Review of JSON Data Interchange Format

Garima Dutt[1], Anup Singh Kushwaha[2]
[1]Mtech Student, Department of Computer Science & Technology
[2]Assistant Professor ,Department of Computer Science & Technology
Manav Rachna College of Engineering, Aravalli Hills, Sector-43, Faridabad

***Abstract-* This paper will compare the XML RPC and OPTIMIZED JSON which are two data interchange formats currently used by industry applications. We need to opt for an appropriate data interchange format which can cause notable impact on the rate at which data is transmitted and its performance. Firstly, we will explain the specific languages and their use. Further then study is conducted to do the comparison of utilized resources and performance of various applications that use the data interchange formats. We find that OPTIMIZED JSON is much faster than XML and design the metrics related to resources for our results.**

***Index Terms*— Cloud-computing, JSON, meta-data, Encryption**

I. INTRODUCTION

The evolution of Data Interchange formats has been done from markup and display oriented to provide the support for encoding of meta-data that explains the structural attributes of information. The need of support for data interchange for Java applications became the reason for the development of standard interchange formats [2]. Among many data interchange formats, JSON and XML are the two data interchange formats with specific purposes. We will use OPTIMIZED JSON with XML for our comparison. Next two Sections will give the brief about JSON and XML background. Section four will describe the study and methodology which will compare performance parameters. Section five shows the results after comparison of both. Conclusion is explained in Section six and suggestion of possible work for future.

A. JSON

As web services performances are imperative and plays an important role in data transmission over the network, We need more faster mediation services to improve the performance. The traditional XML services are there in the market from a very long time but has certain drawbacks as parsing the request or sending more bytes through the internet which can be possible with lower bytes transmission. Therefore, JSON came into picture which is comparatively faster and lighter than XML. JSON (Java Script Object Notation) is a data interchange format just like XML (Extended markup Language) but light weighted and human readable. JSON is smaller, simpler with no data configuration overhead and hence considered as a FAT-FREE alternative to XML.

Example of XML is:

```
  <employee>

     <firstName>David</author>

     <lastName>Guetta</title>

      <empId>"101"</empId>

  </employee>
```

Example of JSON is:

```
{“employee”:
[{“firstName”: “Charles”,
”lastName:”Lee”}]
}
```

II. COMPRESSING JSON WITH CJSON ALGORITHM

CSJON comes with the feature for compress the JSON with automatic type extraction. To handle the problem of pressing: the need to constantly repeat key names over and over. Below an example is given using compression algorithm, the following JSON:

```
[
  { // This is a point
    "x": 200,
    "y": 200
  },
{ // This is a rectangle
    "x": 200,
    "y": 200,
    "width": 300,
    "height": 350
  },
  {}, // an empty object
]
```

Can be compressed as:

```
{
  "templates": [
    [0, "x", "y"], [1, "width", "height"]
  ],
  "values": [
    { "values": [ 1,  200, 200 ] },
    { "values": [2, 200, 200, 300, 350 ] },
    {
}
  ]
}
```

III. ENCRYPTION AND ITS TECHNIQUE

Encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor. In an encryption scheme, the intended communication information or message, referred to as plaintext, is encrypted using an encryption algorithm, generating cipher text that can only be read if decrypted. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, large computational resources and skill are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients, but not to unauthorized interceptors.

The primary purpose of encryption is to protect the confidentiality of digital data stored on computer systems or transmitted via the Internet or other computer networks. The purpose of encryption is to ensure that only somebody who is authorized to access data (e.g. a text message or a file), will be able to read it, using the decryption key.

Modern encryption algorithms play a vital role in the security assurance of IT systems and communications as they can provide not only confidentiality, but also the following key elements of security:

1. Authentication: we can verify the message’s origin.
2. Integrity: The message content will not get change since the time it was sent.

3. Non-repudiation: The denial of message cannot be done by sender.

IV. TYPES OF ENCRYPTION

B. Symmetric key encryption

In symmetric-key schemes ,both keys encryption and decryption has same keys .For both parties, receiving and sending side , both keys must stay same for secure channel.

C. Public key encryption

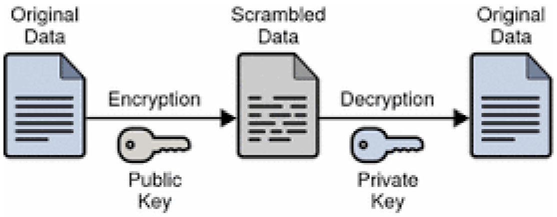

Fig (a)

In public-key encryption schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key that enables messages to be read.

