DNS ENCRYPTION USING INNOVATIVE ALGORITHM

Harsh Savla, Vinesh Mohta, Namita Parwah[1] and Sanjay Deshmukh, Prathamesh Churi[2]

[1]Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering , SVKM's NMIMS University , Mumbai , India

[2]Assistant Professor, Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering , SVKM's NMIMS University , Mumbai , India

Sanjay.deshmukh@nmims.edu, prathamesh.churi@nmims.edu

Abstract - Domain names are always needed to abstract away details of location, authorization and human readability in World Wide Web (WWW). In the world of Internet, it becomes crucial to have naming systems to help the user select or extract the information he requires, in a form that is human readable. Many naming services are implemented, Domain Name System (DNS) being one of them. In this paper, we have suggested an innovative algorithm to encrypt the IP address of the client computers. The Jumbling-salting (JS) algorithm is used which converts the IP address to a randomized jumbled and salted form which is difficult to crack against brute force and dictionary attacks. The main focus lies on random values generated which is difficult to crack and prevents attacks on the DNS.

Keywords -Jumbling-Salting , JS, DNS , Name Servers , AES.

I. INTRODUCTION

The mapping process of IPv4 addresses to host names became a major problem in the Internet and the higher level binding effort went through different stages of development up to the currently used Domain Name System (DNS).The DNS Security is intended to provide high level security by merging the concept of Digital Signature and Asymmetric key (Public key) Cryptography. In the earlier work, the DNS security uses MD5 algorithm for compression of file along with Pseudo Random Number Generator Algorithm for generating Public/ Private Key.

DNS maps domain names to Internet Protocol addresses, and vice versa. DNS is implemented as a globally distributed database supporting a hierarchical structure. The availability and performance of DNS can be enhanced through a process of replication and caching. An effective DNS is critical to Internet access speeds. The bandwidth of your Internet connection is irrelevant if the DNS system is slow. In recent years, a number of DNS exploits have been uncovered. These exploits affect the system in such a way that an end user cannot be certain the mappings he is presented with are in fact legitimate.

The paper suggests an innovative approach for DNS encryption using Jumbling-Salting Algorithm. The Jumbling –Salting algorithm is randomized algorithm. The algorithm is compared with AES algorithm against the parameters like encryption time, decryption time , size of cipher text. The overall throughput of the algorithm is better as compared to AES algorithm.

The organization of the paper is as follows: in the section 2 , literature survey is done . In this section the concept of Recursive DNS server with working is explained. There has been a lot of research work on DNS security solutions. This paper also explains all the solutions as a review of this research. There are 10 research documents has been cited.

In Section 3, possible DNS attacks are explained. There are many DNS attacks on DNS server which are possible. Few of them are relevant for this research DNS Spoofing , DNS ID Hacking etc.

In Section 4 Jumbling-Salting algorithm is explained. The algorithm encrypts DNS IP address through the process of jumbling and salting. The algorithm is compared with the analysis parameters like Encryption time , Decryption time , Size of cipher text , throughput etc. the encryption of 10 different DNS IP addresses is done . The different classes of IP address are taken into the consideration. The results are discussed in section 5.

Section 6 concludes the encryption process of DNS with possible outcomes.

II. LITERATURE SURVEY

2.1. Recursive DNS Server

Recursive name server area unit is just like the phone operator wanting up a telephone number from multiple phone books on behalf of the requesting party (The user's PC on behalf of associate degree application), some phone books can list simply last names, then alternative phone books exist per name, and list initial names. As an example, once creating missive of invitation to an internet site from

 DOI : 10.18231/2454-9150.2018.0437

your browser, the host (computer) can then build missive of invitation to algorithmic DNS server to search out the information processing address related to the website; this can be assumptive your software and application program don't have already got a response cached. From there, the algorithmic server can check to check if it's a cached DNS record from the authoritative name server, and still features a valid time-to-live (TTL). If the algorithmic server doesn't have the DNS record cached, it begins the algorithmic method of prying the authoritative DNS hierarchy.

