Performance Analysis of Information Security Algorithms on Cloud computing Systems

[1]Dr Surjeet Dalal, [2]Pinky Dagar, [3]Manoj Nehra
[1]Associate Professor, Department CSE, SRM University, Haryana
[2]School Information Manager, GSSS Rohtak
[3]Associate Professor, Department CSE, E Max Green Hill Engineering College Solan

Abstract— Computations on cloud networks are increasingly being considered suitable for information security tasks by businesses worldwide. This paper studies the suggested benefits arising from the use of cloud networks, particularly for encrypting data and quantifies them using parameter processing time as the parameter. It has been shown that there are substantial speed-ups achieved by availing the services of a cloud. Results have been obtained using different input sizes to observe the effect of increasing input size on the processing time on a cloud. An experimental evaluation on Google AppEngine has been used to compare and analyze these results for three common security algorithms.

Index Terms— Cloud computing securityData encryption, Distributed applications, Performance attributes, Analysis of algorithms

I. INRODUCTION

A large part of the IT industry today depends on the potential of cloud computing. Both the applications delivered as services over the Internet, as well as the servers and system software in the data centers that provide those services are included in this concept of cloud computing. An on-demand infrastructure is used by a cloud to provide its computing resource in the form of an elastic service. Cloud users only pay for the resources allocated to them.

This paper aims to find in quantitative terms like speed-up ratio the benefits of using cloud resources for implementing security algorithms. Such algorithms are commonly used by businesses to encrypt large volumes of data. Section 2 outlines the platform used for carrying out the concerned observations (Google's AppEngine). A brief overview of the algorithms that have been analyzed is given in section 3. In section 4, experimental results and observations are reported. In Section 5 we have explained the inferences obtained from the results and Section 6 describes the future prospects of our research.

II. PLATFORM USED

Google App Engine is a platform for developing and hosting web applications in Google-managed data centers. It was first released as a beta version in April 2008. Google App Engine is cloud computing technology. It virtualizes applications across multiple servers and data centers. Other cloud-based platforms include offerings such as Amazon Web Services and Microsoft's Azure Services Platform.

Google App Engine is free up to a certain level of used resources. Fees are charged for additional storage, band-width, or CPU cycles required by the application. Compared to other scalable hosting services such as Amazon EC2, App Engine provides more infrastructure 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 re-moves 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.

Per-day and per-minute quotas restrict bandwidth and CPU use, number of requests served, number of concur-rent requests, and calls to the various APIs, and individual requests are terminated if they take more than 30 seconds or return more than 10MB of data.

III. ALGORITHMS ANALYZED

3.1 RSA

RSA is an algorithm for public-key cryptography, involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key.

3.2 MD5

Message-Digest algorithm 5, a widely used cryptographic hash function with a 128-bit hash value, processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit little endian integers); the message is padded so that its length is divisible by 512.

3.3 AES

The Advanced Encryption Standard (AES) is an encryption standard based on a design principle known as a Substitution permutation network. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

EXPERIMENTAL 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.

TABLE 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 (local)	RSA (Cloud)	MD5 (local)	MD5 (cloud)	AES (local)	AES (cloud)
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

The Mean Processing Time is calculated in milliseconds and the Input size is taken in kilobytes.

TABLE 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

IV.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 is obtained in AES- a symmetric encryption algorithm for low input sizes, the Speed-Up falls sharply as the input size is increased
3. For each input size, the speed up 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.

V.CONCLUSION

The results given clearly show that a cloud network can be used for faster encryption/decryption 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 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 and decryption, organizations and individuals who earlier could not use advanced encryption algorithms due to unavailability of fast and parallel computing resources can now do so. Implementation of quotas on data and processing time ensures that the cloud resources are not used to compromise the security of these algorithms.

VI. FUTURE WORK

In this paper we discussed one symmetric algorithm, one asymmetric algorithm and one hashing scheme. More algorithms are being analyzed and their results compared with the ones presented above to discern further and establish more concretely the effect of the nature of the algorithm (symmetric, asymmetric or hashing) on its performance characteristics on a cloud network. We are also investigating further the level of similarity between performance characteristics of an algorithm when run locally and when run on a cloud network.

ACKNOWLEDGMENT

The authors wish to thank Siddhant Goel, student of Department of Information Technology at Delhi College of Engineering, New Delhi for his technical advice on using Google's Appengine architecture.

REFERENCES

[1] L. Lefèvre , A. Orgerie “Designing and evaluating an energy efficient Cloud“

[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”

[3] Y.C. Lee, A.Y. Zomaya, “Energy efficient utilization of resources in cloud computing systems”

[4] Vladimir Stantchev SOA and Public Services Research Group TU Berlin “Performance Evaluation of Cloud Computing Offerings”

[5] X. Li, Y. Li, T. Liu, J. Qiu, F. Wang,”The Method and Tool of Cost Analysis for Cloud Computing” *IBM China Research Lab, BJ*, 100193, China

[6] 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.

[7] 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.

[8] 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.

[9] B. Hayes, “Cloud computing,” *Commun. ACM*, vol. 51, no. 7, pp. 9–11, 2008.

[10] Microsoft, “Comparing Web Service Performance: WS Test 1.1 Benchmark Results for.NET 2.0, .NET1.1, Sun One/ JWSDP 1.5 and IBM WebSphere6.0” *http://www.theserverside.net/tt/articles/content/NET2Benchmarks, 2006*.

[11]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.

[12]M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above

the Clouds: A Berkeley View of Cloud Computing," *EECS, University of California, Berkeley, The University of California, Berkeley, California*, EECS-2009-28, Feb. 2009.

[13] V. Stantchev and C. Schröpfer, "Negotiating and enforcing qos and slas in grid and cloud computing," *Advances in Grid and Pervasive Computing, ser. Lecture Notes in Computer Science*, vol. 5529. Springer, 2009, pp. 25–35.

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

[15] JV Brocke, MA Lindner, "Service portfolio measurement: a framework for evaluating the financial consequences of out tasking decisions," *Proceedings of the 2nd international conference on Service oriented computing*, 2004.