COMPARATIVE ANALYSIS OF DIFFERENT ENCRYPTION TECHNIQUES IN MOBILE AD HOC NETWORKS (MANETS)

Amal Ahmad, Andraws Swidan and Ramzi Saifan

Computer Engineering Department, University Of Jordan, Amman, Jordan

ABSTRACT

In this paper a detailed analysis of Data Encryption Standard (DES), Triple DES (3DES) and Advanced Encryption Standard (AES) symmetric encryption algorithms in MANET was done using the Network Simulator 2 (NS-2) in terms of energy consumption, data transfer time, End-to-End delay time and throughput with varying data sizes. Two simulation models were adopted: the first simulates the network performance assuming the availability of the common key, and the second simulates the network performance including the use of the Diffie-Hellman Key Exchange (DHKE) protocol in the key management phase. The obtained simulation results showed the superiority of AES over DES by 65%, 70% and 83% in term of the energy consumption, data transfer time, and network throughput respectively. On the other hand, the results showed that AES is better than 3DES by approximately 90% for all of the performance metrics. Based on these results the AES was the recommended encryption scheme.

KEYWORDS

MANET, AES, DES, Key management.

1. INTRODUCTION

In recent years, MANETs emerged as a major next generation wireless networking technology. However, the security issues on MANET have become one of the primary concerns. MANETs are vulnerable to attacks more than wired networks. As a result, attacks with malicious goals will always devise to exploit these vulnerabilities and to disrupt the MANET operation. The problem posed by potential breaching of the systems by passive observations and masquerading is further complicated by the varying nature of the wireless environment [1].

Security is provided through security services such as confidentiality. The goal of confidentiality is to control or restrict access to sensitive information to the only authorized individuals. MANET uses an open medium, so usually all nodes within the transmission range can obtain the data. One way to keep information confidential is to use data encryption schemes. Moreover, compromised nodes may be a threat to confidentiality if the cryptographic keys are not encrypted and stored in the node [2]. Another challenge when it comes to MANET security is the key management issue. In order to prevent the malicious nodes from joining in the networks, it's necessary to authenticate the nodes when they are joining in. Due to the restricted energy and computational capability of MANETs, it's necessary to design a light weight and storage efficient key management scheme [3] [4].

Numerous security solutions, key management and cryptographic techniques have been designed to support MANET, some of them are adapted to fit the network requirements (minimum delay, minimum power consumption and maximum throughput) while others are known to be

DOI : 10.5121/ijcnc.2016.8208

computationally demanding. They consume a considerable amount of computing resources such as bandwidth and power [5]. There is no enough information about the efficiency of incorporating different encryption techniques in Ad hoc networks. This study was done to investigate DES, 3DES and AES encryption techniques efficiency and suitability for MANETs. Table 1 shows a comparison between these encryption techniques according to [6].

Table 1. Comparison between DES, 3DES and AES

Factors	**DES**	**3DES**	**AES**
Key Length	56 bits	(k_1,k_2 and k_3) 168 or 112 bits	128,192 or 256 bits
Block Size	64 bits	64 bits	128,192 or 256 bits
Possible Keys	2^{56}	2^{168} or 2^{112}	2^{128} , 2^{192} or 2^{256}
Time Required to Check All Possible Keys at 50 Billion Keys per Second	400 days	For a 112 bits key: 800 days	For a 128 bits key: 5 x 10^{21} years

DH algorithm was the first published public key algorithm by Diffie, and is generally referred to as DHKE. Many commercial products employ this key exchange technique [7]. The purpose of the algorithm is to allow two users to securely exchange a key that can then be used for data encryption. The algorithm itself is limited to the exchange of secret values. The DH algorithm depends for its efficiency on the difficulty of computing discrete logarithms. DHKE algorithm general steps are shown in Figure 1.

