2 Days International Conference on CSIT-2019, ICSD-2019
Mahratta Chamber of Commerce, Industries and Agriculture Tilak Road, Pune (India)
2nd -3rd November 2019 www.conferenceworld.in
ISBN : 978-81-943584-1-1

Secure Data Transmission through IBE and Symmetric Encryption Techniques

[1]Vamsi Mohan V, [2]Dr. Sandeep Malik

[12]Department of Computer Science,
[12]School of Engineering and Technology, Raffles University, Neemrana, India

ABSTRACT

There are many techniques, methods and methodologies in secure data transmission. However, still users are failing to transmit the data securely. Hackers are succeeding in attacking the data for every few minutes. In cyber security attacks, SQL Injection and Cross Site Scripting attacks are very prominent. SQL Injection (SQLI) executes malicious code in the form of SQL statement at database by directly executing the query or changing the requested parameters, changing the triggered URL. In case of Cross-Site Scripting, the attacker targets to execute the malicious code at the client side. In this paper, for secure data transmission, we proposed a model using IBE and AES encryption techniques. we explained the proposed model "How to transmit the data securely?".

Keywords – Symmetric encryption, IBE, AES, Data transmission, Secure transmission, SQL Injections (SQLI), Cross Site Scripting (XSS), Static Analysis, Identity Based Encryption (IEB), Advanced Encryption Standard (AES).

I. INTRODUCTION

In the last two decades, the usage of web and mobile applications are increased in many folds. Web applications usage grown exponentially and mobile apps as well. With the growth of the usage, the cyber security attacks are also equally increased. According to the Security magazine, there is a hacker attack for every 39 seconds. Nesparker web security identified the vulnerability for every 4.59 minutes. In these vulnerabilities, SQL injections and Cross Site Scripting attacks are prominent. In our research, we used IBE encryption for the authenticating the users and AES encryption for the encrypting the sender's data to transmit securely. We have introduced a method for splitting the sender's data into by parts, encrypt it and send it, where SQL injections fail to attack. In our proposed method, the attackers cannot access or hit the application even through the incognito mode and all incognito operations will be recorded through our solution.

II. IBE ENCRYPTION

Identity-based Encryption (IBE) is an alternative to PKI, and involves generating the encryption key from a piece of identity for the recipient. For example, we could use the email address of the recipient to generate the key for a destination.

Boneh and Franklin introduced a practical Identity-Based Encryption system (IBE) that has become very popular in the security research community.

ISBN : 978-81-943584-1-1

In the conventional public-key cryptosystems, it requires to distribute the key before sending the encrypted message to the receiver. Various systems have been designed for distribution, e.g. certificates. Practically, the key distribution is complex.

In case of IBE, the name (email address) of receiver is his public key and there is no need for the distribution. There is a trusted central service, to which you send the email; the central service forwards the message to the receiver, and all communications are encrypted before sending to the receiver. There are two disadvantages with the approach.

1. the central service must be trusted (it has the power to read all emails);
2. the central service will have oversized optic fibers and thousands of processors because every message goes through it.

