www.dmietr.edu.in

ISSN -2277 8659

December 2015

Volume- 1

DMIETR

International Journal on Information Technology Management

Department of Business Management (MBA)

Datta Meghe Institute of Engineering, Technology & Research

ISSN -2277 8659

DMIETR International Journal on Information Technology Management (ejournal)

Volume 1

Issue- DECEMBER 2015

DMIETR, Wardha

INDEX

STUDY AND ANALYSIS OF RECOVERY OF DATA IN CLOUD ENVIRONMENT: A SURVEY

PROF. ANAND R. PADWALKAR
Department of Computer Application
Shri Ramdeobaba College of Engg & Management,Nagpur, India
&
PROF. PRIYANKA P. SAMARTH
Department of Computer Science
S.S.M. College of Computer &Management Nagpur, India

ABSTRACT:

Cloud computing recovery issues requirements have been addressed in publications earlier, but it is still difficult to estimate what kinds of requirements have been researched most, and which are still under researched. This paper carries out a systematic literature review by identifying cloud computing security requirements from publications between last recent years. Cloud Computing (CC) is a new term given to a technological evolution of distributed computing and grid computing. It will categorize these requirements in a framework and assess their frequency of research. The paper will then identify changes in the assessment of requirements and proposed solutions compared to publications prior research work. Backing up our databases to the public cloud is an important strategic focus for us going forward in order to save money, scale our backup and DR operations, and to ensure our applications are always available to customers and business users worldwide. NIST (National Institute of Standards and Technology) defines cloud computing as follows: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Over the past few years, many organizations have started to deploy public cloud for backup and Disaster Recovery (DR). Most enterprises using public cloud find significant cost savings on storage, improved IT productivity, and agility to support new database backup and DR requirements. The paper will be based on the research work carried over the different aspects of DR and surveys based on different modules of designing of an application to recover the data in cloud environment.

Introduction:

Cloud Computing (CC) is a new term given to a technological evolution of distributed computing and grid computing. CC has been evolving over a period of time and many companies are finding it interesting to use. Without the development of ARPANET (Advance Research Projects Agency Network), CC would never have come into existence. The advent of ARPANET, which helped to connect (for sharing, transferring, etc.) a group of computers , lead to the invention of Internet (where bridging the gap between systems became

easy)**[1]**. This Internet helped to accelerate number of activities such as human interaction (social media, instant messaging, etc.), business needs of an organization (online shopping, financial services, etc.). Further advancement in this area of Internet resulted in development of Applications Service Provision (ASP), grid and utility computing and cloud computing. CC introduced a new paradigm which changed the traditional interconnection of systems to a pool of shared resources that can be accessed through internet.

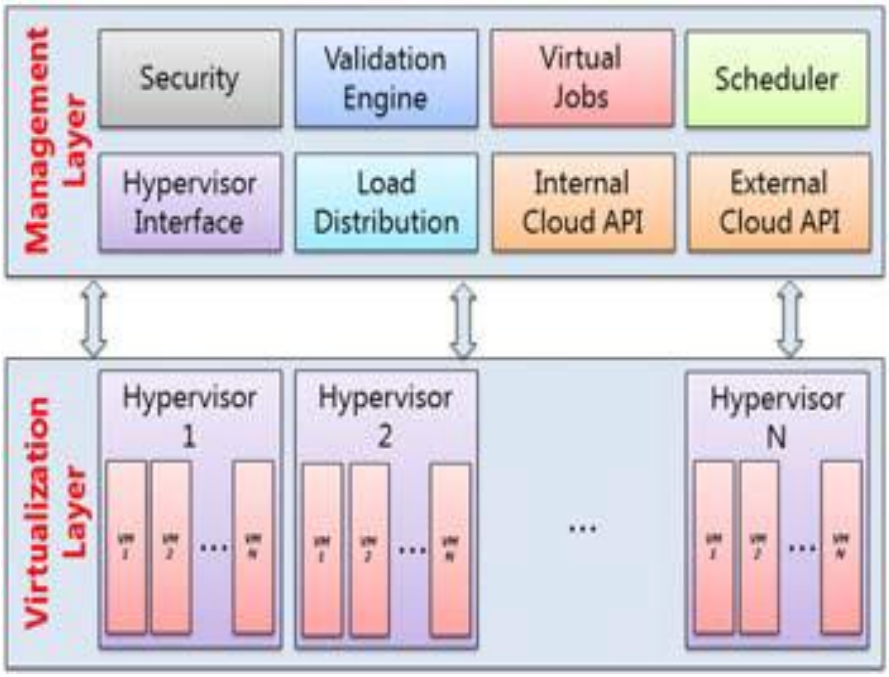

Fig1: Basic Cloud Computing Architecture

NIST (National Institute of Standards and Technology) defines cloud computing as follows: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction", as shown in Fig 1. This definition clearly states that CC helps in minimizing an organization's expenditure towards managing resources and also reduces the burden of maintaining software or hardware by its user **[2]**. When burden of management, maintaining a software/hardware is reduced, the companies' expenditure and time spent towards infrastructure management is reduced and time saved can be utilized in doing some creative work.

Survey Based Study and Analysis

The literature identifies three different broad service models for cloud computing: a) *Software as a Service (SaaS),* where applications are hosted and delivered online via a web browser offering traditional desktop functionality for example Google Docs, Gmail and MySAP. *B) Platform as a Service (PaaS),* where the cloud provides the software platform for systems (as opposed to just software), the best current example being the Google App Engine. c*) Infrastructure as a Service (IaaS),* where a set of virtualized computing resources,

such as storage and computing capacity, are hosted in the cloud,customers deploy and run their own software stacks to obtain services. Current examples are Amazon Elastic Compute Cloud (EC2), Simple Storage Service (S3) and Simple DB. The literature also differentiates cloud computing offerings by scope. In private clouds; services are provided exclusively to trusted users via a single-tenant operating environment. Essentially, an organization's data centre delivers cloud computing services to clients who may or may not be in the premises. Public clouds are the opposite: services are offered to individuals and organizations who want to retain elasticity and accountability without absorbing the full costs of in-house infrastructures [3]. Public cloud users are by default treated as untrustworthy. There are also hybrid clouds combining both private and public cloud service offerings.

This section includes survey conducted by international data corporation (IDC). It shows the strength of cloud computing to be implemented in IT industry and gives the potential inspiration to CSP. The section contains the survey related to the growth of cloud, security aspect, cloud is the first priority to the vendors, revenue report, future and current usage, state of cloud to the IT users and popularity survey of cloud computing.

a) Cloud growth: The Table 1 shows the cloud growth from year 2008 to 2012 [5].

b) Survey on cloud security: This represents security as first rank according to IT executives [8]. This information is collected from 263 IT professional by asking different question related to the cloud [6], and many of the executives are worried about security perspective of cloud.

Year	2008	2012	Growth
Cloud IT Spending	$ 16 B	$42 B	27%
Total IT spending	$383 B	$ 494 B	7%
Total-cloud spend	$367 B	$ 452 B	4%
Cloud Total spend	4%	9%	

Table 1: Cloud Growth

c) *Top ten technology priorities:* This report displayed in Fig 2 collected at the end of 2010 by IDC. This shows that now a days the cloud computing is the first priority by organization in the field of technology [4].

d) *World wide IT cloud services revenue by product/service type:* The Fig. 3 show the survey collected in 2009 by IDC. This survey shows the revenue on cloud in 2009 is 17.4 billion dollars but it will enhance up to 44.2 billion in 2013 **[7]**.

e) Current and future usage of cloud in IT: The Fig 4 shows the graph that is collected by IDC in August 2009.

It shows today's usage and future usage of Cloud in different areas [8].