2.2. Working of DNS

The working of DNS is very vast and critical. The diagram shown below explains the working of DNS Server. When you type in a web address, e.g. www.xyz.com, your Internet Service Provider views the DNS associated with the domain name, translates it into a machine friendly IP address (for example 214.178.225.10 is the IP for jimsbikes.com) and directs your Internet connection to the correct website.

After you register a new domain name or when you update the DNS servers on your domain name, it usually takes about 12-36 hours for the domain name servers world-wide to be updated and able to access the information. This 36-hour period is referred to as propagation.

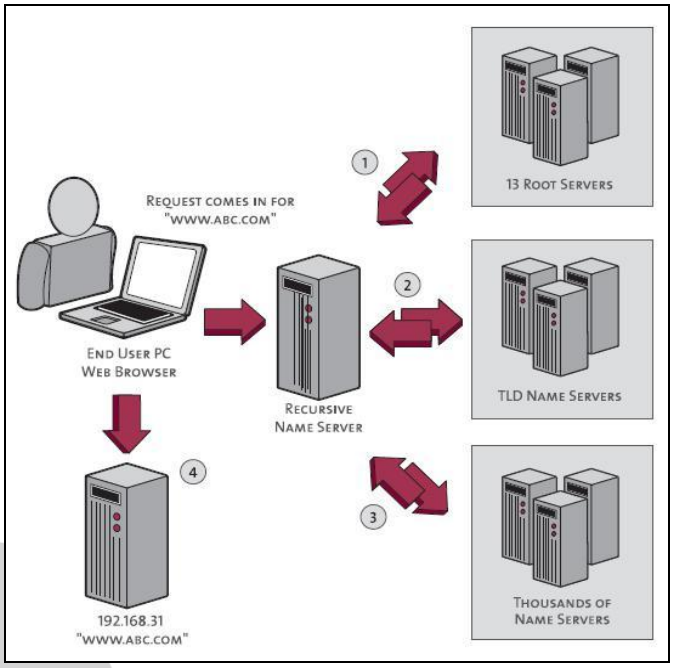

Figure 1. Working in DNS

2.3. Literature Cited

Following papers regarding DNS Security are there in the following table 1.

Table 1. Literature Survey of DNS Security

Sr no .	Paper Title and Author	Year	Inferences
1	Security System for DNS using Cryptography , Naveen Kumar Tiwari , Sanjay Khakil [1]	2015	The paper presents security implementation of domain name systems. It uses ECDSA algorithm and digital signature for encryption and secure transmission. Encryption and Key exchange process is very slow. Signature verification and Generation is very slow.
2	A new approach to DNS security (DNSSEC) , Giuseppe Ateniese , Stefan Mangard [2]	2001	DNNSSES generally uses public key cryptography The paper suggests an improvised approaches for DNS Security. The algorithm basically builds a rust level from root servers to authoritative servers.
3	DNS security introduction and requirements. [7]	2005	The review paper suggests DNS Services that DNS Security extensions must provide / not to provide.
4	Security vulnerabilities in DNS and DNSSEC. [8]	2007	The paper describes the typical vulnerabilities in implementing DNS Security. DNS Vulnerabilities such as Man in the middle attack , Transaction ID Guessing , Caching problems are discussed. The paper also explains DNSSEC with its possible vulnerabilities in network.

III. DNS PROTOCOL ATTACKS

DNS protocol attacks are based on flaws in the DNS protocol implementation the actual way DNS works on the Internet.

3.1. DNS Spoofing [3]

DNS cache poisoning relates to AN attack consisting of constructing a DNS server cache false information: sometimes, a wrong record which will map a reputation to a wrong scientific discipline address. we'll going to see that there are other ways for a hacker to try and do that, which they're usually associated with DNS spoofing. With DNS cache poisoning, the hacker can try and build a DNS answer one thing he desires for a particular request. as an example, try and build the ns.defense.gov DNS to answer with the scientific discipline of the hackers pc to any question regarding the scientific discipline of telnetaccess.defense.gov.