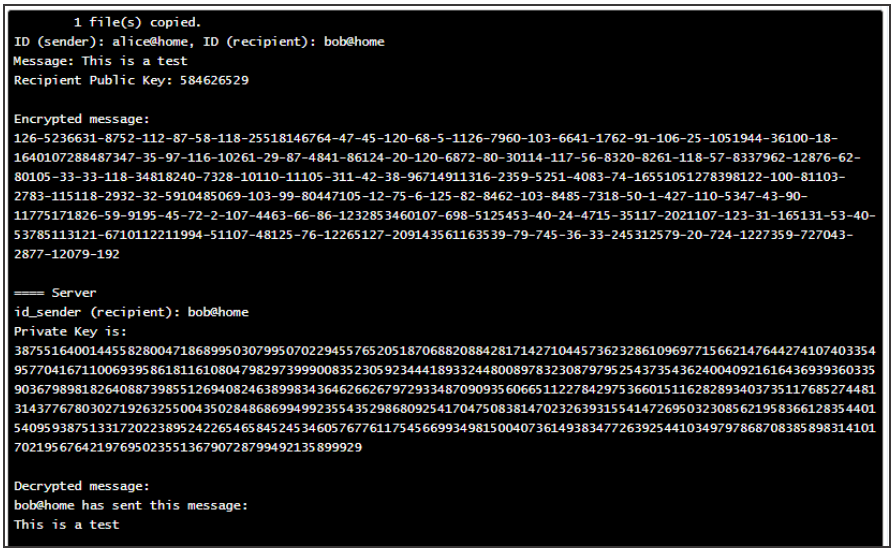

Fig 1.0

IBE uses high-level mathematics to remove the second issue. With IBE, the "central service" exists and need not be online. The real-time action is replaced with the mathematical pairing.

III. SYMMETRIC ENCRYPTION (AES Encryption)

The Advanced Encryption Standard, or AES, is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.

The National Institute of Standards and Technology (NIST) developedthe AES in 1997 when it announced the need for a successor algorithm for the Data Encryption Standard (DES), which was starting to become vulnerable to brute-force attacks.

AES features

NIST specified the new advanced encryption standard algorithm should be a block cipher capable of handling 128-bit blocks, using keys sized at 128, 192, and 256 bits; Security, Cost and Implementation was the factors for evaluating the security algorithm.

ISBN : 978-81-943584-1-1

How AES encryption works

AES comprises 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-, 192- and 256-bits, respectively. The Rijndael cipher was designed to accept additional block sizes and key lengths, but for AES, those functions were not adopted.

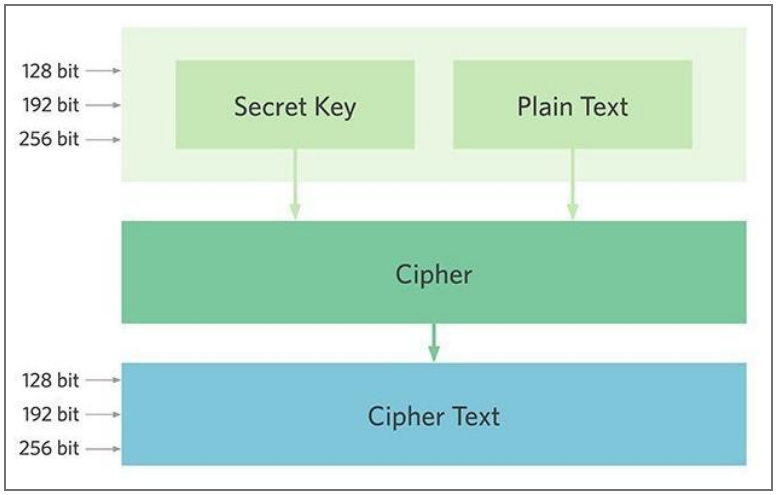

Fig 2.0

Symmetric (also known as secret-key) ciphers use the same key for encrypting and decrypting, so the sender and the receiver both know and use the same secret key. All key lengths are deemed sufficient to protect classified information up to the "Secret" level with "Top Secret" information requiring either 192 or 256-bit key lengths. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of ciphertext.