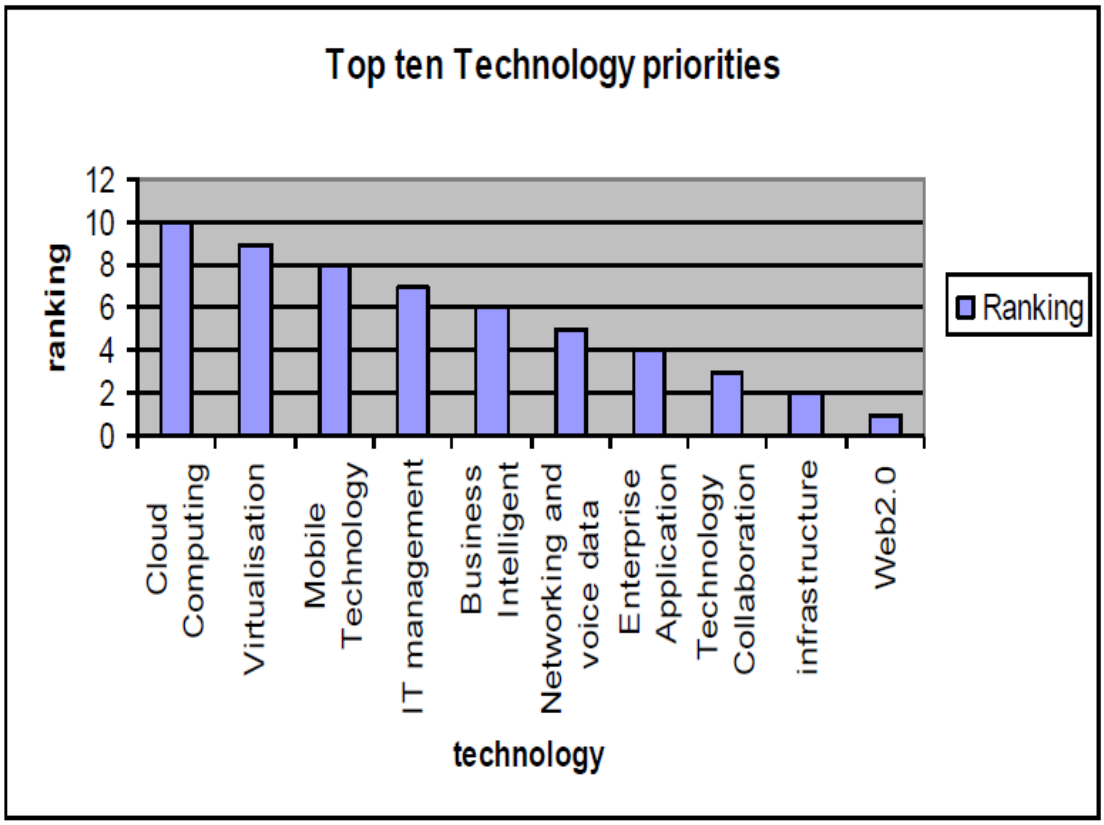

Fig 2: Top Ten Technology Priorities Survey

Existing Solutions for Security Threats

Mirage Image Management System

The security and integrity of VM images are the foundation for the overall security of the cloud since many of them are designed to be shared by different and often unrelated users. This system addresses the issues related to secure management of the virtual-machine images that encapsulate each application of the cloud [12].

The overall architecture of Mirage Image Management System. Mirage Image Management System consists of 4 major components:

1. Access Control. This framework regulates the sharing of VM images. Each image in the repository has a unique owner, who can share images with trusted parties by granting access permissions.

2. Image Transformation by Running Filters. Filters remove unwanted information from images at publishing and retrieval time. Filters at publish time can remove or hide sensitive information from the publisher's original image. Filters at retrieval time may be specified by the publisher or the retriever.

3. Provenance Tracking. This mechanism tracks the derivation history of an image.

4. Image maintenance. Repository maintenance services, such as periodic virus scanning, that detect and fix vulnerabilities discovered after images are published.

Josiah Dykstra and Alan T. Sherman, Cyber Defense Lab, Department of CSEE [13], have emphasized on the model to show the layers of trust required in the cloud. Secondly, they have presented the overarching context for a cloud forensic exam and analyze choices available to an examiner. Also for the first time they have proposed an evaluation of popular forensic acquisition tools including Guidance EnCase and Acces Data Forensic Toolkit, and shown that they can successfully return volatile and non-volatile data from the cloud

Digambar Powar, G. Geethakumari, BITS-Pilani, Jawaharnagar, Shameerpet, Hyderabad[7], emphasized on finding and analyzing digital evidence in virtualized environment for cloud computing using traditional digital forensic analysis techniques. They have also focused on basic services of cloud through which the data recovery considerations can be obtained.

Deoyani Shirkhedkar and Prof. Sulabha Patil [2] have proposed digital forensic technique for cloud environment which will detect two attacks DDOSs and unauthorized file sharing. This paper also emphasizes on forensic investigation techniques of data by taking into consideration of digital object.

Farzad Sabahi, *Member, IEEE,* [9] have focused on new security architecture in a hypervisor-based virtualization technology in order to secure the cloud environment virtualization technology is built on virtualization technology which is an old technology and has had security issues that must be addressed before cloud technology is affected by them. The paper also emphasizes on relation between reliability and security in virtualization, virtual machines security threats and attacks in virtualization

Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina and Eduardo B Fernandez [10] have analyzed the security issues by identifying the main vulnerabilities in this kind of systems and the most important threats found in the literature related to Cloud Computing. In this paper the Systematic review of security issues for cloud computing are discussed like Application Security, Multi-Tenancy, Third party Relationships and solutions to mitigate the treats and vulnerabilities in Cloud environment.

FaridDaryabar, Ali Dehghantanha, NurIzura Udzir, Nor Fazlidabinti Mohd[10], have identified a cross-disciplinary approach between cloud forensics, digital forensics and cloud Computing. This paper also shown the role trusted third parties and the cloud service providers' perspectives and has explained whether they can gain the authority to get access to the evidence. And finally, for service providers' point of view this paper has summarized whether they able to guarantee the safety of the data.

Mohsen Damshenas, Ali Dehghantanha, Ramlan Mahmoud, Solahuddin bin Shamsuddin[15] have discussed about the concept of cloud computing, as well as forensic investigation practices; knowing both, brings the cloud forensic investigation issues to light. This article illuminated some of the conflicts of digital forensic investigation and cloud environment; and then continued with discussing some of the possible solution.

Abdul Wahid Khan, Siffat Ullah Khan, Muhammad Ilyas and Muhammad Ilyas Azeem[16] explored the importance of the cloud computing and risks associated with the cloud computing procedure and process. This paper also illustrated the data privacy problem in cloud computing environment. Different data protection models and techniques have been defined that show their contribution in cloud computing. This paper provided a base for future research work in the field of data security of cloud computing system.

Problem Formulation

The power, edibility and ease of use of CC comes with lot of security challenges. Even though CC is a new intuitive way to access applications and make work simple, there are a number of challenges/issues that can affect its adoption. A non-exhaustive search in this field reveals some issues. They are: Service Level Agreements (SLA), what to migrate, security, etc.. CC has a feature of automatic updates, which means a single change by an administrator to an application would react on all its users. This advertently also leads to the conclusion that any faults in the software are visible to a large number of users immediately, which is a major risk for any organization with little security. It is also agreed up on by many researchers that security is a huge concern for adoption of cloud computing.

A survey by IDC on 263 executives also shows that security is ranked first among challenges in CC . Even though a company boasts to have top class security and does not update its security policies from time to time, it will be prone to security breaches in near future. In this regard, through this detailed study, we propose to update the readers with different distinctions (types of) in security challenges and their solutions. We also include real-time practices to mitigate challenges, include improved solutions proposed by researchers to show which areas of cloud computing need more attention [7].