Poisoning are often done by connected knowledge or unrelated knowledge attacks (we can see what these area unit later), likewise as by mistreatment DNS spoofing. DNS

 DOI : 10.18231/2454-9150.2018.0437

spoofing could be a term pertaining to the action of responsive a DNS request that was supposed for an additional server (a real DNS server). this may be in a very server server exchange (a DNS server asks another for a mapping) or in a very shopper-server dialog (when a client asks a DNS server for a mapping). there's no useful distinction. The hacker spoofs the DNS servers answer by responsive with the DNS servers scientific discipline address within the packets source-address field. however this can be not enough to spoof a DNS reply. DNS uses ID range to spot queries and answer, therefore the hacker has to notice the ID the shopper is looking forward to. For that, he can use DNS ID hacking. With DNS spoofing, the hacker can try and impersonate the DNS reply in order that the requesting shopper is misdirected, however while not touching the DNS cache of the impersonated DNS.

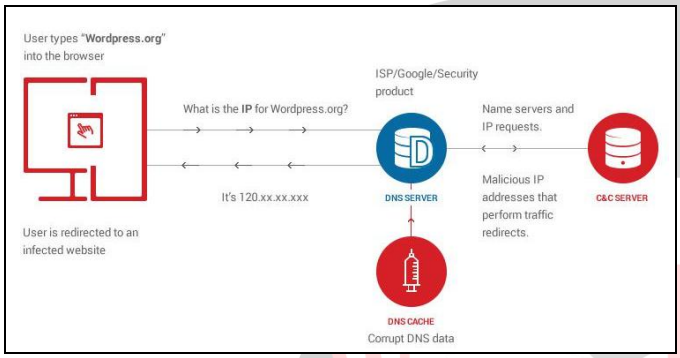

Figure 2. DNS Spoofing

3.2. DNS ID Hacking [3]

The DNS ID hacking could be a necessary technique for a hacker to reach impersonating a DNS server (this is that the basic of DNS spoofing). Indeed, to be ready to forge a faux answer, the hacker should 1st use the DNS servers IP address as a supply for his own IP packet, and so use the right ID variety while not that the consumer wont take the reply into consideration. The consumer can send a question to the DNS server employing a specific ID variety. The server can reply mistreatment an equivalent ID variety. this can be the amount the hacker should notice.

On a LAN, obtaining the ID is pretty simple: all the hacker should do is sniff the network for the initial question and answer faster than the DNS (which is incredibly easy on a LAN). The late reply from the DNS server are going to be discarded. once the hacker isn't on an equivalent local area network because the victim consumer, he has four choices to undertake to guess the ID

- Test all the possible values of the ID flag (or as many random values as you can before the NS replies): this is quite an obsolete method and quite useless since its only advantage is that it will let you know what the ID is
- Flood the DNS server to buy some more time for trying different ID numbers. The hacker can even hope it will crash the server
- Send a couple of hundred replies at constant time to extend his probabilities to search out the nice ID. The hacker will try this many times one once the opposite with completely different ranges till the server replies. Use a vulnerability within the server, knowing that variety of them simply increase the ID number from one request to a different.This works in a server-server dialog (The client in our last figure is a DNS server, and the hacker is trying to poison its cache).In that case, the hacker can first make a request to the client using a host name in a zone controlled by the hacker, and sniff the ID used by the victim DNS server.
- That way the hacker knows the range of IDs currently used by the victim server. All he has to do now is to make a request to the host name he wants to poison the cache of our victim with, and fake the answer using an increased value of the stolen ID

