International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7443-7456

http://www.ripublication.com

A Hybrid Approach for Secure Data Communication in Cluster Based WSN

R.Kayalvizhi[1] and V.Vaidehi[2]

1 & 2 Department of Electronics Engineering, MIT Campus,
Anna University, Chennai,
Tamil Nadu, India – 600044.
kavikkayal@annauniv.edu, vaidehi@annauniv.edu

Abstract

Wireless Sensor Networks are tiny smart devices that are used to sense the physical environment and send the sensed readings to the base station via the intermediate nodes through multi-hop routing. Since they function without any human interaction ones they are deployed they are subjected to various kinds of attacks. Various cryptographic algorithms can be used to prevent these nodes from attacks thus ensuring the confidentiality and integrity of data. The security of the network depends on the strength of the key. In this paper we present a hybrid and efficient key management scheme between the nodes through ECC (Elliptic Curve Cryptography)and ECDH-Elliptic curve Diffie Hellman so as to prevent adversary nodes getting joined in the network and secure data transmission using AES through CBC- Cipher Block Chaining mode encryption and decryption.

Keywords: AES, CBC, ECC, ECDH, Estimation Factor, EWMA, Key Management, TinyViz

1 Introduction

With the rapid growth of Internet and wireless network technologies, many communication services have become the focus for future developments in the military operations, law enforcement, and disaster response domains. Since both Internet and wireless communications are transported over what is considered insecure transmission media, the messages have to be encrypted to prevent eavesdroppers or unauthorized users from capturing the messages. Therefore, secure communications has become a critical design factor when designing networks for the future. However, there are no established guidelines that identify best practices for deploying secure communications in new network topologies. Security services based

on cryptographic mechanisms assume cryptographic keys will be distributed to the communicating parties prior to secure communications.

The secure management of these keys is one of the most critical elements when integrating cryptographic functions into a system, since even the most detailed security concept will be ineffective if the key management is weak. Key Management is the most critical factor of secure communication regardless of the application. Designing and implementing any kind of security mechanism requires a shared secret (usually called the cryptographic key) to construct a trust relationship between two or more communicating parties. Managing these cryptographic keys play a vital role in providing reliable, robust, and secure communication.

Existing key management solutions are primarily developed based on conventional network topologies which are fixed and wired. In such networks, the infrastructure provided supports the underlying mechanisms required for effective key management. In contrast, wireless ad hoc networks by definition have no fixed infrastructure elements. Moreover, the nodes of wireless ad hoc networks, especially sensor network, have several limitations such as memory storage and computational capabilities. These inherent disadvantages make it difficult to employ the tradition solution such as the solution based on a Public Key Infrastructure (PKI). The nodes in a wireless ad hoc network are vulnerable to variety of potential attacks. An adversary only needs to identify and corrupt a single weak node to potentially disrupt the whole network. One way to mitigate this threat is to implement a cryptographic solution with strong key management using Elliptic Curve Diffie Hellman and using Cipher Block Chaining Mode for encryption and decryption of data.

The remaining section of this paper is organized as follows: Section 2 reviews related work; Section 3 tells about the implementation of proposed method. Section 4 consists of experimental results in clustered environment and its comparison with the other methods.

2 Related Work

Keys are important for encryption and decryption. As encryption and decryption algorithms are standards and known to everyone, key generation methodology plays a vital role in security. Keys used for encryption and decryption are of two types. They are symmetric and asymmetric keys.

Symmetric key is nothing but the same key is used at both side (Sender and receiver). Asymmetric key consist of private and public key. Private Key is used for encryption (sender side) and public key for decryption (receiver side) to ensure confidentiality. The disadvantage of symmetric keys are number of keys to be stored are high and scalability is very difficult [5]. Asymmetric algorithm involves high computation but provides high security [9]. Asymmetric key generation algorithms are RSA and Elliptic Curve cryptography (ECC). RSA algorithm is the most widely algorithm. The disadvantage in RSA algorithm is to provide acceptable level of security the key size should be 1024 bits and with the advanced technology the prime number (Private Key) could be factored out. Because of these problems, RSA algorithm could not be used in critical applications like battlefield surveillance as it

may provide a way of revealing the data. In that case, ECC [7] algorithm is preferred and also it is hard to break. The table 1 shows that 160 bit ECC key provides equivalent security of 1024 bit RSA key.

