

Integration of Deffie Hellman Key Exchange Encryption and Advanced Encryption Standard Algorithm for Securing SMS based One-time Password from Man in the Middle (MITM) Attacks

ABOHO D. MOSES.

Directorate of ICT
Akperan Orshi College of Agriculture, Yandev
Benue State, Nigeria

KARIM USMAN

Department of Mathematics and Computer Science
Benue State University, Makurdi, Benue State,
Nigeria

AWUHE T. RICHARD.

Department of Computer Science and Statistics
Akperan Orshi College of Agriculture, Yandev
Benue State, Nigeria

ENGR. IKERAVE A. FREDRICK

Department of Agricultural Engineering Technology
Akperan Orshi College of Agriculture, Yandev
Benue State, Nigeria

ABSTRACT - This paper aims at improving the security feature of One Time Password (OTP). Since all transactions are conducted in an open network, there is a high risk of confidential data attack by Man in the Middle or unauthorized users. Online service providers such as banks have developed a new security framework called One Time Password (OTP) to prevent sensitive information attack by MITM and other unauthorized users. OTP provides additional online protection for users. However, this approach can be hacked by MITM if the OTP is not properly protected. In this paper, a new security model is proposed to enhance the existing OTP approach using Deffie Hellman Key Exchange (DHKE) encryption and Advanced Encryption Standard (AES) algorithm. The proposed security model ensures that the OPT is encrypted using DHKE and AES before it is sent to the user's registered mobile number. The encrypted OTP is decrypted on the user's mobile phone, such that if there is an attack by MITM, it's the encrypted data that will be captured but not the original content.

Keywords - OTP, MITM, DHKE, Transactions, Encryption

I INTRODUCTION

With the recent advancement in e-Commerce technology, people can now buy and pay for services online once they have a computer connected to the internet. E-Commerce literally means any kind of commercial transaction that is conducted electronically using computer networks (such as the internet). It is also regarded as any type of business, or commercial transaction that provides services for buying and selling products or exchanging information across the internet [7]. Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems [11]. Online payment solutions have become progressively famous due to the continuous use of the internet based shopping and banking applications.

However, in developing countries like Nigeria, there exists low patronage of buying and paying for things online due to some fraudulent activities

associated with online payment systems. Credit card holders don't feel convenient giving out their card sensitive data like credit card numbers and secret pins because of the fear that their card information could be captured and used by unauthorized persons. The widespread use of credit card for online transactions has given room for different fraudulent activities. Credit Card Fraud is defined as, "when an individual uses another individuals' credit card for personal use while the owner of the card as well as the card issuer are not aware of the activity that the card is being used for" [12]. Fraud prevention and fraud detection systems are two main mechanisms to avoid frauds and losses due to fraudulent activities [13]. Fraud prevention is the upbeat mechanism with the goal of disabling the happening of fraud. Fraud detection systems come into play when the fraudsters go beyond to the fraud prevention systems and start a fraudulent transaction.

Today, one of the major methods developed by fraudsters to deceive and manipulate E-commerce

users is Man in the middle (MITM) Attack. A man-in-the-middle attack often refers to an attack in which an attacker secretly intercepts the electronic messages given between the sender and receiver and then captures, inserts and modifies message during message transmission [1]. One example of man-in-the-middle attacks is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker [2]. MITM attack is a serious threat to users because the attacker is able to capture and manipulate confidential information in real-time. Online banking systems are the major targeted platforms by MITM in order to capture and manipulate financial transactions.

Almost all the commercial banks in Nigeria today offer online banking services to their customers. Credit and Debit cards provided by these banks

have become the most preferred mode of buying/paying for services online. Since all transactions are carried out in an open network, there is a high risk of confidential information attack by MITM. Most of these banks have developed a new security framework called One Time Password (OTP) to prevent their customers' sensitive information from attack by MITM. OTP provides additional online protection for users. However, this approach can be counter attacked by MITM if the OTP is not properly protected. In this paper, a new security model is proposed to enhance the existing OTP approach using Diffie Hellman Key Exchange (DHKE) encryption and AES algorithm. The proposed security model ensures that the OPT is encrypted using DHKE and AES before it is sent to the user's registered mobile number. The encrypted OTP is decrypted on the user's mobile phone, such that if there is an attack by MITM, the OPT will be visible in an encrypted state.