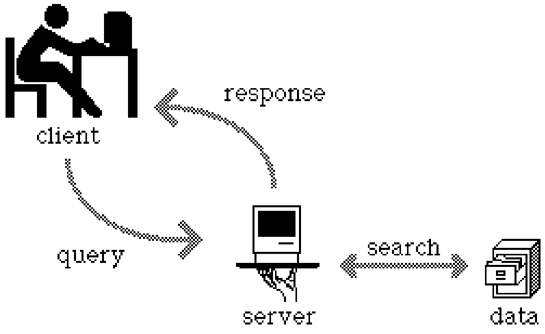

Figure (b)

Figure (a) explains when a user sends a query to the server in the cloud environment , the request in terms of data packets. For this, an encryption technique is needed which can transmit the data in a secure manner.

There are types of encryption techniques available in the market such as DES(Data Encryption Standard), TDES(Triple Data Encryption Standard), AES(Advanced Encryption Standard) and Blowfish Encryption Technique.

DES uses 56 bits key for encryption and decryption. It completes the 16 rounds of encryption on each 64 bits block of data. In all rounds, encryption is done using function F. DES have three modes of operation: ECB (Electronic Code Book), CBC(Cipher Block Chaining), CFB(Cipher Feedback) and OFB(Output Feedback). There is no strong limitation found rather than its small key size which offers less security. Brute Force attack is the only possible and successful attack on DES. Another disadvantage is that, it's encryption speed is very low.

Enhancement to DES is 3DES as an enhancement of DES, a proposed encryption standard. In 3DES, the encryption method is same as it is in original DES, but encryption key is applied three times to further enhance and increase the level of encryption. Although it is slower than other block cipher methods. Encryption strength is directly tied to key size, and 56-bit key lengths have become too small relative to the processing power of modern computers. So, 3DES use same data three times for encryption. To enhance the security of security of encrypted text, 3DES uses the DES encryption algorithm three times.

AES is actually, three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128 bits, 192 bits and 256 bits, respectively. In Advanced encryption standard there are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.

Blowfish is a symmetric key block cipher. It is one of the most common public domain encryption algorithms. Blowfish is a variable length key, 64-bit block cipher. Although, it suffers from weak keys problem, no successful known attack has happened till now.

It operates on block size 64 bits. It is a 16-round Feistel cipher and uses large key dependent S-Boxes. Each S-box contains 32 bits of data.

We opted for Blowfish Encryption Technique because it is much secure compare to others encryption techniques. Another advantage is that, blowfish is fast and easy to implement.

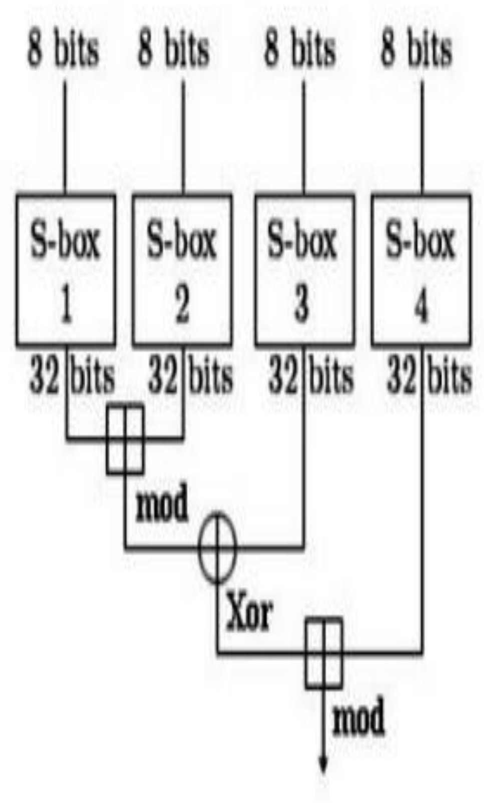

Figure (c)

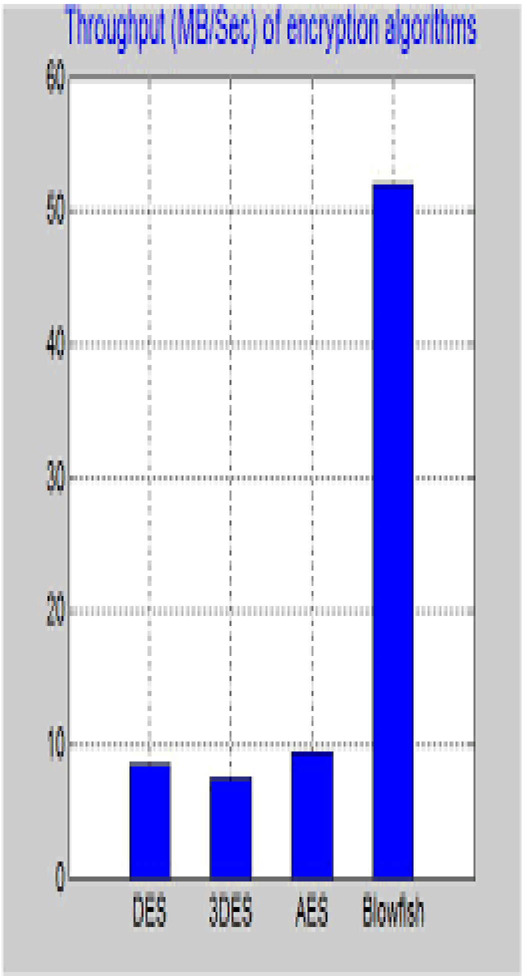

Figure (d)

Figure (d) compares all four algorithms throughput parameter in terms of performance. Throughput is calculated as request per unit time.