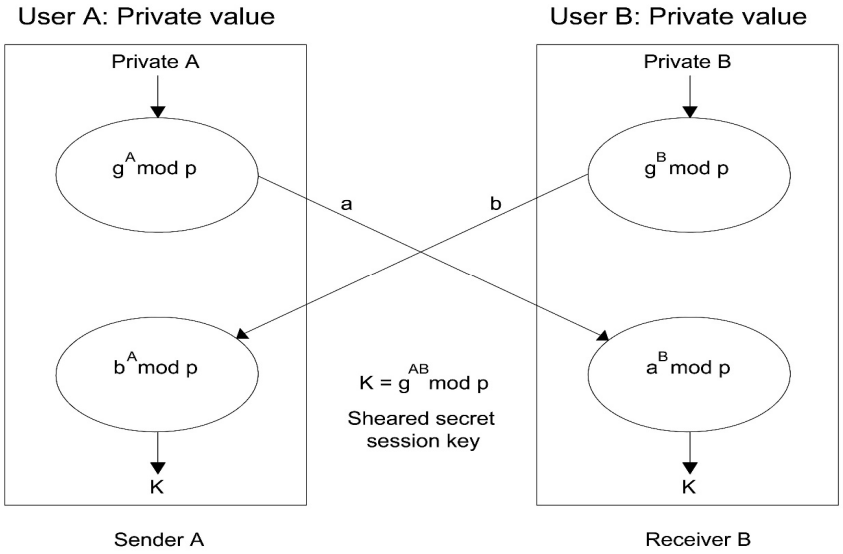

Figure 1. Diffie-Hellman Key Exchange Algorithm General Steps

The rest of this paper is organized as follows; section 2 demonstrates the related work in the field of our study. Section 3 describes the implementation procedure of the cryptographic schemes in NS-2. Section 4 contains experimental results. Finally, this paper is concluded in section 5.

2. RELATED WORK

MANET security issues are very common topic. We will survey some research efforts in this topic. Some researchers focused on the evaluation of the performance of different encryption schemes, others focused on the key management and distribution issues that precede the actual data encryption.

Mandal, et al. [8] proposed a study that investigated the two most widely used symmetric encryption techniques DES and AES. The encryption schemes had been implemented using MATrix LABoratory (MATLAB) software. After the implementation, these techniques were compared on some points, were theses points avalanched the effect due to one bit variation in plaintext keeping the key constant, avalanche effect due to one bit variation in key keeping the plaintext constant, memory required for implementation and simulation time required for encryption. The authors concluded that the DES encryption algorithm has a disadvantage in term of high memory requirement. Moreover, in AES the avalanche effect is very high so that AES is ideal for encrypting messages sent between objects via unsecured channels, and is useful for objects that are part of monetary transactions, and gave a future direction to include experiments on other types of data such as images.

Umaparvathi and Varughese in [9] presented a comparison of the most commonly used symmetric encryption algorithms AES (Rijndael), DES, 3DES and Blowfish in terms of power consumption. A comparison had been conducted for those encryption algorithms using different data types like text, image, audio and video. The various encryption algorithms had been implemented in Java. In the experiments, the software encrypts different file formats with file sizes (4MB - 11MB). The performance metrics like encryption time, decryption time and throughput had been collected. The presented simulation results showed that AES has a better performance than other common encryption algorithms used. Since AES had not showed any known security weak points in the presented study, this makes it an excellent candidate. 3DES showed poor performance results since it requires more processing power. Since the battery power is one of the major limitations in MANET nodes, the AES encryption algorithm is the best choice.

Sahu and Kushwaha in [10] implemented symmetric key encryption algorithms DES, AES and Blowfish using NS-2 network simulator to compare their performance with different data types like text and image based on some performance metrics. In the experiments, the algorithms encrypt a different file types such as text, image and video sizes (0.3KB - 1KB). The performance metrics like encryption and decryption time, battery consumption, residual battery and throughput had been recorded for each file type. The proposed symmetric key encryption algorithms were implemented using NS-2 (v-2.34) with different packet size, the obtainable simulation results showed that AES is simple and better in term of residual battery and encryption time than other implemented algorithms. Blowfish had better performance in term of throughput, but it consumes more battery power compared with the other implemented algorithms.

