i.e. financial investments, get nearer at the price of handing over individual information such as shopping behavior. Another privacy concern found is the stipulation of lawful maliciously-oriented services. This service stipulation in addition is recognized as (In) Security-as-a-Service, provides clients the similar safety measures guarantee as offered services yet their use is malevolent. For example <http://www.wpacracker.com/> is a cloud-based furious service that can be used to `ensure the sanctuary of WPA-PSK confined wireless network. By using a cloud computing application it is claimed that a procedure that would take approximately five plus days now takes on commonly twenty minutes.
* *Insecure Interfaces and Application Programming Interfaces:*Data positioned in the Cloud will be used through Application Programming Interfaces (APIs) and other interfaces. Misbehavior and problems in the boundary software, and also the software used to run the Cloud, can guide to the superfluous introduction of client data and hold responsible in the lead the data's truthfulness. For example a (permanent) law in Apache, a accepted HTTP server, permissible an mugger to increase total control over the network server. Data introduction can also come about when a software break down affects the right of entry policies overriding clients data. This has been observed in more than a few Cloud based services in which a software break down resulted in which a clients privacy settings were overwritten and the client data uncovered to non-authorized entities. Threats can also continue living as a consequence of inadequately designed or implemented security actions. If the commeasures can be bypassed, or are absent, the software can be without difficulty ill-treated by malevolent entities. Not considering of the hazard starting point, APIs and other interfaces necessitate to be completed safe and sound against unintentional and malevolent attempts to get around the APIs and their security measures.

**1.6 Cloud Threat Models:** The source of intimidation towards data surrounded by the cloud is described collectively with two danger models based upon said lifecycle. The first model represents a client- centric view, the other a Cloud Service Provider point of view.

By bearing in mind the Cloud as a inaccessible storeroom system i.e. NAS, one can take obtainable risk models that give the impression of being towards the region of remote storage and become accustomed them, as essential, for the Cloud. The literature provides a discussion of two threat models for inaccessible storeroom systems. The first, categorize intimidation based in the lead their consequence upon the four `conventional security necessities of privacy, truthfulness, approval and accessibility. The second classifies pressure according to how the intimidation affects data for the duration of its lifecycle. The CIAA Threat representation is based upon the four traditional security necessities of: privacy, truthfulness, accessibility and substantiation. Planned threats are off the record according to their relative to each prerequisite. This model is to a certain extent coarse-grained with high opinion to the categorization of intimidation. Intimidation is categorized upon the consequence they have on each constraint and not towards the perspective of their accomplishment. A more grainy model is compulsory that considers where the intimidation start off and when the threat is likely to occur for the duration of service procedure. A threat model based upon the data lifecycle permits for such a model to be formed.

Remote storage systems can be viewed as a single system by consumers. The arrangement of a incorporated boundary from side to side which the customers interrelate can also be used to hide the difficulty of the fundamental infrastructure. This is a significant perception and is also collective by Cloud Computing. Cloud Clients can also analysis the Cloud as a `single' system, though, this view point prohibits for a comprehensive threat model to be recognized for data in the Cloud. It only provides the clients with a view of the flow of their data: into the cloud and back from the cloud. Using these concepts two dissimilar yet interrelated danger models for the cloud can be constructed. The purpose of a threat model is to organize the diverse threats and vulnerabilities into groups so that they can be determined consequently.

Threat Overview: Before the data life cycle threat models are introduced, some time will be exhausted detailing the source and personality of the intimidation.

**1.7 Steganography as a solution of the threats**

Steganography is derived from the Greek for enclosed inscription and fundamentally means “to hide in plain sight”. As defined by Cachin steganography is the art and science of communicating in such a way that the presence of a message cannot be detected. Simple steganographic techniques have is being used since hundreds of years, but with the growing use of documentation in an electronic arrangement new techniques for information thrashing have turn out to be achievable.

This manuscript will inspect some near the beginning examples of steganography and the all-purpose main beliefs following its usage. We will then give the impression of being at why it has turn out to be such an significant issue in current years. There will then be a conversation of some detailed techniques for thrashing information in a assortment of documentation and the attacks that may be used to find a way around steganography. Figure below shows how information thrashing can be broken down into diverse areas. Steganography can be used to hide a communication anticipated for afterward reclamation by a unambiguous personality or assemblage. The aim of the technique is to conceal the data in an unnoticeable way. The other most significant region of steganography is exclusive rights marking, where the communication to be inserted is used to emphasize copyright over a text. This can be additional alienated into watermarking and fingerprinting which will be discussed later.

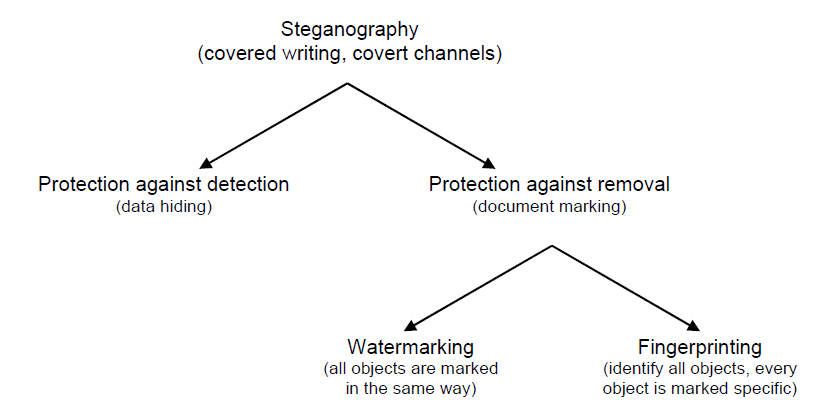


Fig.1.5Types of steganography

Steganography and encryption are both used to make sure data prudence. However the main discrimination among them is that with encryption any person can see that both parties are communicating in secret. Steganography hides the continuation of a clandestine communication and in the best case nobody can see that both parties are communicating in secret. This makes steganography appropriate for a number of responsibilities for which encryption isn’t, such as copyright marking. Adding together encrypted copyright information to a file could be easy to eliminate but embedding it inside the stuffing of the file itself can avoid it being simply recognized and disinterested.

**1.7.1 Embedding and detecting a mark**

The figure presented beneath a straightforward illustration of the common embedding and decoding procedure in steganography. In this illustration, a secret image is being embedded inside a cover image to produce the stego image. The first step in embedding and hiding information is to pass both the secret message and the cover message into the encoder. Inside the encoder, one or several protocols will be implemented to embed the secret information into the cover message. The type of protocol will depend on what information you are trying to embed and what you are embedding it in. For example, you will use an image protocol to embed information inside images. A key is often needed in the embedding process. This can be in the form of a public or private key so you can encode the secret message with your private key and the recipient can decode it using your public key.

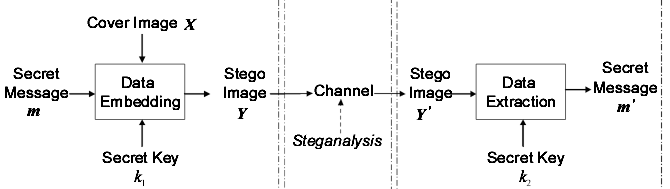


Fig.1.6 image steganography process

In general the embedding process combines a spot, M, in an entity, I. A key, K, usually produced by a random number producer is used in the embedding procedure and the resultant patent article, Ĩ, is generated by the mapping: I x K x M → Ĩ.

Having passed from beginning to end from encoder, a stegano article will be shaped. A stego thing is the innovative wrap object with the underground information embedded inside. This object ought to seem approximately indistinguishable to the cover object as or else a third party mugger can be able to see entrenched information. After producing the stegano image, it will be sent off by means of some communications channel, such as electronic mail, to the proposed beneficiary for decoding. The beneficiary has to make sense of the stego object in order to visualize the secret information. The decoding procedure is basically the reverse of the encoding procedure. It is the taking out of secret information from a stego object. In the decoding procedure, the stego object is feed in to the arrangement. The public or private key which can decode the actual key which is used within the encoding procedure is in addition required so that the secret data can be decoded. Depending on the encoding method, sometimes the actual cover object is as well needed in the decoding procedure. Otherwise, there may be no way of extracting the secret information from the stego object.

After the decoding procedure is finished, the secret data embedded in the stego object can then be recovered. The general decoding procedure again require a key, K, at the instance along with a sophisticatedly marked object, Ĩ’. Also required is either the mark, M, which is being checked for or the original object, I, and the result will be either the retrieved mark from the object or indication of the likelihood of M being present in Ĩ’.

**1.7.2 Criteria for Steganography**

Three common requirements, security, capacity, and imperceptibility, may be used to rate the performance of steganographic techniques.

* ***Security***: Steganography may suffer from many active or passive attacks, correspondingly in the prisoner's problem when Wendy acts as an active or passive warden. If the existence of the secret message can only be estimated with a probability not higher than random guessing in the presence of some steganalytic systems, steganography may be considered secure under such steganalytic systems. Otherwise we may claim it to be insecure.
* ***Capacity:*** To be useful in conveying secret message, the hiding capacity provided by steganography should be as high as possible, which may be given in absolute measurement (such as the size of secret message), or in relative value (called data embedding rate, such as bits per pixel, bits per non-zero discrete cosine transform coefficient, or the ratio of the secret message to the cover medium, etc.).
* ***Imperceptibility***: Stego images should not have severe visual artifacts. Under the same level of security and capacity, the higher the fidelity of the stego image, the better. If the resultant stego image appears innocuous enough, one can believe this requirement to be satisfied well for the warden not having the original cover image to compare.

**1.7.3 Criteria for Steganalysis:** The main goal of steganalysis is to identify whether or not a suspected medium is embedded with secret data, in other words, to determine the testing medium belong to the cover class or the stego class. If a certain steganalytic method is used to steganalyze a suspicious medium, there are four possible resultant situations.

* 1. **Scope and Objective**

***The dissertation present, a system for securing the data on the clouds by the means of image steganographic process. In this process the data in 4-LSBs bits of the RGB components of every pixel. Thus every pixel can accommodate 12 bits of data; we ready two consecutive pixels and embed 24 bits (3 bytes) into RGB components of these pixels. Thus the data carrying capacity of the system increases tremendously as compared to previously used algorithms where 1 pixel of the cover may only carry 3 bits of information means 3 bits were required to embed the data.***

**1.8.1 *Study of cloud computing***

**Cloud computing is an rising technology standard that moves the current technical and computing fundamentals into utility-like solutions similar to electricity and water systems. Cloud computing is a well accepted but still evolving model. It allows customers of cloud providers to avoid start-up costs and/or reduce operating cost.**

**1.8.2 *Study of threats and security issues in cloud computing***

**Security issues come under many guises both technical and socio-technical in origin. To cover all the security issues possible within the cloud, and in-depth, would be herculean a task not suited even for Heracles himself. Existing efforts look to provide a taxonomy over the issues seen. The Cloud Security Alliance1 is a non-profit organisation that seeks to promote the best practises for providing security assurance within the cloud computing landscape. In Hubbard, Sutton et al. [HS+10] the Cloud Security Alliance identify seven threats to cloud computing that can be interpreted as a classification of security issues found within the cloud. They are Abuse and Nefarious Use of Cloud Computing, Insecure Application Programming Interfaces, Malicious Insiders, Shared Technology Vulnerabilities, Data Loss/Leakage, Account, Service and Trafic Hijacking, Unknown Risk Profile For an alternate classification of threats to Cloud Computing, one can consult. This Chapter provides a general view of the security issues within the scope of the threats presented by the Cloud Security Alliance.**

**1.8.3 *Study of image steganography for security in clouds***

**Steganography and encryption are both used to ensure data confidentiality. However the main difference between them is that with encryption anybody can see that both parties are communicating in secret. Steganography hides the existence of a secret message and in the best case nobody can see that both parties are communicating in secret. This makes steganography suitable for some tasks for which encryption aren’t, such as copyright marking. Adding encrypted copyright information to a file could be easy to remove but embedding it within the contents of the file itself can prevent it being easily identified and removed.**

**1.8.4 Encryption of data on cloud using image steganography with LSB technique**

**The four least significant bits of some or all of the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bytes of information in two pixels, four in LSB positions of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. In 8 bit images, one bit of information can be hidden.**

**1.8.5 Decryption of data on cloud using image steganography with LSB technique**

**The cover image contains a hidden message over the cloud. A stego-image is obtained by applying LSB algorithm on both the cover and hidden images. The hidden image is extracted from the stego-image by applying the reverse process.**

* 1. **Organization of dissertation**

***This dissertation covers the design of an automated system for a secure data transmission on cloud. The work focuses on characterization of information hiding possibilities in Cloud Computing. After general introduction to cloud computing and its security we move to brief description of steganography. In particular we introduce classification of steganographic communication scenarios in cloud computing which is based on location of the steganograms receiver. These scenarios as well as the threats that steganographic methods can cause must be taken into account when designing secure cloud computing services.***

**Chapter 1** serves as an introduction of cloud computing as well as image st enographic process. It first introduces the reason, why the clouding is being most popular for data transmission and high speed communication of data. This Chapter demonstrates the fundamental theory of cloud computing, its merits, demerits and its application. The originality of this Chapter is that it proposes the cloud computing secured with image steganography is the important milestone in a microwave communication because of its superior performance with their application.

**Chapter 2** expresses the detail literature survey of cloud computing, cloud security threats and models, steganographic technique for data security and other techniques to achieve a secure data communication. The papers and books which were used as reference for this project work are mentioned with their suggested structure of cloud computing model and steganographic technique its merits and demerits.

**Chapter 3** describes the design and implementation of a secure cloud transmission model using image steganographic process. The data is hidden in a cover image in such a way that o one else than the destination client no one else can detect the data so insuring the security of data. The system design and working is explained in detail.

**Chapter 4** describes the design and implementation of a data recovery system for the data hidden by the image steganographic process. The data is hidden in a cover image is recovered from the image and reformed again. The system design and working is explained in detail.

In **Chapter 5** an example is taken in which a text data file is hidden in an image. The text file is having a larger size then the capacity of the cover image so multiple copies of the cover image are created to hide the complete data. The difference images and stegano images at each and every step are analyzed. The data matrices of each image are produced in order to justify the process results and the graphs are drawn for each image RGB components. The correlation between each stegano image is calculated along with the entropy of each image as well as difference of all the entropies of all the stegano images with respect to cover image.

**Chapter 6** summarizes this dissertation with conclusion and future scope. The techniques and results at different stages are discussed and analyzed.

**Chapter - 2**

**LITERATURE REVIEW**

**2.1 Emergence and growth of cloud computing**

"Cloud computing" fundamentals is sustained to the 1950s when important traditional computers were completely existing to organizations and workplaces. The traditional computer's massive hardware architecture was installed in what could actually be called a "server room", and manifold clients were capable to admittance the traditional computer via "dumb terminals" – stations whose only purpose was to make easy right to use to the traditional computers. Due to the price of import and sustaining traditional computers, an association would not be capable to have enough money for a traditional computer for every client, so it became follow to permit many clients to divide up entrée to the same data storeroom layer and CPU control from any position. By enabling joint traditional computer admission, an group would get a enhanced return on its investments in this complicated portion of machinery.

A couple decades later in the 1970s, IBM released an operating system called VM that permited administration on their System/370 traditional computer systems to have manifold virtual systems, or "Virtual Machines" (VMs) on a particular physical knot. The VM operating system took the 1950s function of shared access of a traditional computer to the next height by permiting numerous different compute environments to exist in the identical physical environment. Most of the essential functions of any virtualization software that you see nowadays can be traced back to this premature VM OS: each VM might run traditional operating systems or visiting operating systems that had their "individual" memory, processors and other peripherals, that even with the truth that all of those resources would be common. "Virtualization" became a equipment driver, and it became a massive medium for several of the prime evolutions in infrastructure and computing.

The term "moving to cloud" also refers to an organization moving away from a traditional [CAPEX](http://en.wikipedia.org/wiki/Capital_expenditure) model (buy the dedicated hardware and depreciate it over a period of time) to the [OPEX](http://en.wikipedia.org/wiki/Operating_expense) model (use a shared cloud infrastructure and pay as one uses it).

Proponents claim that cloud computing permits companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of infrastructure.[[6]](http://en.wikipedia.org/wiki/Cloud_computing#cite_note-aws.amazon-6) Proponents also claim that cloud computing permits enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand[2-4]. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

The term "cloud computing" is mostly used to sell hosted services in the sense of [application service provisioning](http://en.wikipedia.org/wiki/Application_service_provider) that run [client server](http://en.wikipedia.org/wiki/Client_server) software at a remote location. Such services are given popular acronyms like 'SaaS' (software as a service), 'PaaS' (platform as a service), 'IaaS' (infrastructure as a service), 'HaaS' (hardware as a service) and finally 'EaaS' (everything as a service). End clients access cloud-based[applications](http://en.wikipedia.org/wiki/Application_software) through a [web browser](http://en.wikipedia.org/wiki/Web_browser), [thin client](http://en.wikipedia.org/wiki/Thin_client) or [mobile app](http://en.wikipedia.org/wiki/Mobile_app) while the [business software](http://en.wikipedia.org/wiki/Business_software) and client's data are stored on servers at a remote location. Examples include Amazon Web Services and Google App engine, which allocate space for a client to deploy and manage software "in the cloud".

In spite of the rapid expansion of Infrastructure-as-a-Service (IaaS) technologies such as Amazon EC2 1, Microsoft Azure 2, services provided by RackSpace 3 and others, IaaS services continue to be plagued by vulnerabilities at several levels of the software stack, from the web based cloud management console [5] to VM side-channel attacks, to information leakage, to collocated malicious virtual machine instances [6]. The need for secure cloud storage and cloud computing environments has been reiterated on numerous occasions. For example, Molnar et al [7] cite industry decision makers to emphasize the fact that security concerns are among the major factors that prevent businesses from deploying their data and computations into the cloud. Common reasons are unawareness of the state of the data and algorithms once it is in the cloud environment, as well as concerns regarding cloud provider bankruptcy and subsequent lack of clarity and established procedures of data protection and retrieval, along with many other examples. Similarly, Chen et al [8] cite opinions originating from academia, government and industry that point to security concerns as a barrier preventing a quicker adoption of cloud computing. The reasons are both technical, such as the fear of data loss, data breach and data tampering as well as organizational, such as reputation fatesharing. Similar views are reported by other researchers within cloud computing security ([9,10]). The economic benefits of using cloud storage and cloud computing are appealing enough to promote adoption of these technologies, hence their use is likely to increase over time [8]. In this situation, there is a risk that the economic benefits obtained today through the rapid adoption of cloud technologies will in some cases be compensated or even overcompensated by losses resulting from unexpected lack of availability as well as theft and corruption of data. The continuous flow of vulnerabilities discovered in the software stack underlying IaaS platforms has prompted the move towards implementing trust anchors into hardware. Although this move has the potential to greatly reduce the risks posed by software vulnerabilities, it does not guarantee a secure platform out of the box. Rather, the results depend on the correct usage of the trusted hardware. The Trusted Computing initiative and adoption of trusted platform modules (TPM) has been steadily gaining momentum since it's inception [11]. Participation of hardware manufacturing industry leaders in the Trusted Computing Group 4 is likely to accelerate the adoption of this technology across hardware architectures and platforms. Following its initial predominance and narrow focus on laptop computers, trusted computing is making its way into new devices. For example, the use of trusted computing on mobile platforms is already the focus of several recent research projects [12,13] with more to come as increased functionality and ever more information stored on mobile devices become more attractive targets for malware. Another important application domain of trusted computing is its use in virtualized systems and cloud computing [14]. Trustworthy integrity verification of the software components used within the cloud computing infrastructure, as well as information protection using trusted computing techniques can address some of the security concerns related to off-premises computing. While it does not actually offer absolute guarantees, trusted computing raises the complexity bar for attackers by placing the root of trust at the hardware level 5. With a correct implementation, an attacker would need physical access to the hardware in order to subvert the TPM [15]. However, as the technology is still new and in active development, the best practices for the use of TPM are yet to be identified. This is especially relevant for virtualized environments and trusted cloud computing, where the functionality of a single TPM chip needs to be shared between several virtual machines. Solutions like virtualization of TPMs [16] create new possibilities for implementation of secure launch and secure migration of VMs [17,18]. In the same time new attack techniques demonstrate that software implementation of TPM increases the trusted computing base (TCB) and introduces new vulnerabilities [19-20]. This implies that new solutions for secure VM launch and migration need to be found based on the existing components of the TPM and with minimal changes to the TCB.

