ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Data Encryption Using Cloud Computing

Anil Behal[1], Dr. Harish Rohil[2]

[1]M.Tech. Scholar, CSE, Ch. Devi Lal University, Sirsa-125055 (INDIA)
[2]Asst. Professor, Dept. of CSA, Ch. Devi Lal University, Sirsa-125055 (INDIA)

Abstract: ***Cloud computing provides a flexible mechanism for delivering IT services at each level of computing. This research aims to achieve the speed in terms of Encryption time by using cloud computing. Different vendors that provide the cloud computing services are Google App Engine, Amazon Web Services and Microsoft's Azure Service Platform. Different Encryption algorithms that are used in the research are as: (i) Rivest, Shamir and Adleman (RSA), a asymmetric encryption algorithm (ii) Advanced Encryption Standard (AES), a symmetric encryption algorithm (iii) Message-Digest algorithm 5 (MD5), a hashing algorithm. Security algorithms are commonly used by businesses to encrypt large volumes of data. We will use input data of different sizes for better performance evaluation. This research shows the relevance of encryption of huge amount of data by using cloud resources instead of encrypting them on local resources as the encryption time will be reduced to a large extent. Eclipse Juno is the application tool used for encryption of data on local server and cloud server. Google's App Engine is the platform used as cloud server for carrying out the analysis.***

Key Words: Cloud Computing, Data Encryption, RSA, AES, MD5.

1 INTRODUCTION

1.1 Cloud Computing

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications.

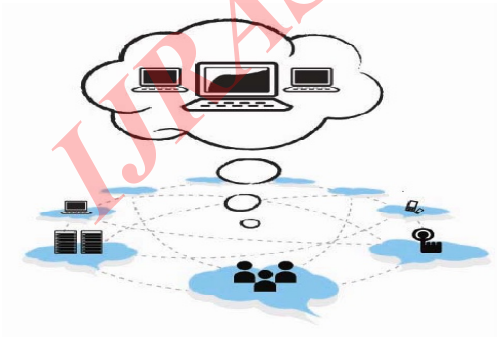

Figure 1.1: Overview of Cloud Network

Cloud computing is a model for enabling 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. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

1.2 Cloud Architecture

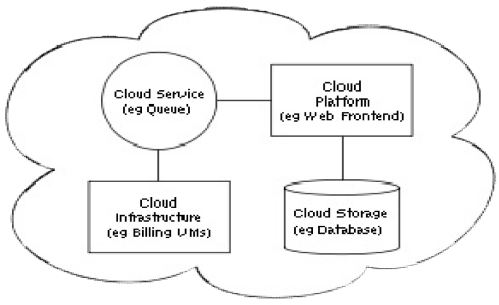

Figure 1.2: Architecture of Cloud Network

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over application programming interfaces, usually web services. This resembles the Unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces. Complexity is controlled and the resulting systems are more manageable than their monolithic counterparts.

1.3 Google App Engine

Google App Engine is a cloud computing platform for developing and hosting web applications in Google-managed data centres. App Engine automatically allocates more resources for the web application to handle the additional demand.

The GAE runtime environment presents itself as the place where the actual application is executed. Google App Engine runs web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as traffic and data storage needs grow

Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, we can use standard Java technologies, including the JVM, Java servlets, and the Java programming language or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a dedicated Python runtime environment, which includes a fast Python interpreter and the Python standard library. The Java and Python runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on system.

With App Engine there are no set-up costs and no recurring fees. The resources the application uses, such as storage and bandwidth, are measured by the gigabyte, and billed at competitive rates. One can control the maximum amounts of resources the app can consumes, so it always stays within budget.

App Engine costs nothing to get started. All applications can use up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month.

1.3.1 Why Using Google App Engine

Compared to other scalable hosting services such as Amazon EC2, App Engine provides more infrastructures to make it easy to write scalable applications, but can only run a limited range of applications designed for that infrastructure.

App Engine's infrastructure removes many of the system administration and development challenges of building applications to scale to hundreds of requests per second and beyond. Google handles deploying code to a cluster, monitoring, failover, and launching application instances as necessary.

While other services let users install and configure nearly any *NIX compatible software, App Engine requires developers to use only its supported languages, APIs, and frameworks. Other competitors include Microsoft's Azure Services Platform, Amazon Web Services, Force.com Platform and Heroku.