Norouzi, et al. [11] focused on the enhancement of security performance in a wireless Ad hoc with an encryption algorithm and transmission rate that predetermined. Simulation had been done using MATLAB the input was text files with minimum size of 50 bytes and maximum size used is 300 bytes, then these data transmitted using two modes; with encryption and without encryption. For the first mode, the data transmitted without using any encryption. Meanwhile for the second method data transmitted with three encryption algorithms; DES, AES and Blowfish. These algorithms were chosen because they were commonly used in previous researches. During the conducted experiments only one key was used to encrypt and decrypt data, which is the largest size key in the particular algorithm. For the encryption, data was encrypted with freeware,

EncryptOnClick for AES Algorithm with 256 bit, Blowfish 2000 for Blowfish algorithm and Kryplite for DES algorithm. Based on the input which is distance and size, time that used to send data to receiver and throughput could be calculate. All of these calculation done in the MATLAB programming and the output produces time of data transfer. Based on the gained results the authors recommended choosing AES to achieve fast delivery of data and high throughput, and choosing Blowfish algorithm when larger size of data sending with smaller transmission rate.

Kashani and Mahriyar in [12] analyzed video streaming characteristics in Ad hoc networks using several cryptography algorithms. The authors presented an application setup for secured video streaming in Ad hoc networks. Public key infrastructure approach was chosen to provide authentication at the network layer. They proposed a fully distributed certification authority (CA) for Optimized Link State Routing (OLSR) based Ad hoc networks. The initial assumption was that the network contains predefined special nodes called shareholders. Shareholders can generate partial signatures. A node joining the network, can obtain a certificate only if it receives at least k partial signatures form k different shareholders ,a shareholder offering service can be identified from the broadcasted HELLO messages. On the other hand, different cryptography schemes were implemented and analyzed in the study; RC4, 3DES, AES-128, AES-256, Salsa20-128 and Salsa20-256 and the time required to encrypt different sizes of data were adopted as a performance metric. The results showed that for RC4, 3DES, AES-128, AES-256, Salsa20-128 and Salsa20-256 took less than 1500 ms to encrypt the 1 MB binary file. 3DES consumes the largest encryption time followed by Salsa20-256, Salsa20-128, AES-256, AES-128 and RC4 respectively.

Sandhiya, et al. [13] proposed an intrusion detection system named Enhanced Adaptive ACKnowledgment (EAACK) which consists of three parts; ACK, Secure ACKnowledgment (S-ACK), and Misbehavior Report Authentication (MRA). All the acknowledgement packets were signed and verified to prevent forged acknowledgement packets. For signing and verifying the acknowledgement packets, keys were generated and distributed in advance. The proposed system uses one-hop ACK which used to enhance the misbehavior of detection rates. To eliminate the requirement of pre-distributed keys the proposed system considered DHKE which depends on the difficulty of computing discrete logarithms and permits user to securely encrypt messages. NS-2 simulator tool was used for running simulation, and the results showed the improvement of misbehavior detection rates which results in lower routing overhead than the existing Intrusion Detection Systems (IDS) when using the DHKE Mechanism.

Du and Xiong in [3] proposed a hop-by-hop authentication and routing driven dynamic key management scheme named HARD-KM. An improved Elliptic Curve Diffie-Hellman (ECDH) protocol with mutual authentication was used to generate two pair keys, which were stored in caches before their expiration. HARD-KM dealing with all nodes in the network equally instead of putting some cluster heads or a base station in the network, the scheme used an off-line certificate authority (CA) to sign certificates and distributed authentication materials matrix for all the mobile nodes.NS2 to simulator was used to evaluate HARD-KM feasibility and efficiency. The results showed that HARD-KM key management scheme was resilient to the adversaries and reduces key storage space. The advantages of the proposed key management scheme were; neighboring pair-wise keys on demand creation to save storage space, the pair-wise keys were derived from an authentication materials matrix to deal with eavesdropping attack and compromised nodes had restricted threats to other uncompromised nodes.