**2.2 Making the clouds secure: issues and remedies**

Cloud computing provides a centralized pool of configurable computing resources and omputing Out sourcing mechanisms that enable different computing services to different people in a way similar to utility-based systems such as electricity, water, and sewage. In electricity, for example, people started to connect with central grids, supported by power utilities rather than relying on their own electricity production capabilities. This migration is beneficial in reducing the cost and time of production and in providing better performance and reliability [21]. Similarly, clouds provide their customers with high performance and more reliable computing services such as e-mail, instant messaging, and web services at a lower cost. Cloud computing does not have a common accepted definition yet [22]. The National Institute of Standards and Technology (NIST) [23] defined five essential characteristics of cloud computing, namely: on-demand self-service, broad network access, resource pooling, rapid elasticity or expansion, and measured service. Also, cloud computing is described as a dynamic and often easily extended platform to provide transparent virtualized resources to clients through the Internet [24]. Cloud computing architecture consists of three layers: (i) Software as a service (SaaS); (ii) Platform as a service (PaaS) and (iii) Infrastructure as a service (IaaS) [25]. The clouds are also viewed as five component architectures that comprise clients, applications, platforms, infrastructure and servers [26]. The current clouds are deployed in one of four deployment models: (a) public clouds in which the physical infrastructure is owned and managed by the service provider; (b) community clouds in which the physical infrastructure is owned and managed by a consortium of organizations; (c) private clouds in which the infrastructure is owned and managed by a specific organization and (d) hybrid clouds which include combinations of the previous three models [27]. Cloud deployment models together with their internal infrastructure (IaaS, PaaS and SaaS). Cloud deployment models have similar internal infrastructure, but vary in their policies and client-access levels. Clouds bring out tremendous benefits for both individuals and enterprises. Clouds support economic savings, outsourcing mechanisms, resource sharing, any-where any-time accessibility, on-demand scalability, and service flexibility. Clouds minimize the need for client involvement by masking technical details such as software upgrades, licenses, and maintenance from its customers. Clouds could also offer better security advantages over individual server deployments. Since a cloud aggregates resources, cloud providers charter expert security personnel while typical companies could be limited with a network administrator who might not be well versed in cyber security issues. Similarly, clouds are more resilient to Distributed Denial of Service (DDoS) attacks due to the availability of resources and the elasticity of the architecture. The clouds support mobile computations where Virtual Machines (VMs) migrate from one physical machine to another. In addition to alleviating dedicated DDoS attacks, mobile computations help to avoid settings in which a single administrator has exclusive control over the computation.

The new concepts introduced by the clouds, such as computation outsourcing, resource sharing, and external data warehousing, increase the security and privacy concerns and create new security challenges. Moreover, the large scale of the clouds, the proliferation of mobile access devices (e.g., smartphones and tablets), and the direct access to cloud infrastructure amplify cloud vulnerabilities and threats. As clouds become more and more popular, security concerns grow bigger and bigger as they become more attractive attack targets due to the concentration of digital assets. Many researchers and practitioners work on identifying cloud threats, vulnerabilities, attacks, and other security and privacy issues, in addition to providing countermeasures in the form of frameworks, strategies, recommendations, and service oriented architectures (e.g., [22,28–33]). Additionally, efforts in other domains such as ad hoc networks have been tuned to address the emerging security problems in the clouds (e.g., [34-38]). Many researchers (e.g., [1, 19–25]) have addressed single attributes of cloud computing security such as data integrity, authentication vulnerabilities, auditing, *etc*. Others (e.g., [23, 42–47]) provide surveys that cover specific areas of cloud security concerns and proposed solutions. In [23, 47], the authors briefly and broadly discuss cloud security issues involving data, applications and virtualization. The authors in [42] discuss similar cloud security issues but with deeper investigations. In [43,45], the authors present surveys on cloud security requirements such as confidentiality, integrity, transparency, availability, accountability, and assurance. In [44], the authors present a survey on the different security issues of the service delivery models of the clouds. In [46], the authors discuss the security challenges specific to the public clouds. In [28], Hashizume *et al*. classify the security issues in the cloud based on the SPI (SaaS, PaaS, IaaS) cloud infrastructure and services model. The authors provide deeper classification of the fourth category (C4) in our classification model (Section 2). Additionally, the authors explain fundamental security concepts including vulnerabilities, threats, and attacks and provide mapping among these concepts. Our work provides a higher classification level and assumes prior knowledge of the fundamental security concepts. In [49], Zissis *et al*. evaluate cloud security requirements. They propose a Trusted Third Party solution that calls upon cryptography to ensure the authentication, integrity and confidentiality of data and communications. In [50], Whaiduzzaman *et al*. present key management and broad aspects of privacy and security issues in the domain of vehicular cloud computing.

**2.3 Steganography for secure data transmission on clouds**

The term steganography came into use in 1500s after the appearance of Trithemius book on the subject Steganographia. The word Steganography technically means covered or hidden writing. Its ancient origins can be traced back to 440 BC. Although the term steganography was only coined at the end of the 15th century, the use of steganography dates back several millennia. In ancient times, messages were hidden on the back of wax writing tables, written on the stomachs of rabbits, or tattooed on the scalp of slaves. Invisible ink has been in use for centuries for fun by children and students and for serious undercover work by spies and terrorists. The majority of todays steganographic systems uses multimedia objects like image, audio, video etc as cover media because people often transmit digital pictures over email and other Internet communication. Modern steganography uses the opportunity of hiding information into digital multimedia files and also at the network packet level. Hiding information into a medium requires following elements

1. The cover medium(C) that will hold the secrat message.

2. The secret message (M), may be plain text, digital image file or any type of data.

3. The stegonographic techniques

4. A stego-key (K) may be used to hide and unhide the message.

In modern approach, depending on the cover medium, steganography can be divided into five types:

1. Text Steganography

2. Image Steganography

3. Audio Steganography

4. Video Steganography

5. Protocol Steganography

* Text steganography Hiding information in text file is the most common method of steganography. The method was to hide a secret message into a text message. After coming of Internet and different type of digital file formats it has decreased in importance. Text stenography using digital files is not used very often because the text files have a very small amount of excess data.
* Image steganography Images are used as the popular cover medium for steganography. A message is embedded in a digital image using an embedding algorithm, using the secret key. The resulting stego-image is send to the receiver. On the other side, it is processed by the extraction algorithm using the same key. During the transmission of stego- image unauthenticated persons can only notice the transmission of an image but cant see the existence of the hidden message.
* Audio steganography Audio steganography is concerned with embedding information in an innocuous cover speech in a secure and robust manner. Communication and transmission security and robustness are essential for transmitting vital information to intended sources while denying access to unauthorized persons. An audible, sound can be inaudible in the presence of another louder audible sound .This property permits to select the channel in which to hide information. Existing audio steganography software can embed messages in WAV and MP3 sound files. The list of methods that are commonly used for audio steganography are listed and discussed below.
* LSB coding
* Parity coding
* Phase coding
* Spread spectrum
* Echo hiding
* Video steganography Video Steganography is a technique to hide any kind of files in any extension into a carrrying Video file.
* Protocol steganography The term protocol steganography is to embedding information within network protocols such as TCP/IP. We hide information in the header of a TCP/IP packet in some fields that can be either optional or are never used.

Applications of Steganography

1. Secret Communications The use steganography does not advertise secret communication and therefore avoids scrutiny of the sender, message, and recipient. A trade secret, blueprint, or other sensitive information can be transmitted without alerting potential attackers.
2. Feature Tagging Elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map. Copying the stego-image also copies all of the embedded features and only parties who possess the decoding stego-key will be able to extract and view the features.
3. Copyright Protection Copy protection mechanisms that prevent data, usually digital data, from being copied. The insertion and analysis of watermarks to protect copyrighted material is responsible for the recent rise of interest in digital steganography and data embedding.

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography. Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words “*stegos*” meaning “cover” and “*grafia*” meaning “writing” [51] defining it as “covered writing”. In image steganography the information is hidden exclusively in images. The idea and practice of hiding information has a long history. In *Histories* the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave’s scalp. When the slave’s hair grew back the slave was dispatched with the hidden message [52]. In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information [53]. Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret [54]. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated [54]. The strength of steganography can thus be amplified by combining it with cryptography. Two other technologies that are closely related to steganography are watermarking and fingerprinting [55]. These technologies are mainly concerned with the protection of intellectual property, thus the algorithms have different requirements than steganography. These requirements of a good steganographic algorithm will be discussed below. In watermarking all of the instances of an object are “marked” in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection [56]. With fingerprinting on the other hand, different, unique marks are embedded in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties [55]. In watermarking and fingerprinting the fact that information is hidden inside the files may be public knowledge sometimes it may even be visible – while in steganography the imperceptibility of the information is crucial [54]. A successful attack on a steganographic system consists of an adversary observing that there is information hidden inside a file, while a successful attack on a watermarking or fingerprinting system would not be to detect the mark, but to remove it [55]. Research in steganography has mainly been driven by a lack of strength in cryptographic systems. Many governments have created laws to either limit the strength of a cryptographic system or to prohibit it altogether [57], forcing people to study other methods of secure information transfer. Businesses have also started to realize the potential of steganography in communicating trade secrets or new product information. Avoiding communication through well-known channels greatly reduces the risk of information being leaked in transit [57]. Hiding information in a photograph of the company picnic is less suspicious than communicating an encrypted file.

**Chapter - 3**

**ENCRYPTION AND DECRYPTION OF DATA IN IMAGE**

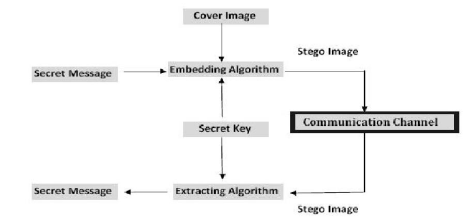
**3.1 Introduction**

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services*.* It brings out a wide range of benefits including configurable computing resources, economic savings, and service flexibility. However, security and privacy concerns are shown to be the primary obstacles to a wide adoption of clouds. The new concepts that clouds introduce, such as multi-tenancy, resource sharing and outsourcing, create new challenges to the security community. Addressing these challenges requires, in addition to the ability to cultivate and tune the security measures developed for traditional computing systems, proposing new security policies, models, and protocols to address the unique cloud security challenges. In this work, we provide a comprehensive study of cloud computing security and privacy concerns. However, cloud computing uniqueness when compared with previous computing approaches launched new security, privacy and trust challenges from which most important are Ensuring that only authorized parties have access to client data even if it is a service provider, Sharing responsibility between cloud provider and its customer for security and privacy and Providing secure multi-tenant environment (secure and efficient partitioning of virtualized and shared infrastructure among different customers). Steganographic techniques can be used to provide a perfect tool for data exfiltration, to enable network attacks or hidden communication among secret parties. The aim of these techniques is to hide secret data (steganograms) in the innocent looking carrier e.g. in normal transmissions of clients. In ideal situation hidden data exchange cannot be detected by third parties. The best carrier for steganograms must possess two features: it should be popular i.e. usage such carrier should not be considered as an anomaly itself and modification of the carrier related to inserting the steganogram should not be “visible” to third party not aware of the steganographic procedure.

In this Chapter first the some common stenographic techniques are described first then the least significant bit technique for encryption of the data is discussed for a 24 bit RGB image pixels and hence a process to embed the secret data file in a cover image is described the data file is broken in parts if it is having a larger size then the capacity of the cover image and each part is send with a new replica of the cover image.

**3.2 Image steganography**

As stated earlier, images are the most popular cover objects used for steganography. In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganographic algorithms exist.

****

**F**ig. 3.1 Generic view of image stegnography

**3.2.1 How the image seems to a computer**

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image [58]. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the Internet consists of a rectangular map of the image’s pixels (represented as bits) where each pixel is located and its colour [59]. These pixels are displayed horizontally row by row. The number of bits in a colour scheme, called the bit depth, refers to the number of bits used for each pixel [16]. The smallest bit depth in current colour schemes is 8, meaning that there are 8 bits used to describe the colour of each pixel [60]. Monochrome and grey scale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour [60]. All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour is represented by 8 bits [58]. Thus in one given pixel, there can be 256 different quantities of red, green and blue, adding up to more than 16-million combinations, resulting in more than 16-million colours [60]. Not surprisingly the larger amount of colours that can be displayed, the larger the file size [59].



Fig. 3.2 Categories of image steganography

**3.2.2 Reducing the size of image**

When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard Internet connection. In order to display an image in a reasonable amount of time, techniques must be incorporated to reduce the image’s file size. These techniques make use of mathematical formulas to analyse and condense image data, resulting in smaller file sizes. This process is called compression [59]. In images there are two types of compression: lossy and lossless [61]. Both methods save storage space, but the procedures that they implement differ. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate [59], resulting in close approximations of the original image, although not an exact duplicate. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group) [58]. Lossless compression, on the other hand, never removes any information from the original image, but instead represents data in mathematical formulas [59]. The original image’s integrity is maintained and the decompressed image output is bit-by-bit identical to the original image input [61]. The most popular image formats that use lossless compression is GIF (Graphical Interchange Format) and 8-bit BMP (a Microsoft Windows bitmap file) [58]. Compression plays a very important role in choosing which steganographic algorithm to use. Lossy compression techniques result in smaller image file sizes, but it increases the possibility that the embedded message may be partly lost due to the fact that excess image data will be removed [62]. Lossless compression though, keeps the original digital image intact without the chance of lost, although is does not compress the image to such a small file size [58]. Different steganographic algorithms have been developed for both of these compression types and will be explained in the following sections.

**3.2.3 Image and Transform Domain**

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain [63]. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image [64]. Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterised as “simple systems” [65]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format [66]. Steganography in the transform domain involves the manipulation of algorithms and image transforms [65]. These methods hide messages in more significant areas of the cover image, making it more robust [67]. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression. In the next sections steganographic algorithms will be explained in categories according to image file formats and the domain in which they are performed.



Fig. 3.3 Data hiding procedure

1. **Image Domain**

**Least Significant Bit:** Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [58]. The four least significant bits (in other words, the 5th to 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, four bits of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 byts in each two pixels. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data [68]. Should an adversary suspect that LSB steganography has been used, he has no way of knowing which pixels to target without the secret key [69].

1. **Transform Domain**

To understand the steganography algorithms that can be used when embedding data in the transform domain, one must first explain the type of file format connected with this domain. The JPEG file format is the most popular image file format on the Internet, because of the small size of the images.

**JPEG compression:** To compress an image into JPEG format, the RGB colour representation is first converted to a YUV representation. In this representation the Y component corresponds to the luminance (or brightness) and the U and V components stand for chrominance (or colour). According to research the human eye is more sensitive to changes in the brightness (luminance) of a pixel than to changes in its colour [71].

1. **Image or Transform domain**

As seen in Figure 2, some steganographic algorithms can either be categorised as being in the image domain or in the transform domain depending on the implementation.

**Patchwork:** Patchwork is a statistical technique that uses redundant pattern encoding to embed a message in an image [59]. The algorithm adds redundancy to the hidden information and then scatters it throughout the image. A pseudorandom generator is used to select two areas of the image (or patches), patch A and patch B [72]. One can embed more bits by first dividing the image into sub-images and applying the embedding to each of them [73].

1. **Spread Spectrum**

In spread spectrum techniques, hidden data is spread throughout the cover-image making it harder to detect. A system proposed by Marvel et al. combines spread spectrum communication, error control coding and image processing to hide information in images [74].

**3.3 Techniques for hiding the data**

**3.3.1 LSB Substitution Method [5]**

The most well-known steganographic technique in the data hiding field is least-significant-bits (LSBs) substitution. This method embeds the fixed-length secret bits in the same fixed length LSBs of pixels. Although this technique is simple, it generally causes noticeable distortion when the number of embedded bits for each pixel exceeds three. Several adaptive methods for steganography have been proposed to reduce the distortion caused by LSBs substitution. For example, adaptive methods vary the number of embedded bits in each pixel, and they possess better image quality than other methods using only simple LSBs substitution. However, this is achieved at the cost of a reduction in the embedding capacity.

**3.3.2 Optimum Pixel Adjustment Procedure [79]**

The proposed Optimal Pixel adjustment Procedure (OPAP) reduces the distortion caused by the LSB substitution method. In OPAP method the pixel value is adjusted after the hiding of the secret data is done to improve the quality of the stego image without disturbing the

* ***Procedure for hiding:***  First a few least significant bits are substituted with the data to be hidden.
* Then in the pixel, the bits before the hidden bits are adjusted suitably if necessary to give less error.
* Let n LSBs be substituted in each pixel.
* Let d= decimal value of the pixel after the substitution.
* d1 = decimal value of last n bits of the pixel.
* d2 = decimal value of n bits hidden in that pixel.
* ***Retrieval:*** The retrieval follows the extraction of the least significant bits (LSB) as hiding is done using simple LSB substitution.
* ***Advantages:***
  + - * Simple methodology.
      * Easy retrieval.
      * Improved stego-image quality than LSB substitution.

**3.3.3 Inverted Pattern Approach (IP) [80]**

This inverted pattern (IP) LSB substitution approach uses the idea of processing secret messages prior to embedding. In this method each section of secret images is determined to be inverted or not inverted before it is embedded. In addition, the bits which are used to record the transformation are treated as secret keys or extra data to be re-embedded.

* ***The embedding procedure is:*** The embedded string is ***S***, the replaced string is ***R,*** and the embedded bit string to divided to ***P*** parts.
* Let us consider n-bit LSB substitution to be made. Then S and R are of n-bits length.
* For P part in i = 1 to P

If MSE (Si, Ri) ≤ MSE (S*’*I, Ri)

Choose Si for embedding

Mark key (i) as logic ‗0‘

If MSE(Si, Ri) ≥ MSE( S‘i,Ri)

Choose S‘ i for embedding

Mark key (i) as logic ‗1‘

MSE – Mean Square Error.

End

where, S is the data to be hidden, S‘ is the data to be hidden in inverted form.

* ***Procedure for retrieval is:*** The stego-image and the key file are required at the retrieval side.
* First corresponding numbers of LSB bits are retrieved from the stego-image.
* If the key is ‗0‘, then the retrieved bits are kept as such.
* Else if the key is ‗1‘, then the bits are inverted.
* The bits retrieved in this manner from every pixel of the stego-image gives the data hidden.

**3.3.4 IP Method Using Relative Entropy [80]**

Relative entropy [81] measures the information discrepancy between two different sources with an optimal threshold obtained by minimizing relative entropy. In this method instead of finding the mean square error for inverted pattern approach, the relative entropy is calculated to decide whether S or S‘ suites the pixel. In probability theory and information theory, the Kullback Leibler divergence (also information divergence, information gain, or relative entropy) is a non-symmetric measure of the difference between two probability distributions *P* and *Q*.

* ***Embedding procedure is:*** Divide the cover image into ***P*** blocks of same size, the embedding string is ***S***, and the replaced string is ***R.***
* For P part in i =1 to P. If rel. entropy (Si, Ri) ≤ rel .entropy (S‘i, Ri).

Choose Si for embedding

Mark key(i) as logic ‗0‘

If rel.entropy (Si,Ri) ≥ rel.entropy (S‘i,Ri)

Choose S‘i for embedding

Mark key(i) as logic ‗1‘

End

where, S is the data to be hidden, S‘ is the data to be hidden in inverted form.

* ***Procedure for retrieval is:*** The stego-image and the key file are required at the retrieval side.
* First corresponding numbers of LSB bits are retrieved from the stego-image.
* If the key is ‗0‘, then the retrieved bits are kept as such.
* Else if the key is ‗1‘, then the bits are inverted.
* The bits retrieved in this manner from every pixel of the stego-image gives the data hidden

**3.3.5 Hiding Streams of 1s and 0s**

The usual steganographic methods fetch few bits from the secret data to be embedded. But this method fetches the 1s or 0s present consecutively for hiding. This is an innovative steganographic method where the data to be hidden is converted to binary. The number of 1s and 0s are counted and stored in the pixels of the cover image in this method. The number of 1s is stored in the odd columns of the pixel and the number of 0s is stored in the even columns.

**3.3.6 Pixel Value Differencing (PVD) [82, 83, 84]**

Pixel Value Differencing is able to provide a high quality stego image in spite of the high capacity of the concealed information. That is, the number of insertion bits is dependent on whether the pixel is an edge area or smooth area. In edge area the difference between the adjacent pixels is more, whereas in smooth area it is less. While human perception is less sensitive to subtle changes in edge areas of a pixel, it is more sensitive to changes in the

smooth areas. Two techniques of PVD are explained here.

