IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP107-112
IMPACT FACTOR 1.719
www.iord.in

"Attribute Based Encryption with Verifiable delegation in cloud storage."

Mr. Pankaj Joshi

[1](PG Student,VITNagpur)

Abstract- ***Many application domains, such as real-time financial analysis, e-healthcare systems, sensor networks, are characterized by continuous data streaming from multiple sources and through intermediate processing by multiple aggregators. Keeping track of data provenance in such highly dynamic context is an important requirement, since data provenance is a key factor in assessing data trustworthiness which is crucial for many applications. Provenance management for streaming data requires addressing several challenges, including the assurance of high processing throughput, low bandwidth consumption, storage efficiency and secure transmission. In this paper, we propose a novel approach to securely transmit provenance for streaming data (focusing on sensor network) by embedding provenance into the interpacket timing domain while addressing the above mentioned issues. As provenance is hidden in another host-medium, our solution can be conceptualized as watermarking technique. However, unlike traditional watermarking approaches, we embed provenance over the interpacket delays (IPDs) rather than in the sensor data themselves, hence avoiding the problem of data degradation due to watermarking. Provenance is extracted by the data receiver utilizing an optimal threshold-based mechanism which minimizes the probability of provenance decoding errors. The resiliency of the scheme against outside and inside attackers is established through an extensive security analysis. Experiments show that our technique can recover provenance up to a certain level against perturbations to inter-packet timing characteristics.***

Keywords— ***(Existing System,proposed System,Technology used, Result Analysis)***

1. INTRODUCTION

The application will be doctor appointment application, in this application receptionist will be able create new patient in account and the doctor will be able to check name and all the other things of the patient and when he will check the patient he will be able to make some changes in the particular patient information, he can add prescription of medicine and blood test request and all. This can all information can be transferred to the medical store and pathology laboratory.

THE emergence of cloud computing brings a revolutionary innovation to the management of the data resources. Within this computing environments, the cloud servers can offer various data services, such as remote data storage [1] and outsourced delegation computation [2], [3], etc. For data storage, the servers store a large amount of shared data, which could be accessed by authorized users. For delegation computation, the servers could be used to handle and calculate numerous data according to the user's demands. As applications move to cloud computing platforms, ciphertext-policy attribute-based encryption (CP-ABE) [4], [5] and verifiable delegation (VD) [6], [7] are used to ensure the data confidentiality and the verifiability of delegation on dishonest cloud servers. Taking medical data sharing as an example (see Fig. 1), with the increasing volumes of medical images and medical records, the healthcare organizations put a large amount of data in the cloud for reducing data storage costs and supporting medical cooperation. Since the cloud server may not be credible, the file cryptographic storage is an effective method

2. Existing Systems

Existing systems which are used in hospitals for communication are traditional where they don't use any softwares to have track of the patients. Hospitals uses hard

IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP107-112
IMPACT FACTOR 1.719
www.iord.in

copies of papers where they store the records of their patients and it is hard to find out their particular patients.

Some Hospitals uses the desktop based softwareto keep records, which is not perfect to have cloud based records of the patients. Using this software runtime data can flow from receptionist, doctor, pathology, medicals etc.

Even the smallest hospitals are massive operations that service hundreds, if not thousands of patients in both inpatient and outpatient services. Each of these patients has a chart, paperwork and billing information. All this information can be difficult for hospital administrators to manage, particularly as it involves bringing together and collecting data from healthcare departments (such as doctors, nurses, and pharmacists) and administrative departments (such as billing). In the past, this data was reported using paper (either printouts or even handwritten charts) and organized manually. As you can imagine, this took a lot of time and effort not to mention a lot of administrative employees.

3. Proposed System

Proposed System will be generalized system to manage the hospital where everything will be managed from the application from the entry of the patient to the hospital to everything. As Patient will get enter to the hospital, He will go to the receptionist, she/he will make entry of the patient with all of his details, like patient information with age, problem,etc this information will go to the doctor. Then doctor will check to the patient and will add some of his prescriptions there and this will get to the medical store or to the pathology lab according to the doctors guidance. Then Patient will have to go to the pathology lab or medical store and then He will get all the things carried out in medical store or in pathology lab.