The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

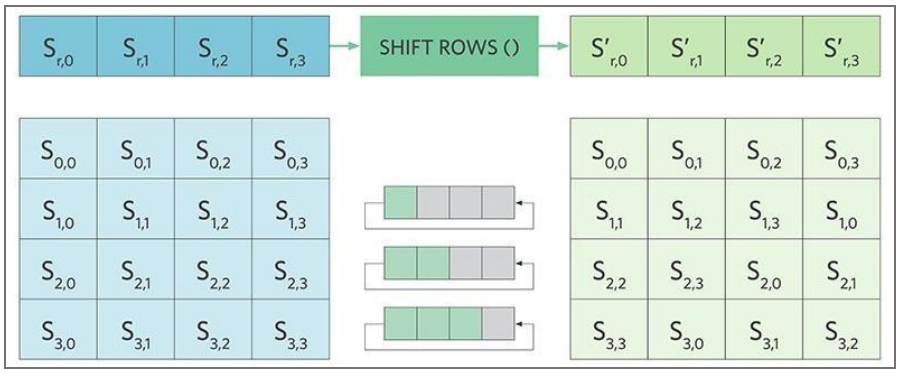

Fig 3.0

The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, the third mixes columns. The last transformation is a simple exclusive or (XOR) operation performed on each column using a different part of the encryption key -- longer keys need more rounds to complete.

IV. SECURE DATA TRANSMISSION–PROPOSED MODEL

For exchanging the communication, data will be transmitted between sender and receiver. In general case, the data will be sent without any restrictions. This will allow hackers and attackers to hack the data without proper authorization. Leaking the data may create different kind of risks and issues.

SQL Injections:

In our proposed system, the data will be sent through AES encryption by splitting the file into different parts. The split parts will be encrypted by IBE and AES encryption techniques. The encrypted file will be sent to the receiver.

Pre encryption

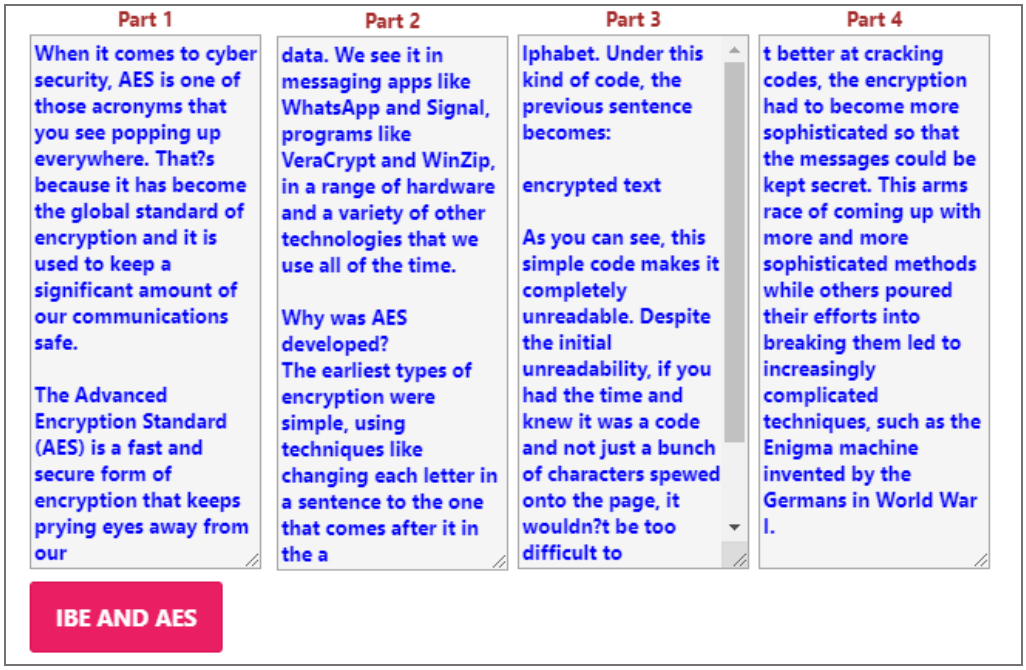

Fig 4.0

Post encryption

Fig 5.0

When the receiver receives the data will be the same which was sent through existing method and proposed method. For the receiver, there will be no change except in the proposed method the secure data transmission.

Usually, the hackers are the unknown and unauthorized users who uses the system. In most of the cases, they will be registered as a normal user to the system and log-in to the system with their credentials. They will try to query the data from their login to extract the other's data.