**3.4 Essential features of steganographic techniques**

All the above mentioned algorithms for image steganography have different strong and weak points and it is important to ensure that one uses the most suitable algorithm for an application. All steganographic algorithms have to comply with a few basic requirements. The most important requirement is that a steganographic algorithm has to be imperceptible. The authors propose a set of criteria to further define the imperceptibility of an algorithm. These requirements are as follows [75]:

* **Invisibility** – The invisibility of a steganographic algorithm is the first and foremost requirement, since the strength of steganography lies in its ability to be unnoticed by the human eye. The moment that one can see that an image has been tampered with, the algorithm is compromised
* **Payload capacity** – Unlike watermarking, which needs to embed only a small amount of copyright information, steganography aims at hidden communication and therefore requires sufficient embedding capacity.
* **Robustness against statistical attacks** – Statistical steganalysis is the practice of detecting hidden information through applying statistical tests on image data. Many steganographic algorithms leave a ‘signature’ when embedding information that can be easily detected through statistical analysis. To be able to pass by a warden without being detected, a steganographic algorithm must not leave such a mark in the image as be statistically significant.
* **Robustness against image manipulation** – In the communication of a stego image by trusted systems, the image may undergo changes by an active warden in an attempt to remove hidden information. Image manipulation, such as cropping or rotating, can be performed on the image before it reaches its destination. Depending on the manner in which the message is embedded, these manipulations may destroy the hidden message. It is preferable for steganographic algorithms to be robust against either malicious or unintentional changes to the image.
* **Independent of file format** – With many different image file formats used on the Internet, it might seem suspicious that only one type of file format is continuously communicated between two parties. The most powerful steganographic algorithms thus possess the ability to embed information in any type of file. This also solves the problem of not always being able to find a sui image at the right moment, in the right format to use as a cover image.
* **Unsuspicious files** – This requirement includes all characteristics of a steganographic algorithm that may result in images that are not used normally and may cause suspicion. Abnormal file size, for example, is one property of an image that can result in further investigation of the image by a warden. The following table compares least significant bit (LSB) insertion in BMP and in GIF files, JPEG compression steganography, the patchwork approach and spread spectrum techniques as discussed in section 3, according to the above requirements:

**3.4 Conventional Least-Significant Bit (LSB) Technique**

The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. In 8 bit images, one bit of information can be hidden. The cover image is shown in Figure 3.1 and a hidden message is shown in Figure 3.1 A stego-image is obtained by applying LSB algorithm on both the cover and hidden images. The hidden image is extracted from the stego-image by applying the reverse process [76, 77]. If the LSB of the pixel value of cover image C(i,j) is equal to the message bit m of secret massage to be embedded, C(i,j) remain unchanged; if not, set the LSB of C(i, j) to m. The message embedding procedure is given below-

S (i,j) = C (i,j) - 1, if LSB(C (i,j)) = 1 and m = 0

S (i.j) = C (i,j), if LSB(C(i,j)) = m

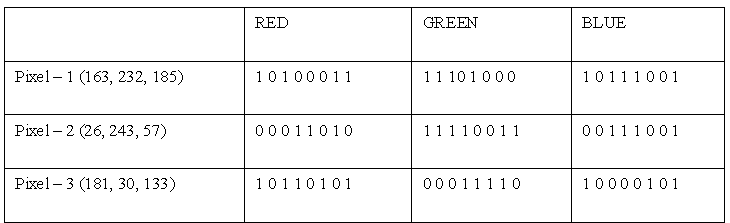
S (i,j) = C (i,j) + 1, if LSB(C (i,j)) = 0 and m = 1

Where LSB(C(i, j)) stands for the LSB of cover image C(i,j) and m is the next message bit to be embedded. S(i,j) is the stego image

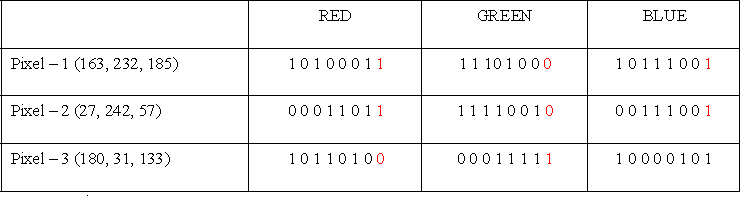
Example:

Let us consider a data byte 1 0 1 1 0 1 0 1

Let the RGB component of 3 pixels read from cover image be:



The RGB component of these 3 pixels in the stegno image shall be:



Bits marked red are the bits from data byte (most significant bit to least significant bit) read. Here blue component of every third pixel remains unchanged.

As we already know each pixel is made up of three bytes consisting of either a 1or a 0. For example, suppose one can hide a message in three pixels of an image (24-bit colors). Suppose the original 3 pixels are:

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

A steganographic program could hide the letter "J" which has a position 74 into ASCII character set and have a binary representation "01001010", by altering the channel bits of pixels.

(11101010 11101001 11001010)

(01100110 11001011 11101000)

(11001001 00100100 11101001)

In this case, only four bits needed to be changed to insert the character succesfully. The resulting changes that are made to the least significant bits are too small to be recognised by the human eye, so the message is effectively hidden. The advantage of LSB embedding is its simplicity and many techniques use these methods [78]. LSB embedding also permits high perceptual transparency.

**3.5 Implemented modified LSB technique**

In current work we embedded the data in 4-lsb bits of the RGB components of every pixel. Thus every pixel can accommodate 12 bits of data, we ready two consecutive pixels and embed 24 bits (3 bytes) into RGB components of these pixels.

Let the image be of size n\*m (where n is number of columns of pixels and m is number of rows of pixels), thus size of file that can be embedded into the image is Floor ((n\*m\*3)/2).

Let the data stream to be embedded is, “PM **may** visit Jaipur on Sunday”. Assume that we have to embed ‘may’ underlined in the above sentence. Let the RGB component of the two pixels (say Pixel(i, j) and Pixel(i+1, j)) where it will embedded are as follows:

Red (Pixel(i, j)) 🡪 (134)10 🡪(10000110)2

Green (Pixel(i, j)) 🡪 (14)10 🡪(00001110)2

Blue (Pixel(i, j)) 🡪 (108)10 🡪(01101100)2

Red (Pixel(i+1, j)) 🡪 (132)10 🡪(10000100)2

Green (Pixel(i+1, j)) 🡪 (18)10 🡪(00010010)2

Blue (Pixel(i+1, j)) 🡪 (110)10 🡪(01101110)2

The ASCII codes for “may” which is to be embedded are

Decimal Binary 2’s Complement

m : (109)10 🡪 (01101101)2 🡪 (10010011)2

a : (97)10 🡪 (01100001)2 🡪 (10011111)2

y : (121)10 🡪 (01111001)2 🡪 (10000111)2

The 4 MSB of ‘m’ (marked red) will be embedded into Red component of Pixel(i,j) while 4 LSB of ‘m’ will be embedded into Green component of Pixel(i,j). same for others, thus after the embedding process the contents of the RGB components of the Pixel(i, j) and Pixel(i+1, j) shall be as shown below:

Red (Pixel(i, j)) 🡪 (131)10 🡪(10001001)2

Green (Pixel(i, j)) 🡪 (3)10 🡪(00000011)2

Blue (Pixel(i, j)) 🡪 (105)10 🡪(01101001)2

Red (Pixel(i+1, j)) 🡪 (143)10 🡪(10001111)2

Green (Pixel(i+1, j)) 🡪 (24)10 🡪(00011000)2

Blue (Pixel(i+1, j)) 🡪 (103)10 🡪(01100111)2

* + 1. **Embedding the data into image**

Embedding the data in image is done using a very simple LSB technique in which the last four bit of the each component RGB 24 bit pixel image. The embedding process is as follows. Fig.3.2 shows the procedure of embedding the data in image

* Inputs : Cover image, stego-key and the text file
* Output: stego image
* Procedure

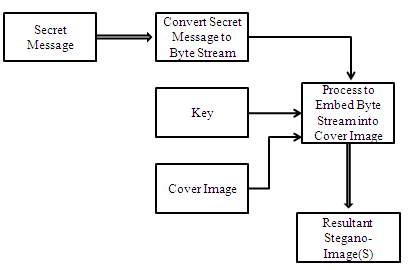


Fig.3.3 embedding the data using image steganography

Step 1: Extract the pixels of the cover image. Fig 3.2 shows the process involved in LSB encoding.

Step 2: Extract the characters of the text file.

Step 3: Extract the characters from the Stego key.

Step 4: Choose first pixel and pick characters of the Stego key and place it in first component of pixel.

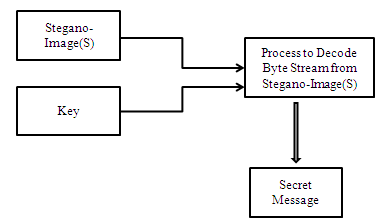


Fig 3.2 process of LSB encoding

Step 5: Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.

Step 6: Insert characters of text file in each first component of next pixels by replacing it.

Step 7: Repeat step 6 till all the characters has been embedded.

Step 8: Again place some terminating symbol to indicate end of data.

Step 9: Obtained stego image.

* + 1. **Process of image dividing**

The algorithm used can be summarized as like we consider a cover image Image\_1 of size 120\*140, i.e. width of image is 120 pixels and height is 140 pixels, and a data file data.txt of size 200 KB. Here total number of pixels in the image = 120 \* 140 = 16800. In every pixel 12-bits of data can be embedded, last four bit at least significant bit of RGB component of pixel, so total data bits that can be embedded in the cover image = 16800 \* 12 = 201600 bits or 25200 bytes.

Total byte of the data file = 200 \* 1024 = 204800

Here number of bytes in the data file is greater than byte that can be accommodated in the cover image. Thus we need multiple copies of the cover image to embed the data file, number of Ceiling (204800 / 25200) = 33. So 33 copies of cover image will be created rather denying the client as size of data file is larger than the number of bytes that can be accommodated in the cover image.

Now the data file will also be broken down in 33 small chunks by the process, where first 32 chunks will of size 6300 bytes each and last chunk will be of 3200 bytes (204800 – 6300 \* 32). Now iterative process will be called to embed every bit of data file into the 33 copies of the cover image to generate the stegno images which shall then be uploaded over the cloud.

In such way the files having a large size can also be uploaded by parts practically having no limit over the data size to be uploaded. The Fig 3.5 below shows the process in a sequential manner.

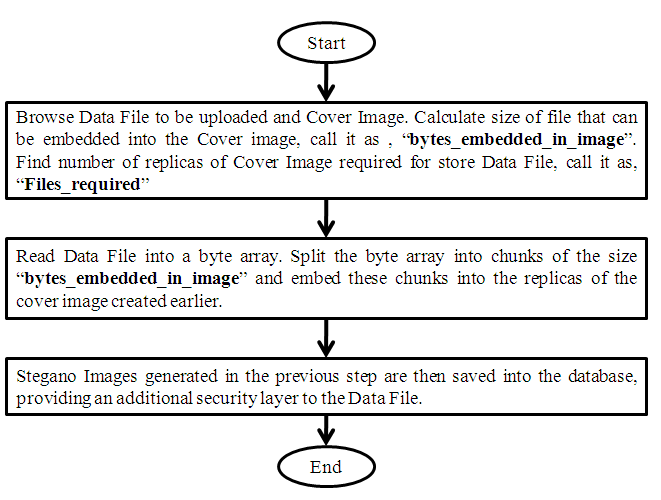
****

Fig 3.5 the flow chart of data uploads over cloud

**3.4.3 Encryption process (Cover Image, Data File)**

// Here Cover\_Image is the image in which data is to be stegnographed, and Data\_File is the data file which is to be stegnographed.

**Step 1**. Initialize length\_data\_file = Length(Data\_File).

// length\_data\_file store the number of bytes in Data\_File

**Step 2**. Initialize byte\_array[length\_data\_file].

//byte\_array is a byte type array.

**Step 3**. Read the Data\_File into byte\_array[].

//Data\_file is read into byte\_array[] as a byte stream.

**Step 4**. Initialize bytes\_embedded\_in\_image = Floor((Cover\_Image.width \* Cover \_Image. height) \*3/ 2)

//byte\_embedded\_in\_image shows the maximum number if byte that can be stored in Cover\_Image,

//as per our algorithm

**Step 5**. Initialize file\_required = Ceiling(length\_data\_file/ bytes\_embedded\_in\_image)

//file\_required shows total number of replicas of the Cover\_Image in order to accommodate the

//Data\_File, as our algorithm does not apply any restriction on the size of Data\_File, thus multiple

//copies may be required to embed the complete Data\_File.

**Step 6**. Initialize a byte\_2\_array[][] = [file\_required][].

//Here we initialized a byte\_2\_array a 2-dimensional byte array to accommodate the byte\_array[].

//We used a 2-dimensional array of byte in order reduce the space complexity, so that only a part of

//the array may be passed to the function (Stegnography explained later). Here number of rows in

//the 2-dimensional array is equal to number of Cover\_Image required to embed the Data\_File

**Step 7**. Initialize number of columns of byte\_2\_array[][]

For i = 0 to (file\_required-1) step up 1

Loop

If(I < (file\_required-1))

byte\_2\_array[i][ bytes\_embedded\_in\_image]

Copy byte\_array[(i\* bytes\_embedded\_in\_image)…..(( i+1)\* bytes\_embedded\_in\_image – 1)] into byte\_2\_array[i].

Call Stegnography (Cover\_Image, byte\_2\_array\_[i], i+1)

Else

byte\_2\_array[i][ length\_data\_file – (i \* bytes\_embedded\_in\_image)]

Copy byte\_array[(i\* bytes\_embedded\_in\_image)…..(data\_file\_length – 1)] into byte\_2\_array[i].

Call Stegnography (Cover\_Image, byte\_2\_array\_[i], i+1)

End If

End Loop

//Here column length of byte\_2\_array for each row is initialized and respective bytes are copied.

**Step 8**. Exit

* + 1. **Stegnography (Cover\_Image, Byte\_Array, Num)**

// Here Cover\_Image is image in which Byte\_Array shall be embedded, Num shall be used to as name for stegno-image to be uploaded over cloud.

**Step 1**. Initialize byte\_index = 0, read\_byte\_flag = 0, bit\_array []. //Here byte\_index is used to keep track of the index of Byte\_Array currently being precessed. i = 0, j = 0.

**Step 2.** While byte\_index < Length(Byte\_Array)

//Length(Byte\_Array) finds the length of the Byte\_Array

Loop

**Step 3.** Set byte1 = null, byte2 = null, byte3 = null

//Initialize byte variables to null

Set pixel\_color1 = null, pixel\_color2 = null

//Initialize pixel\_color variables to null

**Step 4.** Read byte1 = Byte\_Arra[byte\_index++].

//Read three consecutive bytes from Byte\_Array (if available)

If(byte\_index < Length(Byte\_Array)

Read byte2 = Byte\_Array[byte\_index++].

If(byte\_index < Length(Byte\_Array)

Read byte3 = Byte\_Array[byte\_index++].

**Step 5.** If(i == Width(Cover\_Image)) Then

Set i = 0; j = j + 1;

Initialize pixel\_color1 = image1.Pixel(i++, j)

//Retrieve the Pixel(i,j) into color\_pixel

Set red\_array1 = pixel\_color1.Red

//Retrieve the Red component for color\_pixel as a 8-bit array (range 0 -255)

Set green\_array1 = pixel\_color1.Green

//Retrieve the Green component for color\_pixel as a 8-bit array (range 0 -255)

Set blue\_array1 = pixel\_color1.Blue

//Retrieve the Blue component for color\_pixel as a 8-bit array (range 0 -255)

End if.

**Step 5.** If(i == Width(Cover\_Image)) Then

Set i = 0; j = j + 1;

Initialize pixel\_color2 = image1.Pixel(i++, j)

//Retrieve the Pixel(i,j) into color\_pixel

Set red\_array2 = pixel\_color2.Red

//Retrieve the Red component for color\_pixel as a 8-bit array (range 0 -255)

Set green\_array2 = pixel\_color2.Green

//Retrieve the Green component for color\_pixel as a 8-bit array (range 0 -255)

Set blue\_array2 = pixel\_color2.Blue

//Retrieve the Blue component for color\_pixel as a 8-bit array (range 0 -255)

End if.

**Step 6.** Set red\_array1[4…7] = byte1[0…3]

//Set least significant 4 bits of red component of pixel to byte[0], byte[1], byte[2] and byte[3] Set green\_array1 [4…7] = byte1[4…7]

// Set least significant 4 bits of green component of pixel to byte[4], byte[5], byte[6] and byte[7]

**Step 7**. If (byte2 != null) Then

If (pixel\_color2 == null) Then

Set blue\_array1[0…7] = byte2[0…7]

//Set blue component of pixel to byte2 value

Else

Set blue\_array1[4…7] = byte2[0…3]

//Set least significant 4 bits of blue component of pixel to byte[0], byte[1], byte[2] and byte[3] Set red\_array2[4…7] = byte2[4…7]

//Set least significant 4 bits of red component of pixel to byte[4], byte[5], byte[6] and byte[7]

End If.

End If.

**Step 8**. If (byte3 != null) Then

Set green\_array2[4…7] = byte3[0…3]

//Set least significant 4 bits of green component of pixel to byte[0], byte[1], byte[2] and byte[3]

Set blue\_array2[4…7] = byte3[4…7]

//Set least significant 4 bits of blue component of pixel to byte[4], byte[5], byte[6] and byte[7]

End If.

**Step 9.** Set pixe1\_color1 = color(red\_array, green\_array, blue\_array)

//Set pixel\_color1 to updated value of Red, Green, Blue component.

**Step 10.**If (pixel\_color2 != null)

Set pixe1\_color2 = color(red\_array, green\_array, blue\_array)

// Set pixel\_color1 to updated value of Red, Green, Blue component.

End If.

**Step 11.** End While.

**Step 12.** Save Cover\_Image as Num.bmp

//Cover\_Image is uploaded as Num.bmp over cloud.

**Step 13.** Exit

**3.6 Data Extraction**

The extraction process is as follows.

**Inputs** : Stego-image fille, stego-key

**Output** : Secret message.

**Procedure**:

* Step 1: Extract the pixels of the stego image.
* Step 2: Now, start from first pixel and extract stego key characters from first component of the pixels. Follow Step3 up to terminating symbol, otherwise follow step 4.
* Step 4: If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.
* Step 5: If the key is correct, then go to next pixels and extract secret message characters from first component of next pixels. Follow Step 5 till up to terminating symbol, otherwise follow step 6.
* Step 6: Extract secret message [18, 20].

**3.6.1 Decription (Size\_data\_file, Number\_file)**

**Decription (Size\_data\_file, Number\_file)**

// Here Size\_data\_file is original size of the data file, Number\_file is the number of Stegno-images created while encryption.

**Step 1.** Initialize byte\_array[Size\_data\_file], byte\_index = 0

//Initialize byte\_array[] of bytes equal to number of bytes in the original data file.

**Step 2.** For i = 1 to Number\_file step by 1

Loop

**Step 3.** Set x = 0, y = 0.

//x and y are variables to read pixels of Stegno\_image

pixel\_color1 = null, pixel\_color2 = null

//Initialize pixel\_color variables to null

**Step 4**. Retrieve Stegno\_image(i)

//Read ith Stegno\_image

**Step 5**. While(byte\_index < Size\_data\_file) Loop

//Loop to read all data byte

**Step 6**. If(x== Width(Stegno\_Image(i))) Then

Set x = 0, y = y + 1;

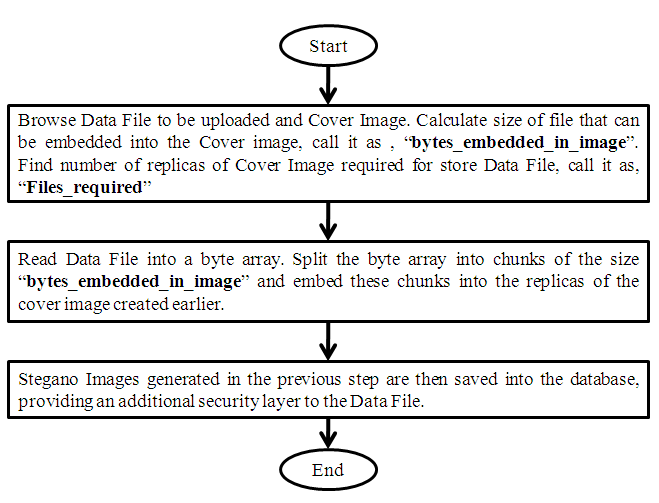


Fig. 3.9 Flowchart for data download from cloud

End If;

**Step 7**. If(y == Height(Stegno\_Image(i))) Then

break.

Else

pixel\_color1 = Stegno\_Image(i).Pixel(x++, y)

//Read Pixel(x,y) of stegno\_image(i) to pixel\_color1

End If.

**Step8.** If(x== Width(Stegno\_Image(i))) Then

Set x = 0, y = y + 1;

End If;

**Step 9**. If(y == Height(Stegno\_Image(i))) Then

break.

Else

pixel\_color2 = Stegno\_Image(i).Pixel(x++, y)

//Read Pixel(x,y) of stegno\_image(i) to pixel\_color2

End If

**Step 10.** If (pixel\_color1 != null) Then

red\_array1 = pixel\_color1.red.

//Retrieve Red, Green, Blue component of pixel\_color1

green\_array1 = pixel\_color1.green.

blue\_array1 = pixel\_color1.blue.

byte\_array[byte\_index][0…3] = red\_array1[4…7] //Set Data bytes

byte\_array[byte\_index++][4…7] = green\_array1[4…7]

If(byte\_index== (Size\_data\_file-1))Then

byte\_array[byte\_index][0…7] = blue\_array1[0…7]

Else

byte\_array[byte\_index][0…3] = blue\_array1[4…7]

End If.