References:

[1] Santosh Bulusu Kalyan Sudia School of Computing Blekinge Institute of Technology SE-371 79 Karlskrona, "A Study on Cloud Computing Security Challenges

[2] Kaleem Ullah and M. N. A. Khan ," Security and Privacy Issues in Cloud Computing Environment: A Survey Paper", International Journal of Grid and Distributed Computing Vol.7, No.2 (2014), pp.89-98 http://dx.doi.org/10.14257/ ijgdc.2014.7.2.09

[3] M. Usha *a. "A Study on Forensic Challenges in Cloud Computing Environments" Journal of Nano Science and Nano Technology, Vol 2 | Issue 3 | Spring Edition | DOI : February 2014 | Pp 291-295 | ISSN 2279 – 0381.

[4] F. A. Alvi1, Ψ, B.S Choudary2 ,N. Jaferry3 , E.Pathan4 , " A review on cloud computing security issues & challenges"

[5] Alecsandru Pătrașcu, Victor-Valeriu Patriciu "Logging System for Cloud Computing Forensic Environments", CEAI, Vol.16, No.1 pp. 80-88, 2014

[6] N.M. Karie1,2, H.S. Venter1" An Ontological Framework for a Cloud Forensic Environment" ,Proceedings of the European Information Security Multi-Conference (EISMC 2013)

[7] Josiah Dykstra*, Alan T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques", Digital Investigation 9 (2012) S90–S98, journal homepage: www.elsevier.com/locate/diin

[8] George SIBIYA1, Hein S. VENTER2 and Thomas FOGWILL1, "Digital Forensic Framework for a Cloud Environment", IST-Africa 2012 Conference Proceedings Paul Cunningham and Miriam Cunningham (Eds) IIMC International Information Management Corporation, 2012 ISBN: 978-1-905824-34-2

[9] F. A. Alvi , B. S. Choudary,N. Jaferry,E. Pathan, " A Review on cloud Computing Security Issues in & Challenges".

[10] Santosh Bulusu, Kalyan Sudia," A Study on Cloud Computing Security Challenges", Masterarbete/ Master's Thesis (120 credits), Dec 2013.

[11] Popovic K, Hocenski Z (2010) Cloud Computing Security issues and challenges. In: Proceedings of the 33rd International convention MIPRO. IEEE Computer Society Washington DC, USA. pp 344-349

[12] Abdul Wahid Khan, Siffat Ullah Khan, Muhammad Ilyas and Muhammad Ilyas Azeem, "A Literature Survey on Data Privacy/ Protection Issues and Challenges in Cloud Computing" *IOSR Journal of Computer Engineering (IOSRJCE) ISSN : 2278-0661 Volume 1, Issue 3 (May-June 2012), PP 28-36 www.iosrjournals.org*

[13] Josiah Dykstra and Alan T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques", Cyber Defense Lab, Department of CSEE University of Maryland, Baltimore County (UMBC) 1000 Hilltop Circle, Baltimore, MD 21250 April 18, 2012.

[14] Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina and Eduardo B Fernandez," An analysis of security issues for cloud computing" , Hashizume et al. Journal of Internet Services and Applications 2013, http://www.jisajournal.com/content/4/1/5

[15] Mohsen Damshenas, Ali Dehghantanha, Ramlan Mahmoud, Solahuddin bin Shamsuddin, "Cloud Computing and Conflicts with Digital Forensic Investigation", International Journal of Digital Content Technology and its Applications(JDCTA) Volume7,Number9,May 2013 doi:10.4156/jdcta.vol7.issue9.65

[16] Abdul Wahid Khan, SiffatUllah Khan, Muhammad Ilyas and Muhammad IlyasAzeem, " A Literature Survey on Data Privacy/ Protection Issues and Challenges in Cloud Computing", *IOSR Journal of Computer Engineering (IOSRJCE) ISSN : 2278-0661 Volume 1, Issue 3 (May-June 2012), PP 28-36 www.iosrjournals.org*

EXTRACTIVE AND ABSTRACTIVE TEXT SUMMARIZATION: A CONTRASTING APPROACH TOWARDS CREATING TEXT SUMMARIES

PROF. APARNA M. GURJAR
Assistant Professor
RCOEM, Nagpur

&

PROF. RUPA P. PATEL
Assistant Professor
RCOEM, Nagpur

ABSTRACT

Text summarization plays a vital role in this age where hundreds and thousands of terabytes of digital information is being produced in all spheres of life. Text summarizers condense original information which helps drive many natural language applications which need information in a summarized form. They can either be of extractive or abstractive type, depending upon the kind of output generated as a summary. This paper studies the extraction based summarizer "SUMMARIST" and compares it with the methodology used for developing abstractive summarizers. It also discusses various evaluation parameters for summary generated by both the types and scope for future research.

Keywords: *Text summarization, Natural Language Processing, Extractive or Abstractive summarization*.

Introduction

Summarized information has always played a vital role in the process of communication through the ages. In the prehistoric days too when cave men drew pictures on the walls to communicate with their fellow beings; they were subconsciously using the concept of summarization, albeit through the use of pictures. As different natural languages developed around the world and became richer in content and expression, the representation of information also became detailed and complex. With the advent of the Web there was a literal explosion of information. Today one has terabytes of data at the finger tips; and searching and extracting relevant information has become the biggest challenge. This has fuelled the interest in the research area of text summarization. Text Summarizers can create summaries which retain important parts of the original document. Search can be performed on multiple summarized documents created in this way in order to identify relevant documents first and then, extract the relevant information from the original full version. This is the approach used by many search engines. Text Summaries are also required for creating literature reviews and condensed news articles.

Text summarization has been defined in numerous ways by many authors. Lloret E. (2012) cites various sources to define text summarization as follows: It is the process of distilling the most important information

from a source(or sources) to produce an abridged version for a particular user (or users) and task (or tasks). When computer programs are written to create abridged version of the original document it is called as automatic text summarization. It requires an input text which is processed, to produce an output which is also in text form.

Text summarization can be broadly classified into two types based on the kind of output generated as a summary. It can either be an extract or an abstract. If the summary is an extract it contains significant parts of the original text and the process of summarization is called as extractive summarization. If the summary is an abstract then a text, having words and phrases different from the original text is created using Natural Language Generation techniques. This text which is semantically related to the original text acts as a summary. This process of summarization is called as abstractive summarization. Abstracts can be further divided into indicative and informative abstracts. Indicative abstracts provide a basis for selecting documents for closer study whereas an informative one covers all salient information in the source at some level of detail, Witten, I.

This paper studies both the extractive as well as abstractive summarization process in terms of their contrasting approach towards summary generation and discusses various evaluation parameters for summary generated by both the types.

The extractive text summarization process

The process is described using an extraction based summarizer "SUMMARIST" Chin-Yew, L., Hovy, E. (2000). Most of the earlier extractive summarizers calculated term frequency in one form or the other to identify important words and sentences in the original text. SUMMARIST uses the concept of topic signatures which are conceptually more related with methodology used by abstractive summarizers. Hence it is used as a reference for comparing the two methodologies.

SUMMARIST is an automated text summarization system designed to generate summaries of multi lingual input texts. It has three stages Topic Identification, Topic Interpretation and Summary generation.

Topic Identification: This stage identifies the key aspect of the text. It uses positional importance, topic signature, and term frequency to identify the main topic. Topic signature acts as a group of terms that are conceptually related to each other. It consists of two terms: a topic which is the central concept and a signature vector consisting of individual terms and their associated weights. A set of training document is used. For a given topic, this set is divided into two disjoint sets; with one set classified as relevant to the topic and other non relevant. A frequency measure which uses most likely hood ratio of term occurrence is established and given to each term in the document collection. All the single, double and triple terms occurring together are ranked in descending order with top rank being the one with highest term occurrence.