Table 1: Comparison of Key Size

ECC Key Size (bits)	RSA Key Size (bits)	Key Size Ratio
160	1024	1:6
224	2048	1:9
256	3072	1:12
384	7680	1:20
512	15360	1:30

Elliptic Curve Cryptography (ECC) is a public key cryptography [8]. The mathematical operations of ECC is defined over the elliptic curve $y^2 = x^3 + ax + b$ where, $4a^3 + 27b^2 \neq 0$. Each value of the 'a' and 'b' gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. Elliptic curve cryptography is difficult to break because its inverse involves discrete logarithm problem.

Apart from key management scheme the security of data also depends on the encryption/decryption scheme applied in it. The two ways for cipher text generation is using stream cipher and block cipher. Stream ciphers are typically faster than block but are more difficult to implement correctly, and it is more vulnerable to use. Also, stream ciphers do not provide integrity protection or authentication. Because of all the above, stream ciphers are usually best for cases only where the amount of data is small. For larger chunks of data block ciphers are preferred since they can provide integrity protection, in addition to confidentiality. Block ciphers can be broadly classified as Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher feedback (CFB), Output Feedback (OFB) modes as in figure 1.

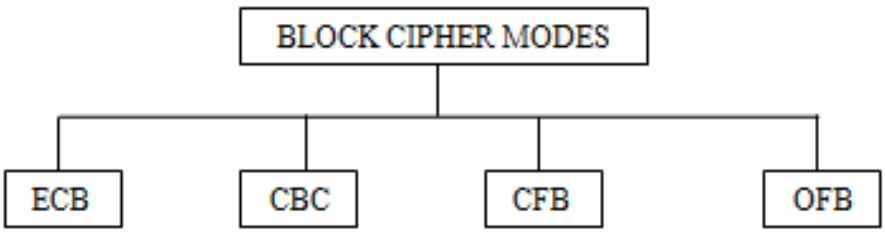


Figure 1: Types of Block Cipher modes

The problem with ECB is that it leaks information. If two messages are encrypted which has two blocks of plaintexts in common, then with ECB mode the

corresponding cipher text blocks will be the same. The hacker can immediately deduce the relationship between the plaintext blocks; even if this doesn't give immediate information about what the plaintext block might be, but it tells something about the plaintext. This doesn't happen with CBC mode; the previous cipher text block (or IV-Initialization Vector) is effectively random (and independent of the plaintext block), and the block cipher is an effectively random string; which makes CBC better than other modes. The different block ciphers are DES, Triple DES, AES, RC5, and Blowfish. AES and Blowfish have the best performance among others. The figure below shows that AES is more efficient in CBC mode [4].