**Step 11.** If(pixel\_color2 != null) Then

red\_array2 = pixel\_color2.red.

//Retrieve Red, Green, Blue component of pixel\_color2

green\_array2 = pixel\_color2.green.

blue\_array2 = pixel\_color2.blue.

byte\_array[byte\_index++][4…7] = red\_array2[4…7] //Set Data bytes

If(byte\_index !=Size\_data\_file) Then

byte\_array[byte\_index][0…3] = green\_array1[4…7]

byte\_array[byte\_index][4…7] = blue\_array1[4…7]

Else

byte\_array[byte\_index][0…3] = blue\_array1[4…7]

End If.

**Step 12.**End While.

**Step 13.** End For

**Step 14.** Exit

* 1. **Summary**

In this Chapter first the some common stenographic techniques were described first then the least significant bit technique for encryption of the data is discussed for a 24 bit RGB image pixels and hence a process to embed the secret data file in a cover image is described the data file is broken in parts if it is having a larger size then the capacity of the cover image and each part is send with a new replica of the cover image. The algorithms and step wise procedure for encryption as well as decryption of data was presented along with the automated system for embedding the data in image.

**Chapter – 4**

**THE AUTOMATED SYSTEM TO UPLOAD THE DATA ON CLOUD**

**4.1 Introduction**

The data bits were hidden in the pixels of the cover image in the last Chapter now the data is practically not available as an individual file on the system but its bits are available in a fixed encrypted format. The encryption format used is very simple LSB technique described in the last Chapter. Now the task is to recover the data from the cover image the process is very and follow back of the encryption process.

Let us have a view of the complete process once again (Fig. 4.1)

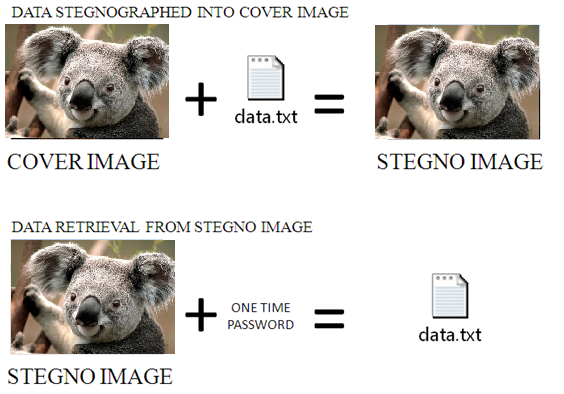


Fig. 4.1 The steganographic process

**4.2The automated system to upload the data on cloud**

The automated system designed for uploading the data on cloud is demonstrated as follows. An image corresponding to each process indicated.

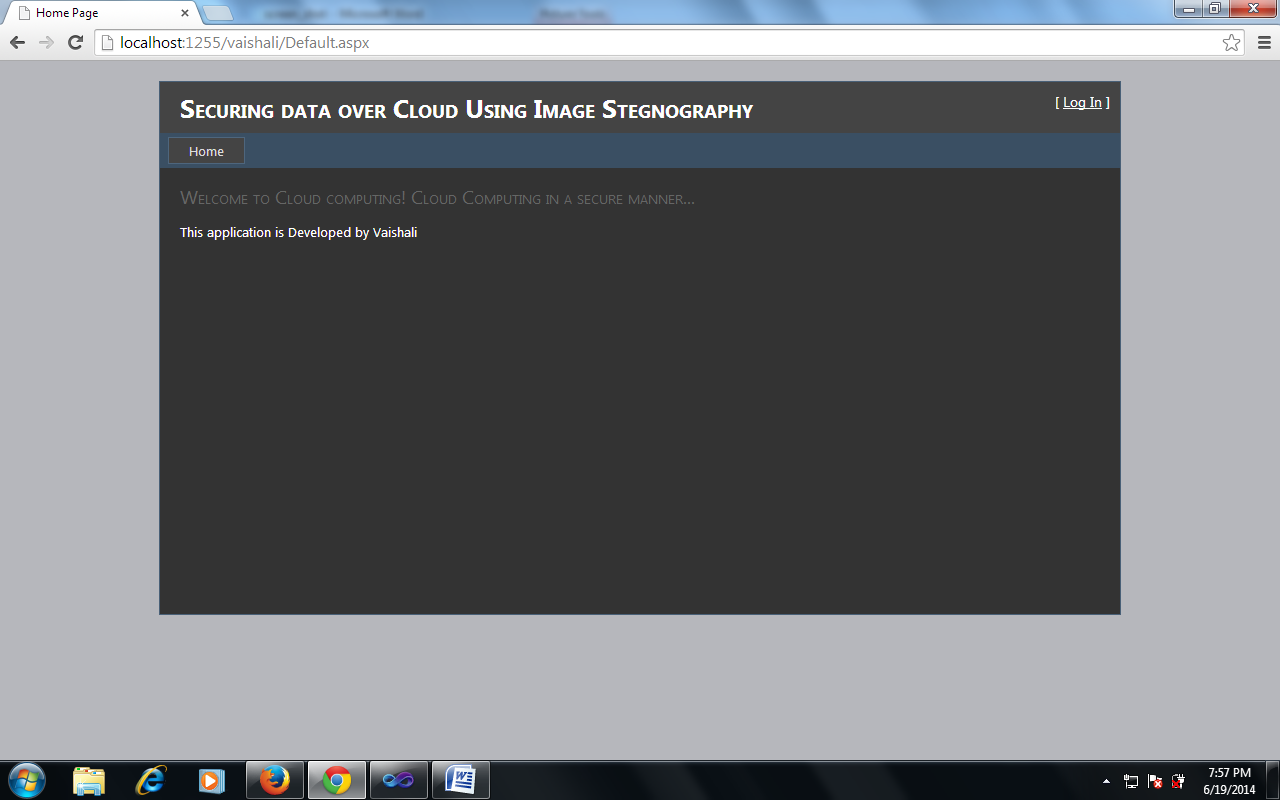


Fig 4.2 Home page of the designed system

Fig 4.2 demonstrates the home page of the cloud. The home page can be browsed online anywhere the client requires

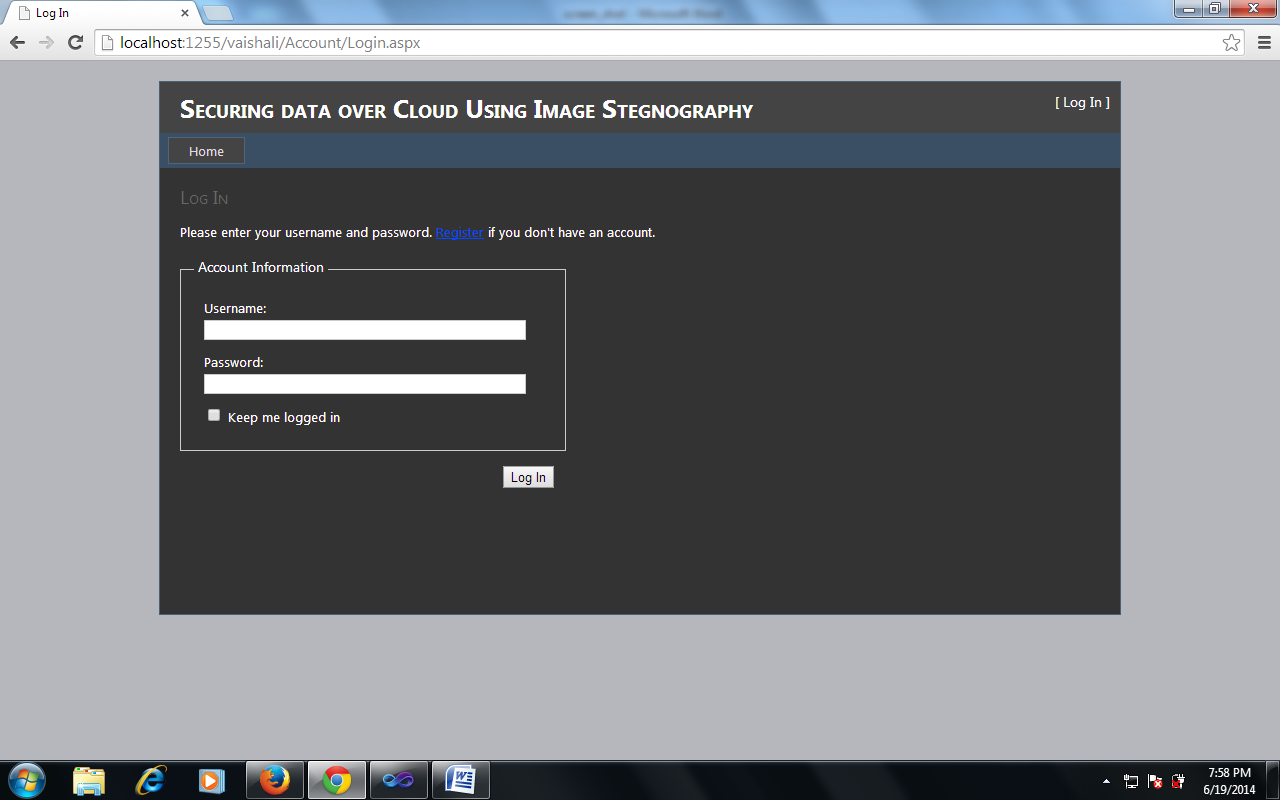


Fig 4.3 (a) Login page of the designed system

This Fig.4.3 shows login page of the cloud; registered client may login using their valid credentials in into the cloud through this page.

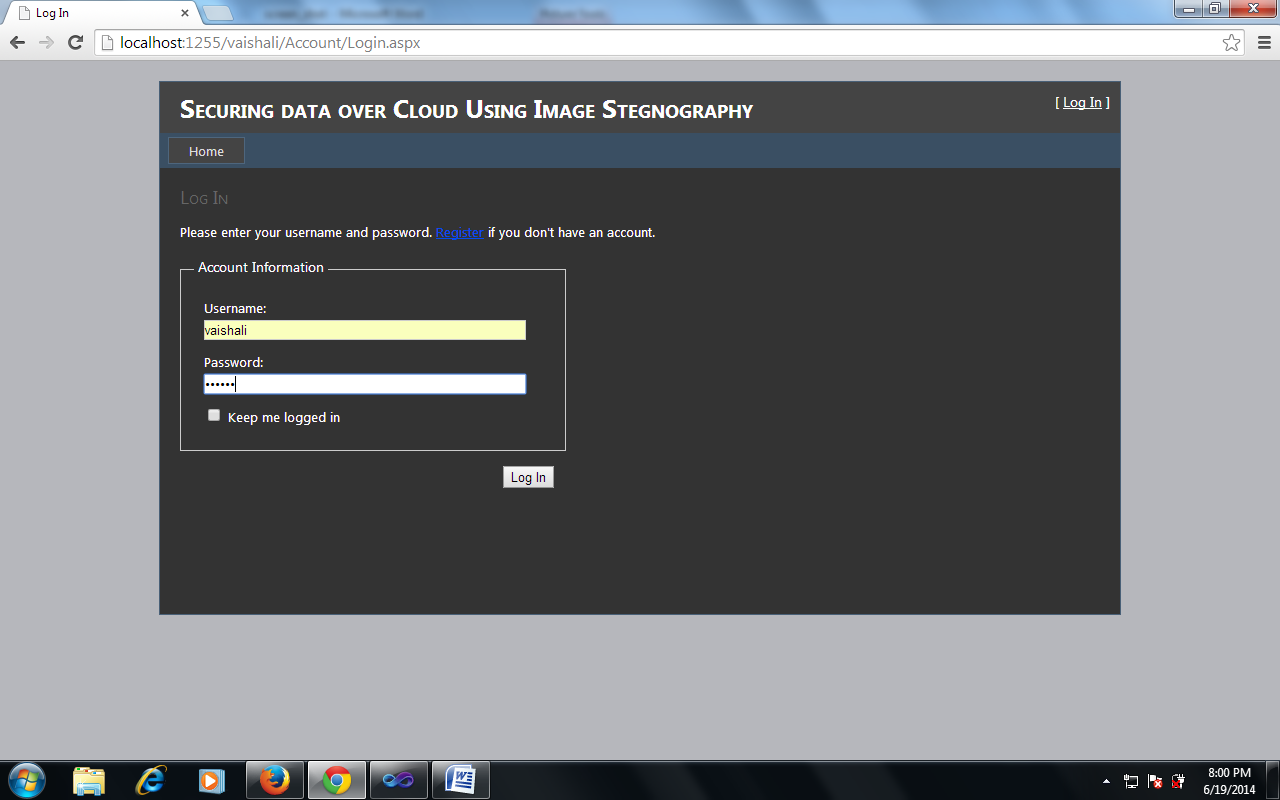


Fig. 4.3 (b) Login process of the designed system

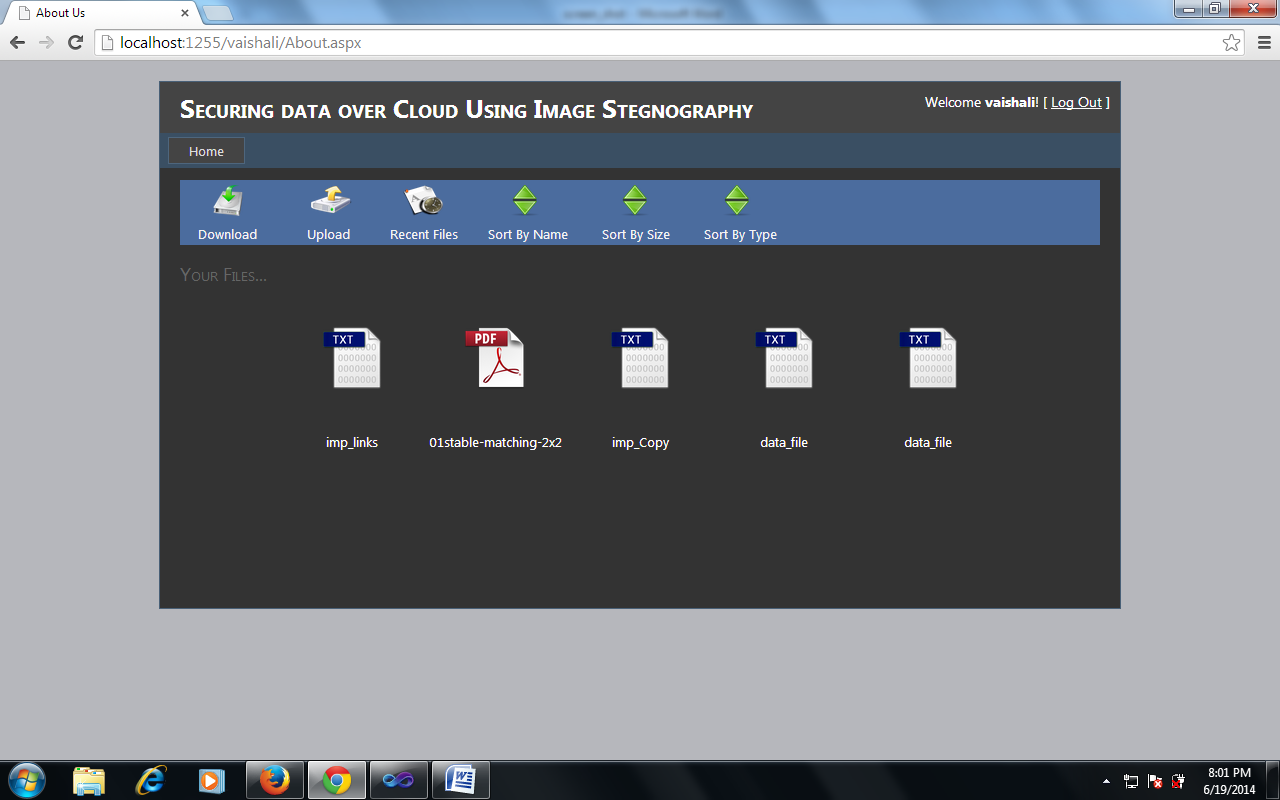


Fig 4.4 Client account window

After entering valid credentials client may click on the login button (Fig.4.3 (b)), if credentials are found valid permission is granted to the client. Once client is successfully logged in, this screen is displayed (Fig. 4.4) which is client account window where all the files uploaded by client earlier (if any) is displayed on the screen. Options available on this screen are:

1. Download
2. Upload
3. Recent Files
4. Sort By Name
5. Sort by Size
6. Sort by Type

When client clicks on upload button, this screen is shown to the client where client has to browse image file (Fig. 4.5 ) to be used as “Cover Image” and a data file to be uploaded over the cloud. This data file will be embedded into the cover image and then uploaded over the cloud.

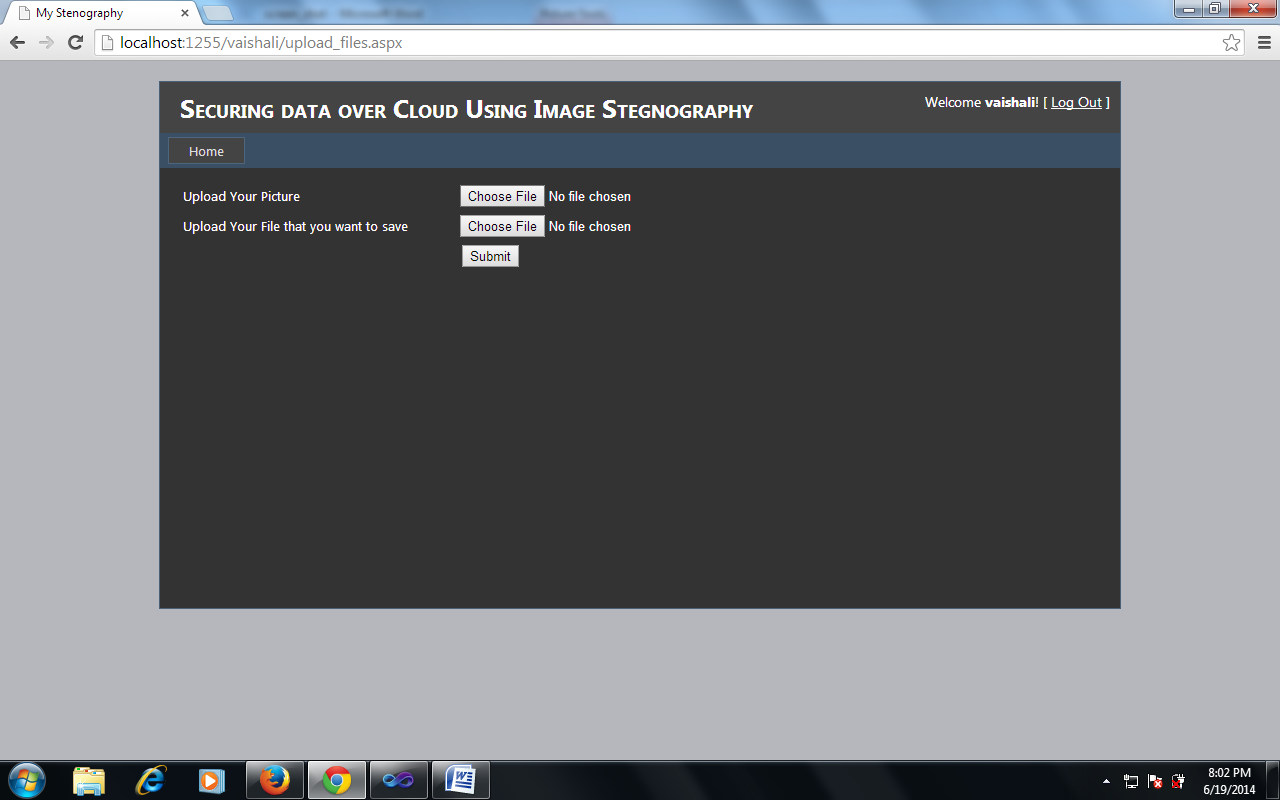


Fig 4.5 Selection of file for upload

When client clicks on upload button, this screen is shown to the client where client has to browse image file to be used as “Cover Image” and a data file to be uploaded over the cloud (Fig. 4.6). This data file will be embedded into the cover image and then uploaded over the cloud.

