Int. J. Advanced Networking and Applications
Volume: 03, Issue: 04, Pages:1245-1252 (2012)

Comparison of RSA-Threshold Cryptography and ECC-Threshold Cryptography for Small Mobile Adhoc Networks

Dr. (Mrs). G.Padmavathi,
Professor and Head
Department of Computer Science
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore-641043
Email: ganapathi.padmavathi@gmail.com
Ms. B. Lavanya
Department of Computer Science
Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore-641043

----------ABSTRACT----------

A mobile ad hoc network is a special type of wireless network in which a collection of mobile hosts with wireless network interfaces may form a temporary network. Without the aid of proper fixed infrastructure, providing secure communications is a big challenge. The strength of the security solutions very much depends on the cryptographic keys used for communication. Efficient key management is an important requirement of such networks. For networks like MANET which are basically constrained networks with minimum resources, identification of suitable asymmetric cryptosystem is a vital one. Hence an attempt has been made in this paper to identity a suitable asymmetric-threshold based cryptosystems for small MANETs. The study focuses on the comparison of Rivest Shamir Adelman-Threshold Cryptography and Elliptic Curve Cryptography Threshold Cryptography in terms of the performance parameters like key generation time, Encryption time, Decryption time and communication cost. Different small network scenarios with variable node sizes and key sizes are experimented and the results show that ECC-TC is the most desirable asymmetric-threshold cryptosystem for small MANET.

Key Words--MANET, Threshold Cryptography, Elliptic Curve Cryptography, RSA.

Date of Submission: September 17, 2011 Date Accepted: November 19, 2011

I. INTRODUCTION

Mobile Ad Hoc Network (MANET) is a collection of communication devices or nodes that wish to communicate without any fixed infrastructure and pre-determined organization of available links. A MANET is referred to as an infrastructure less network, because the mobile nodes in the network dynamically set up paths among themselves to transmit packets. Application of MANET includes battlefield applications, search and rescue operations as well as civilian applications such as e-commerce, business, vehicular services and shopping and other networking applications. Since MANET can be deployed rapidly with relatively low cost, it becomes an attractive option for commercial uses in sensor network applications or virtual classrooms. The main challenges of MANET are Absence of infrastructure, Wireless links between nodes, Limited physical protection, Lack of centralized monitoring, Security, Routing, Quality of Services (QoS) and Reliability. Of which, Security is an important issue for Mobile ad hoc Network. Basic security requirements of MANET are Authentication, Confidentiality, Integrity, Non repudiation and availability. Security is considered as an important requirement due to the reason that many upcoming applications demand high security infrastructure. Key management is the core component of the security infrastructure.

The organization of the paper is as follows: Chapter 2 discusses review of literature; Chapter 3 discusses the proposed method, Chapter 4 gives experimental setup and simulation study and Chapter 5 contains the conclusion of the paper.

II. REVIEW OF LITERATURE

Menezes et al.[9] introduce a set of techniques and procedures to support the establishment and maintenance of keying relationships between authorized parties. A keying relationship is the process by which network nodes share keying material to be used by cryptographic mechanisms. The keying material can include public/private key pairs, secret keys, initialization

parameters, and other secret parameters supporting key management in various instances. Key management should also define methods to revoke keys from compromised nodes and update keys from non-compromised ones.

Edward S. Rogers.[3] points out that key management is an essential cryptographic primitive upon which other security primitives are built. However, none of the traditional key management schemes are suitable for ad hoc networks. They have limited caliber due to non functional on an arbitrary or unknown network topology, or not tolerant to a changing network topology or link failures.

The various key management schemes [1, 2, 7, 8] designed based on cryptographic techniques for MANET are classified and shown in figure 1.