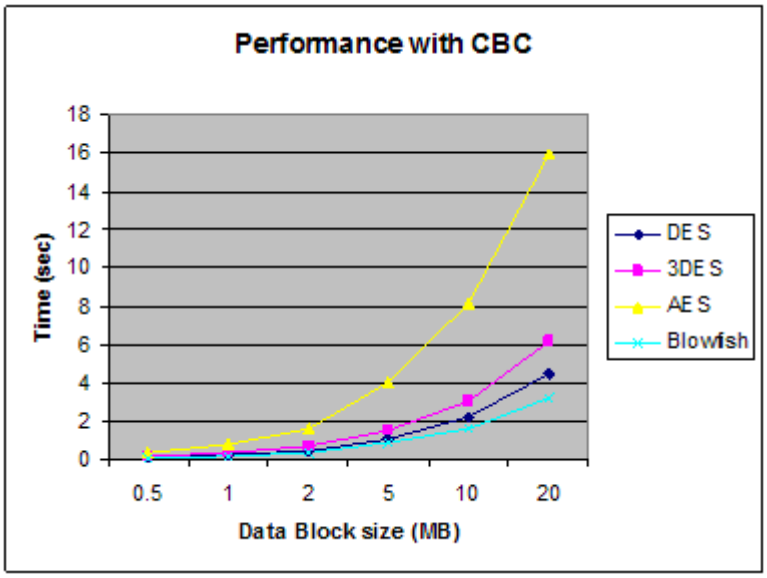


Figure 2: AES performance in CBC mode

3 Proposed Work

The security of data in WSN is a major issue since they are transferred through insecure wireless channels. So there is chance for the data to get lost or injection of false data into the network. In order to prevent these attacks we move on to efficient and hybrid key management which is a combination of both asymmetric key (ECDH) and symmetric key (AES) in CBC mode of block cipher for encryption of data in clustered environment. Clusters are formed in order to prolong the network life time since sensors are low power devices and their batteries are not rechargeable. The architectural diagram of the proposed model is shown in the figure 3 below:

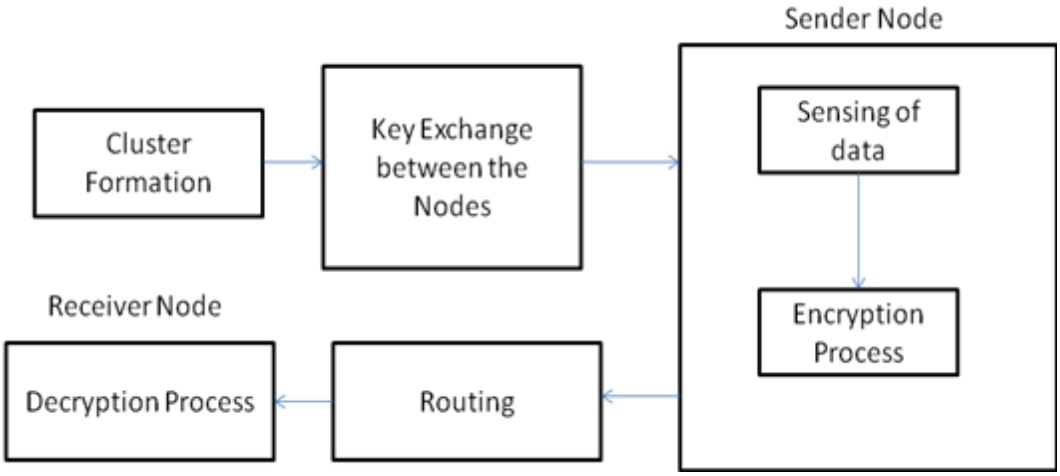


Figure 3: Architectural Diagram

3.1 Cluster Formation:

Once sensors are randomly deployed, clusters are formed in the network. The node that has high link quality becomes the cluster head. This is done with a two way link estimation technique. Node with high link quality becomes the cluster head. This is calculated using EWMA (Exponential Weighted Moving Average) technique where recent samples weigh more when compared to old samples. Nodes which are in the coverage range of cluster head will be included as cluster member. Each node in the cluster will communicate with its cluster head and cluster head in turn will communicate with the base station. Every node in the cluster transmits its location claim to the cluster head.

3.2 Key Exchange:

After clusters are formed key exchange should be done between the nodes and the base station. Key generation algorithm is ECC algorithm. The disadvantage of symmetric key is it requires secure key exchange and more storage; shared key with every individual has to be kept in memory. Asymmetric key is a combination of public and private key. The disadvantage is generation of keys involves higher computation than symmetric key cryptosystem. ECDH is used for shared secret key generation and exchange between the nodes. Elliptic Curve Diffie-Hellman (ECDH) [1] is a key agreement protocol that allows two parties to establish a shared secret key over an insecure channel. Using the public data and their own private data these parties calculates the shared secret. Any third party node, which doesn't have access to the private detail of each device, does not be able to calculate the shared secret key from the available public information. Suppose a node Alice A wants to establish a shared key with another node Bob B, but the only channel available for them may be eavesdropped by a third party. Both have agreed to a common and publicly known curve *K* over a finite field.