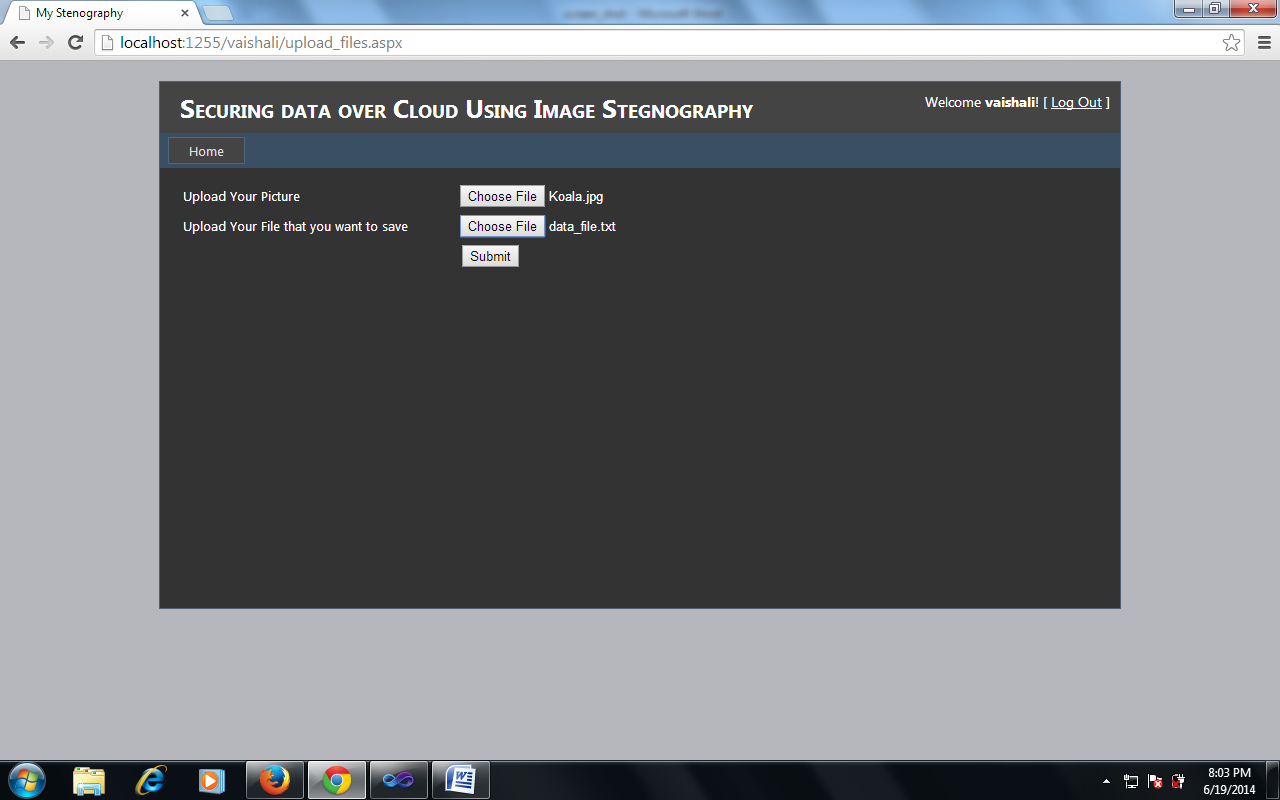


Fig 4.6 Submission process of file and cover image

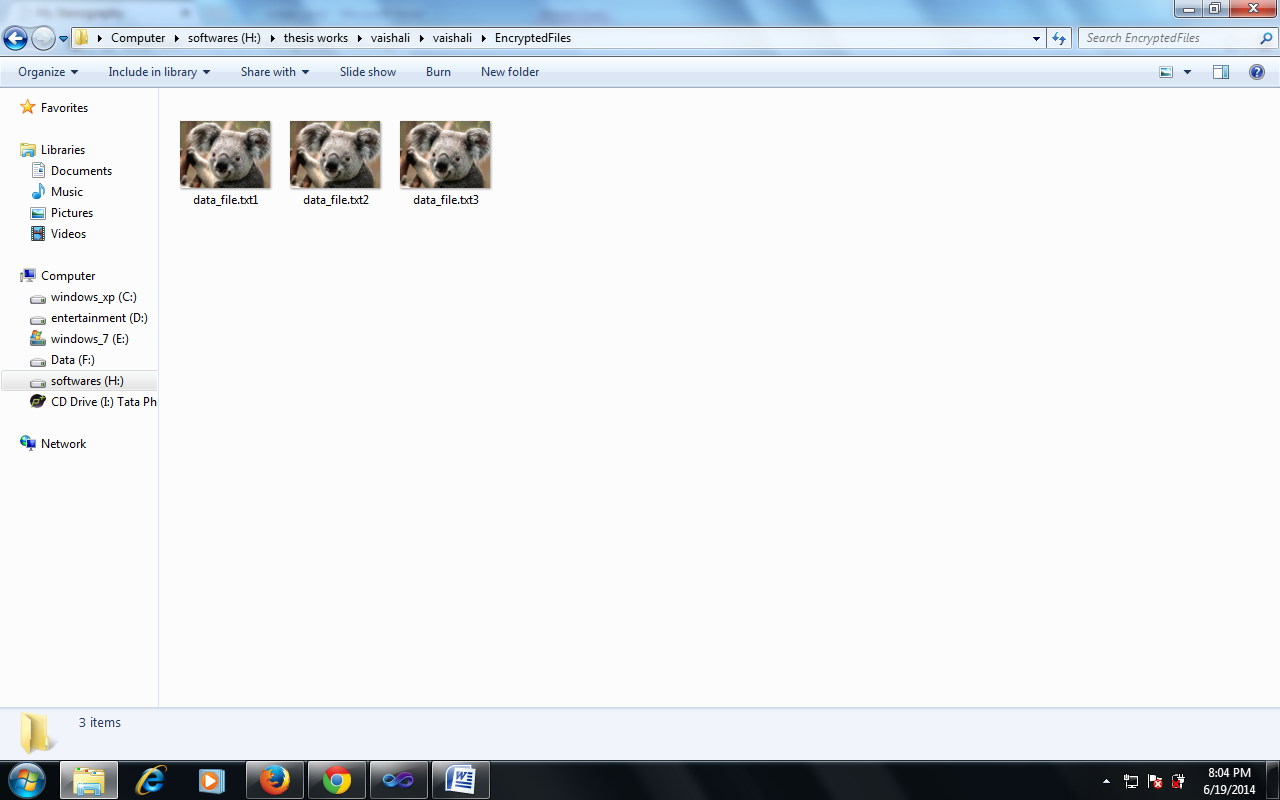


Fig 4.7 Created copies of the cover image

Once client uploads the Cover Image and the Data file, he needs to press the submit button to upload file over the cloud. The screen shot in Fig. 4.7 shows the “Stegano-Images” generated once the submit button is pressed intermediate “Stegano-Images” are generated and if data file is large i.e. size of the file is too large than that can be embedded into one cover image, multiple stegano-images are generated.

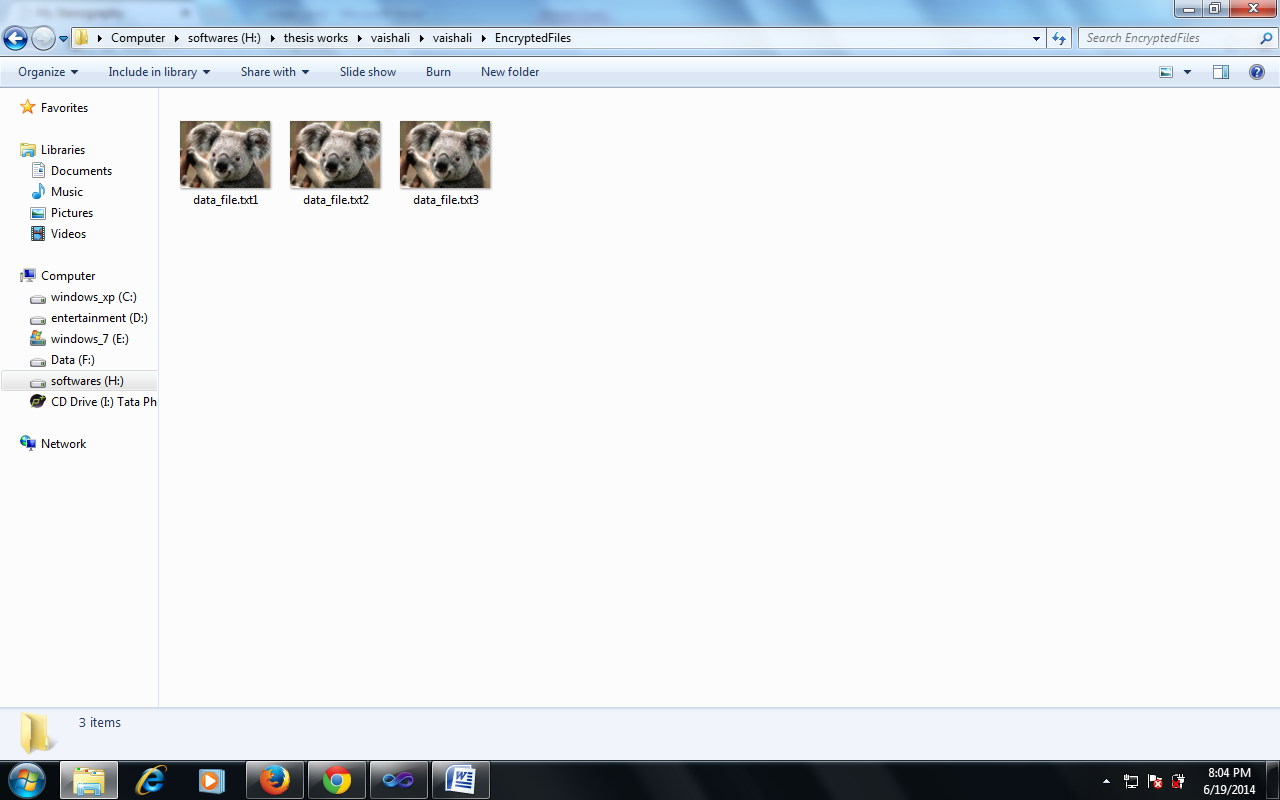


Fig 4.8 Uploading of images to cloud

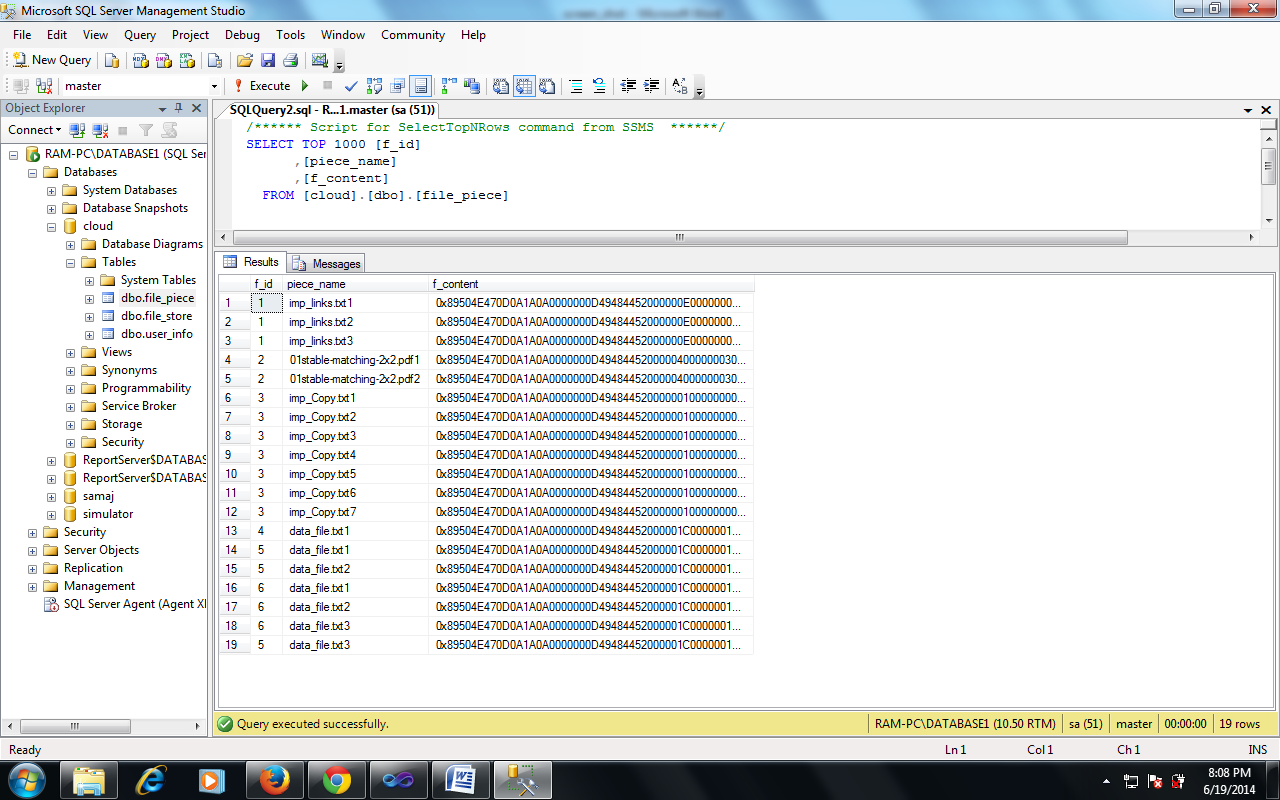


Fig 4.9 Creation of multiple stegao images

The stegno-images shown last screen (Fig 4.8) are then moved to the cloud. The window after the movement is as shown above. If size of file is large multiple stegano-images are generated and uploaded over the cloud (Fig 4.9).

The data encrypted in the cover is available as the LSB’s of the stegano image pixels. The retrieval process described in this Chapter is very easy and straight forward simply the LSB’s of the pixels of the stegano image are encrypted so the data bits are recovered and reformed in the form of the data. If the data was hidden by parts in the form of more than one stegano image then the data bits are recovered from different images and the parts are combined to form the data file again back. In this Chapter first the procedure algorithm for recovering the data is explained. Then the automated system designed for the data recovery is demonstrated.

**4.3 Automated procedure to recover the image**

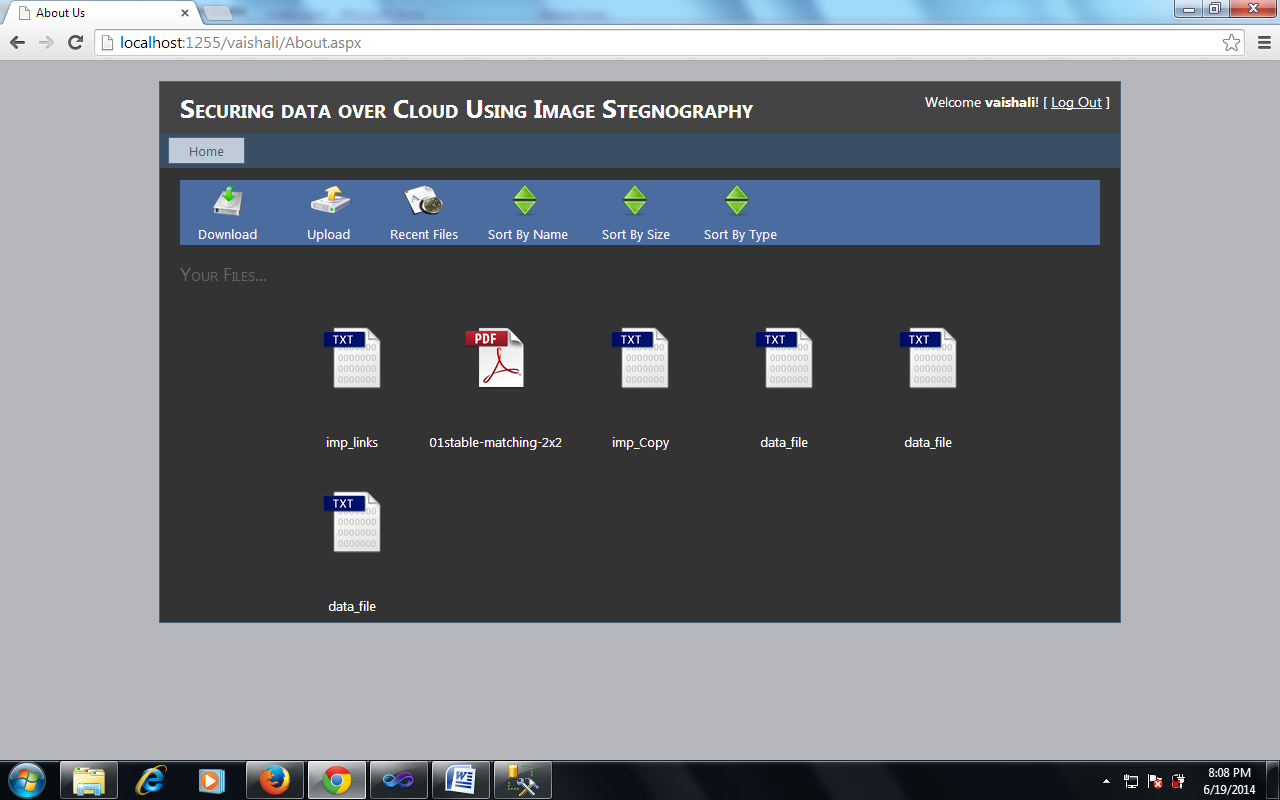


Fig. 4.10 The download window

To download a file, client has to log in again and the window and the files which were uploaded earlier are shown in the window. In the download window the client has to click on the file icons of the file to be downloaded as shown on the screen (Fig. 4.10).

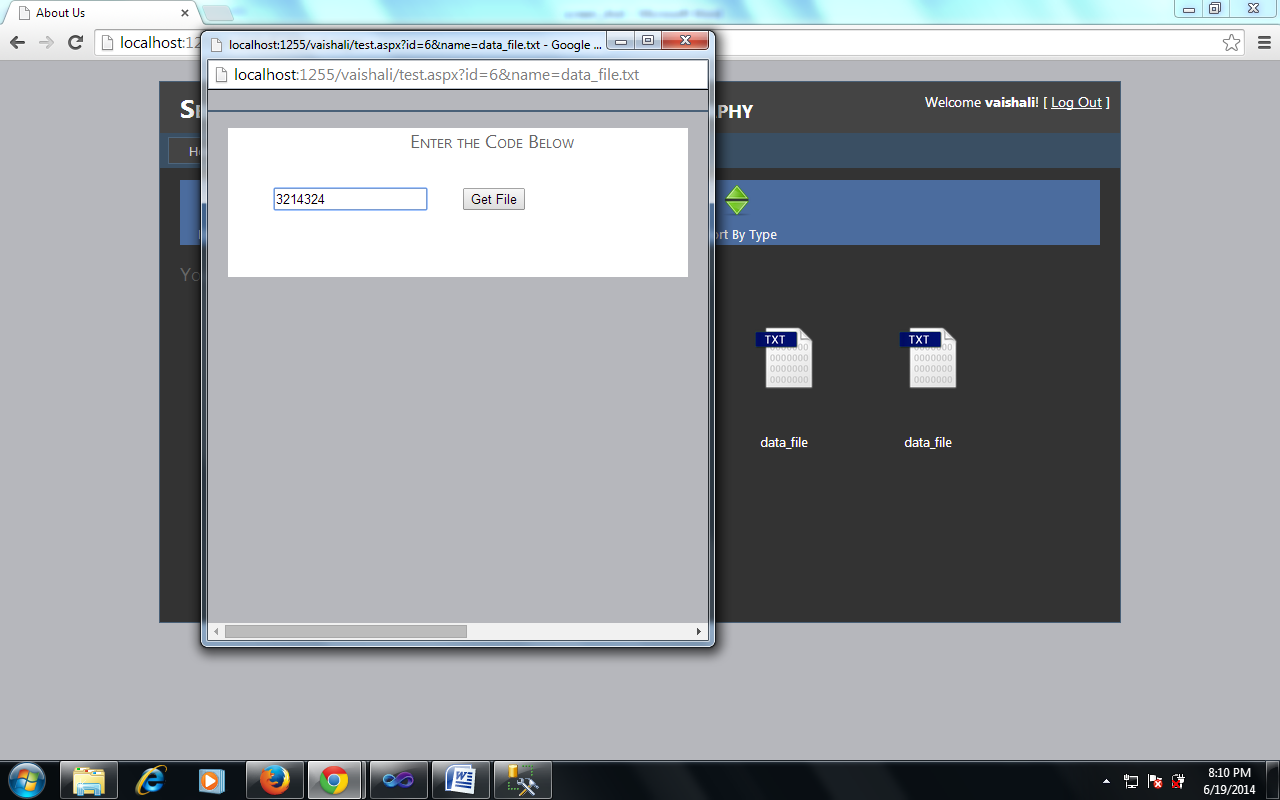


Fig. 4.11 The verification procedure using onetime pass word

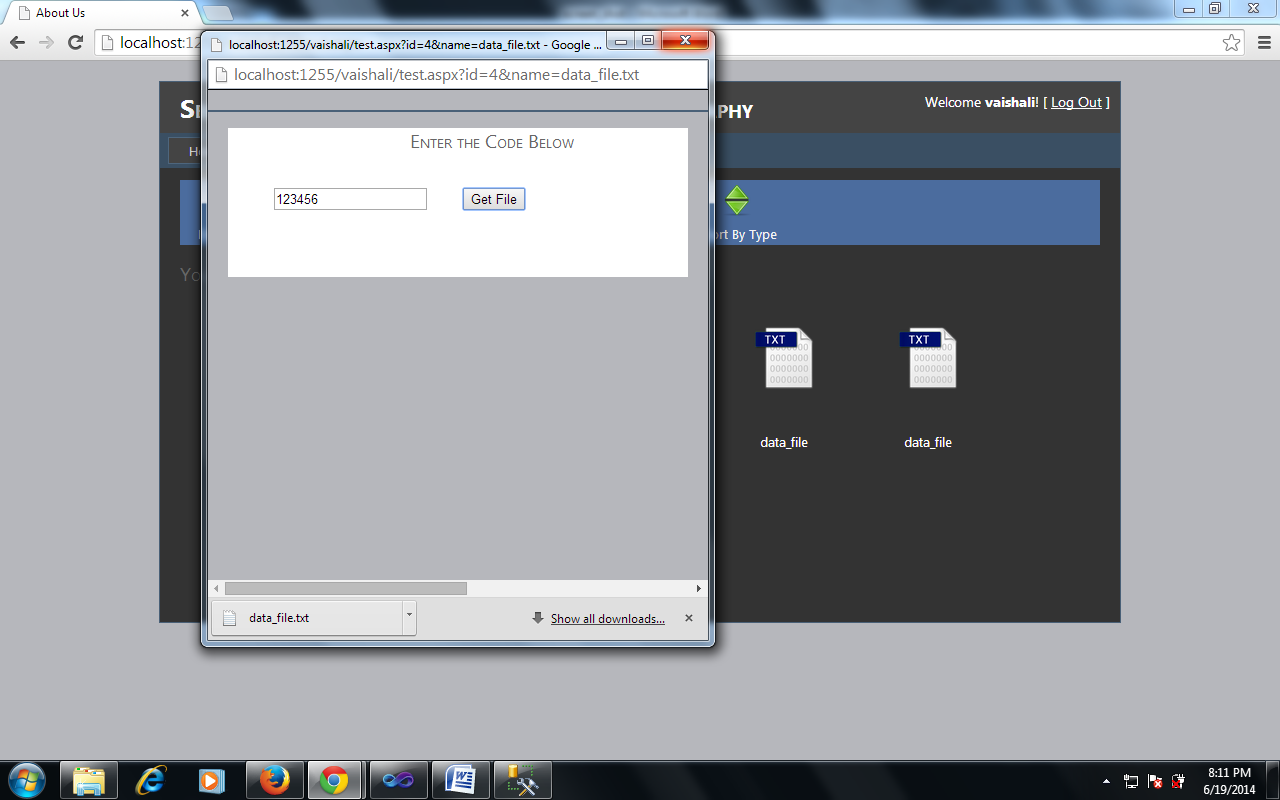


Fig. 4.12 Download options after verification

Once client clicks on any of the icon a onetime password (OTP) is generated and mailed to the client, which has to be enter by the client on this screen (Fig.4.12). This OTP is valid for the time period of the current session, once session expires the OPT will not be accepted by the system, a separated OTP will generated for every file to be downloaded by the client. If OTP entered by client is correct, file is available for download (Fig 4.13).