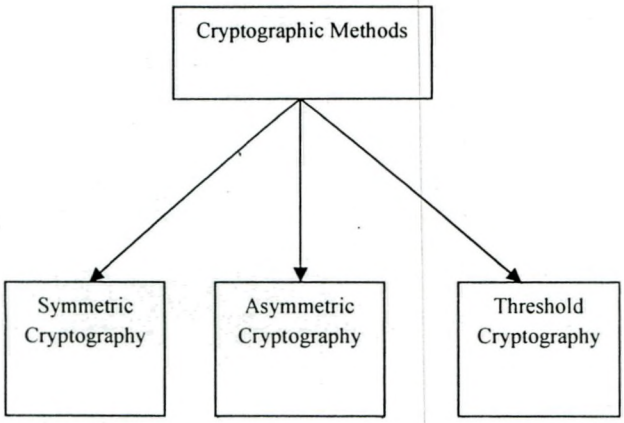

Fig. 1 Key Management techniques for MANET

Symmetric and Asymmetric cryptography are two conventional mechanisms used in a variety of situations. Asymmetric cryptosystems are also known as public cryptosystems and it has been used in some wireless applications. Threshold cryptography is a secret sharing scheme allows a so called dealer to distribute a secret among 'n' parties, where at least $k + 1 \leq n$ of the parties need to collude to reconstruct the secret; 'k' or less secret shares do not reveal any information about the secret. One of the first secret sharing schemes is the (k, n)-threshold scheme proposed by Shamir. This approach is based on the property, that a polynomial of degree k can be described by $k + 1$ data points.

Zhou. Et.al [10] used certificate based cryptography (CBC) and (t,n)-threshold cryptography for MANET. Let N be overall number of nodes and t, n be the two integers of threshold parameters, and $t \leq n < N$. Prior to network deployment, the certificate authority CA's public key is furnished to each node, while each node's private key is divided into 'n' shares, each uniquely assigned to one of 'n' chosen nodes denoted as D-CAs. During network operations, any 't' D-CAs can work together to perform certificate generation and revocation using their secret share, while any less than 't' D-CAs cannot restore the secret key.

L. Ertaul and W. Lu [6] propose a new approach to provide reliable data transmission in MANET with strong adversaries. They have combined Elliptic Curve Cryptography and Threshold Cryptosystem to securely deliver messages in 'n' shares. As long as the destination receives at least 'k' shares, it can recover the original message. The seven ECC mechanisms explored are El-Gamal, Massey-Omura, Diffie- Hellman, Menezes-Vanstone, Koyama-Maurer-Okamoto-Vanstone, Ertaul, and Demytko. For secure data forwarding, they considered both splitting plaintext before encryption, and splitting cipher text after encryption. Keys are exchanged between a pair of mobile nodes using Elliptic Curve Cryptography Diffie-Hellman method. The algorithm is tested using simulation.

According to L. Ertaul and N. Chavan [5] threshold cryptography is sought in computer networks to provide security in terms of availability, confidentiality, and secure key or data distribution. They investigated the difficulties to implement TC in ad hoc networks and propose RSA-based Threshold Cryptography (RSA-TC) for MANET.

Every method discussed in the literature has its own advantages and disadvantages. The suitability of a particular crypto system must be experimented. The next section presents two public cryptosystems that are desirable for MANET.

III. METHODOLOGY

This chapter briefly describes RSA Based Threshold Cryptography (RSA-TC) and Elliptic Curve Cryptography-Threshold Cryptography (ECC-TC)

A. RSA Based Threshold Cryptography (RSA-TC)

RSA is a highly secure, public key encryption algorithm which uses a public key and private key to encrypt and decrypt a message. In public crypto systems there are two different keys: a public key that is released publically so anyone can find it, and a private key is the one that is kept secret. The public key is used to encrypt the message, and private key is used to decrypt it. It is very difficult to find out what private key is used for a public key.

The working of RSA-TC cipher is explained in this section. First a public key and private key pair is randomly generated. As is always the case in cryptography, it is very important to generate keys in the most random and unpredictable manner possible. Then the private key is shred among all the parties. The data is encrypted with public key and decrypted with private key

using the RSA-TC algorithm. The following section shows the steps involved in the RSA-TC.

