Research Result

An Improved Load Balancing Algorithm for Secure Key Exchange and Traffic-Free Data Transmission and Reception

R. Pitchandi[1], M. Sundhararajan[2], B. Kameshwaran[3]

[1]*Associate Professor and Head, Department of MCA, Madha Engineering College, Chennai-69*
[2,3]*Third Year CSE Student, Department of CSE, Madha Engineering College, Chennai-69*

ABSTRACT

There are several really serious challenges with the formulation and use of efficient legislation for load balancing in Content Delivery Networks (CDNs). The internet has a positive impact on security. To encrypt and decode the data during communication between sender and recipient, a shared key is required. Deffie-Hellman Key Exchange (DHKE), in particular, is one of the primary cryptographic mechanisms for guaranteeing network security. Security and anonymity are desired for key exchange via the Internet. This class of key exchange protocols offers a certain amount of privacy protection and forms the backbone of the development of several significant industry standards, most notably the IKE and SIGMA protocols. This outcome is then used to create a time continuous.

KEYWORDS

DHKE, DIKE, CDNs, IKE, Load balancing method, Security, SIGMA, Authenticated, Deniable

1. INTRODUCTION

The difficult task of formulating and putting into practice efficient legislation for load balancing in Content Delivery Networks is the topic of this study. Our suggestion is based on a formal analysis of a CDN system that was completed using a fluid flow model to characterize the server network. From such characterization, we formulate and demonstrate a lemma regarding the equilibrium of the network queues. This outcome is then used to create a brand-new distributed load-balancing method. The privacy feature from the top levels is preserved through a deniability service provided at the IP layer. To prevent the receiver from demonstrating to a third party that the communication came from the sender, a Deniable authentication mechanism is utilized. A communication protocol's security is predicated on one or more hypotheses. One or more cryptographic presumptions form the foundation of a key agreement procedure. A protocol with many independent assumptions and an OR logic connection is analogous to a home with numerous exterior doors equipped with various security features. If the protocol makes more independent OR related assumptions, the attacker will have more options for how to attack it. Simulations used to demonstrate the success of the suggested method in terms of both fair load distribution and limited-service time are used to validate the overall strategy. For all communication systems, including e-commerce, wireless, wired, and Internet applications, authenticated key establishment is crucial. This kind of protocol is built utilizing several cryptographic algorithms based on different cryptographic hypotheses.