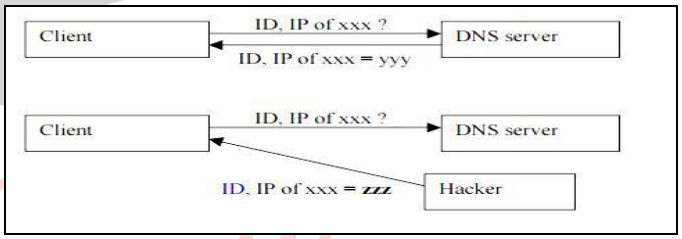

Figure 3. DNS ID Hacking

IV. JUMBLING-SALTING DNS ENCRYPTION ALGORITHM [4][5]

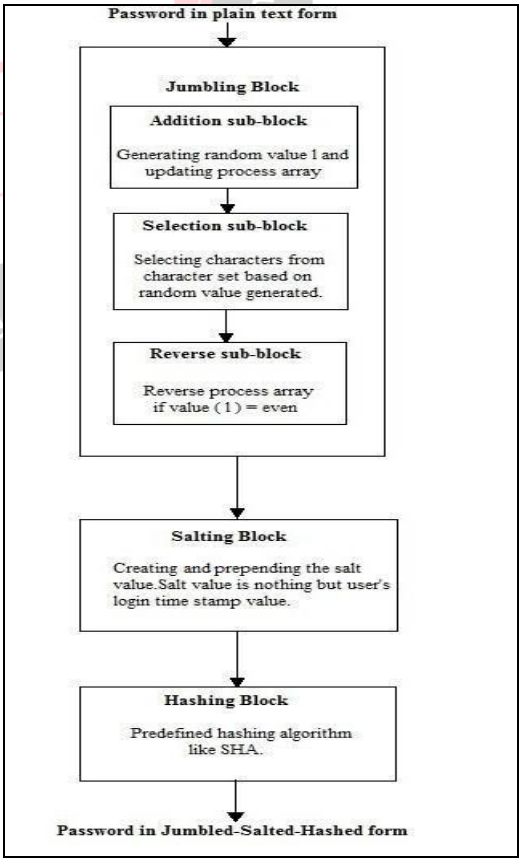

Figure 4. Block Diagram of Jumbling Salting Algorithm

 DOI : 10.18231/2454-9150.2018.0437

V. RESULT ANALYSIS

As, we mentioned JS algorithm is totally random i.e. even if same input are repeated twice the character set length will be totally random as we see in figure 9 and 10 where input is 8.8.4.4 but the random length is 11 and 19 respectively . Same goes with figure 11 and 12.