Generate random numbers

i. Each party 'i' picks two secret numbers pi and qi.

ii. All parties determine whether or not, the sums $p = \sum_{i=1}^{n} pi$ and $q = \sum_{i=1}^{n} qi$ are divisible by any prime number between 0 and some bound. Note that the values of p and q remain totally unknown to all parties.

N – Computation

All parties come together to implement the distributed computation of N as,

$$N = \left(\sum_{i=1}^{n} pi\right)\left(\sum_{i=1}^{n} qi\right)$$

N is public but p1, · · · , pn and q1, · · · , qn remain private.

Private Key Generation

Having computed 'N' and a public encryption exponent 'e', each party now computes its own private additive share, di, of the decryption key 'd' as,

$$d = \sum_{i=1}^{n} di + x^{*}de = 1(modN)$$

The concept of threshold is applied by splitting 'd'. Split the key d into n shares of secret 't', so that at least t+1 cryptographic operations can be successfully performed.

Encryption

In shared RSA scheme, having N = pq, the public components are moduli N and the encryption exponent e is co-prime to N.

$$C = mi^{ei} mod\ N$$

Decryption

Decryption is more complicated as there are more parties who get involved in the scheme. Assuming there are 'n' parties and the prime factors of N remain unknown to every person, each party Pi now only knows the tuple < pi, qi, di > and keeps it secret to any other parties, and they are also required to satisfy the four following conditions:

a. p is an unknown big prime number and

$$p = p1 + p2 + \cdots + pn = \sum_{i=1}^{n} pi$$

b. q is an unknown big prime number and

$$q = q1 + q2 + \cdots qn = \sum_{i=1}^{n} qi$$

c. The unknown decryption exponent

$$d = d1 + d2 + \cdots dn = \sum_{i=1}^{n} di$$

d. $ed = 1(mod\emptyset(N))$.

Each party computes

$$mi = c^{di} mod\ N$$

B. Elliptic Curve Cryptography-Threshold Cryptography (ECC-TC)

Elliptic curve (ECC) is emerging as an attractive public-key cryptosystem for mobile and wireless environments. Compared to traditional cryptosystems like RSA, ECC offers equivalent security with smaller key sizes, which results in faster computations, as well as memory and bandwidth savings.

An ECC-TC operates over points on an elliptic curve. The way that the elliptic curve operations are defined is what gives ECC its higher security at smaller key sizes. An elliptic curve is defined in a standard, two dimensional (x,y) Cartesian coordinate system by an equation of the form:

$$y^2 = x^3 + ax + b(mod\ p)$$

The graph, when plotted with the above equation, turns out to be gently looping lines of various forms. In ECC-TC, the key is not shared because the public key as well as private keys are in form of points. The working of ECC-TC is explained in the following steps:

a. Let the finite field be GF(p) and the elliptic curve be E.

b. Choose randomly a base point (x,y) lying on the elliptic curve.

c. Code the plaintext into an elliptic curve point (xm , ym).

d. Threshold cryptography is applied by splitting the message 'm'. Split the message 'm' into 'n' shares of secret $m_t \quad 1 \leq t \leq n$ Convert each shares m_t to a point on EC. With at least 't' shares of p, if possible to recover message.

e. Each user selects a private key 'n' and compute his/her public key

$$p = n(x, y)$$

For example, user A's private key is nA and the public key is

$$PA = nA(x, y).$$

For anyone to encrypt and send the message point (xm,ym) to user A, he/she needs to choose a random integer k and generate the cipher text,

$$c_m = \{k(x, y), (xm, ym) + kpA \ \}$$

The cipher text pair of points uses A's public key, where only user A can decrypt the plain text using his/her private key.

f. To decrypt the ciphertext Cm, the first point in the pair of Cm, K(x,y), is multiplied by A's private key to get the point : nA(k(x,y)). Then this point is subtracted from the second point of Cm and the result will be the plaintext point (xm,ym). The decryption operation is summarized as below:

g. $((xm, ym) + kPA - nA(k(x, y)\ ) = (xm, ym) + k(nA(x, y)\ ) - nA(k(x, y)\ ) = (xm, ym)$

C. Comparison of ECC-TC with RSA-TC

This section compares ECC public key sizes and encryption lengths with that of RSA. Typical usage scenarios will be used to describe the effect of these on various implementations. The ECC system under consideration will use an odd characteristic 192-bit elliptic curve, which is equal to 1536 bit key size in RSA. The equivalent ECC and RSA key sizes are shown in table 1. The two cryptosystems are implemented and the results are compared in the next section.