1.3.2 Applications of Google App Engine

Google App Engine makes it easy to build an application [10] that runs reliably, even under heavy load and with large amounts of data. App Engine includes the following features:

- Dynamic web serving, with full support for common web technologies.
- Persistent storage with queries, sorting and transactions.
- Automatic scaling and load balancing.
- APIs for authenticating users and sending email using Google Accounts.
- A fully featured local development environment Simulates Google Engine on your computer.
- Task queues for performing work outside of the scope of a web request.
- Scheduled tasks for triggering events at specified times and regular intervals.

2 LITERATURE REVIEW

The concept of cloud computing has been evolving for more than 40 years. In the 1960s, J.C.R.Licklider introduced the term “intergalactic computer network” at the Advanced Research Projects Agency [1]. This concept served to introduce the concept that the world came to know as the Internet. The underlying premise was a global interconnection of computer programs and data. The concept of cloud computing dates back to 1960, when John McCarthy opined that "computation may someday be organized as a public utility"; indeed it shares characteristics with service bureaus that date back to the 1960s. The actual term "cloud" borrows from telephony in that telecommunications companies, who

until the 1990s primarily offered dedicated point-to-point data circuits, began offering Virtual Private Network (VPN) services with comparable quality of service but at a much lower cost [2]. The cloud symbol was used to denote the demarcation point between that which was the responsibility of the provider from that of the user. Cloud computing [3] extends this boundary to cover servers as well as the network infrastructure.

From a technical perspective, cloud computing includes Service Oriented Architecture (SOA) and virtual applications of both hardware and software. Within this environment, it provides a scalable services delivery platform. Cloud computing shares its resources among a cloud of service consumers, partners, and vendors. By sharing resources [8] at various levels, this platform offers various services, such as an infrastructure cloud (for example, hardware or IT infrastructure management), a software cloud (such as software, middleware, or traditional customer relationship management as a service), an application cloud (application, UML modeling tools, or social networks as a service), and a business cloud. Cloud computing itself is a field within service computing, a cross-discipline that bridges the gap between business and IT services.

Cloud computing [7] basically comes to focus on IT, a way to increase capacity or add potentiality on the fly without investing in new infrastructure, training new personnel, or licensing new software. It encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends its existing capabilities. It is often provided "as a service" over the Internet, typically in the form of infrastructure as a service (IaaS), platform as a service (PaaS), or software as a service (SaaS).Microsoft Azure and Google App Engine are the examples of platform as a service. The fast growth in field of “cloud computing” also increases rigorous security concerns.

The UC Berkeley [1] Space Sciences Laboratory’s SETI@home (Search for Extra-Terrestrial Intelligence) project began in 1999 as an attempt to implement distributed computing through computers connected via the Internet to search for intelligent life beyond Earth. This implementation’s success demonstrated the viability of using the Internet as a host for grid computing applications. Concurrent with this project, others were also developing their own variants of cloud computing [9]. Salesforce.com introduced one of the first practical cloud computing based services, including storage, computation, and even human intelligence through the Amazon [4] Mechanical Turk. It followed up this accomplishment in 2006 with its Elastic Compute Cloud (E2C) service, which provides a commercial service through which users can rent computers and run their own applications. AT&T [6] also entered the cloud computing realm when it acquired US internetworking (USi) in 2006. USi was an application service provider for more than 30 countries. In 2008, AT&T introduced Synaptic, which combined USi’s five Internet data centers in the US, Europe, and Asia to serve as regional gateways within its cloud. Today, the latest example of cloud computing is Web 2.0; Google, Yahoo, Microsoft, and other service providers now offer browser-based enterprise service applications. Now that cloud computing [5] has emerged as a viable and readily available platform, many users from disparate Backgrounds are sharing virtual machines to perform their daily activities. This environment requires an implicit level of trust as well as an explicit level of vigilance to ensure success. Implementations in 1999 and established the concept of delivering enterprise services through a Web site. In 2002, Amazon Web Services launched a suite of cloud network.

3 SECURITY ISSUES

While cloud computing is much attractive because of its flexibility and cost effectiveness, certain challenges must be addressed in order to provide a viable option to traditional data services. First and foremost is the issue of security. Data is travelling over the Internet and is stored in remote locations. In addition, cloud providers often serve multiple customers simultaneously. All of this may raise the scale of exposure to possible breaches, both accidental and deliberate.

