Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 4, April 2017, pg.13 – 16

API Development for On-Demand Content Delivery

Nandita Krishnamurthy, Heena Nagaonkar, Rohit Patil, Karan Sahu
K.K.Wagh Institute of Engineering Education & Research, Savitribai Phule Pune University, Nashik
nanditakrishnamurthy7@gmail.com, nagaonkarheena786@gmail.com, thisisrohit009@gmail.com karansahu.sahu@gmail.com

Abstract— Application program interface (API) is a set of routines, protocols and tools for building software application. An API specifies how software component should interact with each other. A good API makes it easier to develop a program by providing all the building blocks. In existing systems, APIs are used by various applications to access and communicate information and data. Major problems faced by these applications are regarding the security of contents provided by the API. Different surveys show that encryption and decryption algorithm used by API is not efficient. Earlier API didn't deal with cloud data. Our system aims to provide better security using efficient encryption and decryption algorithm for accessing huge content repository. It will allow fetching of text & audio content and processing it along with animation to integrate E-learning content. The work includes developing licensing, content protection, tracking usability, billing and analytical reports. The system will act as an epicentre for content delivery, access as well as efficient, protected and monitored usage.

Keywords— AP, Compression, Decompression, Decryption, Encryption, On-Demand, Tracking, Billing.

I. INTRODUCTION

We propose an API which will integrate and provide e-learning content as per the demand of user. And moreover provides additional features like tracking, billing, content protection, analytical report generation etc.

The idea of the system is to prepare an API which delivers the E-learning content according to the user's request. This E-learning content is basically swf (i.e. Shock Wave Flash) animation and audio.

First user authentication is carried out with license key and with its successful validation the content is delivered in encrypted and compressed from. Due to encryption, the content can only be played on embeddable player of the API. The embeddable player at the client side converts the content in its original form and lets the client view it. The system will keep track of the usage and will generate analytical report accordingly and with this data it will also generate bills.

We develop a system which enhances the content delivery along with protection and authentication. Basically, it is about creating an API system which provides integrated E-learning content according to the demand of the user in such a way that the provided contents cannot be copied. API requires less bandwidth for data transmission as compression is applied.

The client will have to pay for the content exactly according to the usage. As per the content usage by the client , API will track this usage and generate analytical report accordingly. This analytical report will be available to view afterwards. The Admin views this analytical report to analyse the usage of the API and keep a track of the client and on the other hand client receives the bill generated through the tracking of its usage of the API.

Monitoring the billing module for accurate bill generation. The billing scheme is predefined by company. It would mostly be of a kind of Rs.1 Per module Per Client Per usage. Tools and interface should be designed for precise calculation of bills. The bill is then sent to the client after a certain time span. The client has to pay the said amount within the deadline to continue the usage of the software.

II. GOALS AND OBJECTIVES

[1]To introduce licensing: As no one should be able to use the content freely, the authentication of client is required to be able to use the content. Unique license key will be allotted to clients.

[2]To introduce encryption: In order to prohibit the downloaded content to be copied and played by a third party player free of cost , the content is encrypted and only embeddable player can play the content by decrypting it.

[3]To introduce tracking: The analytical information of usage of content by the client is an important information. This information can give insights of detail usage of content and will also help in billing.

III. SYSTEM ARCHITECTURE

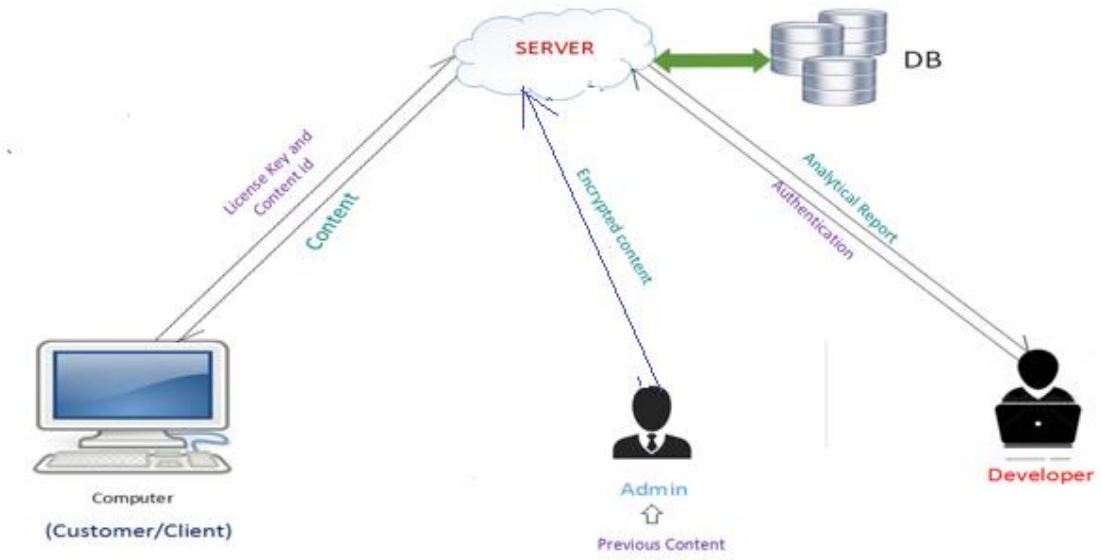

Fig. 1: Block Diagram of API System

Block diagram consists of client machine and admin machine. Admin machine log in to the server and update the new content. This content is in encrypted and compressed form. Client machine send a request to the server which include license key and content ID. After processing on request, server send particular course content in compressed and encrypted form to the requesting client application. Only client player can decompressed and decrypt the contents provided by the server and play it accordingly. Contents are compressed at server side and decompressed at client side by using G stream algorithm. The contents are encrypted at the server side and decrypted at the client side by using RSA algorithm. Admin is also able to logs in to the server and get analytical report.

