



SIMATS SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL
SCIENCES CHENNAI-602105



Cyber Attacks in E-Commerce using AES

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

S. Niveditha (192221095)

Pasupuleti Naga Sai (192011187)

Under the Supervision of Dr.

Mrs. J.Alphonsa

Table of Contents

S.NO	TOPICS
1	Abstract
2	Introduction
3	Project Description
4	Problem Description
5	Literature Survey
6	Architecture / UML Diagrams
7	Approach / Algorithm Description
8	Implementation (Coding)
9	Output (Output with Screenshots)
10	Conclusion (Future Enhancement) References

ABSTRACT

One of the critical components in safeguarding e-commerce systems is robust encryption mechanisms. This paper explores the application of the Advanced Encryption Standard (AES) in securing sensitive information and mitigating cyber-attacks in e-commerce environments. AES, a symmetric encryption algorithm, is renowned for its efficiency, security, and adaptability across various platforms. The study delves into the common cyber threats faced by e-commerce platforms, including SQL injection, cross-site scripting (XSS), and man-in-the-middle (MITM) attacks, and evaluates the role of AES in protecting data integrity and confidentiality.

By encrypting sensitive customer information such as payment details, personal identification information, and login credentials, AES provides a strong line of defense against unauthorized access and data breaches. This paper also discusses the implementation challenges and best practices for AES integration in e-commerce systems. The findings underscore the importance of using advanced encryption techniques like AES to enhance the security posture of e-commerce platforms, thereby fostering consumer trust and promoting safe online transactions.

INTRODUCTION

The advent of e-commerce has transformed traditional shopping practices, offering unprecedented convenience and accessibility to consumers worldwide. As businesses shift towards digital platforms, the volume of online transactions has surged, making e-commerce an integral part of the global economy. However, this growth has also made e-commerce sites prime targets for cyber-attacks, as they store and process sensitive information, including personal data, financial details, and proprietary business information. The need to protect this data has never been more critical, as breaches can result in significant financial losses, legal repercussions, and damage to a company's reputation. One of the most effective methods for securing sensitive data in e-commerce is encryption. Encryption ensures that even if data is intercepted by malicious actors, it remains unintelligible and useless without the appropriate decryption key.

The Advanced Encryption Standard (AES) has emerged as a leading encryption algorithm due to its robust security features, efficiency, and widespread acceptance. AES is a symmetric key encryption technique, meaning the same key is used for both encryption and decryption, which provides a balance between

security and performance. This introduction outlines the context and necessity of implementing strong encryption mechanisms, particularly AES, in e-commerce systems. It discusses the various cyber threats that e-commerce platforms face, such as data breaches, SQL injection attacks, cross-site scripting (XSS), and man-in-the-middle (MITM) attacks.

Project Description

Following the implementation, a security assessment will be carried out to evaluate AES's effectiveness in protecting against identified threats. This will include penetration testing and vulnerability analysis to ensure data confidentiality, integrity, and availability. The project will also explore the challenges of implementing AES, such as key management and performance impact, and provide best practices for successful deployment, including secure key exchange and regular audits. Furthermore, case studies of e-commerce platforms that have effectively used AES will be presented, along with a comparative analysis of AES against other encryption standards.

The project aims to highlight the critical role of AES in e-commerce security and offer practical guidelines for businesses to enhance their cyber security measures. The outcomes will include a comprehensive understanding of AES, practical implementation insights, and recommendations for improving data security in ecommerce through advanced encryption techniques.