This research work shows no compromise with security issue. The data encrypted in this research work is completely safe as used in traditional way, because the security algorithms are not altered in any manner. The working pattern of security algorithms is completely same when using on local resources and cloud resources.

Cloud services are safe option in term of security to get the IT services. Data needs to be encrypted at all times, with clearly defined roles when it comes to who will be managing the encryption keys. For a highly confidential data to be encrypted and keep it truly secure on cloud provider’s storage, the most trusted way for client is to own and manage the data encryption keys.

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

4 CRYPTOGRAPHIC ALGORITHMS

4.1 Message Digest 5 (MD5)

MD5 is a message digest algorithm developed by Ron Rivest. MD5 actually has its roots in a series of message digest algorithms, all developed by Rivest. The original message digest algorithm was called as MD. He soon came with next version MD2, but it was found to be quite weak. Therefore, Rivest began working on MD3. This was a failure and never released. Then Rivest developed MD4. However, soon, MD4 was also found to be weak. Consequently, Rivest released MD5.

In cryptography, MD5 (Message-Digest algorithm 5) [11] is a widely used cryptographic hash function with a 128-bit hash value. Specified in RFC 1321, MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. However, it has been shown that MD5 is not collision resistant; as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property. An MD5 hash is typically expressed as a 32-digit hexadecimal number.

4.2 Advanced Encryption Standard (AES)

In cryptography, the Advanced Encryption Standard (AES) [12] is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES).

According to its designers, the main features of AES are as follows:

1) It gives the implementers of the algorithm a lot of flexibility. It also stands up well against cryptanalysis attacks.
2) The algorithm works well with modern processors (Pentium, RISC, Parallel).
3) The algorithm can work well with smart cards.

4.3 RSA Algorithm

In cryptography, RSA (which stands for Rivest, Shamir and Adleman who first publicly described it) [11] is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations.

The RSA algorithm was publicly described in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT; the letters RSA are the initials of their surnames, listed in the same order as on the paper.

Operation

The RSA algorithm involves three steps: key generation, encryption and decryption.

5 PROPOSED APPROACH

We will work with the following steps:

1 Create some input data samples of sizes, 2kb, 5kb, 10kb, 20kb and 50 kb.
2 Run the encryption algorithms with all input data sizes on local server using the application tool and note the observations.
3 Make a cloud server instance on application tool and then make a dynamic web project.
4 Run the encryption algorithms on cloud server with all input data sizes and note all observations again.
5 Compare the both kind of results. The performance difference will be clearly shown.

6 EXPERIMENTAL EVALUATION

6.1 Implementation

6.1.1 Running the Algorithms on Local server

The algorithms were run locally on a uni-processor system using Eclipse JEE JUNO for Java.

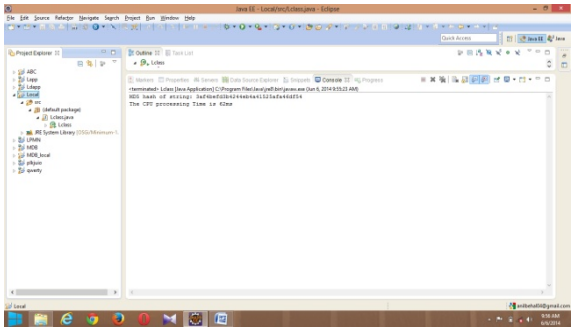

Figure 5.1: Sample output (Running the algorithms on local server)

ISSN: 2321-9653

6.1.2 Running the Algorithms on Cloud

The App Engine software development kits (SDKs) for Java and Python [10] each include a web server application that emulates all of the App Engine services on the local computer. Each SDK includes all of the APIs and libraries available on App Engine. The web server also simulates the secure sandbox environment, including checks for attempts to access system resources disallowed in the App Engine runtime environment.

Each SDK also includes a tool to upload applications to App Engine. Once the application's code, static files and configuration files are created, the data can be uploaded. The tool prompts for Google account email address and password.

When a new major release of an application that is already running on App Engine is built, it can be uploaded as a new version. The old version will continue to serve users until a switch is made to the new version. One can test the new version on App Engine while the old version is still running.

The Administration Console is the web-based interface for managing applications running on App Engine. It can be used to create new applications, configure domain names, change versions applications while they are live, examine access and error logs, and browse an application's datastore.