Taneja, et al. [14] proposed a common secret key establishment for symmetric encryption over Ad hoc networks using DH key agreement protocol. The concept can be used to develop a new routing protocol for MANETs to provide maximum security against all kinds of attacks. While DH key agreement protocol uses symmetric system to encrypt the data and an asymmetric system to encrypt the symmetric keys, the authors proposed a protocol consists of five stages; the key

generation and exchange, shared secret creation, encrypting using symmetric key and encrypted data transmission. CrypTool simulator had been used in modeling and testing the DH key agreement protocol which is an open source e-learning application, used in the implementation and analysis of cryptographic algorithms. As a first step in simulation, public parameters must be set. Since the public parameters were freely accessible to all and therefore, not only source and destination are able to access these parameters rather every third party too can observe the same. Once the public parameters set, secret numbers of the source and the destination are chosen by pushing the button choose secrets in CrypTool. Then the source sends the shared key to the destination and vice versa. As a last step, the source and destination create common and secret session key by pushing the button generates common session key in CrypTool.

3. IMPLEMENTATION OF THE CRYPTOGRAPHIC SCHEMES IN NS-2

The implementation of a new security extension and cryptographic schemes are written as a new implementation in the NS-2 [15]. This section discusses the new security agent and functions that been used to simulates the performance of the encryption schemes of our interest. The NS-2 is a popular discrete event simulator developed mainly for networking research. NS-2 is an open source software provides wide simulating network types, network applications, routing protocols, data sources and network elements. In NS-2, the system is modeled as sequential events that take an arbitrary amount of time. NS-2 is designed having two basic building blocks; C++ for the core functionality which handle data processing and the Object TCL (OTCL) for scripting purposes which is simply a special purpose language used for writing control script to run the simulation. In general the protocol implementation requires the C++ language for packet processing. And the use of script language makes the change of simulation configuration faster and freely adjustable with dynamic parameters [15].

NS-2 is also supported with the Network AniMator (NAM) that gives a GUI of the network that is simulated. For MANET, NS-2 provides a large library for Ad hoc routing, topology generators, propagation models, mobility models and data sources. To run any simulation scenario in NS-2, it must be written using TCL script in the OTCL file [15]. Although NS-2 provides numerous design alternatives it does not provide all. Our implemented cryptographic schemes and security extensions was not included in the original NS-2, we have implemented our source codes and compiled executable files and record results based on some network metrics [15].

The security agent file during the security establishment process needs to be feed with the encryption type from the source and destination nodes through the TCL file. The encryption type received from the TCL file attached with the encryption type variable type using the bind statement. When a node receives the encryption type and the key value the actual encryption get started by reading a data file with varying size using the following pseudocode:

```
get pointer to file ("test.txt");
if (not permitted access file)
return (error);
read data items from ("test.txt");
read data as a separate block
test for end of file:
if yes end with read data;
return (done);
```

4.SIMULATION AND RESULTS DISCUSSION

The two main purposes of the implemented encryption schemes performance evaluation we had done in the Ad hoc network were; to perform a brief study of the implemented symmetric encryption performance, and to determine the overhead that the DH algorithm adds to the overall network performance. In this Chapter we will present the simulation results that we had recorded according to different performance metrics.

By considering different sizes of data files (2 KB to 64KB) the DES, 3DES and AES (128 key) encryption algorithms were evaluated in terms of the energy consumption, data transfer time and network throughput. All the implementations were balanced to make sure that the results will be relatively fair and accurate. The Simulation program accepts four inputs: the encryption algorithm, encryption mode, key and an input data file. After a successful execution, the ciphertext generated.

4.1 Simulation Parameters

Along with usual configuration of the wireless network simulation in NS-2, we had set the routing protocol as AODV using the command, set val(rp) AODV the Mac layer, data rate, transmission range, simulation area, simulation time, number of nodes and other details also set in the network configuration TCL file. We used the AODV routing protocol for power optimization, because it requires less control packets. The details of the computer system that we have used to compile NS-2 and run the simulation are presented in Table 2, and the NS-2 simulation parameters that we used in our experiments are shown in Table 3.