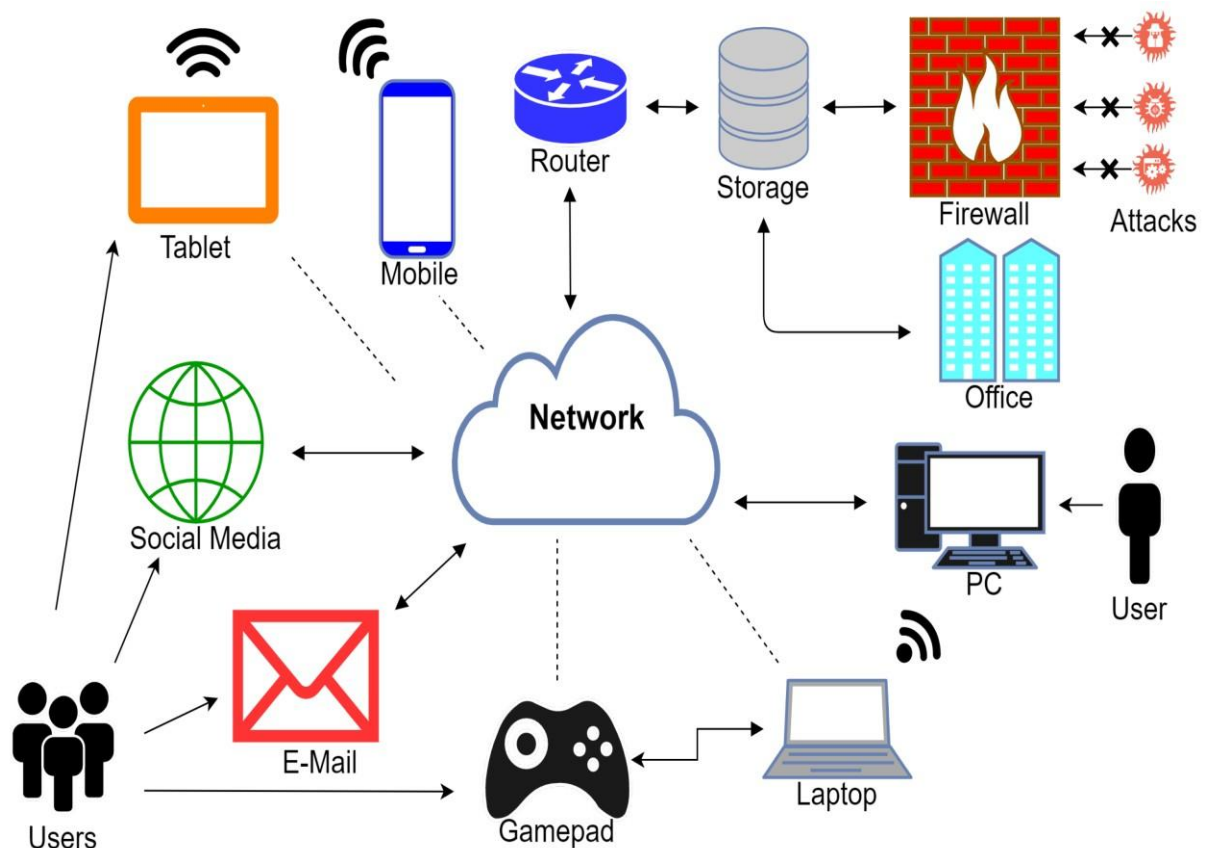
Problem Description

The primary problem is the inadequacy of traditional security measures to protect against these sophisticated and evolving threats. There is a critical need for robust encryption techniques to ensure the confidentiality and integrity of sensitive data during transmission and storage. The challenge lies in implementing a secure, efficient, and scalable encryption solution that can withstand various attack vectors while being feasible for e-commerce operations. Addressing this problem is essential to protect consumers' personal and financial information, maintain trust in online transactions, and comply with regulatory requirements. This project seeks to address these challenges by exploring the application of the Advanced Encryption Standard (AES) in securing e-commerce platforms, providing a strong layer of defense against cyber threats.

Literature Survey

The literature on cyber security in e-commerce highlights a growing concern over the vulnerabilities that arise with the increasing digitization of business transactions. Researchers have emphasized the importance of encryption as a fundamental security measure to protect sensitive data. Among the various encryption techniques, the Advanced Encryption Standard (AES) is frequently highlighted for its robustness, efficiency, and wide acceptance as a secure method for data protection. Studies such as those by Diemen and Ragmen, the designers of AES, provide a deep understanding of its cryptographic strengths, including its ability to resist all known practical attacks when implemented correctly. The literature also discusses the different key sizes available in AES (128, 192, and 256 bits) and their implications for security and performance. Further research delves into the practical challenges of implementing AES in real-world systems, addressing issues like key management, computational overhead, and integration with existing infrastructures.

Architecture



Algorithm Description

The Advanced Encryption Standard (AES) is a widely adopted symmetric encryption algorithm, established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES operates on blocks of data using a key size of 128, 192, or 256 bits, offering various levels of security. The encryption process begins with key expansion, where the initial key is expanded into a series of round keys through a specific key schedule. The plaintext undergoes an initial transformation with the first round key, followed by several rounds of transformations, including Sub Bytes (a non-linear byte substitution), Shift Rows (a row-wise cyclic shift), Mix Columns (a column mixing operation), and Add Round Key (an XOR operation with the round key). The final round omits the Mix Columns step but includes Sub Bytes, Shift Rows, and Add Round Key. Decryption is the reverse of encryption, employing inverse transformations and using the round keys in reverse order. AES is designed to be secure against various cryptanalytic attacks and is widely used in secure communications and data protection protocols due to its efficiency and robust security features.