II ANALYSIS OF ONE TIME PASSWORD (OTP) VIA SMS

A one-time password (OTP) is a password that is valid for only one login session or transaction on a computer system or other digital devices [3]. It is a unique code that can only be used once and is sent to the user's registered mobile number during an interaction with online banking systems. The

unique code is generated based on a changeable parameter, such as time or a random number. The phone number of the user must be registered for the service that provides SMS OTPs for authentication or authorization.

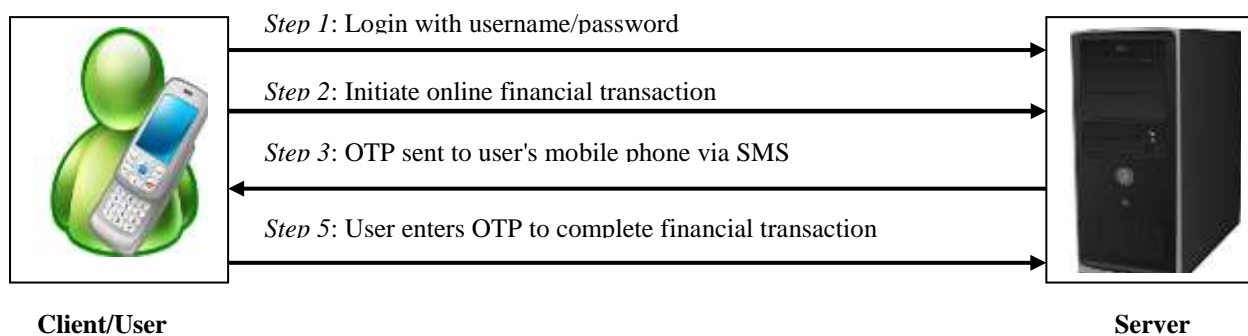


Figure 1 Analysis of One-Time Password

Whenever a user wishes to carry out an online financial transaction, the first step is to enter username and password for authentication. After successful authentication, the user initiate a financial transaction for processing. The server

responds by sending OTP to the user's registered mobile phone. Finally, the user completes the financial transaction by entering the OTP sent to his registered more phone via SMS.

III RELATED WORKS

Abdul *et al.* [4] conducted a research using Hybrid Model For Securing E-Commerce Transactions. They used cipher method to improve the Diffie-

Hellman key exchange by using truncated polynomial in discrete logarithm problem (DLP) to increases the complexity of this method over

unsecured channel, also combines the hashing algorithm of MD5, the symmetric key algorithm of AES and the asymmetric key algorithm of Modification of Diffie-Hellman (MDH). The experimental results showed that the model improved the interacting performance, while providing high quality of security service for desired e-commerce transactions.

Kumar and Chaudhary [5] presented a technique to prevent Hackers and Trackers in on-line-Transactions. In this paper, they designed an algorithm for improving a security level for communicating between the client and the server in SSL. They implemented the modified RSA algorithm which integrates bit-stuffing into a particular algorithm. With the implementation of this algorithm, trespassers cannot access data in an easy manner because this bit-stuffing method increases the level of security. So, this technique improves the security level of the algorithm thus expanding its range with a trusted user.

Rupali and Unmukh [6] presented a security authentication technique to overcome the problem of man in the middle attack in e-commerce. In this paper, they focused on client and server authentication. They examined the phishing problem, Man-In-The-Middle Attack, The main challenge in the design of a security system for high security, and how to prevent the attacks against data modification and authentication. In this work, they tried to solve a very big problem in transaction security of MIMA. First they authenticated a server to a user then finally authenticated user for the same server.

IV METHODOLOGY

In our proposed model, two algorithms are implemented to encrypt the OTP. First, the OTP is encrypted using DHKE; it is then further scrambled using AES to enhance the encryption. The flow diagram of the proposed model and the schematic diagram are presented in figure 2 and figure 3 below respectively.