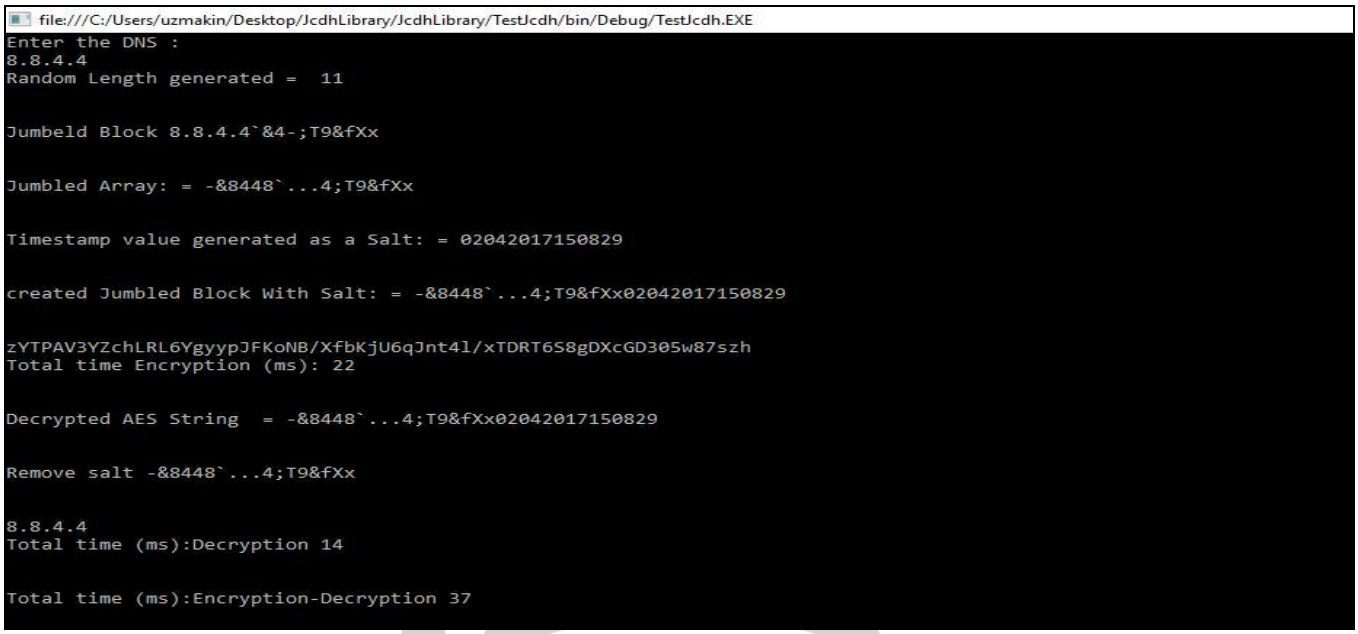

Figure 5. Screenshot of Execution of DNS Encryption

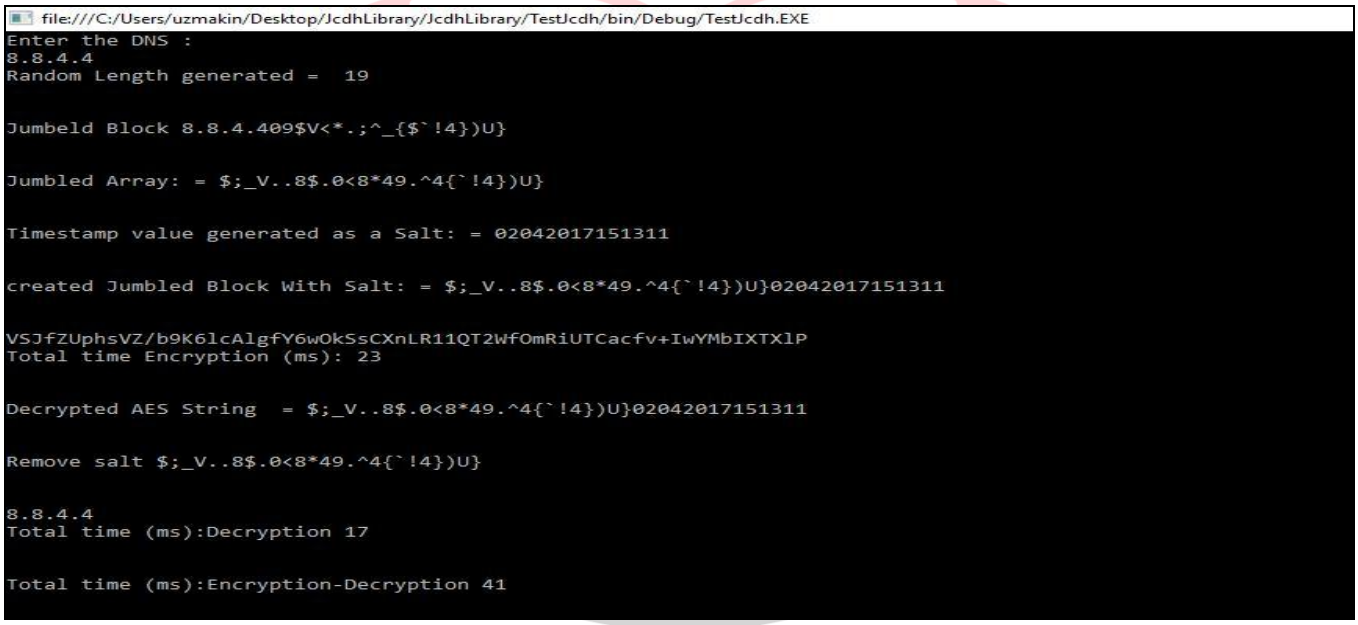

Figure 6. Screenshot of Execution of DNS Encryption

Thus from the output screen one can see that JS algorithm is completely random in selecting length. Further are tables in which one can compare key sizes length of normal AES Encryption and JS decryption

The reverse process is complete reverse of jumbling process. firstly the string is decrypted then salt is removed to get the jumbled string same mod process(mod value =2, then swap) is repeated until one get the inputted number

From the four table figure JS algorithm completely beats the encryption length which makes it much harder to crack thus making attacker jobs more complicating

The bar graph can give you the difference between the lengths. Though encryption times get increased .Encryption and decryption time sometimes is same that is because of the random generated character set.