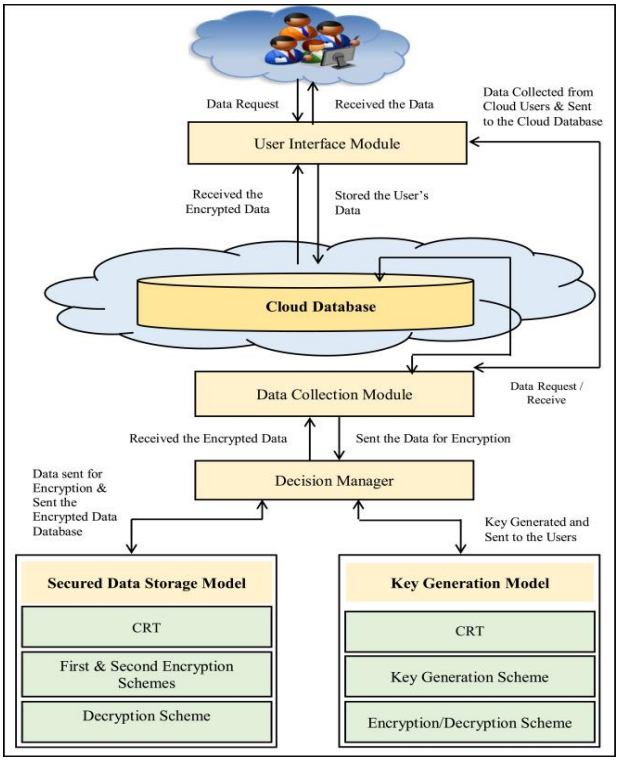

Figure 1 Cloud Storage Architecture

2. SYSTEM MODEL

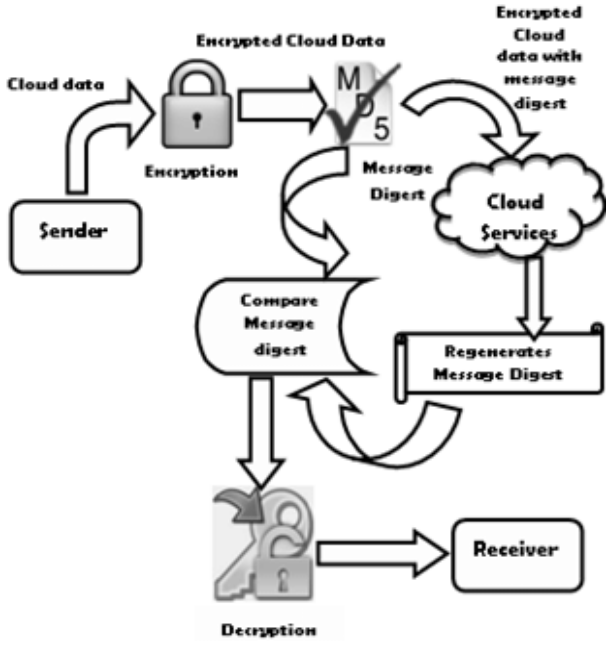

Figure 2 Encrypting Data

This gives a thorough architectural overview of the system and illustrates various system components utilizing a variety of architectural perspectives. It aims to document and communicate the key system architectural choices that have been made.

3. PREVIOUS WORK

The scheduler is put right before the server in a queue-adjustment technique, following the queue. Depending on the condition of the system queues, the scheduler may allocate the request that was taken from the queue to either a local server or a distant server. In a rate-adjustment paradigm, the scheduler is situated right before the local queue and determines whether to add a new request to the local queue or send it to a distant server as it arrives. The scheduler is permitted to manage both the local queue length and the pace of incoming requests at a node in a hybrid adjustment load balancing method. Therefore, in current systems, a CDN server can, upon receiving a new request, either elaborate the request locally or divert it to other servers by a certain decision rule, which is based on the state information shared by the servers. Such a strategy restricts the overhead of state swapping to just local servers.

3.1 Demerits of Previous Work

The request routing mechanism is a crucial part of CDN design. Based on a predetermined set of criteria, it enables the proper server to receive requests from users for content. The proximity principle, which states that the server nearest to the client will always fulfill a request, can occasionally go wrong. To deliver the greatest performance in terms of time of service, latency, etc., the routing procedure connected with a request may take into consideration several factors (including traffic load, bandwidth, and servers' processing capacity).

To avoid impacting how other users perceive the quality of the service, an efficient request routing mechanism should also be able to handle brief and sometimes localized high request rates (the so-called flash crowds). To avoid impacting how other users perceive the quality of the service, an efficient request routing mechanism should also be able to handle brief and sometimes localized high request rates (the so-called flash crowds).

4. PROPOSED METHODOLOGY

Similarly, to that, we construct appropriate load-balancing legislation in this study that ensures queue equilibrium in a balanced CDN by employing a fluid flow model for the network of servers. Then, we go over the major implementation problems connected to the suggested load-balancing technique. Using shared keys, a more powerful kind of deniability may be accomplished. Not a thorough approach to huge data. Processing incrementally is inefficient. The request routing mechanism is a crucial part of CDN design. Based on a predetermined set of criteria, it enables the proper server to receive requests from users for content.

We provide a brand-new method for rerouting client requests to the most suitable server and balancing the volume of requests arriving into the system as a whole. To accomplish global balance, our approach makes use of local balancing. This is accomplished by the system nodes periodically interacting with one another. There are four ways to execute an OTcl command using the tcl instance. Their calling arguments are where they mostly diverge. The interpreter receives a string from each function, which it then uses to evaluate the string globally. If the interpreter returns TCL_OK, these methods will return to the caller. The methods will instead call terror if the interpreter returns TCL_ERROR.

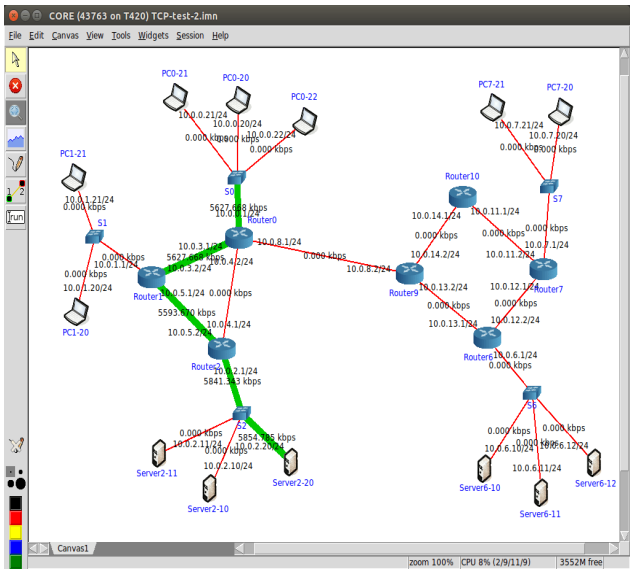

Figure 3 Network Traffic

4.1 Advantages of the Proposed System

Analyzing the performance metrics will allow you to better evaluate the quality of our product.