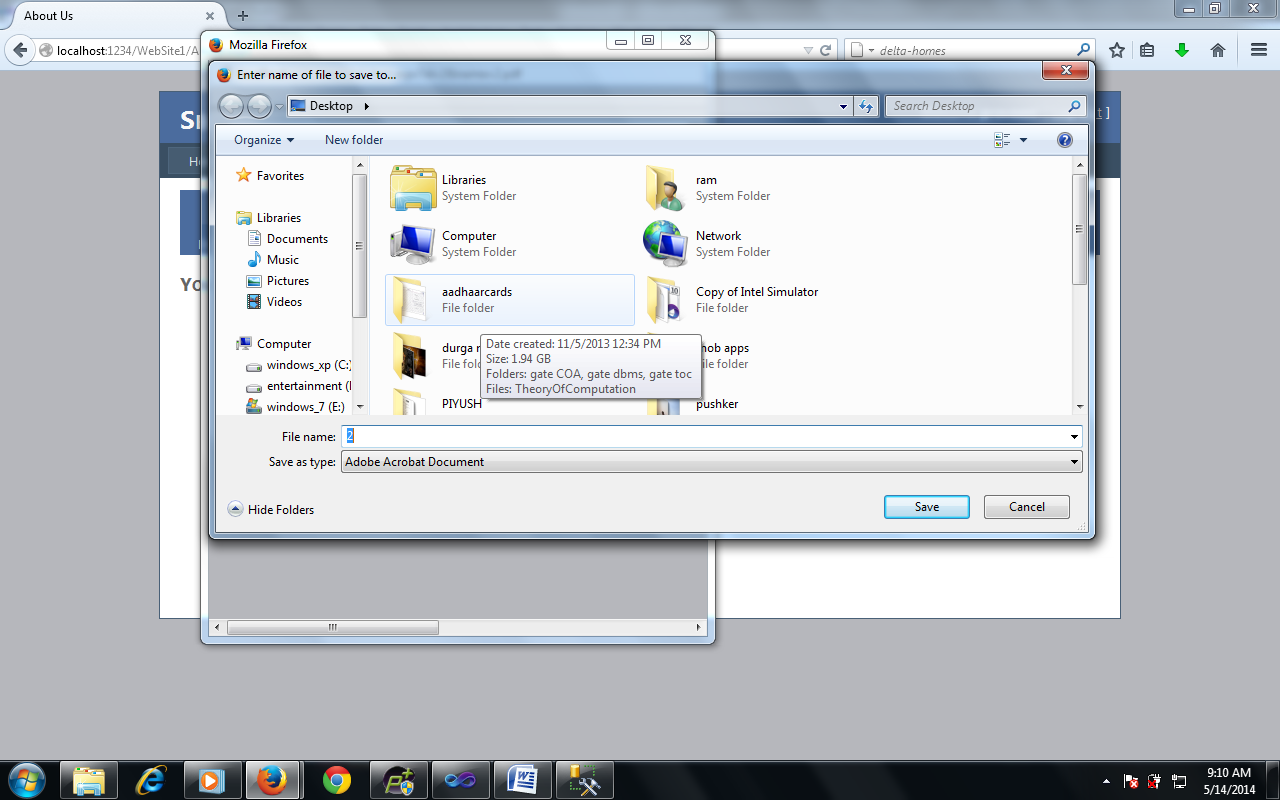


Fig. 4.13 Download options after verification

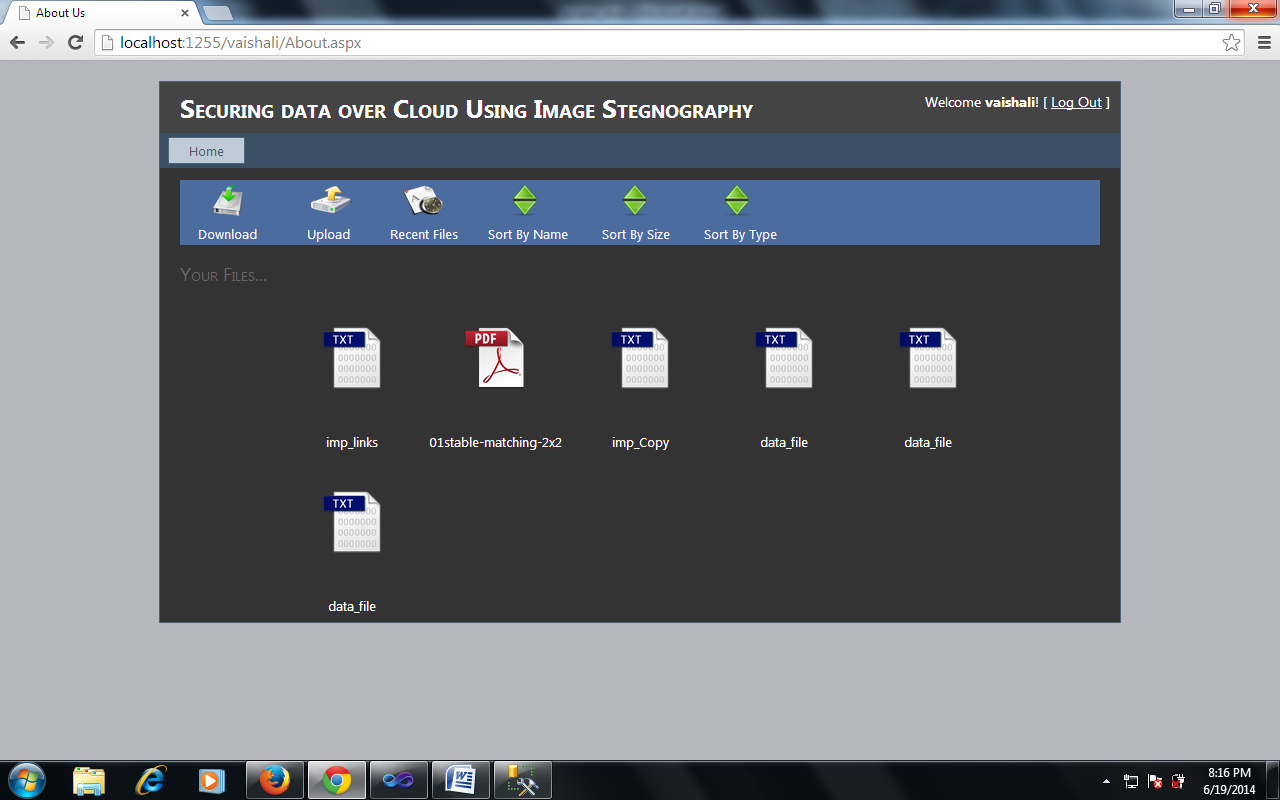


Fig. 4.14 File options in the window

Client may download the file to the desired location of his/her system (Fig 4.14). Client may sort the files by time of uploading of file (Fig 4.15).

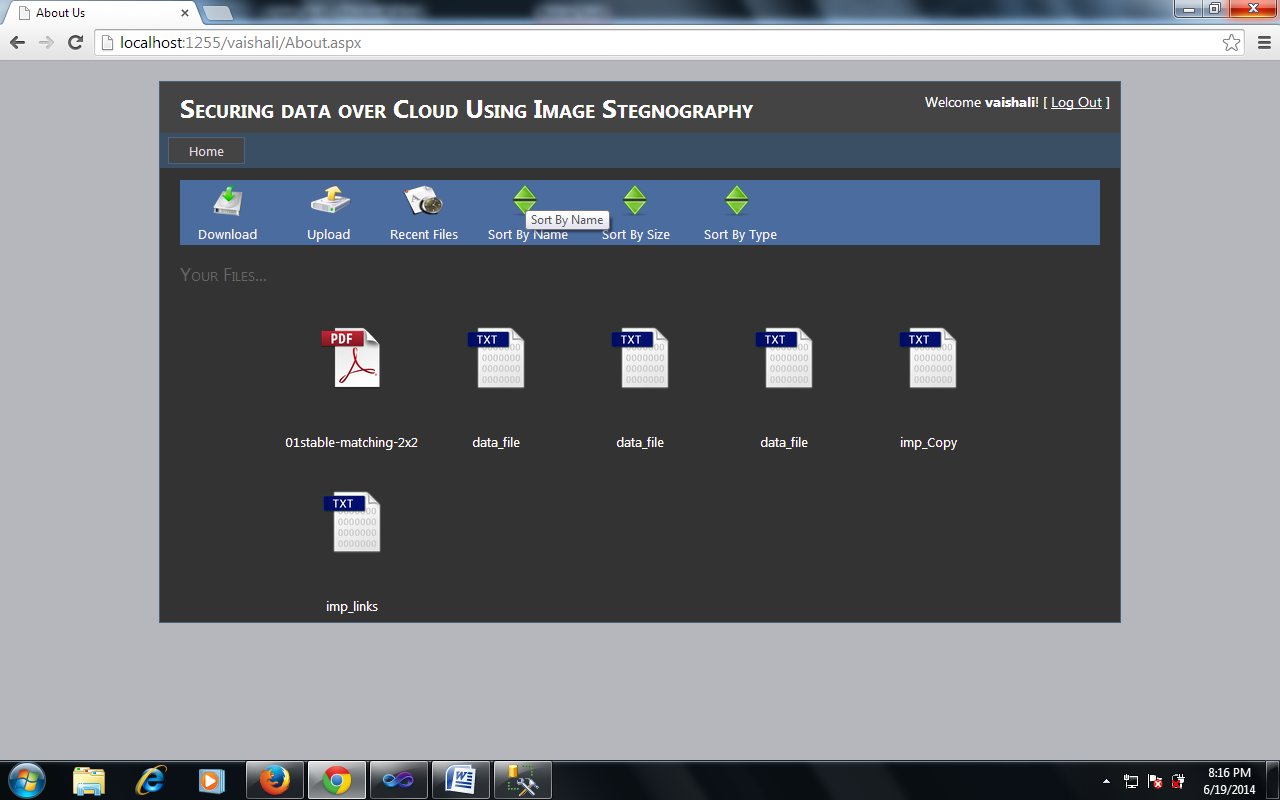


Fig. 4.15 Name wise sorting of the files

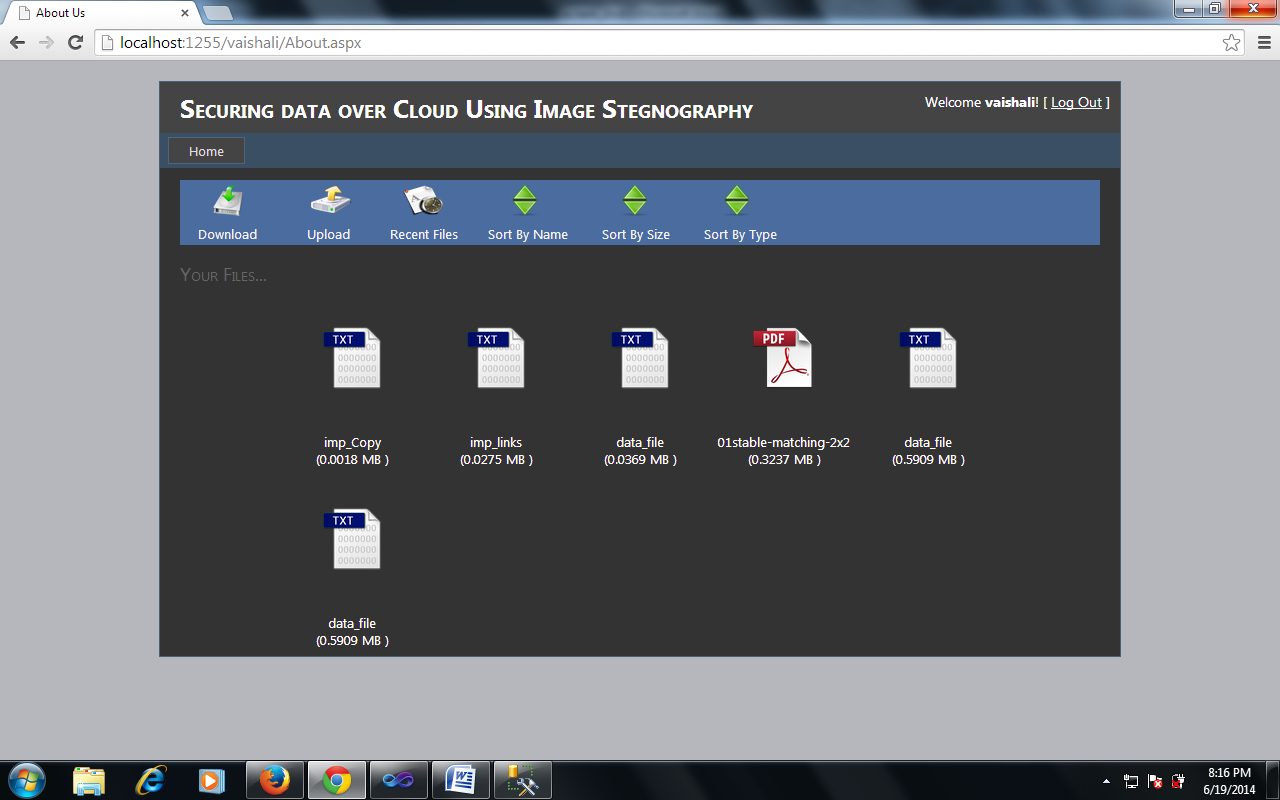


Fig. 4.16 Size wise sorting of the files

Client may sort the files by name of file (Fig. 4.16). Client may sort the files by size of file (Fig.4.9).

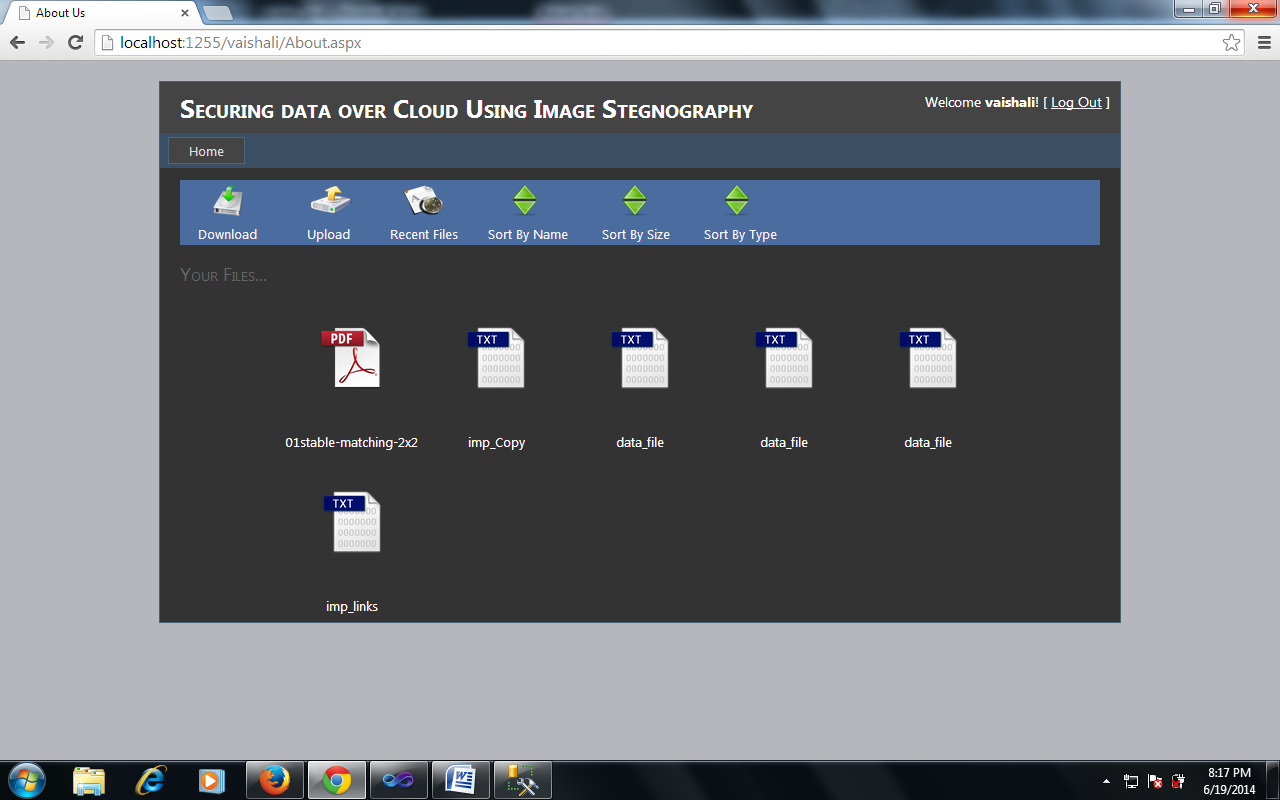


Fig. 4.17 Type wise sorting of the files

Client may sort the files by type of the file (4.17)

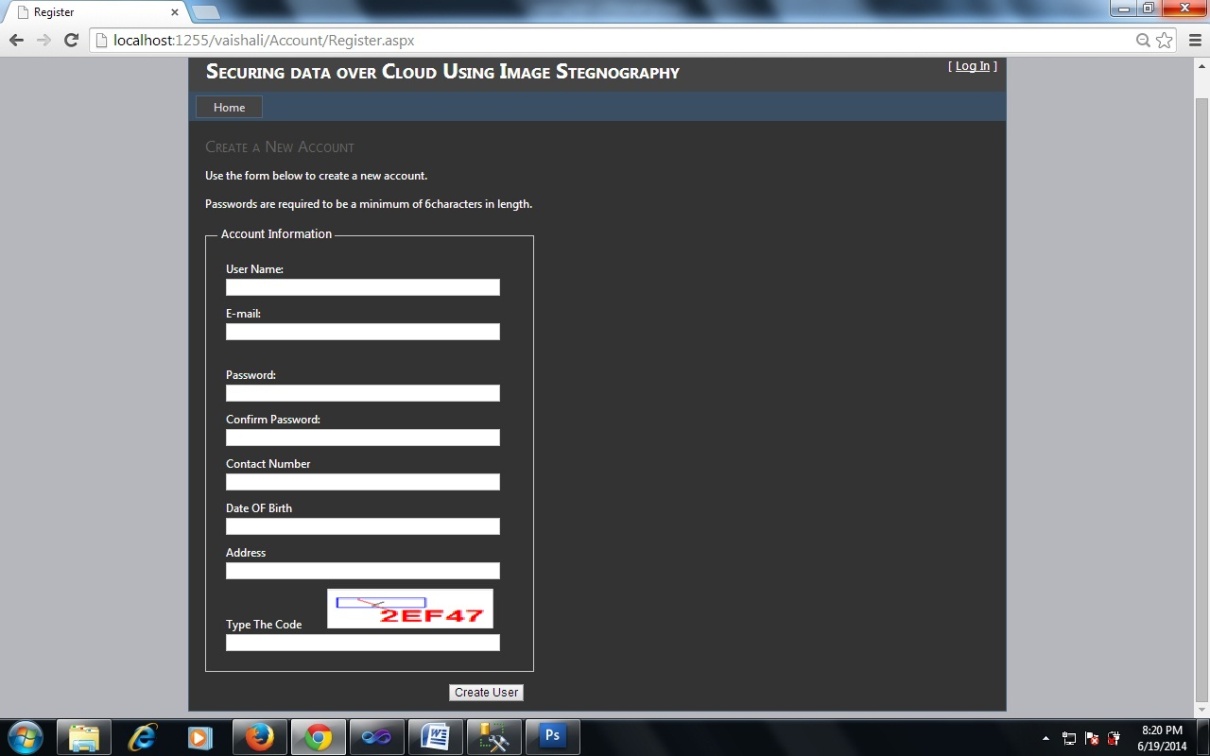


Fig. 4.18 Regestration window to be authorized client

To be an authorized client, the client may create his/her account if he/she is not registered client using registration screen (Fig. 4.18). The client has to be a registered client for using the system. There is a multilevel security system as at the first level the client should enter the authorized client ID and pass word to login to system and in case of pass word theft or unauthorized use condition the false client can’t download any file as for downloading the file each time a onetime password is to be entered which will be sent only on the registered mail ID.