TABLE I
EQUIVALENT KEY SIZES IN RSA AND ECC

ECC Key Sizes	RSA Key Sizes
112	512
160	1024
192	1536
224	2048
256	3072

IV. EXPERIMENTATION AND RESULTS

RSA-TC and ECC-TC are implemented using JAVA 1.6.0 in windows environment. Implementation involved simulations of MANET consisting of a sender 'S', receiver 'R' and other nodes called share holders(SH).

For RSA-TC, the prime numbers p and q are generated using available functions in JAVA [4] for key sizes 512, 1024 and 2048 bits. Then the private key 'd' is split using shamir's t-out-of-n scheme to generate partial keys over modulus N. For ECC-TC the parameters a, b and p are taken from accepted NIST curve with 112, 160 and 224 bits. Conversion of message to and from ECC point discussed by kobiltz is used.

Here (tM) mod $p < x < t(M + 1) mod$p where (x,y) is a point on the elliptic curve , 't' threshold, 'p' prime number. The message is retrieved from an ECC point (x,y) using

$$M = x/tmodp$$

Based on 'n' available nodes, the threshold is randomly generated such that $(t \geq (n + 1)$and t< n, where n ≥ 2. In this (n,t) values are fixed as (10,10), (15,15), (20,20) as shown in table 2.

TABLE 2
PARAMETERS USED

Group Sizes	Threshold Values
10	6,7,8,9,10
15	8,9,10,11,12,13,14,15
20	11,12,13,14,15,16,17,18,19,20

A. Key Generation Time Comparison

The results of the experimentation of key generation time for different node sizes, threshold values and key sizes for both RSA-TC and ECC-TC are completed.

The key sizes of ECC 112 are equivalent to RSA 512, the key size of ECC 160 is equivalent to RSA 1024 and the key sizes of ECC 224 are equivalent to RSA 2048. This originally seems that ECC is storage efficient. From the experimental results it is clear that key generation time increases gradually for a given key size with increase in node 'n' and threshold 't'. As the key size increases, the generation time increases exponentially. Figure 2 and table 3 show the key generation time for RSA-2048 and ECC -224. Here RSA-TC shows desirable results compared to ECC-TC.

TABLE 3
KEY GENERATION TIMES IN SECONDS FOR RSA-2048 AND ECC-224.

Key size (t,n)	RSA-2048	ECC-224
(10,10)	1.031	1.04
(15,15)	1.568	1.56
(20,20)	2.031	2.01

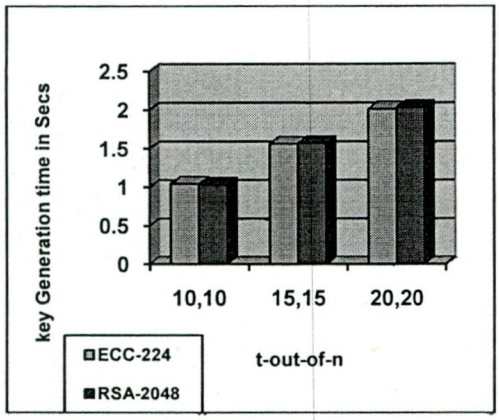

Fig. 2 Key Generation times in secs. for 2048 and 224 bit keys.

B. Encryption Times for Different Key Sizes and Nodes

This section analyses the performance of RSA-TC and ECC-TC for MANET in terms of encryption time for various network sizes and different threshold values. The results of the experimentation of encryption time for different node sizes and threshold values for both RSA-TC and ECC-TC are shown in the following figure 3, figure 4 and figure 5. Tables 4, 5 and 6 show the values due to simulation.