Topic Interpretation: In this stage each word of the given topic is assigned weights of that particular keyword (arrived at from the training data). The word scores in the sentence are added up to form sentence score. A

low sentence score indicates low relevance and a high score indicates more relevance to the topic. Based on this score it is decided whether or not the particular sentence is included in the summary.
Summary generation stage then uses concepts of natural language generation to create document summary. SUMMARIST is capable of creating keyword and extractive type summaries.

Abstractive summarization

Abstractive summarizers produce summaries which are closer to those produced by humans. Humans, while writing summaries, first understand the topic in order to identify the theme and then apply various techniques for generating conciseness. The Purdue University Writing Lab© has a documented strategy for pruning sentences; which throws light on how human beings generate summaries. The various processes of replacing vague words with specific words, eliminating words which provide excessive details, and making less use of expletives- phrases that do not contribute to the semantics of the sentence but are used for structuring the sentence etc. are used during summary generation. These pruning activities, which are easily performed by humans, require too much effort for machine simulation, e.g. "replacing a verbose phrase with a single effective word that captures the essence of the phrase, requires access to resources like a tagged word corpus and involves automatic Natural Language Generation" , Jurafsky, D., Martin, J.(2000). But the basic problem with human summary generation is that it is time consuming and impossible to handle voluminous documents. Therefore automatic text summarization becomes a viable option.
The abstractive summarization process is an ongoing research area in which many methodologies are being explored. Atif, K. et al (2014) identify various techniques both structure and semantic based, which help in generating abstractive summaries. Structure based methods focus mainly on the templates and specialized structure like trees to encode important topics; whereas semantic based methods require some kind of representation for describing concepts, their relationships, and then create semantic models, as reported by MIT. After the representation is selected the important topics related with the text need to be identified. This is done through the process of content selection. This is achieved through the use of extraction rules and various algorithms and metrics. This selected content is then combined and arranged to get an output summary. The authors Genest, P.E., Lapalme, G. et al (2011) propose the concept of Information Items (INIT) to help define the abstract representation of a concept. They develop an abstractive summarization framework based on INIT Retrieval, INIT Selection, summary planning for creating structure of the text, and lastly summary generation with coherent syntax and punctuation. Thus the problem of abstractive summarization can be thought to be one of Text Representation, Content Selection and Summary Generation.

Comparative Analysis of Extractive and Abstractive Methodology

This section compares the two methodologies based on certain parameters some of which are intrinsic and others quantitative in nature.

Deciding Relevance and Granularity: In case of extractive summarization the granularity of the document can be decided in the beginning and summary size can correspond to that value. For e.g. a summary size of 25% would condense the original document to one fourth of its actual size. The condensation is done by identifying important sentences and retaining them in the final summary where as the rest of the sentences omitted. The criterion for deciding the importance of a particular sentence to a summary varies for different types of models used for creating extractive summaries. In the tf.idf model [8] the frequency of each term is calculated and an associated weight is attached to it; denoting a word's importance to a specific document in a set of training documents. The SUMMARIST as explained before calculates the sentence score to decide its inclusion in the final summary. The cut-off value of the term frequency or the sentence score can then decide the size of the summary. A low cut-off will generate large summaries whereas a high value will generate a concise one.

In case of abstractive summarizers the term frequency cannot form the basis of deciding importance of a word or a sentence as actual sentences from an original document are not used directly for creating a summary. The granularity has to be decided on the centrality of a concept in a topic; and strength of the relationship of secondary concepts to the central concept. After the central concept and associated concepts are identified they are associated with words and sentence representations are formed. These are semantically rearranged and sentences are generated by a natural language generation system. A related methodology which is based on "concept" uses RSG(Rich Semantic Graphs) as described in Moawad, I., Mostafa, A. (2011), Leskovec, J. After deep syntactic analysis, the triples called subject-predicate-object are identified and converted into a semantic graph. The nodes in the graph correspond to subjects and objects, and predicates form the link between them. The generated graph is then reduced using graph reduction techniques which focus on retaining the centrality of concept. The text generation phase finally generates the summary from this reduced semantic graph.

Precision Recall and F-score: They are the extrinsic evaluation parameters used by Information Retrieval (IR) systems. Precision is defined as the ratio of the number of sentences correctly identified by the system to the total number of sentences identified by the system. Recall is defined as the ratio of the number of sentences correctly identified by the system to the total number of sentences in the gold standard (Reference Standard). The F score is the harmonic mean of recall and precision. These metrics are also used for evaluating the extractive type of summaries in which the original document acts as a Reference Standard against which the summaries can be evaluated. Precision measures the degree of correctness of the generated summary with reference to the sentences in the original text. Both precision and recall values based on current definitions, are difficult to interpret for abstractive summary as the generated summary sentences can be entirely different from the sentences in the original text but the concept conveyed can be correct.

Redundancy: The amount of information that gets repeated in the summary is called redundancy. In case of extractive summarizers due to extraction of important sentences based on some relevancy metric, it is possible that different sentences with high relevancy score but conveying the same concept can get selected in the final summary. The abstractive summarizers focus instead on relevancy of a concept to the summary and hence

generate less redundant information. Even human summarizers select only one among the many equivalent sentences, described by Nenkova, A., McKeown, K. (2011).

Coherence: During extraction of important sentences from original document it is quite possible to extract sentences which are out of context or which have unresolved linguistic ambiguities. "When such sentences are concatenated the coherence of the generated summary is affected", Hassel, M. (2007).Thus extractive summarizers inherently face the problem of coherence but abstractive summarizers work on concepts which are semantically and linguistically resolved before language is generated. As a result they show less problems related with context and linguistic ambiguity.

Linguistic Quality: Most metrics evaluate the information content of a summary but linguistic quality in terms of grammaticality, low redundancy etc is also important, quoted by Nenkova, A., Mc Keown, K. (2011). If the original document has a fine quality of the language then the linguistic quality of the extractive summary tends to be good as original sentences from the document are extracted for summary generation. The linguistic quality of the abstractive summarizer on the other hand depends upon the analyzing and generating capacity of the natural language generating system. "Therefore both the content as well as generation needs to be controlled", in Genest, PE. Lapalme, G. et al (2011).

Scope for future research:

The abstractive approach towards summarization presents an open area for future research with recent effort focused on rule based, ontology based and rich semantic graph based techniques. The cognitive aspect is also gaining ground which aims to create a summarization model based on the techniques used by human summarizers. Another current topic for research is in the area of query-focused summarization and multi document summarization which create task based summaries according to specific user task.

Conclusion:

Text summarization is an important requirement for many NLP applications. Initially most of the research was being done on building extractive type of summarizers as the basic focus was only condensation of information in a given text. The summary evaluating parameters like compression ratio, term frequency and precision were measured with respect to the original text and some reference standard. Natural language generation was not required as important sentences were directly picked up from original text. As the scope of NLP has widened, summary not only in terms of its size but in terms of abstract or gist of an original text is required. Abstractive summarization techniques which focus on semantic relationships between words and sentences, are being developed, which along with text summarization also have to perform natural language generation. Consequently, evaluation parameters which measure intrinsic concepts like relevance, coherence, linguistic quality etc. are needed to evaluate the quality of the summary.

Bibliography

[1] Lloret, E. (2012) Text Summarization: An Overview

[2] Witten, I.Text mining Computer Science, University of Waikato, Hamilton, New Zealand

[3] Chin-Yew, L., Hovy, E.(2000)The Automated Acquisition of Topic Signatures for Text Summarization

[4] Purdue University Online Writing Lab [Online] Available from: https://owl.english.purdue.edu/owl/

[5] Jurafsky, D., Martin, J.(2000) Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.