Additionally, the suggested method has a fantastic average response time that is only on par with the 2RC algorithm in terms of value.

Our mechanism's high performance may be compensated by a sizable number of redirections. Since all the methods under consideration use the same redirection mechanism, we only compare the proportion of requests that are sent more than once to the overall number of requests created.

utilizing secure encryption and load balancing techniques, the data sent are transported without any data loss or traffic utilizing Network Simulator.

To avoid impacting how other users perceive the quality of the service, an efficient request routing mechanism should be able to handle brief and sometimes localized high request rates (the so-called flash crowds).

5. SIMULATION

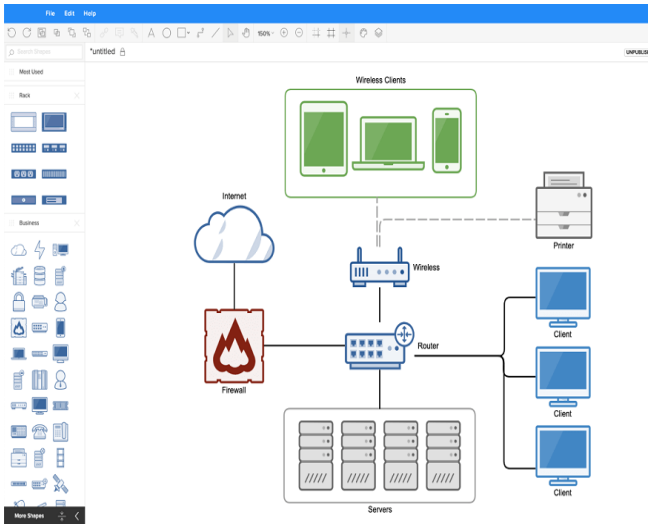

Figure 4 Network Simulation

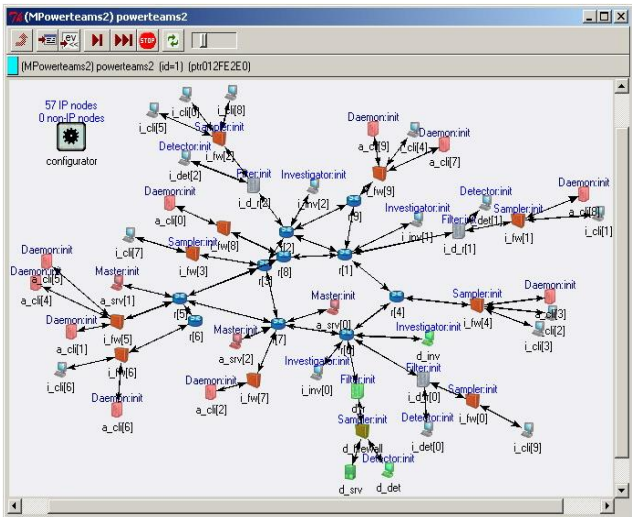

Figure 5 Traffic configuration

utilizing security encryption and load balancing techniques, the data transmitted are transported without any data loss or reduced traffic utilizing Network Simulator.

6. CONCLUSION

A new load-balancing law for collaborative CDN networks was presented. Based on a fluid flow characterization, we first defined a model of these networks. We then went on to the definition of an algorithm that tries to provide load balancing in the network by eliminating local conditions that cause queue instability through the transfer of potentially surplus traffic to the set of servers that are neighbors of the clogged server.

The method is initially described in its time-continuous formulation, and then it is presented in a discrete variant created especially for its implementation and deployment in an operational environment.