Technology Used

Node :

Node.js is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML, to be run client-side by a JavaScript engine in the user's web browser. Node.js enables JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js has become one of the foundational elements of the "JavaScript everywhere" paradigm,[5] allowing web application development to unify around a single programming language, rather than rely on a different language for writing server side scripts.

Though .js is the conventional filename extension for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in Web applications with many input/output operations, as well as for real-time Webapplications (e.g., real-time communication programs and browser games).

The Node.js distributed development project, governed by the Node.js Foundation, is facilitated by the Linux Foundation's Collaborative Projects program.

Node.js was originally written by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web.The initial release supported only Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored by Joyent.

Dahl was inspired to create Node.js after seeing a file upload progress bar on Flickr. The browser did not know how much of the file had been uploaded and had to query the Web server. Dahl desired an easier way.

Dahl criticized the limited possibilities of the most popular web server in 2009, Apache HTTP Server, to handle a lot of concurrent connections (up to 10,000 and more) and the most common way of creating code (sequential programming), when code either blocked the

IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP107-112
IMPACT FACTOR 1.719
www.iord.in

entire process or implied multiple execution stacks in the case of simultaneous connections.

Dahl demonstrated the project at the inaugural European JSConf on November 8, 2009. Node.js combined Google's V8 JavaScript engine, an event loop and a low-level I/O API. The project received a standing ovation.

Angular Js :

AngularJS (commonly referred to as "Angular.js" or "AngularJS 1.X") is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, the framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. In 2014, the original AngularJS team began working on Angular (Application Platform).

The AngularJS framework works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScriptvariables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSONresources.

According to JavaScript analytics service Libscore, AngularJS is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and approximately 12,000 other sites out of 1 million tested in October 2016.[3] AngularJS is the 10th most starred project of all time on GitHub.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic.[5] The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit DOM manipulation with the goal of improving testability and performance.

AngularJS's design goals include:

- to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.
- to decouple the client side of an application from the server side. This allows development work to progress in parallel, and allows for reuse of both sides.
- to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

Angular implements the MVC pattern to separate presentation, data, and logic components.[6] Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced.

HTML :

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered

IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP107-112
IMPACT FACTOR 1.719
www.iord.in

page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997

CSS :

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

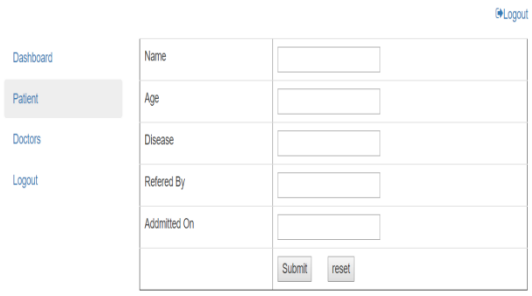
CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

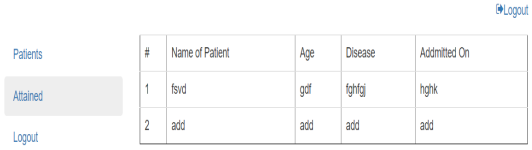
Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

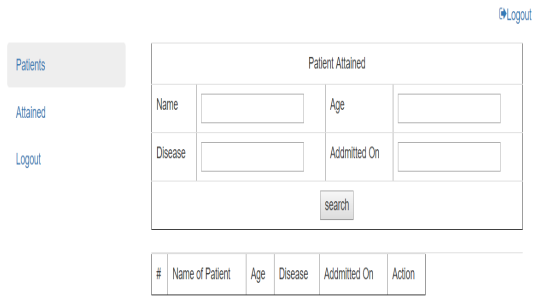
The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

Result Analysis



IORD Journal of Science & Technology
E-ISSN: 2348-0831
Volume 3, Issue 3 (MAR-APRIL 2017) PP107-112
IMPACT FACTOR 1.719
www.iord.in





References

1. Mazhar Ali, Saif U. R. Malik,Samee U. Khan," DaSCE: Data Security for Cloud Environment with Semi-Trusted Third Party", IEEE Transaction on journal name, manuscript ID IN 2015.