1. A randomly chooses k_A, $1 < k < 2^p$ and B accordingly k_B, $1 < k < 2^p$. Now k_A is considered as A's private key, k_B is B's private key.
2. A computes its public key: $T_A = k_A G$, B does: $T_B = k_B G$.
3. A sends T_A to B, B sends T_B to A.
4. A can now compute the shared secret for it and B by equation secret = $k_A T_B$ and B also by secret = $k_B T_A$. An eavesdropper knows only Q_A and Q_B but he is not able to compute the secret from that.

3.3 Encryption /Decryption:

The key generated in the above process will be used for encryption and decryption of data. This is done by Tinysec [2] a link layer security for sensor networks. The advantage is that bandwidth, latency and energy costs of the implementation are minimal for sensor network applications.

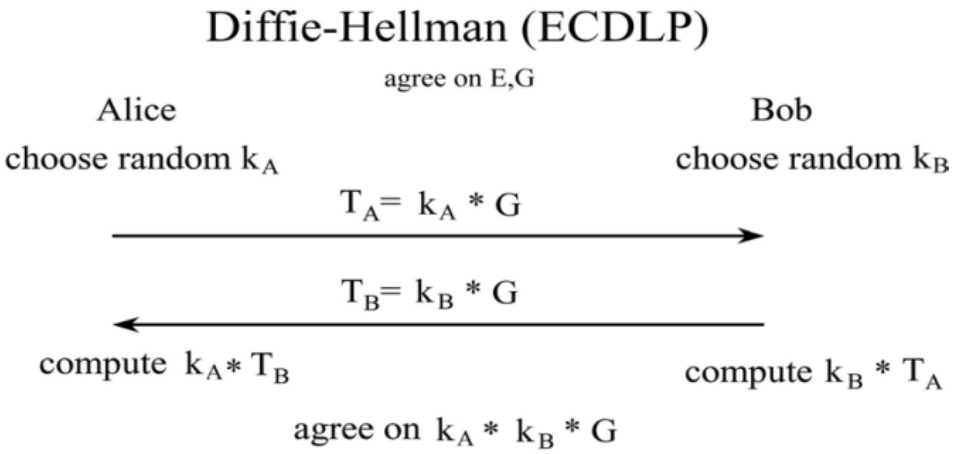


Figure 4: ECDH (Elliptic Curve Diffie Hellman)

Link-layer security mechanisms guarantee the integrity, authenticity and confidentiality of messages between neighboring nodes, while permitting in-network processing. Block cipher modes are useful for encryption of large amount of data. Among various block cipher modes for encryption we use cipher block chaining mode (CBC) because it degrades more gracefully [2] in the presence of repeated IV's (Initialization Vectors). The block diagram for CBC is given below in figure 5.

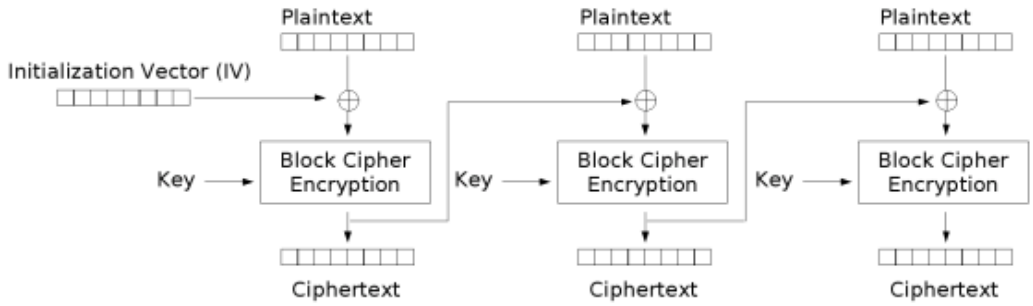


Figure 5: Cipher Block Chaining Mode

For data encryption CBC over AES (Advanced Encryption Standards) is used [3]. The 160 bit key generated by ECDH algorithm is converted to 128 bit using MD5 (Message Digest 5) algorithm. This key is used for AES encryption/decryption of the data.