Table 2. System Configuration

Processor	Intel® Core ™ Duo CPU 2.1 GHz
Operating System	Redhat version 6.0.52 Linux 2.2.x Kernel
Memory	2 GB
C++ Compiler	gcc version 4.3.0
TCL/TK version	8.4.11
NAM version	1.11
MATLAB version	7.12.0.635 (R2011a)

Table 3. Simulation Parameters in NS-2

Parameter	**Value**
Simulator	NS-2 (V- 2.29)
MAC Layer	802.11 datarate_ 11 MB
Simulation Time	150 sec
Simulation Area	2000 m * 2000 m
Transmission Range	250 m
Routing Protocol	AODV
Packet Size	1 KB
Number of Nodes	10

4.2 Simulation Factors and Metrics

The performance of implemented cryptographic schemes in the Ad hoc network depends upon several factors:

1. Encryption schemes: This study evaluates three different symmetric encryption algorithms; DES, AES (128 key) and 3DES.
2. Number of hops: In the conducted experiments the performance of the implemented cryptographic schemes was evaluated separately upon three main scenarios; a single hop, two hops and three hops between the source and the destination nodes.
3. Data file size: the implemented algorithms encrypt different file sizes; 2KB, 4KB, 8KB, 16KB, 32KB and 64KB.
4. Simulation modes: In our study we applied two simulation modes; the first mode simulates the network behavior assuming the availability of the common key, and the second mode simulates the network behavior including the key management phase in the link sensing between the source and the destination nodes to ensure a reliable and secure key management that precedes the actual encryption.

We have performed several tests on our implemented cryptographic schemes to observe its performance using several performance metrics which are defined in Table 4.

Table 4. Simulation Metrics

Metric	**Definition**
The energy consumption (Joule)	The energy consumption is the average amount of energy consumed by the encryption and decryption during algorithm processing.
The data transfer time (sec)	The time from starting the encryption of the first packet in a selected data file till the end of the decryption of the last encrypted packet that reached the destination node including the End-to-End delay time.
End-to-End delay time (sec)	The time taken for a packet to be transmitted across a network from source to destination.
The network throughput (Kb/sec)	The network throughput that evaluated by dividing the total plaintext size that been encrypted on the total encryption time consumed during encryption.

Performance evaluation assumptions:

1. Free space network with no multipath and/or fading
2. No noise affecting the network
3. 20 repetitions for each experiment

4.3 Results and Discussion

This Section discusses the performance based on the selected metrics upon the varying factors that detailed in the previous section.

4.3.1 Energy Consumption

In our experiments the energy consumption was evaluated using the same technique described in [16]. We present a basic cost of encryption and decryption presented by the product of the total number of clock cycles taken by the encryption and the average current drawn by each CPU clock cycle. The author in [17] showed the cost of some encryption algorithms on Pentium processor as clock cycles per byte, which we used in our calculations as shown in Table 4. To calculate the total energy cost, we divide the cost in Amperes for all encryption and decryption clock cycles by the processor clock speed in cycles/sec. For a Pentium processor the clock speed is 7590 cycle/sec as shown in [18] which used in our calculations as shown in Table 4. The energy cost calculations per byte done using the following equation, and the Energy consumption for different data file sizes are shown in figure 2 for DES, 3DES and AES encryption schemes.

$$E = \frac{(CC/B)}{CS} * I * V$$

Where:

- E is the energy consumption (Joule)
- (CC/B) is the clock cycles/byte during encryption and decryption (Cycles/B)
- CS is the processor clock speed (Cycles/sec)
- I is the current drawn in the total encryption and decryption cycles (Amp)
- V is the processor operating voltage (V)