2. Seung-Hyun Seo, Mohamed Nabeel, Xiaoyu Ding, and Elisa Bertino," An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds", IEEE Transaction on knowledge and data engineering VOL. 26, NO. 9, SEPTEMBER 2014.

3. Ning Cao, Cong Wang, Ming Li, Kui Ren and Wenjing Lou," Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data", IEEE Transaction on parellel and distributed system, VOL. 25, NO. 1, JANUARY 2014.

4. Luca Ferretti, Michele Colajanni, and Mirco Marchetti," Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases" IEEE Translation on parallen and distributed system, VOL. 25, NO. 2, FEBRUARY 2014.

5. Khoi-Nguyen Le-Huu, Diem Ho, Anh-Vu Dinh-Duc," Towards a RISC Instruction Set Architecture for the32-bit VLIW DSP Processor Core" ", IEEE Transaction on knowledge and data engineering VOL. 24, NO. 2, NOVEMBER 2014.

6. Ayad F. Barsoum and M. Anwar Hasan," Provable Multicopy Dynamic Data Possession in Cloud Computing Systems" IEEE transaction on informations forensics and security, VOL. 10, NO. 3, MARCH 2015.

7. Arjun Kumar , Byung Gook Lee, HoonJae Lee", Secure Storage and Access of Data in Cloud Computing" 978-1-4673-4828-7/12/$31.00 ©2012 IEEE.

8. Jianbing Ni, Yong Yu, Yi Mu, Qi Xia", On the Security of an Efficient Dynamic

Auditing Protocol in Cloud Storage" IEEE on parallel and distributed system, VOL. 25, NO. 10, OCTOBER 2014.

9. Sean Thorpe, Tyrone Grandison, Arnett Campbell, Janet Williams, Khalilah Burrell," Towards a Forensic-based Service Oriented Architecture Framework for Auditing of Cloud Logs" IEEE transaction on information forensics and security ,January 2013.

10. Shichao Guan, Robson Eduardo De Grande, and Azzedine Boukerche," A Multi-layered Scheme for Distributed Simulations on the Cloud Environment" IEEE Transactions on Cloud Computing, april 2015 .

11. Luca Ferretti, Michele Colajanni, and Mirco Marchetti," Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases" IEEE transaction on parallel and distributed system, VOL. 25, NO. 2, FEBRUARY 2014 .

12. Cong Wang, , Kui Ren, Senior , and Jia Wang, "Secure Optimization Computation Out sourcing in Cloud Computing: A Case Study of Linear Programming" IEEE Transactions on Computers, may 2015.

13. Kirtiraj Bhatele, Prof Amit Sinhal, ProfMayankPathak," A Novel Approach to the Design of a New Hybrid Security Protocol Architecture" IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) 2012.

14. Ramgovind S, Eloff MM, Smith E," The Management of Security in Cloud Computing" ©2010 IEEE.

15. Weiwei Jia‡, Haojin Zhu†, Zhenfu Cao, Lifei Wei, Xiaodong Lin," SDSM: A Secure Data Service Mechanism in Mobile Cloud Computing" IEEE worshop on Security in Computers, Networking and Communications 2011.

16. Farzad Sabahi," Cloud Computing Security Threats and Responses"IEEE conference on cloud comuting , 2011.

17. Ling Li Lin Xu Jing Li Changchun Zhang," Study on the Third-party Audit in Cloud Storage Service" IEEE 2011 International Conference on Cloud and Service Computing.

18. Daojing He, Sammy Chan, and Mohsen Guizani," Accountable and Privacy-Enhanced Access Control in Wireless Sensor Networks" IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 14, NO. 1, JANUARY 2015.