In order to use the platform, the user is first authenticated after which a financial transaction is initiated. The server responds by sending an OTP to the user's registered mobile number. Before the OTP is sent to the user, it is encrypted using DHKE and AES. The encrypted OTP is finally decrypted as soon as it hits the user's phone. If the user wishes to continue the transaction, he enters the OTP on the payment platform. The data entered by the user

Safa *et al.* [7] proposed an idea for securing e-commerce transaction from phishing attack. The proposed idea is more secure compared to the existing online payment system using OTP. In this mechanism, OTP is combined with the secure key and is then passed through RSA algorithm to generate the Transaction password. A Copy of this password is maintained at the server side and is being generated at the user side using a mobile application; so that it is not transferred over the insecure network leading to a fraudulent transaction.

Jason *et al.* [8] presented a security model for Preventing Man-in-the-Middle Attacks in Authentication, Data Entry and Transaction Verification. This paper has three main purposes. The first is to detail the current threats and vulnerabilities to online financial systems and in particular online banking, from the selected literature. The second is to present the known prevention techniques for protecting against these attacks. The third is to present a conceptual model for authentication, data entry and transaction verification. It is suggested that the design added another layer of security to existing methods to either prevent a MitM attack or to make the procedure of capturing and reassembling customer log on and transaction details more computationally and time intensive than what it is worth to an attacker. The model is based on a graphical authentication application previously developed called Authentigraph.

is then encrypted before being transmitted and decrypted at the server side using DHKE and AES respectively. With this security framework, even if the data in transit is intercepted by MITM, the original content will not be captured since both encryption and decryption take place at the server and client sides.

The steps for generating the transaction OTP are presented below:

Step 1: To generate the OTP, a 6-digit randomly generated number and a secret key generated by DHKE algorithm are required.

Step 2: The 6-digit random number and the secret key are converted into binary forms

Step 3: Process the first encryption by taking the XOR of the binary 6-digit number and the binary secret key

Step 4: Process second encryption by applying AES to the result of Step 3.

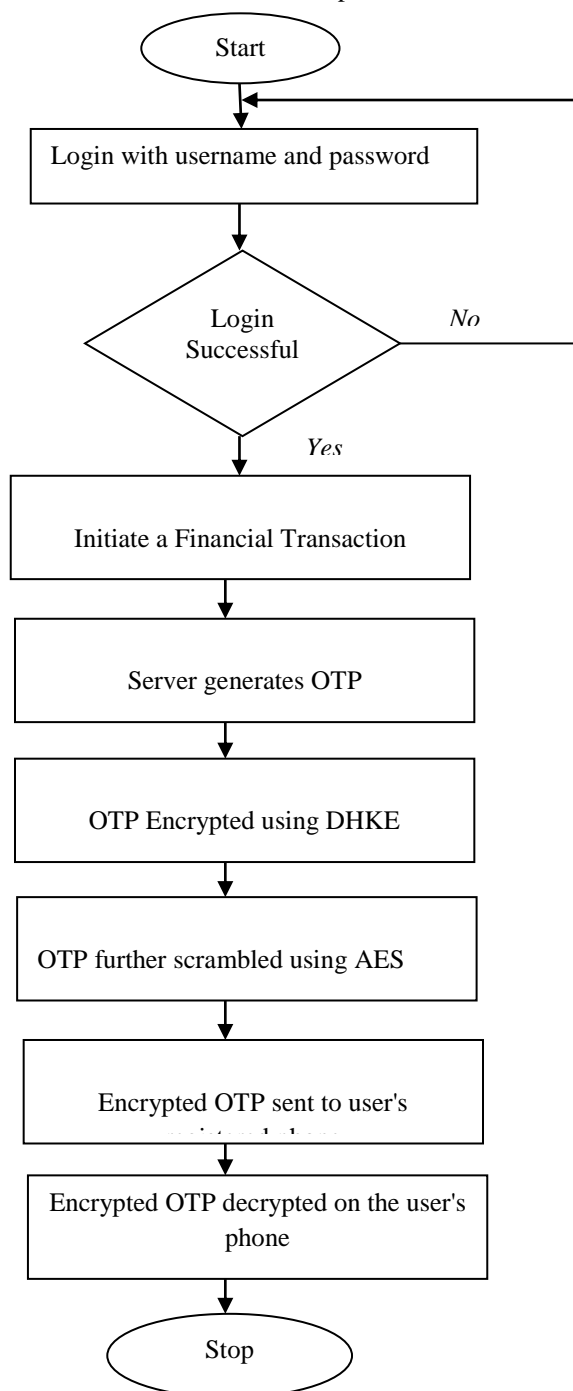


Figure 2 Flow diagram of the proposed model

Step 5: The result of step 4 is finally sent to the user's registered mobile number as the transaction OTP. The OTP is decrypted as soon as it hits the user's mobile phone.