Example:

```
Actual URL: http://localhost:8080/Sql_injection/Shared-Data-AES.jsp?id=8
Modified URL: http://localhost:8080/Sql_injection/Shared-Data-AES.jsp?id=8'
OR id >=1 (which is always true)
```

In case of normal data transmission or existing method, When the modified query triggers, it will show all the data from the database since the modified query is always true and the attacker has access to the system. In case of proposed method, the hacker cannot see the data as it restricts to view the un-authorized data.

Cross Site Scripting:

In case if the attacker uses the URL of the application without authentication, it restricts the user to allow to the application even, if the attacker tries to enter into the system using incognito mode. Simultaneously, the

attacker IP address, access time, date and unauthorized login credentials will be captured by the proposed system. System admin can see the details of the users.

V. ADVANTAGES OF THE PROPOSED MODEL

1. Data transport is quite fast as it is split into different parts and secure from SQL injection and Cross Site scripting attacks.
2. Protects from the unauthorized users by using in cognitive mode, using IBE and AES algorithms.
3. History of the data access users with date and time logs.

VI. CONCLUSION AND FUTURE WORK

In our research, we proposed a system which restricts the SQL Injection attacks and Cross Site attacks for the web applications. However, the scope of the research is limited to web applications only. This can be extended to mobile and hybrid applications.The proposed model is out of scope for Create, Retrieve, Update and Delete (CRUD) transactions.

REFERENCES

[1] https://www.securitymagazine.com/articles/87787-hackers-attack-every-39-seconds
[2] Akbar, Memen, and Muhammad Arif Fadhly Ridha. "SQL Injection and Cross Site Scripting Prevention using OWASP ModSecurity Web Application Firewall." JOIV: International Journal on Informatics Visualization 2.4 (2018): 286-292.
[3] Bisht, Prithvi, A. Prasad Sistla, and V. N. Venkatakrishnan. "Automatically preparing safe SQL queries." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2010.
[4] G. Wassermann and Z. Su, “Static detection of cross-site Scripting vulnerabilities,” In Proceeding of the 30th International Conference on Software Engineering, (2008)May.
[5] I. Hydara, A. B. M. Sultan, H. Zulzalil, N. Admodisastro, Current state of research on cross-site scripting (xss) c a systematic literature review, Information & Software Technology 58 (2014) 170–186.
[6] I. Parameshwaran, E. Budianto, S. Shinde, H. Dang, A. Sadhu, P. Saxena, Dexterjs: robust testing platform for dom-based xss vulnerabilities, in: Joint Meeting, 2015, pp. 946–949.
[7] I. Parameshwaran, E. Budianto, S. Shinde, H. Dang, A. Sadhu, P. Saxena, Auto-patching dom-based xss at scale, in: Joint Meeting, 2015, pp. 272–283.
[8] K. Z. Snow, R. Rogowski, J.Werner, H. Koo, F. Monrose, M. Polychronakis, Return to the zombie gadgets: Undermining destructive code reads via code inference attacks, in: Security and Privacy (SP), 2016 IEEE Symposium on, IEEE, 2016, pp. 954–968.
[9] Makera M Aziz and Dena Rafaa Ahmed (2015), Proposed Method to Prevent SQL Injection Attack. The Annual Conference on Network Security & Distributed Systems (NSDS'2015).

ISBN : 978-81-943584-1-1

[10] Muhammad Saidu Aliero, Abdulhamid Aliyu Ardo, Imran Ghani & Mustapha Atiku (2016), Classification of Sql Injection Detection And Prevention Measure. IOSR Journal of Engineering (IOSRJEN), ISSN (e): 2250-3021, ISSN (p): 2278-8719, Vol. 06, Issue 02 (February. 2016), ||V1|| PP 06-17.

[11] P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Krgel, G. Vigna, Cross site scripting prevention with dynamic data tainting and static analysis., in: Network and Distributed System Security Symposium, NDSS 2007, San Diego, California, Usa, February - March, 2007.