TABLE 4
ENCRYPTION TIMES IN SECONDS FOR RSA-TC WITH KEY SIZE 512 AND ECC-TC WITH KEY SIZE 112 FOR 20 NODES.

(t,n)\key size	RSA-512	ECC-112
(11,20)	0.098	0.089
(12,20)	0.096	0.089
(13,20)	0.097	0.092
(14,20)	0.099	0.095
(15,20)	0.10	0.094
(16,20)	0.099	0.093
(17,20)	0.101	0.092
(18,20)	0.102	0.096
(19,20)	0.102	0.095
(20,20)	0.106	0.097

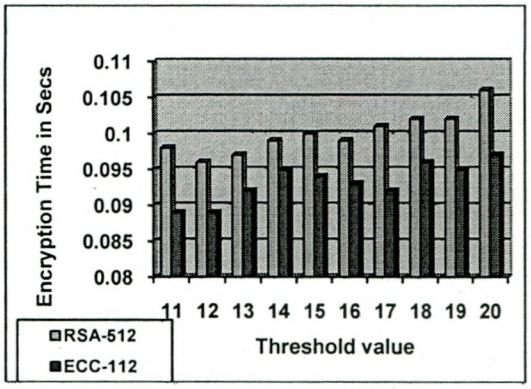

Fig. 3 Encryption times in secs. for RSA-TC and ECC-TC for 20 nodes with key sizes 512 and 112.

TABLE 5
ENCRYPTION TIMES IN SECONDS FOR RSA-TC WITH KEY SIZE 1024 AND ECC-TC WITH KEY SIZE 160 FOR 20 NODES.

(t, n)/ key size	RSA-1024	ECC-160
(11,20)	0.10	0.095
(12,20)	0.099	0.098
(13,20)	0.11	0.1
(14,20)	0.106	0.102
(15,20)	0.105	0.102
(16,20)	0.114	0.1
(17,20)	0.113	0.099
(18,20)	0.113	0.102
(19,20)	0.115	0.102
(20,20)	0.114	0.105

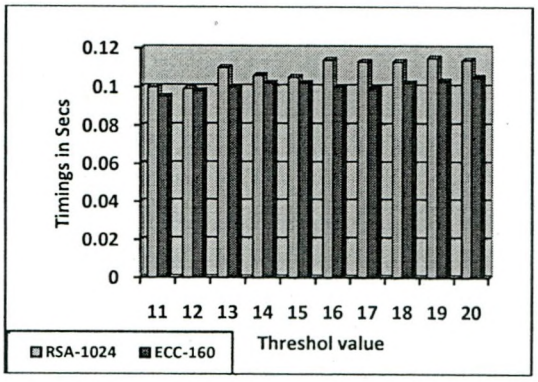

Fig. 4 Encryption times in secs. for RSA-TC and ECC-TC for 20 nodes with key sizes 1024 and 160.

TABLE 6
ENCRYPTION TIMES IN SECONDS FOR RSA-TC WITH KEY SIZE 2048 AND ECC-TC WITH KEY SIZE 224 FOR 20 NODES.

(t, n)/ key size	RSA-2048	ECC-224
(11,20)	0.114	0.112
(12,20)	0.115	0.11
(13,20)	0.117	0.113
(14,20)	0.116	0.113
(15,20)	0.117	0.114
(16,20)	0.12	0.117
(17,20)	0.123	0.117
(18,20)	0.124	0.117
(19,20)	0.123	0.118
(20,20)	0.124	0.12

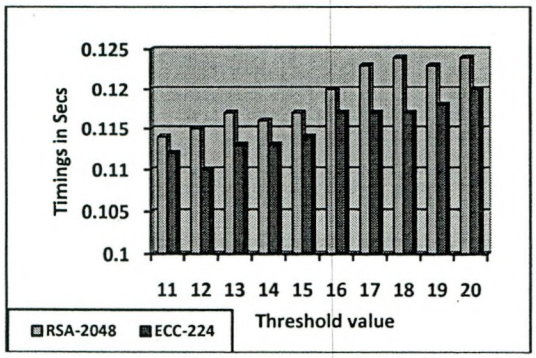

Fig. 5 Encryption times in secs. for RSA-TC and ECC-TC for 20 nodes with key sizes 2048 and 224.