[6] Khan, A., Naomie, S.(2014) A Review on Abstractive Summarization Methods

[7] MIT. [Online] Available from: http://web.mit.edu/cocosci/Papers/topics15.pdf

[8] Hassel, M. (2007) Resource Lean and Portable Automatic Text Summarization Doctoral Thesis Stockholm, Sweden

[9] Genest, PE. Lapalme, G. et al (2011) Framework for Abstractive Summarization using Text-to-Text Generation

[10] Moawad, I., Mostafa, A. (2011) Semantic Graph reduction approach for abstractive text summarization

[11] Extracting Summary Sentences Based on the Document Semantic Graph- Jure Leskovec et al.

[12] Nenkova, A.,McKeown, K. (2011) Automatic Summarization

THE UNIQUE CHALLENGES OF TEST AUTOMATION ON EMBEDDED SYSTEMS

PROF. PRAVIN Y. KARMORE
Assistant Professor
Shri Ramdeobaba College of Engineering & Management, Nagpur, India

DR. PRADEEP K. BUTEY
Associate Professor
Kamla Nehru Mahavidyalaya, Nagpur

ABSTRACT

An embedded system is a computer system designed to interface with, and control, some sort of electromechanical device(s). That amalgamation of computing power with an interface to external devices creates special challenges when it comes time to test the system software. Most software shares certain potential anomalies in common: incorrect logic, math, algorithm implementation, program flow [branching, looping, etc.], bad data, data boundary issues, initialization problems, mode switching errors, data sharing, etc. Embedded systems introduce many factors that can result in anomalous system behavior. While, it is necessary to address more conventional software defects, it is also necessary to consider these additional factors when creating tests. Test automation on an embedded system requires three things: special tools, a customized interface or test "harness" between the tester and the system under test, and special automation techniques to cover not only common software defects but also those that are unique to embedded systems. Developing this test harness can be both complex and expensive. This time and monetary cost has to be factored into the project in order to get an accurate test schedule and return-on-investment calculation. It is necessary to work closely with both software and hardware engineers to design this test harness, particularly if expertise in those disciplines is insufficient. First, choose a tool for test automation on an embedded system. This tool should include provisions to manipulate physical analog and binary inputs and outputs interfaced to the system being tested. An automation tool must also produce manageable and maintainable test artifacts. A number of commercial tools on the market cover some or all the criteria. But the embedded test automation market has some notable gaps and could be better served with specialized tools. Once appropriate automation tools have been selected and designing and building a test harness for the embedded system has been completed, it is time to create some test scripts. Because of the real-time nature of embedded systems, the test professional must also employ specific automation techniques. Being aware of these unique challenges will greatly decrease the time needed to debug automated tests, which will result in successful automation attempts and greater likelihood of management satisfaction. Test automation on an embedded system can greatly expand the scope of testing and eliminate defects that would have been virtually impossible to identify using manual testing alone. Awareness of the unique challenges posed by embedded systems can help the test professional to decide an appropriate scope of automation, avoid pitfalls during test development, and deliver a successful product.

Introduction

Embedded systems introduce many factors that can result in anomalous system behavior. These factors consist of:

- Processor loading
- Watchdog servicing
- Power modes (sleep, low, standby, etc.)
- Bad interfacing to external peripherals
- Peripheral loading (e.g. network traffic, user interface requests)
- Signal conditioning anomalies (e.g. filtering)
- Thread priority inversion
- Noise conditions

While, it is necessary to address more conventional software defects, it is also necessary to consider these additional factors when creating tests. Otherwise the test coverage will be inadequate and the system will likely ship with an unacceptable number of potentially serious defects.

While most, if not all, of these obstacles can be overcome—given enough time and resources—the fact remains that addressing them requires effort above and beyond what would be required in a more conventional computing system. Test automation on an embedded system requires three things: special tools, a customized interface or test "harness" between the tester and the system under test, and special automation techniques to cover not only common software defects but also those that are unique to embedded systems.

Special Tools

First, choose a tool for test automation on an embedded system. This tool should include provisions to manipulate physical analog and binary inputs and outputs interfaced to the system being tested. And often an embedded system will utilize one or more communications protocols—the automation tool will need to be able to support these as well. This may, in fact, require a separate automation tool.

For example, the tester might evaluate the portions of code that manipulate hardware input/output (I/O) using one automation tool and then utilize a different automation tool to test portions of code that communicate using communications protocols such as BACnet or ModBus. The ideal situation, however, is when a given automation tool can handle all of the system inputs and outputs—whether hard wired or communicated—together.

In a similar vein, if the embedded system includes a user display it may be possible to automate here as well, but frequently this will require a separate tool specifically designed to test user interfaces. Debugging an automated test script for an embedded system presents many of the same challenges as debugging the system software. So the same kind of tools that the software developers use to manipulate system inputs and view the outputs in real-time will be needed.

An automation tool must also produce manageable and maintainable test artifacts. Test automation is, after all, just software created to test software, so it runs into many of the same maintenance difficulties faced in

more conventional software development.

Specifically, there are a number of test automation tools that utilize graphical programming languages. These can be extremely useful for rapid prototyping and easy comprehension of specific test steps. But more than one test developer has found that once these graphical programs grow beyond a certain size, the task becomes daunting even for the original developer – let alone somebody else - to understand, modify and extend.

The way inputs and outputs are handled in the test tool should be abstracted from their particular implementation in hardware. A test script should not "care" whether a temperature set point, for example, comes from a thermocouple, a thermistor, or an RTD. It should not "care" if a communicated value comes from a BAC net network, a LAN, or the Internet. Otherwise, a change in system implementation will break all of the scripts.

It is also useful for post-test analysis if the tool bundles both the script and the results from a specific run of that test and archives them in a single file.

This avoid any confusion that might arise as the test script is up dated or expanded—there will always be a record of exactly what test steps yielded a given set of results.

A number of commercial tools on the market cover some or all of these criteria. But the embedded test automation market has some notable gaps and could be better served with specialized tools.

Special Interfaces

Because embedded systems represent an combination of a computer system with external devices, a complete test automation system will require some sort of interface or "harness" between the automated test tool and the system under test.

Developing this test harness can be both complex and expensive. This time and monetary cost has to be factored into the project in order to get an accurate test schedule and return-on-investment calculation. It is necessary to work closely with both software and hardware engineers to design this test harness, particularly if expertise in those disciplines is insufficient.

Be sure to consider one important, overarching principle before planning and beginning work: An automated test harness should not, if at all possible, require any special "hooks" in the software or any special modifications to the hardware. Both software "hooks" and hardware modifications automatically mean that what is being tested is not the same as what the customer will be using.

Special software "hooks" add overhead and therefore affect the performance of the system under test. They also can result in a Catch 22—if the hooks have to be taken out of the software just before shipment, the software has been changed in a fundamental way while the ability to test it has been lost.

And hardware modifications to facilitate interfacing to an automated test system mean that standard, production hardware cannot be used for tests. This can open the door to shipping the product with subtle defects that appear on production hardware but not on the modified system. They also require spending precious time and money acquiring and modifying hardware for use in the test system. This can become especially burdensome if the hardware itself is going through numerous revisions.

Sometimes software "hooks" and hardware modifications cannot be avoided, and the payoff may be more than sufficient to justify their use—as long as the potential pitfalls are fully understood. But in general, try to avoid these technical compromises.

Here are some more challenges that may be encountered when designing a test harness for embedded automation:

- High voltages and currents in the system require due attention to the safety of both human beings and the system under test.
- The interface to each input or output from the system under test may need to be conditioned in order to interface with the available test hardware. For example, an analog voltage may need to be divided down before it is applied to an analog input, or there may need to be connections between the test harness and the system under test.
- Non-linear sensors such as thermocouples can be notoriously difficult to imitate, especially if a very high degree of accuracy is necessary. Achieving accuracy to± 0.5 ºC over the entire operating range may not be too difficult, but0.01 ºC is probably going to be very difficult.
- Presenting a system with a simple DC voltage (e.g. 0-10VDC) or current (e.g. 4-20 mA) is not difficult with off-the-shelf hardware. But presenting it with high voltage, variable capacitance, or variable resistance will be significantly more difficult and will likely require some custom hardware development.
- End-points and extreme values may be difficult to reproduce with the test harness. For example, when using a simple resistor voltage divider to condition an analog output to interface with an analog input, it often is not possible to drive the input all the way to its extremes (especially on the high side) to simulate a "shorted" or "open" input condition.
- Complex and fast communications protocols are a challenge to automate.
- User-intervention is often still necessary via key pads, touch screens, etc. You can automate these things, but may not be cost effective. On the other hand, the user interface may be the only part of an embedded system that can be cost-effectively automated and this may be well worth doing.
- While the subsystems may be manageable on a case-by-case basis, the ability to service all of the system inputs and outputs simultaneously can require a prohibitive amount of processing power in the automated test tool.