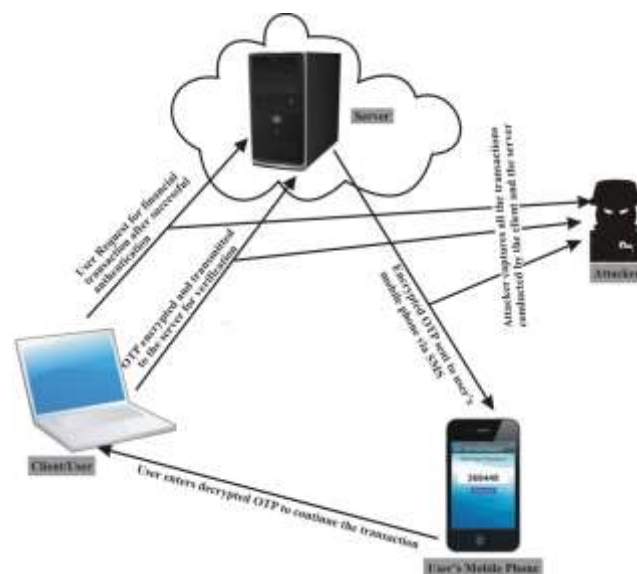


Figure 3 Schematic diagram of the proposed model

V DEFFIE HELLMAN KEY EXCHANGE ENCRYPTION

Global Variables: Prime number q ; $a < q$ and a is a primitive root of q .

Server Side

1. Select a random secret number S_A $S_A < q$
2. Calculate public key Y_A $Y_A = a^{S_A} \text{ mod } q$
3. Calculate Secret Key: $K = (Y_B)^{S_A} \text{ mod } q$

Client/User Side

1. Select a random secret number C_B $C_B < q$
2. Calculate public key Y_B $Y_B = a^{C_B} \text{ mod } q$
3. Calculate Secret Key: $K = (Y_A)^{C_B} \text{ mod } q$

The algorithms above show that the Server and the Client sides exchanged their public keys Y_A and Y_B respectively. Because S_A and C_B are kept private at the server and Client Sides, MITM can only have access to q , a , Y_A , Y_B and is faced with the task of taking a discrete logarithm to generate the key. This algorithm, particularly in its early forms, has a major weakness in the form of man-in-the-middle vulnerability [9].

To enhance the algorithm, the secret key K is further scrambled using AES algorithm.

ADVANCED ENCRYPTION STANDARD ALGORITHM

The Advanced Encryption Standard AES is a symmetric block cipher. This algorithm acts on 128-bit blocks and can use a key of 128, 192 or 256 bits in length [10]. The algorithm consists of 10 rounds (when the key has 192 bits, 12 rounds are used, and when the key has 256 bits, 14 rounds are used). Each round has a round key, derived from the original key. There is also a 0th round key, which is the original key. The round starts with an input of 128 bits and produces an output of 128 bits [4].

For encryption, each round consists of the following four steps [10]:

1. Substitute bytes,
2. Shift rows,
3. Mix columns, and
4. Add round key.

VI IMPLEMENTATION AND RESULTS

The implementation of our proposed security model for securing SMS based One-Time-Password from Man-in-the-Middle attacks was done using visual studio 2010 (C#). The figures below provide details about the encryption and decryption of transaction OTP using DHKE and AES algorithms.

```

=====ENCRYPTION PROCEDURE=====
Prime Number (q) = 19 & Primitive Root of q (a) = 6
Random Secret Number = 14
Public Key (YA) = 5
Secret Key (K) = 16
Binary Digits of the Secret Key (bsk) = 00000000000000010000
-----
Plain OTP = 751214
Binary Digits of OTP (botp) = 10110111011001101110
Cipher1 = DHKE Value of OTP (bsk XOR botp) = 10110111011001111110
Cipher2 = AES value of Cipher =
j8KaJnbRdmVeMHR30sZNePnTk5Cl6+dgYTv/hvtfmYEqW7hgKoSwEC/QHGkEVe7
-----
CIPHER OTP =
j8KaJnbRdmVeMHR30sZNePnTk5Cl6+dgYTv/hvtfmYEqW7hgKoSwEC/QHGkEVe7
    
```