Table 5. Energy Consumption Results

Algorithm	Clock Cycles/B	Energy (mJoule)
DES	90	117.4
AES	32	42
3DES	216	280

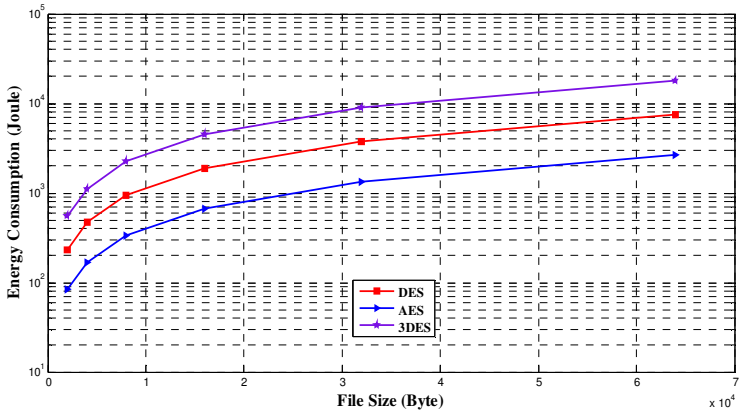

Figure 2 : Energy Consumption for Varying Data File Sizes

In general the results showed the superiority of AES algorithm over DES and 3DES in term of the energy consumption (when encrypt the same data file). Actually, we found that the AES requires approximately 65%, 85% energy less that the energy consumed by DES and 3DES algorithms respectively. DES algorithm consumes approximately 58% energy less than 3DES algorithm

4.3.2 Data Transfer Time

The data transfer time calculations in our conducted experiments were based on the same technique used by [11] which considered as the time from starting the encryption of the first packet in a selected data file till the end of the decryption of the last encrypted packet that reached the destination node including the End-to-End delay time. In order to compute the transfer time the following equation was used:

$$T_r = T_e + T_d + T_{EE}$$
$$T_e \cong T_d \cong \sum_1^{N_p} T_i$$
$$N_p = F_s/P_s$$

Where

- T_r is the transfer time (sec)
- T_e is the encryption time (sec)
- T_d is the decryption time (sec)
- T_{EE} is the End-to-End delay time (sec)
- N_p is the number of packets in single data file
- T_i is the time taken to encrypt a single packet (sec)
- F_s is the data file size
- P_s is the single packet size

For the implemented encryption schemes in our study the transfer time results are shown graphically in Figure. 3.

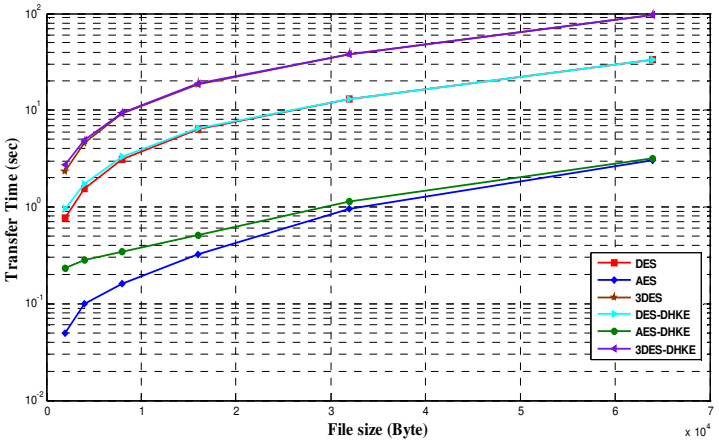

Fig.3 : The Implemented Encryption Schemes Transfer Time Results for the Two Simulation Modes

As we can notice from Figure 4 an advantage of using the AES encryption scheme is that it takes less data transfer time than DES and 3DES encryption schemes. The experimental results showed that the AES transfer time is approximately 90% less than DES encryption when running simulation mode one. On the other hand, AES consumes an approximately 25% transfer time less than DES encryption for small data files and (57%-80%) less than DES for larger data files when applying the DHKE algorithm in simulation mode two applied experiments (loading the same data sizes for both encryption schemes).

4.3.3 Network Throughput

In our study the throughput of the network while running the implemented encryption schemes is calculated using the formula presented by [11], which done by normalizing the total encrypted file size in bytes by the data transfer time using the following formula:

Throughput = size of plain text / time consumed during encryption

For different data file sizes the throughput results while running the two simulation modes are shown in Figure 4.