Table 2. Comparison of DNS string I with AES and JS Algorithm

PARAMETERS	JUMBLING-SALTING STRING	AES STRING
INPUT	8.8.8.4	8.8.8.4

 DOI : 10.18231/2454-9150.2018.0437

Encryption Length	zYTPAV3YZchLRL 6YgyypJFKoNB/XfbKjU6 Jnt41/Xtdrt6S8gDXcGD305w87szh	1eiDWDPzVDK5po V/bqixSg==
Encryption Time	22	21
Decryption Time	17	15
Total Time	39	36

Table 3. Comparison of DNS string II with AES and JS Algorithm

Parameters	**Jumbling-Salting String**	**AES String**
Input	157.178.212.198	157.178.212.198
Encryption Length	7dfAs/Nkrw5BQeDFhs1bYdg8fVwtCwFNF Leb/3cX3yzEPyxzxTsc8Zoj5jkvcssF8yGcqtiI1 My9qZXCVzGF02yWofHu8kxxcdwM qqFu7Q=	g/q17eumEYdEhU Eh6Xgu4Q==
Encryption Time	23	22
Decryption Time	16	15
Total Time	39	37

Table 4. Encryption and Decryption time of DNS string of JS Algorithm

Sr. No	IP Address	Encryption Time (in Ms)	Decryption Time (in Ms)
1	194.182.74.51	26	17
2	110.77.181.20	22	19
3	122.216.120.251	23	18
4	50.233.137.33	27	18
5	167.99.48.250	24	15
6	159.65.110.167	27	15
7	50.233.137.34	21	17
8	103.238.69.139	26	20
9	218.50.2.102	27	19
10	194.182.74.248	25	17