Figure 4 Server side OTP encryption procedure

In order to generate and encrypt the transaction OTP, a Prime Number (q) and a Primitive root of q (a) are generated as 19 and 6 respectively. These two numbers are shared by both the server and client/user. The server generates a Random Secret Number (S_A) of 14 which is only accessible by the server itself and a public key (Y_A) of 5 is calculated using q , a and S_A . The formula to calculate the public key is given as:

$$Y_A = a^{S_A} \text{ mod } q$$

i.e $Y_A = 6^{14} \text{ mod } 19 = 5$

The calculated public key is also shared by both the server and the client. In addition, the server computes a Secret Key K using the Client's Public Key (Y_B), S_A and q respectively. The Secret Key K is computed using the formula:

$$K = (Y_B)^{S_A} \text{ mod } q \text{ where } Y_B = 9$$

Therefore, $K = 9^{14} \text{ mod } 19 = 16$

The Secret Key K is finally converted to binary digits to give 00000000000000010000. The conversion is done in such a way that it matches the length of the OTP.

The server proceeds by generating the transaction OTP and converting same to binary digits. In figure 3 an OTP of 751214 is generated and converted to a binary digits 10110111011001101110.

To achieve the first encryption, the binary value of the secret key is XORed with the binary value of the OTP. The result is finally scrambled using AES to have the encrypted transaction OTP.


```

=====DECRYPTION PROCEDURE=====
Prime Number (q) = 19 & Primitive Root of q (a) = 6
Random Secret Number = 7
Public Key (YB) = 9
Secret Key (K) = 16
Binary Digits of the Secret Key (bsk) = 0000000000000010000
=====
Cipher OTP =
j8KaJnbRdmVeMHR30sZNePnTk5Cl6+dgYTv/hvtfmYEqW7hgKoSwECJQHgkE
P1 = Decryption using AES = 10110111011001111110
P2 = (P1 XOR bsk) = 10110111011001101110
=====
Plain OTP = 751214
=====
    
```

Figure 5 Client side OTP decryption procedure

At the client side also, the decryption process starts by generating a random secret number, here 7 is the

random secret. The public global variables, q and a are the same with that of the server side. The client computes its public key as shown below:

$$Y_B = a^{SB} \text{ mod } q \text{ where } a=6, SB=7, \text{ and } q=19$$

$$Y_B = 6^7 \text{ mod } 19 = 9$$

The client proceeds by computing the secret key using the formula shown below

$$K = (Y_A)^{SB} \text{ mod } q \text{ where } Y_A = 5$$

$$\text{i.e } K = 5^7 \text{ mod } 19 = 16$$

The value of k is then converted to binary digits. In addition, the value of the cipher OTP is first decrypted using AES algorithm, the result is XORed with the binary value of the secret key and finally converted to decimal to have the plain OTP.