From the experimental results it is clear that encryption time increases gradually for a given key size with threshold t= 6 to 20 and node sizes from 10, 15 up to 20. Further increasing key size results in exponential increase in these timings for a given 'n' and ' t'. ECC-TC gives desirable results compared to RSA-TC.

C. Decryption Time for Different Key Sizes and Nodes

This section analyses the performance comparison of RSA-TC and ECC-TC for MANET in terms of decryption time for various network sizes and different threshold values. The results of the experimentation of decryption time for different node sizes and threshold values for RSA-TC and ECC-TC are shown in the following figure 6, figure 7 and figure 8. Tables 7, 8 and 9 show the simulation results.

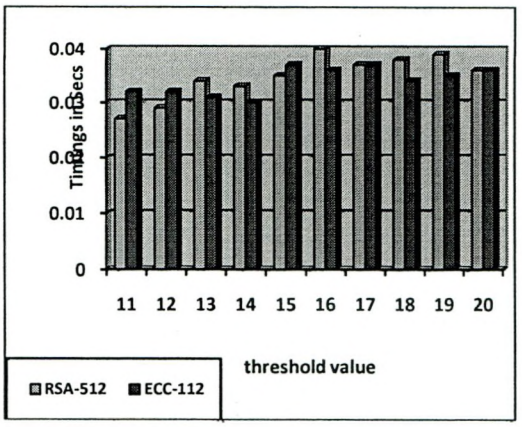

Fig. 6 Decryption times in secs. for RSA-TC and ECC-TC for 20 nodes with key sizes 512 and 112.

TABLE 7
DECRYPTION TIMES IN SECONDS FOR RSA-TC WITH KEY SIZE 512 AND ECC-TC WITH KEY SIZE 112 FOR 20 NODES

(t, n)\key size	RSA-512	ECC-112
(11,20)	0.027	0.032
(12,20)	0.029	0.032
(13,20)	0.034	0.031
(14,20)	0.033	0.03
(15,20)	0.035	0.037
(16,20)	0.04	0.036
(17,20)	0.037	0.037
(18,20)	0.038	0.034
(19,20)	0.039	0.035
(20,20)	0.036	0.036

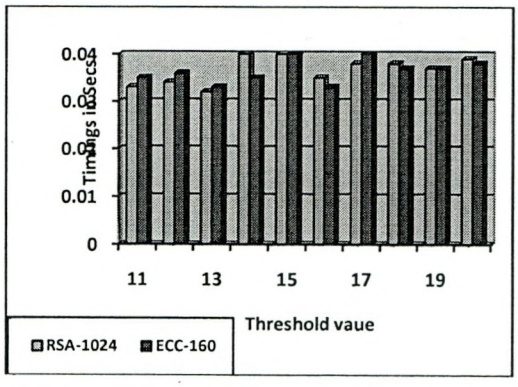

Fig. 7 Decryption times in secs. for RSA-TC and ECC-TC for 20 nodes with key sizes 1024 and 160.