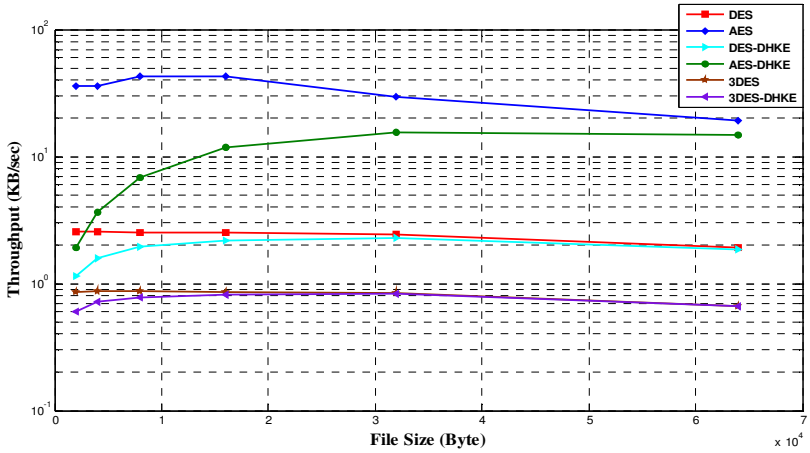

Fig.4 : Network Throughput Results for the Implemented Encryption Schemes

In general, we can notice that the AES throughput was approximately 92% greater than the DES algorithm while running simulation mode one, and approximately 30% when inserting small data file, and ranges from 60% to 80% for large data files when running the simulation mode two by applying the DHKE.

4.3.4 End-to-End Delay Time

The End-to-End delay time in our study measured as the time interval from the moment that the source node sends a first packet of data after encryption procedure completion until the moment that the destination node in the network receives the last encrypted packet. According to the End-to-End delay definition the DHKE transactions adds a certain preprocessing time overhead to the actual End-to-End delay time between source and destination nodes this time is fixed for DES, 3DES and AES because it is related to the transfer packets during session initiation stage and not the actual data encryption. Assuming different number of hops between the source and destination nodes, and using 16KB data file size the End-to-End delay time results are shown in "Fig. 5" for the two applied simulation modes. Generally the file size VS. the percentage of the DHKE overhead is shown in Table 6.

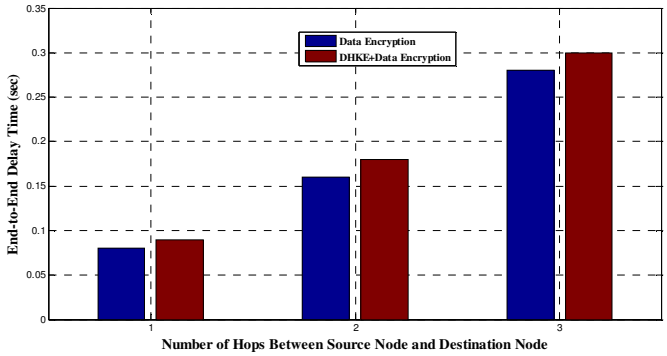

Figure 5 Ad hoc Network End-to-End Delay Time Calculations for 16KB Data

Table 6 The Data File Size VS. DHKE Overhead

File Size (KB)	DHKE Overhead (%)
1	66.7
2	50
4	33.3
8	20
16	11.1
32	5.9
64	3

From the results shown in the above table we can conclude that the overhead caused by applying DHKE protocol to the overall MANET performance is acceptable comparing with its benefits especially for big data files.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this study we tried to evaluate the performance of DES, 3DES and AES symmetric encryption algorithms under MANET environment. On the other hand, we applied a secure key management solution using the DHKE protocol. And finally we offered the ability to choose the encryption type by the user based on the required security level.

Table 7. Performance Evaluation Results Summary

Performance Metric	AES superiority over DES (%)	AES superiority over 3DES (%)	DES superiority over 3DES (%)
Energy Consumption	65	85	59
Transfer Time	70	95	63
Network Throughput	83	95	64

The following conclusions were obtained:

- The results showed the superiority of AES algorithm over DES and 3DES for all parameters of the performance metrics. Analysis results are summarized in Table 7 shown bellow.
- The obtained results seem to be sensible compared with the expected and the results obtained from [11] and [16].
- The overhead caused by applying DHKE protocol to the overall MANET performance is acceptable compared with its benefits especially for big data files, and was approximately 28% in term of processing time during algorithm procedures upon using a 2Kbits prime number.