6.1.2.1 Creating a Project

App Engine Java applications use the Java Servlet standard for interacting with the web server environment [10]. An application's files, including compiled classes, JARs, static files and configuration files are arranged in a directory structure using the WAR standard layout for Java web applications. We can use any development process to develop web servlets and produce a WAR directory. (WAR archive files are not yet supported by the SDK.)

6.1.2.2 The Project Directory

A subdirectory named src/ contains the Java source code, and a subdirectory named WEB-INF/ contains the app cofiurations, JSPs, images, data files etc. And other subdirectory META-INF/ contains other configurations. The complete project directory looks like this:

```
Name/
  src/
    ...Java source code...
    META-INF/
      ...other configuration...
    WEB-INF/
      ...app configuration...
    ...JSPs, images, data files...
    lib/
      ...JARs for libraries...
    classes/
      ...compiled classes...
```

Using Eclipse, a new project can be created by clicking the New Web Application Project button in the toolbar.

6.1.2.3 The Servlet Class

App Engine Java applications use the Java Servlet API to interact with the web server. An HTTP servlet is an application class that can process and respond to web requests. This class extends either the javax.servlet.GenericServlet class or the javax.servlet.http.HttpServlet class.

6.1.2.4 The Web.xml File

When the web server receives a request, it determines which servlet class to call using a configuration file known as the "web application deployment descriptor." This file is named web.xml, and resides in the WEB-INF/ directory. WEB-INF/ and web.xml are part of the servlet specification.

6.1.2.5 The App Engine-Web.xml File

App Engine needs one additional configuration file to figure out how to deploy and run the application. This file is named appengine-web.xml, and resides in WEB-INF/ alongside web.xml. It includes the registered ID of your application (Eclipse creates this with an empty ID for to be filled in later), the version number of application, and lists of files that ought to be treated as static files (such as images and CSS) and resource files (such as JSPs and other application data).

6.1.2.6 Running the Project

The App Engine SDK includes a web server application to test the application. The server simulates the App Engine environment and services, including sandbox restrictions, the datastore, and the services.

ISSN: 2321-9653

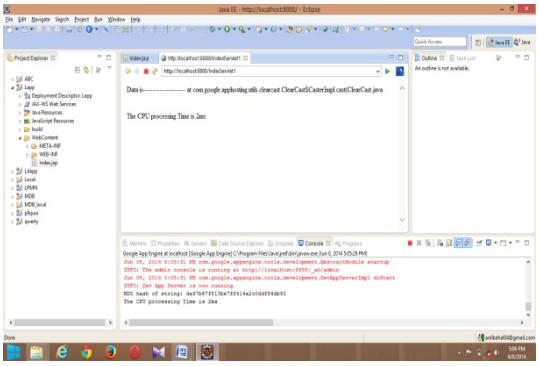

Figure 5.2: Sample Output (Running the algorithms on AppEngine)

6.2 Results

Each of the afore-mentioned algorithms was run locally as well as on cloud. Also, each one was run on different input sizes: 2kb, 5kb, 10kb, 20kb and 50kb. The comparison between local (uni-processor) running time and running time on the cloud was done by calculating the *Speed-Up Ratio*. Speed-Up Ratio is defined as the ration of mean processing time on a single processor to the mean processing time on the cloud.

Each algorithm was run multiple times with each input size and the mean value was used for calculations in each case. In the following tables and figures we are showing individual performance of each algorithm on data of different input sizes and after that we are showing the speed up ratio of these algorithms among themselves.

TABLE 6.1: A Comparison of Mean Processing Time of the Three Algorithms on the Cloud (Appengine) and on a Single Processor (Local) for Different Input Sizes

Input Size	RSA (L)	RSA (C)	MD5 (L)	MD5 (C)	AES (L)	AES (C)
2kb	678.4	380.2	15.6	0.7	425	2.3
5kb	747.3	390.2	15.9	0.9	445.7	8.2
10kb	796.8	400.9	15.9	1	454.2	15.5
20kb	853.4	429	16	1.4	487.4	24.8
50kb	1038.6	507.6	16.3	1.7	514.4	56.1

TABLE 6.2: Speed-Up Ratio of the Three Algorithms for Different Input Sizes

Input Size	RSA	MD5	AES
2kb	1.784324	22.28571	184.7826
5kb	1.915172	17.66667	54.35366
10kb	1.987528	15.9	29.30323
20kb	1.989277	11.42857	19.65323
50kb	2.046099	9.588235	9.16934