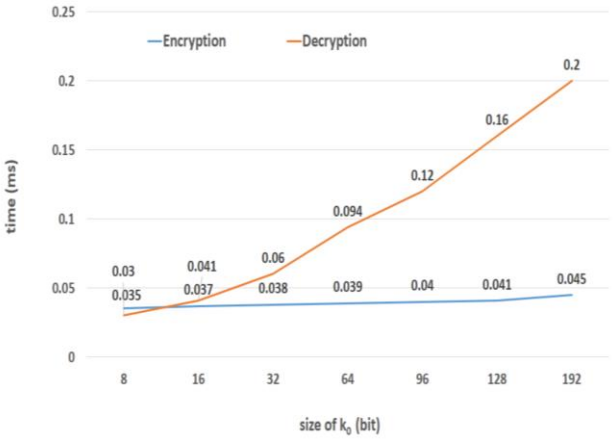

Figure 6 Encryption & Decryption Loss and Time Efficiency

We showed the scalability and performance of our idea, which exceeds the majority of the potential alternatives that have been put up in the past, with the use of simulations. The current study serves as our first step towards realizing a comprehensive load-balancing solution in a collaborative, distributed setting. To create a prototype of a load-balanced, cooperative CDN network that can be used as a proof-of-concept implementation of the simulation results and as a testing ground for additional research in the more general field of content-centric network management, our future work will be devoted to actually implementing our solution in a real system.

7. FUTURE WORK

This work offers a safe and effective method of establishing security between the sender and the receiver such that any data sent by the sender is protected against many sorts of attacks, including spoofing, identity theft, and desynchronization assaults. In comparison to the sign-then-encrypt signature followed by the encryption approach, encryption offers a reduced message size and faster processing performance. Non-repudiation is achievable with Signcryption because it uses asymmetric cryptography, as opposed to safeguards that rely on symmetric keys. Sometimes the key is impossible to reach the destination point owing to network traffic and packet loss. At that point, we may use the Graph Theory approach to determine another path and resend the receiver's key.

REFERENCES

[1] R. Pitchandi, H. Jagadeesan, "An Optimized Secure Key Exchange and Traffic Free Data Transmission and Reception with Load Balancing Algorithm," *IJSPR - Issue 69, Volume 24, Number 05*, pp. 205-208, June. 2016.

[2] H. Yin, X. Liu, G. Min, and C. Lin, "Content delivery networks: A Bridge between emerging applications and future IP networks," *IEEE Netw.*, vol. 24, no. 4, pp. 52–56, Jul.–Aug. 2010.

[3] J. D. Pineda and C. P. Salvador, "On using content delivery networks to improve MOG performance," *Int. J. Adv. Media Commun.*, vol. 4, no. 2, pp. 182–201, Mar. 2010.

[4] D. D. Sorte, M. Femminella, A. Parisi, and G. Reali, "Network delivery of live events in a digital cinema scenario," in *Proc. ONDM*, Mar. 2008, pp. 1–6.

[5] Akamai, "Akamai," 2011 [Online]. Available: http://www.akamai. com/index.html

[6] Limelight Networks, "Limelight Networks," 2011 [Online]. Available: http://. uk.llnw.com

[7] Coral, "The Coral Content Distribution Network," 2004 [Online]. Available: http://www.coralcdn.org

[8] Network Systems Group, "Projects," Princeton University, Princeton, NJ, 2008 [Online]. Available: http://nsg.cs.princeton.edu/projects

[9] A. Barbir, B. Cain, and R. Nair, "Known content network (CN) request- routing mechanisms," IETF, RFC 3568 Internet-Draft, Jul. 2003 [Online]. Available: http://tools.ietf.org/html/rfc3568

[10] T. Brisco, "DNS support for load balancing," IETF, RFC 1794 Internet-Draft, Apr. 1995 [Online]. Available: http://www.faqs.org/rfcs/ rfc1794.html

[11] M. Colajanni, P. S. Yu, and D. M. Dias, "Analysis of task assignment policies in scalable distributed Web-server systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 9, no. 6, pp. 585–600, Jun. 1998.

[12] D. M. Dias, W. Kish, R. Mukherjee, and R. Tewari, "A scalable and highly available Web server," in *Proc. IEEE Comput. Conf.*, Feb. 1996, pp. 85–92.

[13] C. V. Hollow, V. Misra.D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. Autom. Control*, vol. 47, no. 6, pp. 945–959, Jun. 2002.

[14] S. Manfredi, F. Oliviero, and S. P. Romano, "Distributed management for load balancing in content delivery networks," in *Proc. IEEE GLOBECOM Workshop*, Miami, FL, Dec. 2010, pp. 579–583.