**4.4 Summary**

The data encrypted in the cover is available as the LSB’s of the stegano image pixels. The retrieval process described in this Chapter is very easy and straight forward simply the LSB’s of the pixels of the stegano image are encrypted so the data bits are recovered and reformed in the form of the data. If the data was hidden by parts in the form of more than one stegano image then the data bits are recovered from different images and the parts are combined to form the data file again back. In this Chapter first the procedure algorithm for recovering the data is explained. Then the automated system designed for the data recovery was demonstrated.

**Chapter - 5**

**Analysis of image steganographic method for data security on clouds**

**5.1 Introduction**

To analyze the performance of the designed system for securing the data in the cover image and to insure the efficiency of the hiding procedure a correlation method is utilized and the results basis on the correlation method are summarized for a particular example. The cover image is distorted due to hiding of message are hiding the data by changing the pixel values of the image, this distortion or change in the cover image can be recognized and used for analyzing the system performance.

In this Chapter an example is taken in which a text data file is hidden in an image. The text file is having a larger size then the capacity of the cover image so multiple copies of the cover image are created to hide the complete data. The difference images and stegano images at each and every step are analyzed. The data matrices of each image are produced in order to justify the process results and the graphs are drawn for each image RGB components. The correlation between each stegano image is calculated.

**5.2 The taken example**

The cover image taken is shown in fig. 5.1. It is the enlarged copy of the actual cover image used. For the sake of simplicity if the system and to make the data file as small as to illustrated the actual cover image is taken small.

****

Fig. 5.1 The cover image

The RGB pixel matrix of the image is as follows

**R-Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 117 | 138 | 130 | 136 | 143 | 131 | 138 | 146 | 146 | 151 | 156 | 163 | 168 | 174 | 180 | 184 |
| 162 | 153 | 142 | 149 | 151 | 160 | 173 | 151 | 165 | 170 | 180 | 186 | 192 | 196 | 199 | 201 |
| 168 | 106 | 104 | 171 | 190 | 181 | 199 | 194 | 197 | 201 | 208 | 216 | 220 | 225 | 229 | 231 |
| 150 | 154 | 143 | 136 | 132 | 161 | 214 | 225 | 229 | 230 | 234 | 235 | 236 | 237 | 240 | 244 |
| 168 | 221 | 187 | 107 | 57 | 61 | 147 | 239 | 240 | 244 | 250 | 254 | 252 | 249 | 247 | 244 |
| 218 | 244 | 197 | 141 | 85 | 17 | 43 | 144 | 157 | 168 | 184 | 198 | 205 | 204 | 201 | 198 |
| 217 | 245 | 190 | 121 | 90 | 60 | 41 | 29 | 48 | 54 | 67 | 80 | 89 | 95 | 100 | 102 |
| 152 | 144 | 87 | 40 | 35 | 56 | 81 | 71 | 28 | 24 | 21 | 18 | 20 | 27 | 37 | 45 |
| 36 | 44 | 52 | 42 | 26 | 32 | 45 | 43 | 19 | 19 | 16 | 12 | 10 | 12 | 15 | 15 |
| 28 | 32 | 37 | 25 | 10 | 13 | 24 | 19 | 19 | 18 | 16 | 12 | 11 | 10 | 13 | 12 |
| 15 | 16 | 21 | 15 | 7 | 16 | 27 | 22 | 22 | 18 | 14 | 10 | 9 | 7 | 8 | 8 |
| 10 | 10 | 14 | 11 | 13 | 30 | 44 | 36 | 20 | 13 | 10 | 8 | 6 | 3 | 1 | 2 |

**G-Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 133 | 156 | 150 | 159 | 170 | 163 | 172 | 183 | 183 | 187 | 191 | 195 | 196 | 200 | 204 | 206 |
| 172 | 166 | 156 | 167 | 173 | 186 | 201 | 181 | 196 | 201 | 208 | 213 | 215 | 216 | 218 | 219 |
| 170 | 110 | 110 | 181 | 203 | 198 | 217 | 214 | 217 | 221 | 227 | 231 | 234 | 236 | 238 | 241 |
| 141 | 147 | 137 | 134 | 133 | 165 | 220 | 235 | 239 | 239 | 241 | 240 | 239 | 238 | 240 | 242 |
| 146 | 200 | 168 | 92 | 44 | 52 | 141 | 234 | 237 | 240 | 246 | 247 | 243 | 239 | 234 | 231 |
| 186 | 214 | 169 | 117 | 63 | 0 | 27 | 132 | 145 | 156 | 171 | 183 | 187 | 186 | 181 | 178 |
| 182 | 211 | 157 | 92 | 64 | 38 | 19 | 11 | 30 | 38 | 51 | 61 | 69 | 75 | 79 | 81 |
| 116 | 110 | 55 | 11 | 9 | 34 | 60 | 53 | 12 | 9 | 5 | 2 | 4 | 11 | 21 | 29 |
| 7 | 17 | 27 | 18 | 5 | 14 | 29 | 31 | 8 | 9 | 8 | 4 | 4 | 6 | 9 | 9 |
| 8 | 14 | 19 | 9 | 0 | 3 | 15 | 13 | 17 | 17 | 15 | 13 | 12 | 14 | 14 | 16 |
| 5 | 8 | 13 | 7 | 2 | 15 | 28 | 24 | 24 | 23 | 21 | 20 | 19 | 19 | 18 | 20 |
| 6 | 6 | 10 | 10 | 12 | 33 | 47 | 41 | 28 | 23 | 21 | 20 | 20 | 17 | 16 | 17 |

**B-Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 184 | 206 | 201 | 209 | 223 | 214 | 220 | 228 | 225 | 223 | 223 | 220 | 217 | 215 | 216 | 217 |
| 207 | 200 | 191 | 203 | 210 | 223 | 238 | 215 | 227 | 229 | 230 | 230 | 229 | 225 | 224 | 223 |
| 169 | 109 | 110 | 183 | 209 | 205 | 227 | 223 | 226 | 228 | 233 | 234 | 234 | 232 | 233 | 233 |
| 112 | 119 | 111 | 111 | 115 | 150 | 210 | 226 | 231 | 234 | 234 | 233 | 232 | 230 | 230 | 230 |
| 97 | 153 | 125 | 51 | 9 | 23 | 117 | 214 | 220 | 228 | 235 | 237 | 234 | 230 | 226 | 222 |
| 135 | 164 | 122 | 73 | 26 | 0 | 2 | 110 | 129 | 144 | 163 | 178 | 183 | 184 | 180 | 177 |
| 142 | 173 | 122 | 62 | 39 | 17 | 5 | 1 | 26 | 38 | 54 | 67 | 78 | 84 | 88 | 90 |
| 92 | 85 | 34 | 0 | 0 | 23 | 55 | 51 | 13 | 14 | 15 | 13 | 17 | 24 | 34 | 42 |
| 0 | 8 | 20 | 14 | 2 | 14 | 32 | 35 | 16 | 18 | 19 | 15 | 16 | 18 | 19 | 19 |
| 7 | 12 | 19 | 10 | 0 | 4 | 18 | 17 | 22 | 22 | 20 | 17 | 16 | 17 | 18 | 17 |
| 3 | 5 | 11 | 5 | 0 | 11 | 23 | 19 | 19 | 17 | 14 | 11 | 10 | 7 | 7 | 8 |
| 3 | 3 | 7 | 6 | 7 | 26 | 40 | 34 | 17 | 12 | 7 | 6 | 3 | 0 | 0 | 0 |

The text file size used is having a size of 1920B. The cover image taken is having 3x12x16= 576 pixels. To hide one byte of data 2 pixels are required so the given image can handle 576 / 2 = 288 bytes of data. According to the capacity of the cover image seven replicas of cover image are needed to encrypt the complete date. Six Stegano images will be produced to hide the complete data.

The Stegano image 1 produced is shown in Fig. 5.2 (a). It is the enlarged copy of the actual Stegano image produced. The difference image is also created by taking the difference of the cover image and the Stegano 1 image. The difference image is almost black showing no significant difference in the pixel values of the cover image. This makes the hiding of the data at a good level.

** **

* 1. (b)

Fig. 5.2 (a) Stegano Image 1 (b) Difference from cover image

The RGB pixel matrix of the Stegano image1 is as follows. Here R, G,B are matrices for Red, Green, Blue component of the Cover Image respectively.

**R1-Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 121 | 140 | 137 | 134 | 141 | 137 | 136 | 146 | 153 | 154 | 153 | 173 | 169 | 173 | 184 | 190 |
| 169 | 157 | 137 | 155 | 153 | 161 | 173 | 148 | 169 | 173 | 185 | 185 | 201 | 193 | 200 | 207 |
| 169 | 103 | 105 | 173 | 191 | 182 | 200 | 204 | 204 | 193 | 216 | 217 | 221 | 239 | 232 | 239 |
| 153 | 155 | 136 | 129 | 136 | 173 | 217 | 225 | 233 | 226 | 233 | 227 | 233 | 237 | 248 | 252 |
| 169 | 223 | 185 | 105 | 57 | 48 | 152 | 226 | 248 | 240 | 255 | 248 | 248 | 240 | 253 | 241 |
| 216 | 249 | 201 | 140 | 89 | 18 | 41 | 156 | 153 | 171 | 185 | 203 | 201 | 204 | 205 | 205 |
| 217 | 241 | 184 | 126 | 88 | 59 | 41 | 18 | 57 | 62 | 73 | 88 | 93 | 95 | 105 | 99 |
| 153 | 156 | 89 | 38 | 45 | 57 | 88 | 66 | 25 | 26 | 25 | 29 | 25 | 29 | 40 | 46 |
| 41 | 45 | 57 | 43 | 25 | 33 | 45 | 36 | 25 | 29 | 25 | 9 | 9 | 1 | 8 | 15 |
| 25 | 39 | 41 | 29 | 15 | 6 | 24 | 28 | 28 | 17 | 24 | 9 | 13 | 15 | 8 | 15 |
| 9 | 27 | 24 | 1 | 8 | 29 | 25 | 17 | 25 | 18 | 9 | 3 | 9 | 13 | 8 | 12 |
| 9 | 15 | 9 | 9 | 9 | 16 | 40 | 34 | 24 | 0 | 15 | 8 | 8 | 0 | 13 | 1 |

**G1 –Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 136 | 152 | 144 | 157 | 161 | 168 | 169 | 185 | 188 | 185 | 178 | 201 | 204 | 201 | 203 | 201 |
| 171 | 169 | 156 | 173 | 173 | 185 | 193 | 184 | 206 | 200 | 219 | 217 | 210 | 217 | 222 | 217 |
| 168 | 109 | 111 | 185 | 195 | 201 | 220 | 217 | 214 | 221 | 233 | 232 | 226 | 233 | 234 | 249 |
| 141 | 153 | 141 | 136 | 142 | 169 | 221 | 233 | 235 | 233 | 241 | 253 | 228 | 233 | 253 | 249 |
| 153 | 201 | 161 | 88 | 47 | 57 | 135 | 233 | 237 | 255 | 246 | 248 | 252 | 236 | 225 | 232 |
| 185 | 221 | 175 | 120 | 63 | 9 | 27 | 136 | 145 | 152 | 173 | 185 | 177 | 185 | 178 | 185 |
| 179 | 217 | 157 | 88 | 76 | 41 | 31 | 9 | 25 | 41 | 48 | 56 | 66 | 72 | 64 | 95 |
| 120 | 104 | 48 | 13 | 1 | 40 | 57 | 57 | 12 | 9 | 2 | 9 | 12 | 9 | 27 | 25 |
| 11 | 25 | 28 | 29 | 13 | 9 | 17 | 24 | 14 | 8 | 11 | 9 | 2 | 9 | 14 | 9 |
| 8 | 13 | 31 | 9 | 3 | 9 | 12 | 9 | 22 | 29 | 9 | 8 | 2 | 9 | 10 | 25 |
| 13 | 9 | 13 | 8 | 14 | 9 | 29 | 25 | 27 | 25 | 17 | 29 | 20 | 25 | 29 | 25 |
| 9 | 9 | 1 | 8 | 15 | 41 | 39 | 41 | 29 | 31 | 22 | 24 | 28 | 28 | 17 | 24 |

**B1-Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 184 | 204 | 204 | 209 | 216 | 217 | 221 | 239 | 232 | 223 | 217 | 219 | 216 | 209 | 216 | 221 |
| 201 | 193 | 185 | 194 | 217 | 211 | 233 | 221 | 232 | 236 | 233 | 239 | 233 | 233 | 233 | 208 |
| 168 | 98 | 104 | 176 | 223 | 200 | 232 | 208 | 237 | 225 | 232 | 233 | 233 | 236 | 233 | 226 |
| 121 | 124 | 105 | 107 | 121 | 155 | 217 | 236 | 237 | 237 | 233 | 225 | 232 | 238 | 232 | 235 |
| 105 | 146 | 121 | 62 | 9 | 24 | 125 | 223 | 217 | 227 | 233 | 236 | 233 | 230 | 237 | 217 |
| 136 | 162 | 121 | 74 | 25 | 13 | 9 | 109 | 136 | 158 | 169 | 189 | 185 | 187 | 185 | 177 |
| 141 | 164 | 121 | 61 | 41 | 25 | 9 | 1 | 24 | 47 | 57 | 71 | 73 | 93 | 95 | 86 |
| 88 | 92 | 44 | 1 | 8 | 25 | 61 | 63 | 8 | 15 | 9 | 11 | 24 | 17 | 40 | 45 |
| 9 | 1 | 25 | 2 | 9 | 3 | 41 | 45 | 24 | 28 | 25 | 15 | 25 | 25 | 25 | 16 |
| 8 | 2 | 24 | 0 | 15 | 8 | 24 | 16 | 29 | 17 | 24 | 25 | 25 | 28 | 25 | 18 |
| 9 | 12 | 9 | 11 | 9 | 11 | 25 | 28 | 29 | 29 | 9 | 1 | 8 | 14 | 8 | 11 |
| 9 | 2 | 9 | 14 | 9 | 24 | 45 | 47 | 25 | 3 | 9 | 12 | 9 | 6 | 13 | 9 |

Similarly six different Stegano images and respective difference images are obtained and the respective matrices are obtained for the comparison purpose rest of the Stegano images and respective matrices are added in appendix 1. Each stegano image seems to be an perfect copy of the cover image having almost no difference, which can be justified by all the difference images are almost black. Here R1, G1,B1 are matrices for Red, Green, Blue component of the Stegano-Image(1) respectively.

**5.3 Analysis of the designed system**

Here to estimate the similarity of the cover image with the stegano image the correlation of the RGB component is taken with respective components of each stegano image and the results of the correlation are summarized below:

Correlation Coefficient shows amount of similarity between two matrices. Here the calculated value Correlation Coefficients of stegano-images with respect to Cover Image are:

TABLE 5.1 CORRELATION COEFFICIENTS OF ALL IMAGE COMPONENTS

|  |
| --- |
| corr2(R, R1) = 0.9979 |
| corr2(G, G1) = 0.9979 |
| corr2(B, B1) = 0.9975 |
| corr2(R, R2) = 0.9975 |
| corr2(G, G2) = 0.9978 |
| corr2(B, B2) = 0.9978 |
| corr2(R, R3) = 0.9979 |
| corr2(G, G3) = 0.9978 |
| corr2(B, B3) = 0.9980 |
| corr2(R, R4) = 0.9978 |
| corr2(G, G4) = 0.9981 |
| corr2(B, B4) = 0.9978 |
| corr2(R, R5) = 0.9978 |
| corr2(G, G5) = 0.9979 |
| corr2(B, B5) = 0.9978 |
| corr2(R, R6) = 0.9978 |
| corr2(G, G6) = 0.9977 |
| corr2(B, B6) = 0.9978 |
| corr2(R, R7) = 0.9980 |
| corr2(G, G7) = 0.9984 |
| corr2(B, B7) = 0.9936 |

Here R, G, B are matrices for Red, Green, Blue component of the Cover Image respectively, R1, G1,B1 are matrices for Red, Green, Blue component of the Stegno-Image(1) respectively. Similarly R2, G2, B2 are matrices for Red, Green, Blue component of the Stegno-Image(2) respectively and R3, G3,B3 are matrices for Red, Green, Blue component of the Stegno-Image(3) respectively. Here R4, G4, B4 are matrices for Red, Green, Blue component of the Stegno-Image(4) respectively and R5-7, G5-7,B5-7 are matrices for Red, Green, Blue component of the Stegno-Image(5-7) respectively . the value of correlation coefficients show that there is an perfect match between the stegano image RGB components with the respective components o the cover image showing the good hiding of data in cover image and insuring the non detectiblity without prior information.

To extend the possibility of mathematical verification of security of the method the entropy of the cover image and the different stegano images are calculated and the percent difference between the entropy of the cover image and stegano images are calculated

TABLE 5.2 ENTROPY OF DIFFERENT IMAGES

|  |  |  |
| --- | --- | --- |
|  | | Percentage of variation in Entropy of Cover-Image and Stegno-images |
| Entropy of Cover-Image | 7.2350 |  |
| Entropy of Stegno-Image(1) | 6.4703 | 10.5695 |
| Entropy of Stegno-Image(2) | 6.4683 | 10.5971 |
| Entropy of Stegno-Image(3) | 6.5008 | 10.1479 |
| Entropy of Stegno-Image(4) | 6.4690 | 10.5874 |
| Entropy of Stegno-Image(5) | 6.5188 | 9.8991 |
| Entropy of Stegno-Image(6) | 6.4957 | 10.2184 |
| Entropy of Stegno-Image(7) | 6.7909 | 4.9549 |

From the above table of the entropies it very clear that the difference between entropies the cover image and the respective stegano images is very less also the Percentage of variation in Entropy of Cover-Image and Stegano –images is coming very less that is a very less of fraction showing a good amount matching of the images making the difference delectability very less.