GRAPHS SHOWING PROCESSING TIME (IN MILLISECONDS) VS INPUT SIZE (IN KB) OF THE ALGORITHMS WHEN RUN LOCALLY AND WHEN RUN ON THE APPENGINE-THE CLOUD NETWORK

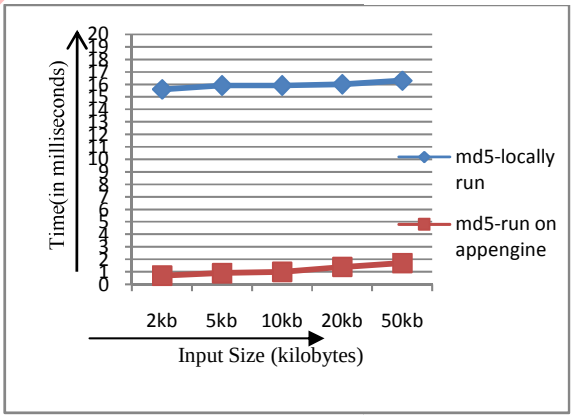

Figure 6.1: Mean processing time for MD5 running locally and running on cloud

ISSN: 2321-9653

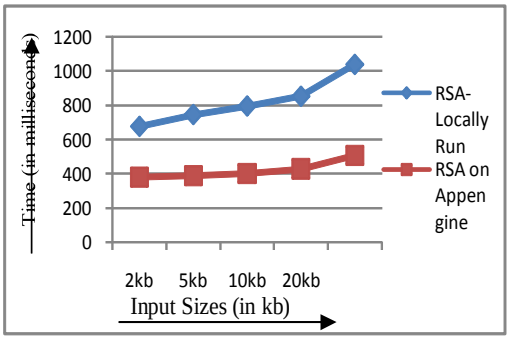

Figure 6.2: Mean processing time for RSA running locally and running on cloud

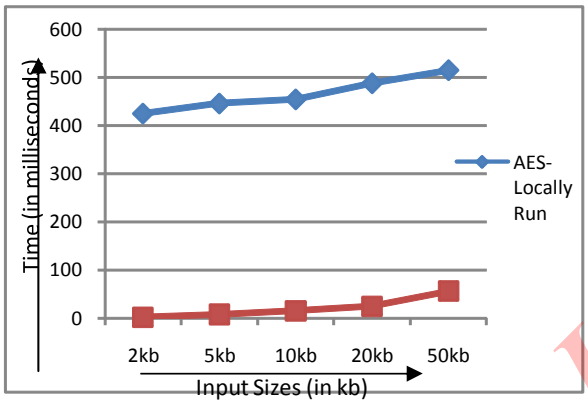

Figure 6.3: Mean processing time for AES running locally and running on cloud

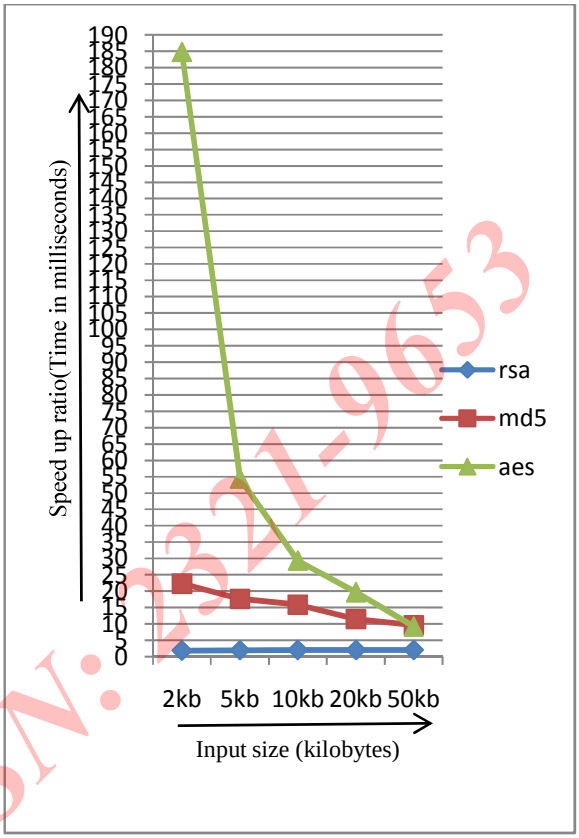

Figure 6.4: Speed-up ratio for RSA, MD5 and AES, X axis: speed up ratio, Y axis: Input size (in kilobytes)

6.3 Discussion of Results and Inferences

From the results tabulated above, the following observations and inferences can be made:

1. Amongst the algorithms RSA- an asymmetric encryption algorithm, is on an average the most time consuming and MD5- a hashing algorithm, the least. This is true in a uni-processor (local) as well as cloud (Appengine) environment.