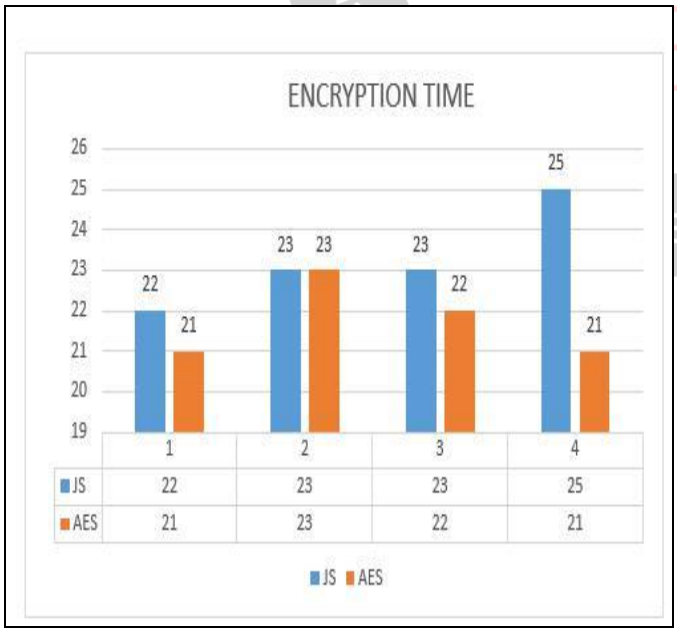

Figure 7. Encryption time of JS and AES Algorithm

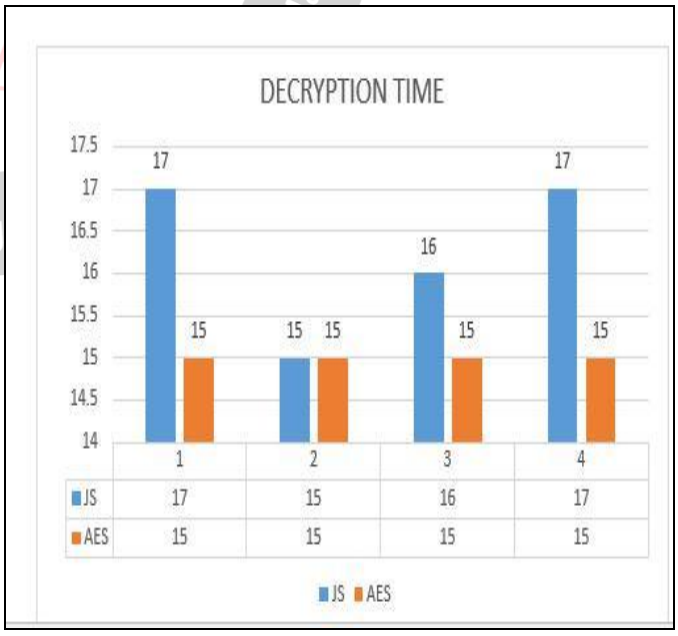

Figure 8. Decryption time of JS and AES Algorithm

 DOI : 10.18231/2454-9150.2018.0437

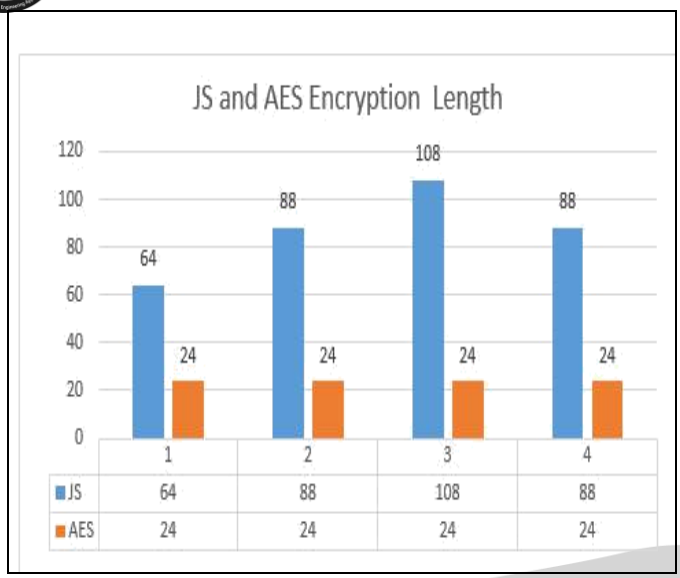

Figure 9. Encrypted text of JS and AES Algorithm

VI. CONCLUSION

Here we conclude that the IP address has been successfully encrypted so that we can prevent attacks. The admin of the network forces the decryption for ensuring the security of the network. As specified earlier we have used the JS algorithm for encrypting the IP address of the entire network, only a single client or multiple clients. Here the size of the cipher text is large as compared to AES. We also conclude that as compared to AES, JS takes almost the same time for encryption and decryption but our focus is on the randomization. Here JS generates more random values which are very difficult to crack thus ensuring us a definite secure network. We also conclude that our algorithm creates random values for the same IP address which adds an extra layer of security. The jumbling and salting process exhibit the feature of randomization. The main reason for increasing the larger text size is the extra overhead of the characters which makes the IP address more difficult to crack.

REFERENCES

[1] Tiwari, N. K., & Khakhil, S. (2015). Security System for DNS using Cryptography. *International Journal of Computer Applications*, *120*(17).

[2] Ateniese, Giuseppe, and Stefan Mangard. "A new approach to DNS security (DNSSEC) . "*Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM, 2001.

[3] Schneider, B. (1996). *Applied cryptography: protocols, algorithms, and source code in C*. John Wiley & Sons.

[4] Prathamesh Churi, Medha Kalelkar, and Bhavin Save, JSH Algorithm: A Password Encryption Technique using Jumbling- Salting-Hashing, International Journal of Computer Applications (0975-8887), Vol. 92-No.02, April 2014.

[5] Churi, P. P., Ghate, V., & Ghag, K. (2015, November). Jumbling-Salting: An improvised approach for password encryption. In *Science and Technology (TICST), 2015 International Conference on* (pp. 236-242). IEEE.

[6] Goldreich, O., Goldwasser, S., & Micali, S. (1984, October). How To Construct Random Functions. In *Foundations of Computer Science, 1984. 25th Annual Symposium on* (pp. 464-479). IEEE.

[7] Larson, M., Massey, D., Rose, S., Arends, R., & Austein, R. (2005). DNS security introduction and requirements.

[8] Ariyapperuma, S., & Mitchell, C. J. (2007, April). Security vulnerabilities in DNS and DNSSEC. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on* (pp. 335-342). IEEE.

[9] Ramkrishna Oruganti, Saurabh Shah, Yohan Pavri, Neelansh Prasad, Prathamesh Churi (2017) . JSSecure: A Secured Encryption Strategy for Payment Gateways in E-Commerce. Circulation in Computer Science, 2, 5(June 2017), 13-17. https://doi.org/10.22632/ccs-2017-252-17

 DOI : 10.18231/2454-9150.2018.0437