**5.4 Graphical analysis of the images**

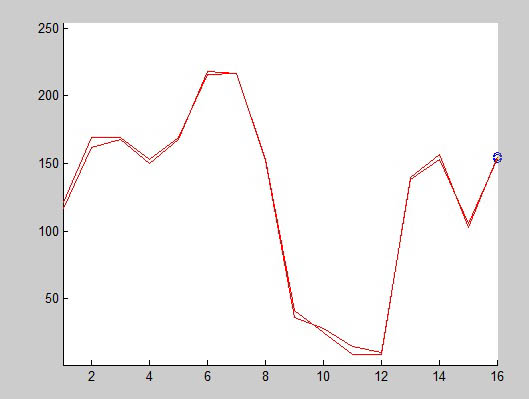
****

Fig. 5.3 Plotting pixel values of R1 v/s R

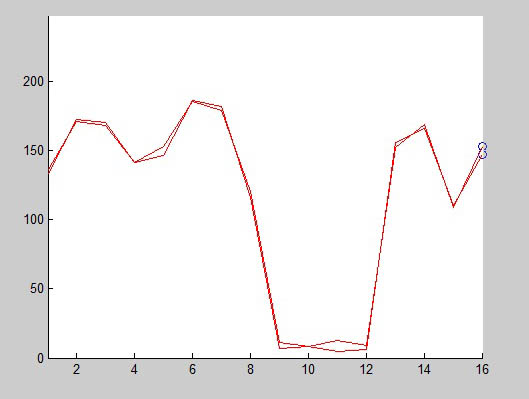


Fig. 5.4 Plotting pixel values of G1 v/s G

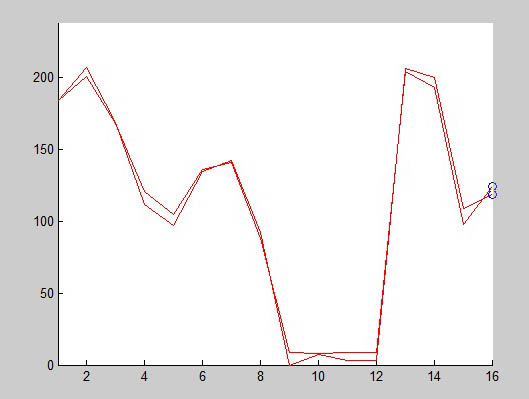
****

Fig. 5.5 Plotting of R1 v/s R

From the above plots of RGB components of cover image (Fig. 5.3) and plots of R1and R which are components of stegano image (1) (Fig. 5.4) and all the graphs of respective RGB components of stegano images (1) shown in Fig.5.5- Fig.5.6 it can be observed there is no significant difference between these images.

**5.5 Summary**

In this Chapter an example is taken in which a text data file is hidden in an image. The text file is having a larger size then the capacity of the cover image so multiple copies of the cover image are created to hide the complete data. The difference images and stegano images at each and every step are analyzed. The data matrices of each image are produced in order to justify the process results and the graphs are drawn for each image RGB components. The correlation between each stegano image is calculated along with the entropy of each image as well as difference of all the entropies of all the stegano images with respect to cover image.

**CONCLUSIONS**

In recent years, the extended growth in cloud computing industry continues to drive the requirements for more and more applications on the clouds.The cloud computing services can be utilized without any investment on bulky IT set up and a large demand to purchase the software. A large pool of the services provided by the cloud computing firms without any investment on set up or IT infrastructure no need to hair IT professionals, no need to train the IT resources it is just pay and get the service at a very reasonable rate. The services provided by the clouds are so cheap and straight forward but at the same time posses the threat of data insecurity. Because in the cloud computing the data is available in an remote server, which clients simultaneously is used by several clients at a time in case of theft of the data or any expose of data to any false client knowingly or unknowingly may arise the questions on the data security.

An image steganographic process using the LSB encryption technique is used to secure the data over the clouds in such scheme the data to be transferred is hidden in the LSBs of pixels of the cover image so that he data or information to be sent is in the form of LSBs of the cover image which cannot be detected without any prior information of availability of data in that image. The conclusions of the work carried out are summarized below

* An automatic system is designed to upload to data on the clouds the data as soon as uploaded on the cloud are encrypted in the LSB,s of a cover image and the data is no longer available as the uploaded format but in the encrypted format now.
* The data containing capacity of the cover image is finite so if the data exceeds the capacity of the cover image a multiple copies of the cover image are created the data file is broken in the comestible size pieces and assembled back at the time of recovery.
* An stegano image is created with respect to each cover image embedded with data and the correlation of that stegano image comes out to be 1 in each case, proving that no one can detect the difference in stegano image and cover image.
* The entropy of each stegano image is having a very less difference with the entropy of the cover image again justifying the a good hiding possibility of data in images.

**Future Scope**

An image steganographic process using the LSB encryption technique used to secure the data over the clouds in such scheme the data to be transferred is hidden in the LSB of pixels of the cover image, in this scheme the ratio of hiding is 4 bit per eight bits so it can be improved so that the data carrying capacity of the cover can be improved. Some sort of data compression may be used to compress the data before hiding it into the cover image so that the minimum copies of the stegano image should be required to embed maximum amount of data.

**REFERENCES**

1. ["The NIST Definition of Cloud Computing"](http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf). National Institute of Standards and Technology. Retrieved 24 July 2011.
2. ["What is Cloud Computing?"](http://aws.amazon.com/what-is-cloud-computing/). *Amazon Web Services*. 2013-03-19. Retrieved 2013-03-20.
3. ["Baburajan, Rajani, "The Rising Cloud Storage Market Opportunity Strengthens Vendors," infoTECH, August 24, 2011"](http://it.tmcnet.com/channels/cloud-storage/articles/211183-rising-cloud-storage-market-opportunity-strengthens-vendors.htm). It.tmcnet.com. 2011-08-24. Retrieved 2011-12-02.
4. Oestreich, Ken, (2010-11-15). ["Converged Infrastructure"](http://www.thectoforum.com/content/converged-infrastructure-0). *CTO Forum*. Thectoforum.com. Retrieved 2011-12-02.
5. J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, \All your clouds are belong to us: security analysis of cloud management interfaces," in Proceedings of the 3rd ACM workshop on Cloud computing security workshop, CCSW '11, (New York, NY, USA),

pp. 3{14, ACM, 2011.

1. T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, \Hey, you, get o\_ of my cloud: exploring

information leakage in third-party compute clouds," in Proceedings of the 16th ACM conference on Computer and communications security, CCS '09, (New York, NY, USA), pp. 199{212, ACM, 2009.

1. D. Molnar and S. Schechter, \Self hosting vs . cloud hosting : Accounting for the security impact

of hosting in the cloud," in Workshop of the economics of cloud security, pp. 1{18, 2010.

1. Y. Chen, V. Paxson, and R. Katz, \The hybrex model for con\_dentiality and privacy in cloud

computing," Technical Report UCB/EECS-2010-5, EECS Department, University of California,

Berkeley, January 2010.

1. N. Santos, K. P. Gummadi, and R. Rodrigues, \Towards trusted cloud computing," in Proceedings of the 2009 conference on Hot topics in cloud computing, HotCloud'09, (Berkeley, CA, USA), USENIX Association, 2009.
2. D. Kuhlmann, R. Landfermann, H. V. Ramasamy, M. Schunter, G. Ramunno, and D. Vernizzi,

\An open trusted computing architecture { secure virtual machines enabling user-de\_ned policy

enforcement," Work, pp. 1{14, 2006.

1. N. Pohlmann and H. Reimer, \Trusted computing - eine einf~A 14 hrung," in Trusted Computing

(N. Pohlmann and H. Reimer, eds.), pp. 3{12, Vieweg+Teubner, 2008. 10.1007/978-3-8348-9452-6 1.

1. M. Nauman, S. Khan, X. Zhang, and J.-P. Seifert, \Beyond kernel-level integrity measurement: Enabling remote attestation for the android platform," in Trust and Trustworthy Computing (A. Acquisti, S. Smith, and A.-R. Sadeghi, eds.), vol. 6101 of Lecture Notes in Computer Science, pp. 1{15, Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-13869-01.
2. I. Bente, G. Dreo, B. Hellmann, S. Heuser, J. Vieweg, J. von Helden, and J. Westhuis, \Towards permission-based attestation for the android platform," in Trust and Trustworthy Computing (J. Mc- Cune, B. Balache\_, A. Perrig, A.-R. Sadeghi, A. Sasse, and Y. Beres, eds.), vol. 6740 of Lecture Notes in Computer Science, pp. 108{115, Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642- 21599-58.
3. R. Neisse, D. Holling, and A. Pretschner, \Implementing trust in cloud infrastructures," in Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on, pp. 524{533, may 2011.
4. B. Parno, \Bootstrapping trust in a "trusted" platform," in Proceedings of the 3rd conference on Hot topics in security, (Berkeley, CA, USA), pp. 9:1{9:6, USENIX Association, 2008.
5. V. Scarlata, C. Rozas, M. Wiseman, D. Grawrock, and C. Vishik, \Tpm virtualization: Building a general framework," in Trusted Computing (N. Pohlmann and H. Reimer, eds.), pp. 43{56, Vieweg+Teubner, 2008. 10.1007/978-3-8348-9452-64.
6. A.-R. Sadeghi, C. St~A 14 ble, and M. Winandy, \Property-based tpm virtualization," in Information Security (T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, eds.), vol. 5222 of Lecture Notes in Computer Science, pp. 1{16, Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-85886-71.
7. B. Danev, R. J. Masti, G. O. Karame, and S. Capkun, \Enabling secure vm-vtpm migration in private clouds," in Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC '11, (New York, NY, USA), pp. 187{196, ACM, 2011.
8. F. Rocha and M. Correia, \Lucy in the sky without diamonds: Stealing con\_dential data in the cloud," in Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, DSNW '11, (Washington, DC, USA), pp. 129{134, IEEE Computer Society, 2011.
9. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, \A break in the clouds: towards a cloud definition," SIGCOMM Comput. Commun. Rev., vol. 39, pp. 50{55, Dec. 2008.
10. 1. Tripathi, A.; Mishra, A. Cloud computing security considerations. In Proceedings of the 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xi’an, China, 14–16 September 2011; pp. 1–5.
11. Wang, J.-J.; Mu, S. Security issues and countermeasures in cloud computing. In Proceedings of the 2011 IEEE International Conference on Grey Systems and Intelligent Services (GSIS), Nanjing, China, 15–18 September 2011; pp. 843–846.
12. Final Version of NIST Cloud Computing Definition Published. Available online:

http://www.nist.gov/itl/csd/cloud-102511.cfm (accessed on 25 August 2013).

1. Lv, H.; Hu, Y. Analysis and research about cloud computing security protect policy. In

Proceedings of the 2011 International Conference on Intelligence Science and Information Engineering (ISIE), Wuhan, China, 20–21 August 2011; pp. 214–216.

1. Mell, P and Grance, T. *The NIST Definition of Cloud Computing*, NIST, USA. available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, USA, 2009.
2. Jain, P.; Rane, D.; Patidar, S. A survey and analysis of cloud model-based security for computing secure cloud bursting and aggregation in renal environment. In Proceedings of the 2011 World Congress on Information and Communication Technologies (WICT), Mumbai, India, 11– 14 December 2011; pp. 456–461.
3. Gowrigolla, B.; Sivaji, S.; Masillamani, M.R. Design and auditing of cloud computing security. In Proceedings of the 2010 5th International Conference on Information and Automation for Sustainability (ICIAFs), Colombo, Sri Lanka, 17–19 December 2010; pp. 292–297.
4. Houmansadr, A.; Zonouz, S.A.; Berthier, R. A cloud-based intrusion detection and response system for mobile phones. In Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), Hong Kong, China, 27–30 June 2011; pp. 31–32.
5. M. Taifi, J. Y. Shi, A. Khreishah, “SpotMPI: A Framework for Auction-based HPC Computing Using Amazon Spot Instances”, in Proc. of the International Symposium on Advances of Distributed Computing and Networking (ADCN), 2011.
6. Sabahi, F. Virtualization-level security in cloud computing. In Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN), Xi’an, China, 27– 29 May 2011; pp. 250–254.
7. Wang, C.; Wang, Q.; Ren, K.; Lou, W. Towards secure and dependable storage services in cloud computing. *IEEE Trans. Serv. Comput.* **2012**, *5*, 220–232.
8. Lingfeng, C.; Hoang, D.B. Towards scalable, fine-grained, intrusion-tolerant data protection models for healthcare cloud. In Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Changsha, China,

16–18 November 2011; pp. 126–133.

1. Morin, J.; Aubert, J.; Gateau, B. Towards cloud computing SLA risk management: Issues and challenges. In Proceedings of the 2012 45th Hawaii International Conference on System Science (HICSS), Maui, HI, USA, 4–7 January 2012; pp. 5509–5514.
2. Khalil, I.M. ELMO: Energy aware local monitoring in sensor networks. *IEEE Trans. Dependable Secur. Comput.* **2011**, *8*, 523–536.
3. Khalil, I.; Bagchi, S. MISPAR: Mitigating stealthy packet dropping in locally-monitored multihop wireless Ad Hoc networks. In Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks (SecureComm ’08), Istanbul, Turkey, 22–25 September 2008; ACM: New York, NY, USA, 2008; article 28, pp. 1–10.
4. Khalil, I. MCC: Mitigating colluding collision attacks in wireless sensor networks. In Proceedings of the 2010 IEEE Global Telecommunications Conference (GLOBECOM 2010), Miami, FL, USA, 6–10 December 2010; pp. 1–5.
5. M. Hayajneh, I. Khalil and Y. Gadallah, “An OFDMA-based MAC protocol for under water acoustic wireless sensor network,” Proceedings of the 2009 ACM International Conference on Wireless Communications and Mobile Computing (IWCMC’09), Leipzig, Germany, June 21 – 24 2009, pp. 810-814.
6. Khalil, I.; Hayajneh, M.; Awad, M. SVNM: Secure verification of neighborhood membership in static multi-hop wireless networks. In Proceedings of the IEEE Symposium on Computers and Communications, 2009, ISCC 2009, Sousse, 5–8 July 2009; pp. 368–373.
7. Sengupta, S.; Kaulgud, V.; Sharma, V.S. Cloud computing security—Trends and research directions. In Proceedings of the 2011 IEEE World Congress on Services (SERVICES), Washington, DC, USA, 4–9 July 2011; pp. 524–531.
8. Chow, R.; Golle, P.; Jakobsson, M.; Shi, E.; Staddon, J.; Masuoka, R.; Molina, J. Controlling data in the cloud: Outsourcing computation without outsourcing control. In Proceedings of the 2009ACM Workshop on Cloud Computing Security, Chicago, IL, USA, 13 November 2009; ACM Press: New York, NY, USA, 2009; pp. 85–90.
9. Samarati, P.; di Vimercati, S.D.C. Data protection in outsourcing scenarios: Issues and directions. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS ’10), Chicago, IL, USA, 4–8 October 2010; ACM: New York, NY, USA, 2010; pp. 1–14.
10. Popovic, O.; Jovanovic, Z.; Jovanovic, N.; Popovic, R. A comparison and security analysis of the cloud computing software platforms. In Proceedings of the 2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), Nis, Serbia, 5–8 October 2011; Volume 2, pp. 632–634.
11. Gul, I.; ur Rehman, A.; Islam, M.H. Cloud computing security auditing. In Proceedings of the 2011 The 2nd International Conference on Next Generation Information Technology (ICNIT), Gyeongju, Korea, 21–23 June 2011; pp. 143–148.
12. Kandukuri, B.R.; Paturi, V.R.; Rakshit, A. Cloud security issues. In Proceedings of the IEEE International Confer ence on Services Computing, 2009 (SCC ’09), Bangalore, India, 21–25 September 2009; pp. 517–520.
13. Chen, Z.; Yoon, J. IT auditing to assure a secure cloud computing. In Proceedings of the 2010 6th World Congress on Services (SERVICES-1), Miami, FL, USA, 5–10 July 2010; pp. 253–259.
14. Ryan, G.W.; Bernard, H.R. Data Management and Analysis Methods. Available online:

http://www.rand.org/pubs/external\_publications/EP20000033.html (accessed on 25 August 2013).

1. Holloway, I.; Todres, L. The status of method: Flexibility, consistency and coherence. *Qual. Res.* **2003**, *3*, 345–357.
2. Hashizume, K.; Rosado, D.G.; Fernández-Medina, E.; Fernandez, E.B. An analysis of security issues for cloud computing. *J. Internet Serv. Appl.* **2013**, *4*, 5.
3. Zissis, D.; Lekkas, D. Addressing cloud computing security issues. *Future Gener. Comput. Syst.* **2012**, *28*, 583–592.
4. Whaiduzzaman, M.; Sookhak, M.; Gani, A.; Buyya, R. A survey on vehicular cloud computing. *J. Netw. Comput. Appl.* **2013**, doi:10.1016/j.jnca.2013.08.004.
5. Moerland, T., “Steganography and Steganalysis”, *Leiden Institute of Advanced Computing Science*, www.liacs.nl/home/ tmoerl/privtech.pdf
6. Silman, J., “Steganography and Steganalysis: An Overview”, *SANS Institute*, 2001
7. Jamil, T., “Steganography: The art of hiding information is plain sight”, *IEEE Potentials*, 18:01, 1999
8. Wang, H & Wang, S, “Cyber warfare: Steganography vs. Steganalysis”, *Communications of the ACM*, 47:10, October 2004
9. Anderson, R.J. & Petitcolas, F.A.P., “On the limits of steganography”, *IEEE Journal of selected Areas in Communications*, May 1998
10. Marvel, L.M., Boncelet Jr., C.G. & Retter, C., “Spread Spectrum Steganography”, *IEEE Transactions on image processing*, *8:08*, 1999
11. Dunbar, B., “Steganographic techniques and their use in an Open-Systems environment”, *SANS Institute*, January 2002
12. Johnson, N.F. & Jajodia, S., “Exploring Steganography: Seeing the Unseen”, *Computer Journal*, February 1998
13. “Reference guide: Graphics Technical Options and Decisions”, <http://www.devx.com/projectcool/Article/19997>
14. Owens, M., “A discussion of covert channels and steganography”, *SANS Institute*, 2002
15. Moerland, T., “Steganography and Steganalysis”, *Leiden Institute of Advanced Computing Science*, www.liacs.nl/home/ tmoerl/privtech.pdf.
16. Dunbar, B., “Steganographic techniques and their use in an Open-Systems environment”, *SANS Institute*, January 2002
17. Silman, J., “Steganography and Steganalysis: An Overview”, *SANS Institute*, 2001
18. Lee, Y.K. & Chen, L.H., “High capacity image steganographic model”, *Visual Image Signal Processing*, 147:03, June 2000
19. Johnson, N.F. & Jajodia, S., “Steganalysis of Images Created Using Current Steganography Software”, *Proceedings of the 2nd Information Hiding Workshop*, April 1998
20. Venkatraman, S., Abraham, A. & Paprzycki, M., “Significance of Steganography on Data Security”, *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2004
21. Wang, H & Wang, S, “Cyber warfare: Steganography vs. Steganalysis”, *Communications of the ACM*, 47:10, October 2004.
22. Krenn, R., “Steganography and Steganalysis”, <http://www.krenn.nl/univ/cry/steg/article.pdf>
23. Anderson, R.J. & Petitcolas, F.A.P., “On the limits of steganography”, *IEEE Journal of selected Areas in Communications*, May 1998
24. Chandramouli, R., Kharrazi, M. & Memon, N., “Image steganography and steganalysis: Concepts and Practice”, *Proceedings of the 2nd International Workshop on Digital Watermarking*, October 2003
25. Currie, D.L. & Irvine, C.E., “Surmounting the effects of lossy compression on Steganography”, *19th National Information Systems Security Conference*, 1996.
26. Bender, W., Gruhl, D., Morimoto, N. & Lu, A., “Techniques for data hiding”, *IBM Systems Journal*, Vol 35, 1996.
27. Petitcolas, F.A.P., Anderson, R.J. & Kuhn, M.G., “Information Hiding – A survey”, *Proceedings of the IEEE*, 87:07, July 1999.
28. Marvel, L.M., Boncelet Jr., C.G. & Retter, C., “Spread Spectrum Steganography”, *IEEE Transactions on image processing*, *8:08*, 1999
29. T Morkel, JHP Eloff and MS Olivier, "An Overview of Image Steganography," *in Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sandton, South Africa, June/July 2005.
30. R.Anderson and F. Petitcolas, "On the limits of steganography" IEEE Journal of Selected Areas in Communications, Vol. 16, No. 4, May 1998.
31. A. Cheddad, J. Condell, K. Curran and P.M. Kevitt. (2010). "Digital image steganography: survey and analysis of current methods." Signal Processing Journal.
32. International Journal of Computer Science Engineering Technology (IJC- SET) "Modern Steganographic technique: A Survey" by Pratap Chandra Mandal Asst. Prof., Department of Computer Application B.P.Poddar Institute of Management Technology.
33. C.K. Chan, L.M. Chen, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.
34. C.H. Yang, Inverted pattern approach to improve image quality of information hiding by LSB substitution **Pattern Recognition** 41 (2008) 2674–2683
35. S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.
36. C.C. Thien, J.C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition 36 (11) (2003) 2875–2881.
37. C.M. Wang, N.I. Wu, C.S. Tsai, M.S. Hwang, A high quality steganography method with pixel-value differencing and modulus function, J. Syst. Software 81 (1) (2008) 150–158.
38. Young-Ran Park, Hyun-Ho Kang, Sang-Uk Shin, and Ki-Ryong Kwon, An Image Steganography Using Pixel Characteristics Y. Hao et al. (Eds.): CIS 2005, Part II, Springer-Verlag Berlin Heidelberg LNAI 3802, (2005) 581– 588.