That last point brings up yet another factor that must be considered when designing a complete embedded test automation system. The automation system will have to run fast enough to sample inputs at a sufficient rate and assert outputs in a timely fashion.

What that means varies from system to system

In the HVAC industry, for example, being able to respond within one second is usually quite sufficient, with many events taking place in the5 to 10 second range. This makes test automation very feasible. On the other hand, something like an automobile engine controller or a flight guidance system may need to process inputs hundreds or thousands of times per second and assert outputs within milliseconds of detecting a given condition. A test automation system capable of that level of performance may be prohibitively difficult and expensive.

But even faced with such a scenario, can useful testing be done at reduced speeds? If so, some automation may still be possible and warranted.

The bottom line is that it is necessary to factor in test harness development, fabrication, and testing of the harness itself into the project schedule. It is an added bonus if the test harness is designed to be generic and/or expandable, so that it can be applied to more than one product. This can enhance the long-term return on investment, so watch for these opportunities.

And given that there may be technical obstacles that would prevent test automation on the entire system, it may still be worthwhile to automate even a portion of a project, provided that the return on the investment of time and effort promises a payoff.

Special Automation Techniques: Some Typical "Gotchas" in Embedded Software Test Automation

Once appropriate automation tools have been selected and designing and building a test harness for the embedded system has been completed, it is time to create some test scripts.

Here again there are a number of special considerations that should be factored in to the automation effort on an embedded system.

First, embedded systems can be vulnerable to initialization problems. You can write scripts and have them pass ordinarily, just because some system input is typically sitting at a given value. But if a prior test script left that system input at a non standard value, suddenly a subsequent script may fail. So the same test on different test set-up/facility/etc. can fail unexpectedly because a less than comprehensive initialization has been performed.

To address this, try to have a comprehensive initialization sequence that can be called by all test scripts. Make it a matter of policy that this initialization sub-script is called at the start of each script. Yes, people are going to complain that it seems to be a waste of time to execute all of these steps at the start of every single test script. But in the end the time will be well spent, since chasing errant conditions caused by initialization problems will be avoided.

Managing tolerances is crucial to successful embedded automation. It is not practical for a system requirement to say that a system needs to control to a set point of 72 F. It is only helpful to say that the control must control to the set point plus or minus some tolerance. Automated tests need to be written to handle the tolerances rather than absolutes. Otherwise numerous testing errors will be logged when the real world system deviates, even slightly, from those absolutes.

Race conditions are caused specifically by timing tolerances. A race condition is a fault in an electronic system or process whereby the result or output of the process is critically and unexpectedly dependent on the sequence or timing of other events. The term originates with the idea of two signals racing each other to influence the output first. In the case of test automation, it most often manifests itself in a condition in which the test script execution gets to a checkpoint first—perhaps even by just a millisecond—and fails the step because the process it's checking has not caught up.

Conversely, the process on the embedded system may have just completed and moved on—so the test script fails to detect the desired process state because it has already enthused on. Fortunately, race conditions are comparatively easy to avoid.

Tests can use a simple "Wait While "followed by a "Wait For" construct. As long the timing necessities for the event that's being tested are understood, this combination will not only prevent false errors because of the race condition, but will also verify that the system is working inside of its formal timing requirements.

A very big potential "gotcha" in automated testing on an embedded system occurs when the test is not

comprehensive enough to catch an unexpected glitch on a system output that might seem to fall outside of the specific test case. The difficulty is that a given system may have dozens or even hundreds of outputs. It is usually impossible to check the status of all of them in every test step.

At the very least, be sure that each test case explicitly checks the status of all known critical values. But on the flip side, so as not to add unnecessary execution overhead, if it's truly a "don't care" then don't include it.

Formal script reviews are the solution here—other test engineers, hardware engineers and software developers may identify system outputs that were not considered, but that really should be included in the test case.

To Automate or Not to Automate: Finding the Return on Investment

The first question is whether the embedded system testing should be fully automated. The answer is no, generally not. At the very least, relying completely on automated tests is probably a bad idea. As mentioned above, in a system of any significant complexity there are simply too many inputs and outputs for the tests to be absolutely comprehensive. Many times manual tests run by individuals with significant understanding of the system will catch defects that would have been missed by a more narrowly scripted automated test.

Total reliance on automated testing will generally not result in sufficient coverage. There are aspects to most embedded systems that will defy full coverage through automation without enormous effort. And in certain embedded systems there are human and machine safety considerations—in these cases, although the safety tests can be automated, they should also be run manually so that a human being verifies the safety of the system.

Regression is the Primary key to ROI

So how does one decide whether to automate or not to automate?

Ultimately this will be determined by calculating the return on investment (ROI) for the automation effort.

Remember, first, that almost every obstacle can be overcome: it is purely a function of how much time, money, and effort can be expended for an ROI. The better the metrics available about your automation process, the more information can be provided management concerning the ROI when automating new systems.

There are many models available to calculate ROI for test automation. Any of these can applied to test automation on an embedded system. The main difference will be to factor in the time it takes to design, build and troubleshoot the test harness(es).

It is also necessary to be aware of any extensions to the test tool that maybe required providing coverage for parts of the embedded system that the tool does not already support, such as a new communications protocol or hardware I/O type.

Here are some good rules of thumb to maximize return on investment from embedded test automation:

- First and foremost, regression is the primary key to ROI. Repetition pays the bills. Automate tests that will be run numerous times over multiple test cycles.
- Intelligent selection of the scope of automation is the secondary key to ROI. Don't bite off more than can be handled (or paid for).The low-hanging fruit would be tests that require large amounts of time to execute and where catastrophic results could result if the software is defective. For example, signal conditioning algorithms such as piece-wise filtering and linearization applied to analog inputs can have

bugs at the transition points that are relatively difficult to detect but can throw the input value wildly out of range. It is easy to create a test that sweeps the entire range of analog values in small increments looking for these anomalies. Such a test would be daunting to run manually, is easy to automate, and can catch software defects that could have catastrophic problems in the embedded system. (But note that, in this example at least, a good code inspection would go pretty far in eliminating the risk of such a software defect.)

- Another way embedded system test automation can have a huge payoff is to reproduce faults that require large numbers of iterations to occur, so many that manual testing would be impractical or impossible. For example, I once worked on a serious field issue that occurred very in frequently and at just a few job sites. The software engineers eventually came up with a set of conditions they thought could reproduce the problem. An automated test was developed to repeatedly present those conditions to the system and it turned out that on average the error would occur approximately every 300 presentations. The ability to reproduce the error enabled the software engineers to craft a fix. The test was then run for thousands of cycles and we were able to calculate, to a statistically exact level of confidence, just how certain we were that there mediation actually fixed the problem. The payoff of the automation was a little difficult to quantify in dollar terms, but the payoff in increased management confidence in the competence of the engineering group was very high.

- Remember that partial automation of a given test may still be worthwhile. Even if an automated test has to stop execution to prompt a user for certain intervention, the test might still provide better coverage, better reporting, better consistency, and be less mind-numbing—and therefore more prone to being run accurately during regression—than a fully manual test.