TABLE 8
DECRYPTION TIMES IN SECONDS FOR RSA-TC WITH KEY SIZE 1024 AND ECC-TC WITH KEY SIZE 160 FOR 20 NODES

(t, n)\key size	RSA-1024	ECC-160
(11,20)	0.033	0.035
(12,20)	0.034	0.036
(13,20)	0.032	0.033
(14,20)	0.04	0.035
(15,20)	0.04	0.04
(16,20)	0.035	0.033
(17,20)	0.038	0.04
(18,20)	0.038	0.037
(19,20)	0.037	0.037
(20,20)	0.039	0.038

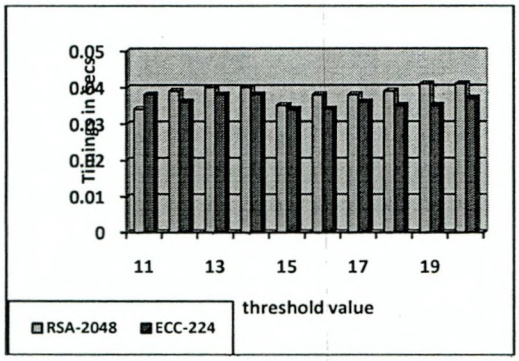

Fig. 8 Decryption times in secs. for RSA-TC and ECC-TC for 20 nodes with key sizes 2048 and 224.

TABLE 9
DECRYPTION TIMES IN SECONDS FOR RSA-TC WITH KEY SIZE 1024 AND ECC-TC WITH KEY SIZE 160 FOR 20 NODES.

(t, n)/ key size	RSA-2048	ECC-224
(11,20)	0.034	0.038
(12,20)	0.039	0.036
(13,20)	0.04	0.038
(14,20)	0.04	0.038
(15,20)	0.035	0.034
(16,20)	0.038	0.034
(17,20)	0.038	0.036
(18,20)	0.039	0.035
(19,20)	0.041	0.035
(20,20)	0.041	0.037

In all the test cases, there are certain fluctuations in the results. Though initially RSA showed better results in terms of decryption time, in all other cases ECC-TC had given better results.

D. Communication Cost Analysis for Varying Number of Nodes

This section analyses the communication cost for RSA-TC and ECC-TC in terms of varying transmission ranges in different networks. The experiments are conducted for the node size varying from 10, 15 and 20 and transmission range from 50 to 100. The results are shown in figure 9 and table 10.

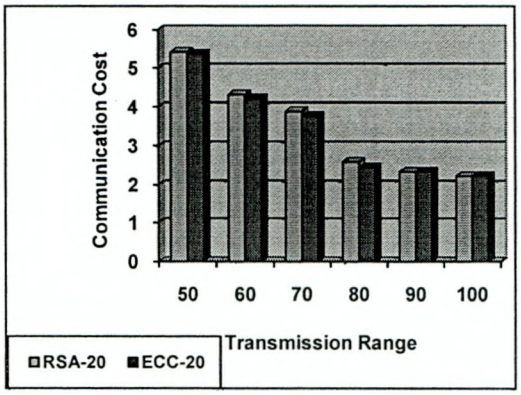

Fig. 9 Communication Cost for RSA-TC and ECC-TC for 20nodes

TABLE 10
COMMUNICATION COST FOR RSA-TC AND ECC-TC FOR 20 NODES

Nodes Transmission Range	RSA-20	ECC-20
50	5.407	5.360
60	4.313	4.731
70	3.876	3.756
80	2.581	2.436
90	2.328	2.331
100	2.219	2.227

When the transmission range is low the communication cost becomes high. So it is observed that the ECC-TC key management scheme has low communication cost for small MANET suitable for constrained devices. However, there are certain fluctuations observed in the table. Generally, the transmission range will be low for constrained devices. Hence taking that aspect into consideration, ECC-TC can be taken as the desired option for MANETs.

By comparing the performance of different schemes, (i.e) the execution times of the key generation, encryption and decryption it is necessary to find which key size provides a comparable level of security. From the results, it is evident that in most of the situations, the performance of ECC outperforms the performance of RSA. Hence ECC is the desirable choice for implementing asymmetric cryptosystems in MANETs. The recommended RSA key size for most application is 2048 bits whereas, for equivalent security using ECC, the key size is 224 bits.

Considering timings for RSA-TC and ECC-TC algorithms, it is observed that with increasing key size and node size (t=6 to 20, n=10, 15 and 20), the encryption and decryption timings increases gradually.