2. The highest Speed-Up ratio is obtained in AES- a symmetric encryption algorithm for low input sizes, the Speed-Up ratio falls sharply as the input size is increased.

3 For each input size, the speed up ratio achieved is highest for AES- a symmetric encryption algorithm, followed by MD5- a hashing algorithm and the least for RSA- an

asymmetric encryption algorithm.

4 For both MD5- a hashing algorithm and AES- a symmetric encryption algorithm, the speed up ratio decreases with increase in input size whereas for RSA- an asymmetric encryption algorithm, it remains almost constant (showing a minute decrease) with increase in input size.

7. CONCLUSION

Data has to be encrypted for security purposes. There are a number of encryption algorithms which can be used to encrypt the data. When data to be encrypted is large, the process of encryption becomes too much time consuming. This paper purposes that instead of encrypting data on local machine, the cloud server can be used which is generally fast in terms of efficiency in comparison to local machine.

In support of our proposed idea, we conducted an experiment. For experiment, sample data of different sizes (2, 5, 10, 20 and 50kb) was encrypted using three encryption algorithms RSA, AES and MD5 on local machine and the cloud server and the results were compared.

The results clearly show that a cloud network can be used for faster encryption of data. Using these cryptographic algorithms on a cloud network is thus more efficient than using them on single systems. We also conclude that the speed up ratio achieved and the performance of an algorithm on a cloud network varies according to the nature of the algorithm (symmetric, asymmetric or hashing) and also with the size of the input.

By using a cloud network for encryption, organizations and individuals who earlier could not use advanced encryption algorithms due to unavailability of fast and parallel computing resources can now do so. There is no compromise with security by using these security algorithms on cloud.

REFERENCES

[1] L. Lefèvre, A. Orgerie, "Designing and evaluating an energy efficient Cloud", The Journal of Supercomputing, Volume 51, Issue 3, Pages 352 – 373, March 2010.

[2] T. Rings, G. Caryer, J. Gallop, J. Grabowski, T. Kovacikova, S. Schulz, I. Stokes-Rees "Grid and Cloud Computing: Opportunities for Integration with the Next Generation Network", Springer Netherlands, Volume 7, Issue 3, Pages 375 – 393, Aug. 2009.

[3] Vladimir Stantchev SOA and Public Services Research Group TU Berlin "Performance Evaluation of Cloud Computing Offerings", Third International conference AECAS, Pages: 187 – 192, Oct. 2009.

[4] S. Garfinkel, "An evaluation of Amazon's grid computing services: Ec2, s3 and sqs," *School for Engineering and Applied Sciences, Harvard University, Cambridge, MA, Technical Report* TR-08-07, July 2007.

[5] P. Bodık, A. Fox, M. Jordan, D. Patterson, A. Banerjee, R. Jagannathan, T. Su, S. Tenginakai, B. Turner, and J. Ingalls, "Advanced Tools for Operators at Amazon.com," in *The First Annual Workshop on Autonomic Computing*, 2006.

[6] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: the search for invariants," *SIGMETRICS Perform. Eval.Rev.*, vol. 24, no. 1, pp. 126–137, 1996.

[7] B. Ripley, "The R project in statistical computing," *The newsletter of the LTSN Maths, Stats & OR Network*, vol. 1, no. 1, pp. 23–25, 2001.

[8] Vouk, M.A. "Cloud Computing - Issues, research and implementations, *"IEEE Information Technology Interfaces 30th International Conference*, page(s): 31~40, June, 2008.

[9] I. Raicu, Y. Zhao, I. Foster, A. Szalay. "Accelerating Largescale Data Exploration through Data Diffusion", International Workshop on Data-Aware Distributed Computing 2008.

[10] Developing with Google App Engine by Eugene Ciurana, Springer.

[11] Cryptographic and Network Security: Practice and Principle, William Stallings

[12] Applied Cryptographic: Protocols, Algorithms and source code in C, Bruce Schneier.