Conclusion

A unique set of software tools required for test automation on embedded systems. And since embedded systems involve an amalgamation of a specific tester-to-controller, hardware and software interface is required.

Developing this interface can be complicated, challenging, and costly. The test professional must factor in the cost and time needed to create the automation interface, or the testing schedule is incomplete. Because of the real-time nature of embedded systems, the test professional must also employ specific automation techniques. Being aware of these unique challenges will greatly decrease the time needed to debug automated tests, which will result in successful automation attempts and greater likelihood of management satisfaction.

Test automation on an embedded system can very much increase the scope of testing and eliminate defects that would have been virtually impossible to identify using manual testing alone. Awareness of the unique challenges posed by embedded systems can help the test professional to decide on an appropriate scope of automation, avoid pitfalls during test development, and deliver a successful product.

References

[1] Testing embedded systems: Paul Szymkowiak, www.embeddedsystem.com
[2] Singpurwalla and Wilson, Software reliability modeling: Statistical methods in Software Engineering.

[3] Pravin Y. Karmore and Dr. Pradeep K. Butey, "Technical View of Testing Methodologies for Diverse Designs of Embedded System", International Journal on Information Technology Management, vol. 2, ISSN2277 8659, 2013, pp. 123–131.
[4] Goel and Okumoto: Singpurwalla, to determine an optimal time interval for testing and debugging software. IEEE Trans. Software Engineering, 17, 313 – 319.
[5] Harmen Sthamer, Joachim Wegener, and Andre Baresel. Using Evolutionary Testing to improve Efficiency and Quality in Software Testing. In Proc. of the2nd Asia-Pacific conference on Software Testing Analysis & Review. Melbourne, 2002.

[6] Mary Shaw and David Garlan, Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, 2010.
[7] Automation Software testing magazine, April, 2013.

[8] J. Lygeros. Lecture Notes on Hybrid Systems. Cambridge, 2003.
[9] H. Giese and S. Henkler, "A Survey of Approaches for the Visual Model-Driven Development of Next Generation Software-Intensive Systems," J. Visual Languages and Computing, vol. 17, no. 6, 2006, pp. 528–550.

Analysis of Randomness in Graphical Authentication System

Satyajit S. Uparkar

Department of Computer Application

Shri Ramdeobaba College of Engineering & Management Nagpur, India

uparkarss@rknec.edu

&

Purushottam D. Shobhane

Department of Applied Mathematics

Rajiv Gandhi College of Engineering

& Research, Nagpur, India

pds267@gmail.com

Abstract: Graphical password, the proper alternative to text based passwords can offers improved security. The picture-based techniques come in the core area of knowledge-based authentication and can be further divided into three categories: recognition-based and recall based graphical techniques and Cued Recall based scheme. A click based approach, is proposed where users click on image or a sequence of images to create passwords. A lab study is conducted to generate a database of password based on users' images. The probability of the attacker to guess the password of the user is studied. This paper is an attempt to analyze the randomness using J-statistics concept that can be used to measure the security aspect of the proposed system.

Keywords: graphical authentication, J-statistics, security aspect.

1] Introduction:

Computer systems have always faced the concern of security. Secured systems must be usable to maintain intended security. Password Authentication Systems have either been usable and not secure or vice-versa. Today, authentication is the principal method to guarantee information security and the most common and convenient method is password authentication. The synonymous word to any authentication method used is a 'Password', that is generally is text based an alphanumeric word which can be easily remembered by the user. Traditional alphanumeric passwords are strings of letters and digits, which are easy and familiar to essentially all users. Text based passwords are nothing but string of characters. For text passwords, peoples always creates password which is easy to remember but these passwords are easy for attackers to break. Due to the limitation of human memory, most users tend to choose short or simple passwords which are easy to remember.

Graphical passwords, an alternative to text based passwords can offers improved security. In graphical password, images are the tools to provide venerable range of selection that generate the randomness in any system.

2] Study of Relative concepts:

The literature survey for a graphical authentication is highlighted under following dimensions:

2.1] Knowledge based Authentication:

The picture-based techniques come in the core area of knowledge-based authentication and can be further divided into three categories: recognition-based and recall based graphical techniques and cued recall based scheme. The cued recall based scheme can be further sub-divided into three categories as pass-point technique, cued click point (CCP) technique and Persuasive Cued Click-Points (PCCP) Technique.

The three above techniques have been critically studied on various dimensions and come up with their limitations. The major concern of our interest was the randomness of every technique. The previous studies provide the following graphs of randomness.

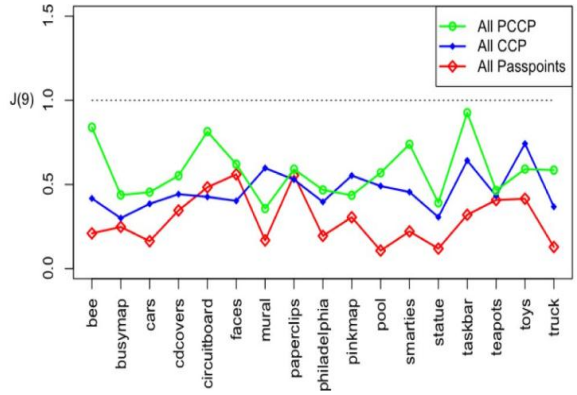

Figure 1: Comparative Study of Randomness

The randomness of PCCP technique is more as compared to the other two techniques. The randomness is calculated using the J-statistics which provide an estimate of the guessing by the attacker.

2.2] Attacker Models:

No system offers perfect security; therefore schemes must be evaluated according to their vulnerabilities. For a particular attack strategy, it is possible to compare the susceptibility of different schemes. In practice following are some attacking model.

- Dictionary Attack
- Exhaustive (brute-force) Attack
- Shoulder-surfing
- Phishing
- Social Engineering

2.3] J-statistics:

Exploratory analysis of point patterns is based largely on summary statistics. To measure the estimate randomness J-statistics is used which given in following formula:

$$J(r) = 1\text{-}G(r) / 1\text{-}F(r) \quad \text{.. (1)}$$

where,

- G(r) is the distribution function of the distance from a point of the process to the nearest other point of the process.
- F(r) is empty space function of the process.
- r is the distance of given point.

This is a nonparametric measure of the type of spatial interaction. The J- function quantified the range and strength of point interactions in a spatial point process function. For a stationary point process F is the distribution function of the distance from arbitrary fixed point to the nearest random point of the process, and G of the distance from a point the process to the nearest other point of the process. For define the empty space function F of X (event of points), has to calculate Euclidean distance from arbitrary point y. To define G find the nearest neighbor from arbitrary point using Euclidean distance. For homogeneous Poisson point process of Intensity λ, the nearest neighbor distance distribution function G is known is

$$G(r)=1\text{-}exp(-\lambda\pi r^2) \quad \text{.. (2)}$$

where,

- r is the distance from an arbitrary point of process.
- λ is intensity (expected number of points per unit area).

This is identical to the empty space function F for Poisson process.

2.4] Inferences of Randomness using J-Statistics:

A result of J(r) closer to 0 indicates that all of the data points cluster at the exact same coordinates, J(r)=1indicates that the data set is randomly dispersed, and J(r) > 1shows that the points are increasingly regularly distributed. For passwords, results

closer to J(r)=1are desirable since this would be least predictable by attackers.

3] Design & Lab Study:

In proposed system, multi-object images are provided to user .The images of size 400x300 pixels and tolerance square is 20x20 pixel size. Images has grid like structure to provide wide better password space. Following figure indicates the user selecting the click point to create the graphical password. The form design and the database is generated using MATLAB tool.

Figure 2: User Registration form

After the completion of registration process successfully, the final data is stored in the database, where the name of the user, along with image number and the corresponding x and y coordinates are stored. e.g.