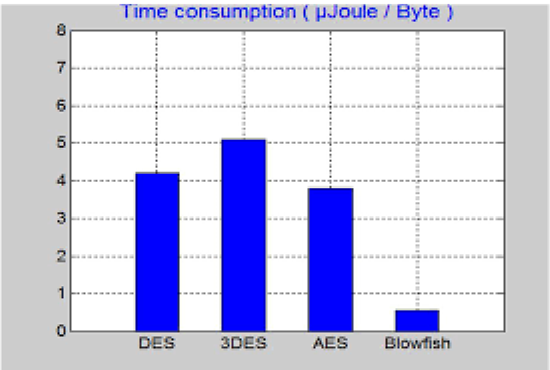

Figure (e)

Figure (c) describes all four algorithms in terms of time consumption for encryption technique.

So, it can be concluded that blowfish is better in terms of throughput and also time efficient.

V. TOOL EXPLANATION

Performance Testing is an imperative task for the web services to satisfy high load requirements. There are several tools available in the market for performance testing such as Allmon, Benerator, Apache J Meter etc. For our review and analysis work we will use Apache Jmeter tool which is designed to measure performance and to load test functional behavior. It can be used to test the strength and analyze the performance under different load types to simulate a heavy load on a object, network or servers. We can also do graphical analysis of performance or to test your server, script or object behavior under heavy load. Furthermore it is an open source software which can be easily used for Performance of and functional testing.

Apache J Meter features include:

- Ability to load and performance test many different server/protocol types:
  - Web - HTTP, HTTPS
  - SOAP / REST
  - FTP
  - Database via JDBC
  - LDAP
  - Message-oriented middleware (MOM) via JMS

VI. CONCLUSION

This paper gives a short review of the web services of JSON which is comparatively better than XML which is in the use in industry since few decades. Also, compressed JSON is an enhancement to JSON which is much more convenient. As, In Cloud Computing we require services which are better in performance and simple to understand. To provide the users end to end services over the network in cloud environment, we need to have services which are more efficient in time and with improved performance.

REFERENCES

1. Daniel Aslan , Wolfgang Thronicke, Evaluation of XML-RPC interoperability between the .NET and JAVA framework
2. Cesare Pautasso, Restful Web services:principles, patterns, emerging technologies.

3. Abdel-Karim AlTamimi, Performance Analysis of Data Encryption Algorithms
4. JSON. json.org. http://www.json.org
5. Extensible markup language (xml) 1.0 (fourth edition). W3C, 2006. http://www.w3.org/TR/2006/REC-xml-20060816
6. Alexander (2007). "JSON Pros and Cons ". Retrieved April 25, 2012,from http://myarch.com/json-pros-and-cons.
7. Sporny, M. (2010). "Web Services: JSON vs. XML." Retrieved June 02,2012, from http://digitalbazaar.com/2010/11/22/json-vs-xml/.
8. Rajdeep Bhanot and Rahul Hans, A Review and Comparative Analysis of Various Encryption Algorithms, International Journal of Security and Its Applications Vol. 9, No. 4 (2015), pp. 289-306.
9. E. Biham and A. shamir, "A differential cryptoanalysis of data encryption stamdard", Springer-verlag, **(1993)**.
10. B. Schneier, “Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)”, [online] Available at: http://www.schneier.com/paper-blowfishfse.html.
11. S. Basuin, “International data encryption algorithm (idea) – a typical illustration”, Journal of global research in computer science (JGRCS), vol. 2, no 7, **(2011)**.
12. A. Kakkar and M. L Singh, “Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network”, Published in International Journal of engg, and technology(IJET), vol. 2, no. 1, **(2012)**.
13. J. Daemen, R. Govaerts and J. Vandewalle, “Weak Keys for IDEA”, Springer-Verlag, **(1998)**.
14. M. Abutaha, M. Farajallah, R. Tahboub and M. Odeh, “Survey Paper: Cryptography Is the Science of Information Security”, published in International Journal of Computer Science and Security (IJCSS), vol. 5, no. 3, **(2011)**.
15. M. Thaduri, S. Yoo and R. Gaede, “ An Efficient Implementation of IDEA encryption algorithm using VHDL”, Elsevier, **(2004)**.
16. L. Singh and R. K. Bharti, "Comparative perfomance analysis of cyptographic algorithms", International journal of advanced research in computer science and software engineering (IJARCSSE), vol. 3, no. 11, **(2013)**.