The figures below show the source codes of the proposed security framework.

```

SMSBasedOPT.UHInfoA.Controls
    genPublicKey()
    {
        int prime = Constants.PRIME_NUMBER;
        int int_val = Constants.INTEGER_VALUE;
        int random_secret = randomNumber();

        BigInteger client = (BigInteger) Math.Pow(int_val, random_secret);
        BigInteger yclient = client % prime;

        return yclient + ", " + random_secret + ", " + prime + ", " + int_val;
    }

    randomNumber()
    {
        Random ran = new Random();
        return ran.Next(Constants.RANDOP_VALUE);
    }

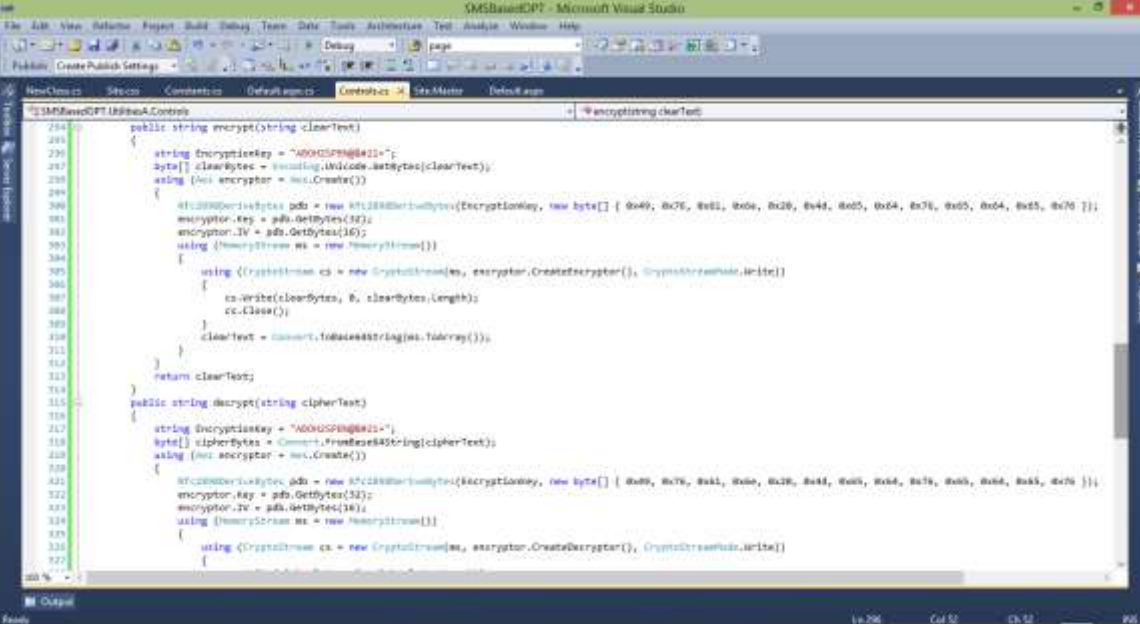
    randomNumberServer(int n)
    {
        return randomNumber();
    }

    computePrivateKey(int public_key, int random_secret)
    {
        int prime = Constants.PRIME_NUMBER;
        BigInteger client = (BigInteger) Math.Pow(public_key, random_secret);
        BigInteger yclient = client % prime;
        return yclient;
    }

    generateOTP()
    {
        Random ran = new Random();
        int otp = ran.Next(100000, 999999);
        return otp;
    }

    makeBinaryEqual(string otp_binary, string skey_binary)
    public BigInteger[] takeXOR(string binary_OTP, string binary_Secret)
    public string sendSMS(string otp, string phone)
    
```

Figure 6 Codes to implement DHKE algorithm



```
public string encrypt(string clearText)
{
    string encryptionKey = "A00G0P0H@0011=";
    byte[] clearBytes = Encoding.Unicode.GetBytes(clearText);
    using (Aes encryptor = Aes.Create())
    {
        RijndaelCipherKey key = new RijndaelCipherKey(EncryptionKey, new byte[] { 0x49, 0x7E, 0x41, 0x04, 0x20, 0x44, 0x05, 0x04, 0x76, 0x05, 0x04, 0x05, 0x76 });
        encryptor.Key = key.GetBytes(32);
        encryptor.IV = key.GetBytes(16);
        using (MemoryStream ms = new MemoryStream())
        {
            using (CryptoStream cs = new CryptoStream(ms, encryptor.CreateEncryptor(), CryptoStreamMode.Write))
            {
                cs.Write(clearBytes, 0, clearBytes.Length);
                cs.Close();
            }
            clearText = Convert.ToBase64String(ms.ToArray());
        }
    }
    return clearText;
}

public string decrypt(string cipherText)
{
    string encryptionKey = "A00G0P0H@0011=";
    byte[] cipherBytes = Convert.FromBase64String(cipherText);
    using (Aes encryptor = Aes.Create())
    {
        RijndaelCipherKey key = new RijndaelCipherKey(EncryptionKey, new byte[] { 0x49, 0x7E, 0x41, 0x04, 0x20, 0x44, 0x05, 0x04, 0x76, 0x05, 0x04, 0x05, 0x76 });
        encryptor.Key = key.GetBytes(32);
        encryptor.IV = key.GetBytes(16);
        using (MemoryStream ms = new MemoryStream())
        {
            using (CryptoStream cs = new CryptoStream(ms, encryptor.CreateDecryptor(), CryptoStreamMode.Write))
            {
                cs.Write(cipherBytes, 0, cipherBytes.Length);
                cs.Close();
            }
            clearText = Convert.ToBase64String(ms.ToArray());
        }
    }
    return clearText;
}
```