Ramesh:16.70.145:16.165.100:16.210.100:8.90.65:5.375.215

In above string Name of the user is Ramesh and has used image no.16 thrice and images no.8 and 5 once, to generate a graphical password under the proposed system. A lab study is conducted using 20 images, where 100 user database has been generated. The password length was restricted up to less than or equal to 5 click points. The user can select any image from given set for desire number of times. The study was carried out thrice after the duration of one week gap where the remembrance of the user was put on test to calculate the success rate.

The following graph shows the distribution of click points on the images used by the users to generate the password.

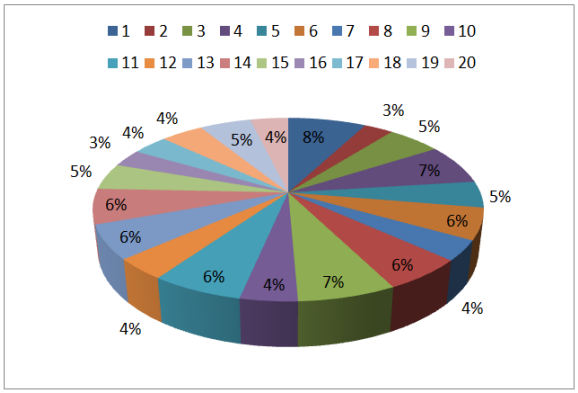

Figure 3: Click point distribution on the images

Total 462 click points were noted in which the following graphs shows 100 passwords distribution using number of images.

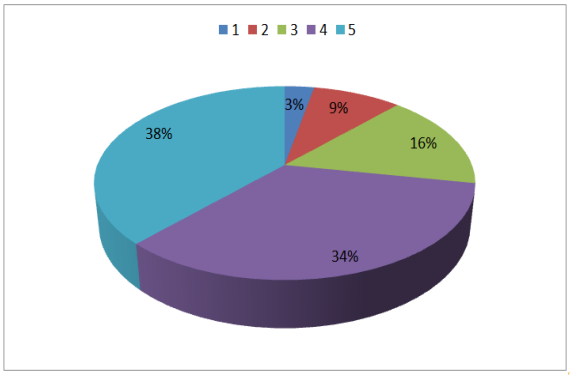

Figure 4: Password distribution using number of images

4. Analysis of Experimental result:

A huge database is process to analyze the randomness of the proposed system. The calculations are carried out by using the mathematical tool to avoid any errors.

4.1] Determination of J Statistics:

The calculation of J statistics, using MATLAB is given below:

Step 1: Define an array containing the set of x and y coordinates of the image under consideration:

Step 2: Find minimum Euclidean distance say r.

Step 3: For given r of step 2, G(r) for image is then calculated suing equation 2.

Step 4: For F function, according to pixel value, and clustering of hot stops value of is calculated for expected value of r'.

Step 5: For expected r', F(r) for the image is determined.

Step 6: Finally J(r) is calculated using equation (1)

For example J(r) =1.0865 for image no. 2, where r=5, r'=0.07 and λ=0.000012. J(r) =1.0865 reflect the randomness and hence provide a high degree of security using the proposed system. Similar process can be applied on remaining 19 images. Following graph shows the J(r) for each image.

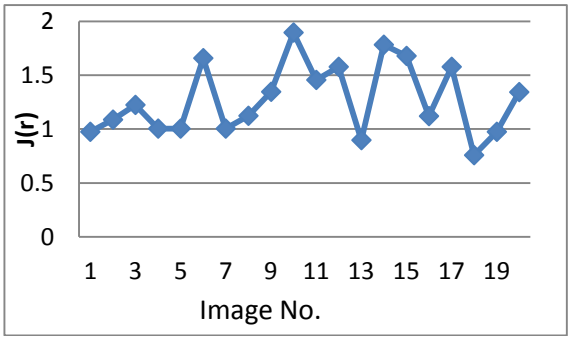

Figure 5: Distribution of Randomness in the images

Comparing the Figure 1 and Figure 5 the J value of all the images are estimate near or above 1, which means that the complexity in guessing any of passwords set by the user is very high. It also shows that there are less clustering in click point pixels and reduces the hotspot problem. In case of different attack is expensive to possible occurs with this system. This increases the workload for attackers by forcing them to first acquire image sets for each user, and then conduct hotspot analysis on each of these images. The proposed system can provide as secured system as compared to the previous techniques.

4.2] Limitations of the proposed system:

Some of the difficulties are analysis and discussed below:

Images Selection: In the proposed system the images provided during registration is user defined. Users are allowed to select image from any sections of system. User has to click anywhere in images. From analysis it is shown click points are randomly dispread. Existing methods used system defined image for user password creation.

Storage Space of System: In the proposed system, user's defined images are select for registration. This scheme provided number of multiple objects images at the time of registration images. Multi object images give more clickable areas to user. Existing method has stored multiple images to registration which increase system load.

5] Conclusion:

The time required during registration phase in proposed system provides good usability. It takes less time than existing systems for password creation and image selection. In view of tolerance size the proposed system provides 20 x 20 pixel size. It makes high usability as compare to existing method. In this system user get specific area around an original click-point accepted as correct. In this grid structure is provide artificial predefined boundaries around areas of the image within which the user can click.

The proposed system offers a more secure alternative to Pass-Points, Cued Click Point, and Persuasive Cued Click Point method. This proposed system increases the workload for attackers by forcing them to acquire images sets for each user. It works for hotspot reduction on each of used images and increase randomness in case of click points.

6] References:

[1] K. Renaud; Evaluating authentication mechanisms; InL.Cranor and S. Garnkel, editors,and Usability: Designing Secure Systems That People Can Use, chapter 6,pp.103-128. O'Reilly Media,(2005).

[2] X. Suo, Y. Zhu, and G. Owen;Graphical passwords: A survey;In Annual ComputerSecurity ApplicationsConference (ACSAC), (December 2005).

[3] R. Biddle, S. Chiasson, and P. van Orschot;Graphical Passwords: Learning from TheFirstTwelve Years; to be published inACM Computing Surveys, vol. 44, no. 4, (2012).

[4] G. Blonder; Graphical passwords;United States Patent 5,559,961, (1996).

[5] A. Paivio;Mind and Its Evolution:A Dual Coding Theoretical Approach; LawrenceErlbaum: Mahwah, N.J,(2006).

[6] C. Kuo, S. Romanosky, and L. Cranor;Human selection of Mnemonic Phrase-based Passwords; In 2nd ACM Conference on Symposium on Usable Privacyand Security (SOUPS), (July 2006).

[7] PatricElftmann, Diploma Thesis;Secure Alternatives to Password-Based Authentication Mechanisms; Aachen, Germany,(October 2006).

[8] A. Salehi-Abari, J. Thorpe, and P. van Oorschot; On Purely Automated Attacks and Click-Based Graphical Passwords; Proc. Ann. Computer Security Applications Conf. (ACSAC),"(2008).

[9] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin; The Design and Analysis of Graphical Passwords; in Proceedings of the 8th USENIX Security Symposium, (1999).

[10] Harsh Kumar Sarohi1, Farhat Ullah Khan; Graphical Password Authentication Schemes: Current Status and Key Issues; IJCSI International Journal of Computer Science Issues, Vol. _10, Issue_ 2, No .1,(March 2013).

[11] Everitt, K.Bragin, T. Fogarty, J. Kohno; T: A comprehensive study of frequency, interference, and training of multiple graphical passwords; In: Proceedings of the 27th international conference on Human factors in computing systems, Boston, MA, USA, pp. 889-898. ACM (2009).

[12] D. Davis, F. Monrose, and M. Reiter; On user choice in graphical passwordschemes;In 13th USENIX Security Symposium, (August 2004).

[13] M. Orozco, B. Malek, M. Eid, and A. El Saddik; Haptic-based sensible graphical password; In Proceedings of Virtual Concept, (December 2006).