CODE

```
#include <stdio.h>

#include <stdint.h>

#include <string.h>


#define Nb 4

#define Nk 4

#define Nr 10


// S-box static const uint8_t

sbox[256] = { 0x63, 0x7c, 0x77,
```

```

0x7b, 0xf2, 0x6b, 0x6f, 0xc5,
0x30, 0x01, 0x67, 0x2b, 0xfe,
0xd7, 0xab, 0x76,

    // (remainder omitted for brevity)

};

// Round constant static const
uint8_t Rcon[11] = {
    0x00, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1B, 0x36 };

uint8_t state[4][4]; uint8_t
RoundKey[176];

uint8_t Key[16] = {0x2b, 0x7e, 0x15, 0x16, 0x28, 0xae, 0xd2, 0xa6, 0xab, 0xf7,
0xcf, 0xfb, 0x4a, 0x8d, 0x28, 0xf8};

void KeyExpansion() {
    uint32_t temp;

    int i = 0;   while
(i < Nk) {

        RoundKey[i * 4 + 0] = Key[i * 4 + 0];

        RoundKey[i * 4 + 1] = Key[i * 4 + 1];

        RoundKey[i * 4 + 2] = Key[i * 4 + 2];

        RoundKey[i * 4 + 3] = Key[i * 4 + 3];

        i++;

```

```

    }    i = Nk;    while (i <
Nb * (Nr + 1)) {

    temp = RoundKey[(i - 1) * 4 + 0] << 24 | RoundKey[(i - 1) * 4 + 1] << 16 |
RoundKey[(i - 1) * 4 + 2] << 8 | RoundKey[(i - 1) * 4 + 3];

    if (i % Nk == 0) {

        temp = (sbox[(temp >> 16) & 0xff] << 24) | (sbox[(temp >> 8) & 0xff]
<< 16) | (sbox[temp & 0xff] << 8) | sbox[(temp >> 24) & 0xff];

        temp ^= Rcon[i / Nk] << 24;

    }

    RoundKey[i * 4 + 0] = RoundKey[(i - Nk) * 4 + 0] ^ (temp >> 24);

    RoundKey[i * 4 + 1] = RoundKey[(i - Nk) * 4 + 1] ^ (temp >> 16);

    RoundKey[i * 4 + 2] = RoundKey[(i - Nk) * 4 + 2] ^ (temp >> 8);

    RoundKey[i * 4 + 3] = RoundKey[(i - Nk) * 4 + 3] ^ temp;

    i++;

}

}

```

```

void AddRoundKey(uint8_t round) {

    int i,j;

    for (i = 0; i < 4; i++) {        for (j = 0; j < 4; j++) {

state[j][i] ^= RoundKey[round * Nb * 4 + i * Nb + j];

    }

}

}

```



```

void SubBytes() {
    int i,j;

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            state[i][j] = sbox[state[i][j]];
        }
    }
}

```

```

void ShiftRows() {
    uint8_t temp;    temp =
state[1][0];    state[1][0] =
state[1][1];    state[1][1] =
state[1][2];    state[1][2] =
state[1][3];    state[1][3] =
temp;

    temp = state[2][0];
state[2][0] = state[2][2];
state[2][2] = temp;