Figure 7 Codes to encrypt and decrypt data using AES

VII CONCLUSION AND FUTURE WORK

In this paper, we presented two algorithms to enhance SMS based One-Time-Password from Man-in-the-Middle attacks and other unauthorized users in e-commerce. We implemented DHKE and AES algorithms to prevent confidential data attacks by MITM in an open network environment during a financial transaction. With the integration of the two algorithms, communication between the user and the online payment solution providers will be conducted in an encrypted form and hence making it difficult for MITM to intercept and manipulate the data exchanged by the actual parties involved in a genuine financial transaction. Therefore, this security framework enhances OTP approach developed by Online service providers such as banks to prevent sensitive information attacks by a non-trusted user.

As technology changes, fraudsters acquire new techniques to gain access to users confidential data in an online financial transaction. In addition, it requires using greater effort or exertion to track the pattern and behaviour of deceitful transactions. Future research will investigate this designed security model to determine where and how the system can be enhanced to prevent online financial transactions from MITM attacks.

References

- [1] Nazreen M. and Munawara S. (2013). A Comprehensive Study of Phishing Attacks. International Journal of Computer Science and Information Technologies, 4(6), 783-786.
- [2] Man-in-the-middle attack From Wikipedia, the free encyclopedia. Accessed 23/04/2016 from https://en.wikipedia.org/wiki/Man-in-the-middle_attack.
- [3] One-time password From Wikipedia, the free encyclopedia. Accessed 23/04/2016 from https://en.wikipedia.org/wiki/One-time_password.
- [4] Abdul S., Rabah N. and Hussam J. (2011). Hybrid Model For Securing E-Commerce Transaction. International Journal of Advances in Engineering & Technology, 1(5), 14-20
- [5] Kumar N. and Chaudhary P. (2015). Prevention Technique from Hackers and Trackers in on-line-Transactions. Research Journal of Recent Sciences, 4(IVC-2015), 53-59
- [6] Rupali S. and Unmukh D. (2015). Two Way Authentication in MITM Attack to Enhance Security of E-commerce Transactions. International Journal of Security and Its Applications 9(9), 265-274
- [7] Safa H., Varsha N. and Jayashri M. (2014). Securing SMS Based One Time Password Technique from Man in the Middle Attack. International Journal of

- Engineering Trends and Technology (IJETT)-11(3), 154-158
- [8] Jason W., Damien H. and Justin P. (2006). Enhanced Security for Preventing Man-in-the-Middle Attacks in Authentication, Data Entry and Transaction Verification. Proceedings of the 6th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia
- [9] Raymond, J.F. and Stiglic, A. (2000) Security Issues in the Diffie-Hellman Key Agreement Protocol. [Online] Available at <http://crypto.cs.mcgill.ca/~stiglic/publications.html> [Accessed 17 March 2016].
- [10] Prerana C., Vikas K., S K. (2014) Security Enhancement Algorithms for Data Transmission in 4G Networks. International Journal of Applied Information Systems. ISSN : 2249-0868
- [11] Niranjnamurthy M., Kavyashree N., Jagannath S. and Dharmendra C. (2013). Analysis of E-Commerce and M-Commerce: Advantages, Limitations and Security issues. *International Journal of Advanced Research in Computer and Communication Engineering* 2(6), 2360-2370
- [12] Rinky D. P. and Dheeraj K. S. (2013) Credit Card Fraud Detection & Prevention of Fraud Using Genetic Algorithm . *International Journal of Soft Computing and Engineering (IJSCE)*, 2(6), 292-294
- [13] Jyoti R. G., Amruta B. D., Harshada V. S., Snehal V. P. and Rinku A. B. (2014) Credit Card Fraud Detection using Decision Tree Induction Algorithm. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 4(6), 66-69