IV. METHODOLOGY

[1] Licensing tool**:** It validates the license key to provide access to the player and content. The license key will be different for different users.

[2] Content delivery**:** After authentication of the user the actual content in different form is provided to the embeddable player at client side as per the demand. The content is sent in encrypted and compressed form.

[3] Encryption, Decryption**:** To avoid unauthorized access of the content, encryption should be provided which can only be decrypted in the embeddable player of API.

[4] Compression**:** Compression is done so that large size of content can be easily delivered on a small bandwidth of network.

[5] Tracking and billing**:** Tracking is used to keep track of the usage of contents made by client. The billing is done on the basis of tracking only. Billing is done on the basis of use of per module per user per unit of time. (for example Rs. 1 for use each module by each user one time).

[6] Analytical report generation **:**-Finally analytical report is generated which will show the full analysis of the usage of contents by client. This will be helpful to the admin.

V. Algorithm

Encryption and decryption algorithm:

Advanced Encryption Standard (AES) Algorithm:
The Advanced Encryption Standard (AES), also known by its original name Rijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST).

The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. AES is based on a design principle known as a substitution permutation network, a combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits. AES operates on a 4 4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a particular finite field.

AES achieves the goal of being both secure and practical for real systems. The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data the data to be encrypted. This array we call the state array.

Following are the aes steps of encryption for a 128-bit block:
1. Derive the set of round keys from the cipher key.
2. Initialize the state array with the block data (plaintext).
3. Add the initial round key to the starting state array.
4. Perform nine rounds of state manipulation.
5. Perform the tenth and final round of state manipulation.
6. Copy the final state array out as the encrypted data (ciphertext).

The order of operation in decryption is:

Perform initial decryption round:
1. XorRoundKey
2. InvShiftRows
3. InvSubBytes

Perform nine full decryption rounds:
1. XorRoundKey
2. InvMixColumns

3. InvShiftRows
4. InvSubBytes
5. Perform final XorRoundKey

VI. CONCLUSION

The outcome of the API is nothing but the contents which the user has requested.
These contents will be fetched from the database (may be from single module or different modules as per the demand of user) then it will be integrated and delivered to the user. The delivered content will be of swf along with animation and audio in encrypted and compressed form. The major output of the API is the integrated E-learning content. Besides this API will also provide billing on the basis of tracking the usage of contents and finally analytical report is generated which shows the usage of contents by each user. An admin can refer this report to analyze the overall use of contents by each user. Thus for our project the input is API request and output should be particular module requested by user.

REFERENCES

[1] Juneyoung Park and Mun Y. Yi , Graph-based Retrieval Model for Semistructured Data, IEEE , 6-5-2016.

[2] Yang Luo, Hongbo Zhou, Qingni Shen, AnbangRuan, Zhonghai Wu Peking University, China, RestPL: Towards a Request-Oriented Policy Language for Arbitrary RESTful APIs, 2016 IEEE International Conference on Web Services, 5.3.2016.

[3] Moon Soo Cha, So Yeon Kim, Jae HeeHa, Min-June Lee, Young-June Choi, Kyung-Ah Sohn , CBDIR: Fast and Effective Content Based Document Information Retrieval System , 2015 IEEE ICIS 2015, June 28-July 1 2015

[4] NadeemAbji, Ali Tizghadam and Alberto Leon-Garcia Energy Efficient Content Delivery in Service Provider Networks with Content Caching IEEE Online Conference, 09.05.2015

[5] Tukasa Ikeda, Makoto Ikeda , Comprehensive Study on Higher Order Radix SYNOPSIS RSA Cryptography Engine, IEEE, 5.5.2015

[6] LuminiaScripcariu Faculty of Electronics, Telecommunications and Information Technology A Study of Methods Used To Improve Encryption Algorithms Robustness 2015 IEEE

[7] Hua Deng , Jifu Zhang and Xiaoli Chai , The Design and Implementation of Flash Animation Watermarking, IEEE, 5.08.2014

[8] MayssaJemel and Ahmed Serhrouchni , Content protection and secure synchronization of HTML5 local storage data, IEEE, 3-7, 2012.

[9] JeffreyStylos and Brad Myers The Implications of Method Placement on API Learnability ACM, November 915, 2008

[10] Brian Ellis, Jeffrey Stylos, and Brad Myers The Factory Pattern in API Design: A Usability Evaluation 29th International Conference on Software Engineering(ICSE'07)28.07.2007