While comparing the performance of RSA-TC and ECC-TC algorithms, it is obvious that RSA-TC is expensive in terms of key generation timings and total encryption timings irrespective of 'n' and 't' values as compared to ECC-TC. With increase in key size for ECC-TC the security provided increases significantly. The increase in the timing is gradual as the key size and 'n' increase. As against this, the timings in RSA-TC increase exponentially with increase in key size.

ECC is known to provide equivalent security as RSA at much smaller key sizes. ECC-TC would also provide equivalent security as RSA-TC. It is also evident that ECC-TC is much efficient algorithm compared to RSA-TC. Due to smaller key size, the storage requirements during the encryption are very less for ECC-TC compared to RSA-TC. This would result in less bandwidth consumption during transmission which is a vital requirement of constrained devices. Hence ECC-TC is communication efficient also. The experimental results strongly acknowledge the above discussions.

V. CONCLUSION

In this work an attempt has been made to compare the performance of two asymmetric cryptographic systems Rivest-Shamir-Adelman based threshold cryptography (RSA-TC) and Elliptic Curve cryptography – threshold cryptography (ECC-TC). The comparison is done based on the efficiency analysis with key generation, encryption, decryption and communication overheads for small MANETs. The conclusion derived from the experimental work is that the security of Elliptic Curve Cryptosystem depends on the efficiency of finite field and the size of the ECC-TC key compared to size of RSA-TC key is less but still provides a similar level security. This concept has been proved through experimentation using simulation with different network sizes for different thresholds. As a further scope, the fluctuations in decryption cost may be studied with respect to other network parameters.

REFERENCES

[1]. Adi Shamir,"*How to Share a Secrect*", Communication of the ACM, vol.22,no.11, Nov 1979, pp. 612-613.

[2]. Burmester,M and Van Le,T,"*Secure Communication in Ad hoc Networks*",Proceedings of the IEEE Workshop on Information Assurance and Security', West Point, NY, pp. 234–241, June 2004. (pdf version) http://www.cs.fsu.edu/burmeste/iaw04.pdf

[3]. Edward S.Rogers, "*Distributed Symmetric Key Management for Mobile Ad hoc Networks*", IEEE Transactions on Information Theory, 2004, pp-2414-2424.

[4]. Eodh, K "*Elliptic Curve Cryptography: Java Implementation,*" Proceedings of the 1st Annual Conference on Information Security curriculum development, October 2004, pp. 88-93.

[5]. Ertaul. L and N. Chavan, "*Security of Ad Hoc Networks and Threshold Cryptography*", in MOBIWAC 2005.

[6]. Ertaul. L and W. Lu, "*ECC Based Threshold Cryptography for Secure Data Forwarding and Secure Key Exchange in MANET (I),*" Networking 2005, LCNS 3462, University of Waterloo, Canada, May 2005, pp. 102-113.

[7]. Fokine, K., "*Key Management in Ad hoc Networks*",LITH-ISY-EX-3322-2002. 2002-09-11.

[8]. Gemmell P. S. "*An Introduction to Threshold Cryptography*", *Cryptobytes,* 1997, pp. 7-12.

[9]. Menezes .A., Oorchat.P., and Vanstone .S Hand book of Applied Cryptography, CRC Press,1996.

[10]. Zhou,L. and Haas, J.Z., "*Securing ad hoc networks*" .IEEE Networks, vol.13no.6,1999, pp.24-30.

Authors Biography

Dr Padmavathi Ganapathi is the professor and Head of the department of Computer Science in Avinashilingam Institute for Home Science and Higher education for Women, Coimbatore. She has 24 years of teaching and industrial experience. She has authored around 165 papers at National and International levels. She has 5 funded projects with UGC, DRDO. She is a life member of many professional organizations. She is serving as resource person at Regional, National and International level in different organizations in different capacities. Her areas of interest include Network Security, Cryptography, Real time Communication and Image Processing.

B. Lavanya is the Research scholar and Research Associate of the Department of Computer Science in Avinashilingam Institute for Home Science and Higher education for Women, Coimbatore.