Security in Ad hoc networks is an open research issue, and investigative work is still ongoing for new security solutions. The cryptographic solutions, and their suitability with Ad hoc limitations, will always be a challenge in order to provide protection from malicious attacks. The followings are some future work suggestions:

- Analyze and evaluate the performance of another symmetric block cipher such as the Blowfish cipher.
- Analyze and evaluate the performance of stream cipher encryption such as the RC4 and SEAL ciphers. A comparative analysis of stream cipher encryption with block cipher encryption is assumed to be valuable.
- Evaluate the performance of the network using another network simulator such as Opnet network simulator in order to validate the obtained thesis results.
- Evaluate the performance of the network with different network topologies.
- Evaluate the performance of the network assuming new nodes joining/leaving the network.

REFERENCES

[1] Nadeem, A. and Howarth, M. P. (2013), A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks. IEEE Communications Surveys & Tutorials, 15(4), 2027-2045.

[2] Chen, J. and Wu, J. (2010), A Survey on Cryptography Applied to Secure Mobile Ad hoc Networks and Wireless Sensor networks. Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice, IGI Global, AH ALTALHI, 5, 2414-2424.

[3] Du, D. and Xiong, H. (2011), A Dynamic Key Management Scheme for MANETs. In Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), IEEE, 1, 779-783.

[4] Mokhtarnameh, R. Muthuvelu, N. Ho, S. B. and Chai, I. (2010), A Comparison Study on Key Exchange-Authentication Protocol. International Journal of Computer Applications IJCA, 7(5), 5-11.

[5] Abdul, D. S. Elminaam, H. M. A. K. and Hadhoud, M. M. (2009), Performance Evaluation of Symmetric Encryption Algorithms. International Journal of Computer Science and Network Security, 8(12), 78-85.

[6] Alanazi, H., Zaidan, B. B., Zaidan, A. A., Jalab, H. A., Shabbir, M. and Al-Nabhani, Y. (2010). New comparative study between DES, 3DES and AES within nine factors. arXiv preprint arXiv:1003.4085.

[7] Stallings, W. (2006), Cryptography and Network Security: Principles and Practice, (5^th ed.). India: Pearson Education.

[8] Mandal, A. K. Parakash, C. and Tiwari, A. (2012), Performance Evaluation of Cryptographic Algorithms: DES and AES. In Electrical, Electronics and Computer Science (SCEECS), 2012 IEEE Students' Conference, IEEE, 1-5.

[9] Umaparvathi, M. and Varughese, D. K. (2010), Evaluation of Symmetric Encryption Algorithms for MANETs. In Computational Intelligence and Computing Research (ICCIC), IEEE International Conference, 1-3.

[10] Sahu, S. K. and Kushwaha, A. (2014), Performance Analysis of Symmetric Encryption Algorithms for Mobile Ad hoc Network. In International Journal of Emerging Technology and Advanced Engineering IJETAE, 4(6).

[11] Norouzi, M. esmaeel Akbari, M. and Souri, A. (2012), Optimization of Security Performance in MANET. Journal of American Science, 8(6).

[12] Kashani, A. A. and Mahriyar, H. (2014), A New Method for Securely Streaming Real-time Video in Ad hoc Networks. Advances in Environmental Biology, 8(10), 1331-1338.

[13] Sandhiya, D. Sangeetha, K. and Latha, R. S. (2014), Adaptive ACKnowledgement Technique with Key Exchange Mechanism for MANET. In Electronics and Communication Systems (ICECS), 2014 International Conference, IEEE, 1-5.

[14] Taneja, S. Kush, A. and Hwang, C. J. (2011), Secret Key Establishment for Symmetric Encryption over Adhoc Networks. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 2).

[15] Fall, K. and Varadhan, K. (2002). The NS Manual. Notes and Documentation on the Software NS2-Simulator.

[16] Elminaam, D. S. Kader, H. M. A. and Hadhoud, M. M. (2009), Energy Efficiency of Encryption Schemes for Wireless Devices. International Journal of Computer Theory and Engineering, 1, 302-309.

[17] Biham, E. (Ed.). (2006), Fast Software Encryption. 4th International Workshop, FSE'97, Haifa, Israel, January 20-22, 1997, Springer, Proceedings (Vol. 1267).

[18] Rhett, (1999), x86 CPU Reference, Part 2. Retrieved May 25, 2015, from http://alasir.com/x86ref/index.html.