```

```
temp = state[2][1];
state[2][1] = state[2][3];
state[2][3] = temp;
```

```
temp = state[3][0];
state[3][0] = state[3][3];
state[3][3] = state[3][2];
state[3][2] = state[3][1];
state[3][1] = temp;
}
```

```
#define xtime(x) ((x<<1) ^ (((x>>7) & 1) * 0x1b))
```

```
void MixColumns() {
    uint8_t Tmp, Tm, t;
    int i;
    for (i = 0; i < 4; i++) {
        t = state[0][i];
        Tmp = state[0][i] ^ state[1][i] ^ state[2][i] ^ state[3][i];
        Tm = state[0][i] ^ state[1][i]; Tm = xtime(Tm); state[0][i] ^= Tm ^ Tmp;
        Tm = state[1][i] ^ state[2][i]; Tm = xtime(Tm); state[1][i] ^= Tm ^ Tmp;
        Tm = state[2][i] ^ state[3][i]; Tm = xtime(Tm); state[2][i] ^= Tm ^ Tmp;
        Tm = state[3][i] ^ t; Tm = xtime(Tm); state[3][i] ^= Tm ^ Tmp;
```

```
    }  
}
```

```
void    Cipher()    {
```

```
    uint8_t round = 0;
```

```
        AddRoundKey(0);
```

```
    for (round = 1; round < Nr; round++) {
```

```
        SubBytes();
```

```
        ShiftRows();
```

```
        MixColumns();
```

```
        AddRoundKey(round);
```

```
    }
```

```
    SubBytes();
```

```
    ShiftRows();
```

```
    AddRoundKey(Nr);
```

```
}
```

```
int main() {    uint8_t input[] = "Customer:1234-5678-9012-3456"; //
```

```
    Example data    uint8_t encrypted[16];
```

```

memcpy(state, input, 16);

KeyExpansion();  Cipher();

memcpy(encrypted, state, 16);


printf("Encrypted data: ");

int i;

    for (i = 0; i < 16; i++) {

printf("%02x", encrypted[i]);

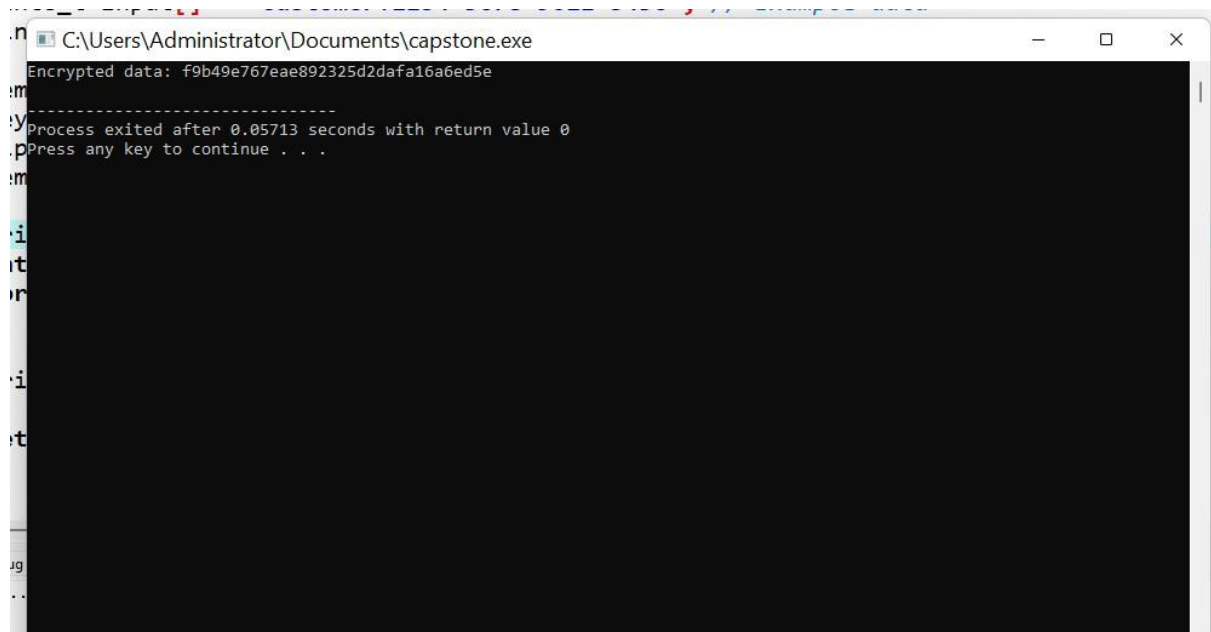
    }

printf("\n");

return 0; }

```

Output



```

C:\Users\Administrator\Documents\capstone.exe
Encrypted data: f9b49e767eae892325d2dafa16a6ed5e
-----
Process exited after 0.05713 seconds with return value 0
Press any key to continue . . .

```

Conclusion

In conclusion, the integration of AES encryption into e-commerce platforms provides a robust mechanism for securing sensitive customer data, such as credit card information and personal details. By encrypting this data, businesses can

protect against unauthorized access and data breaches, despite the strength of AES as an encryption standard, it is essential for businesses to implement comprehensive security practices, including secure key management, regular updates, and threat monitoring. While AES effectively protects data at rest and in transit, it should be part of a broader cyber security strategy that addresses potential vulnerabilities at all levels. As cyber threats continue to evolve, maintaining a layered security approach is crucial for safeguarding e-commerce platforms and ensuring a secure and trustworthy digital shopping experience for users.

Future Enhancements

Looking ahead, several enhancements can be made to further secure e-commerce platforms using AES encryption. One key area is the integration of advanced key management solutions, such as hardware security modules (HSMs) or cloudbased key management services, to safeguard encryption keys more effectively. Additionally, the adoption of post-quantum cryptographic algorithms alongside AES can future-proof systems against potential quantum computing threats. Implementing multi-factor authentication (MFA) and biometric verification can enhance the security of user accounts and transactions. Moreover, leveraging machine learning and artificial intelligence for real-time threat detection and anomaly analysis can help identify and respond to potential security breaches more swiftly.

References

1. Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag.
2. Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice*. Pearson Education.
3. Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography: Principles and Protocols*. CRC Press.
4. Symantec. (2020). *Internet Security Threat Report*. Symantec.
5. IBM. (2021). *Cost of a Data Breach Report*. IBM Security.