3.4 Routing:

The routing done here is SCAR (Sensor Context Aware Routing) [10]. It exploits movement and resource prediction techniques to smartly forward data towards the right direction at any point in time and uses probabilistic approach. This approach also can be named as adaptive routing. Mobile neighbor nodes can be the best carriers to forward information to the sink. The two basic routing algorithms used are broadcast (Bcast) and multi-hop routing (MultiHopRouter). Bcast is a flooding protocol. BS floods "commands" to all the nodes, such as to sleep, wakeup or set the interval for how often sensors are read. MultiHopRouter – transfers packets from any sensor to the BS. It forms a dynamic spanning tree. The spanning tree contains paths which have the least number of hops over "reliable" links. Reliable means the best link quality with one of the neighbor nodes. Every 10 secs, spanning tree is refreshed again by Bcast and MultiHopRouter. All nodes are then detected in a network. Every single node tends to send a message to a root node (BS). It takes readings and sends or routes received reading up the tree. Every node memorizes the network address of its parent in the tree and also the depth. Address and depth are specified from received messages. The metric is the hop count over reliable links. In figure 6, link quality is indicated by color and value from 0 to 100. Green and red color indicates best and worst link quality respectively, whereas other color indicates acceptable link quality values.

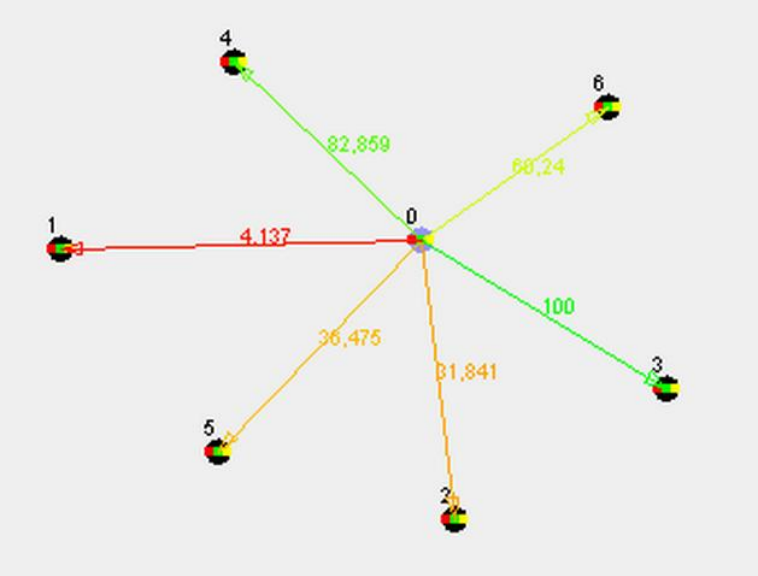


Figure 6: Link Quality Values

For every 50 seconds the routing table is updated. The routing table format is given in table 2.

Table 2. Routing Table Format

Routing Info [in bytes]							
2	2	2	2	2	1	1	1
Id	Parent Node Id	Missed packets	Received packets	Last sequence number	Hop count	Received Estimation Factor	Send Estimation factor

4. Result and Discussion

Fifty nodes are deployed in grid plus random topology. The results are simulated in TinyViz [6] [11]. Clusters are formed based on the link quality between the nodes. The node nearby with best link quality forms a cluster with the help of goodness factor. The node with the best two way link estimation becomes the cluster head. The figure 7 shows the initial deployment of 50 nodes.

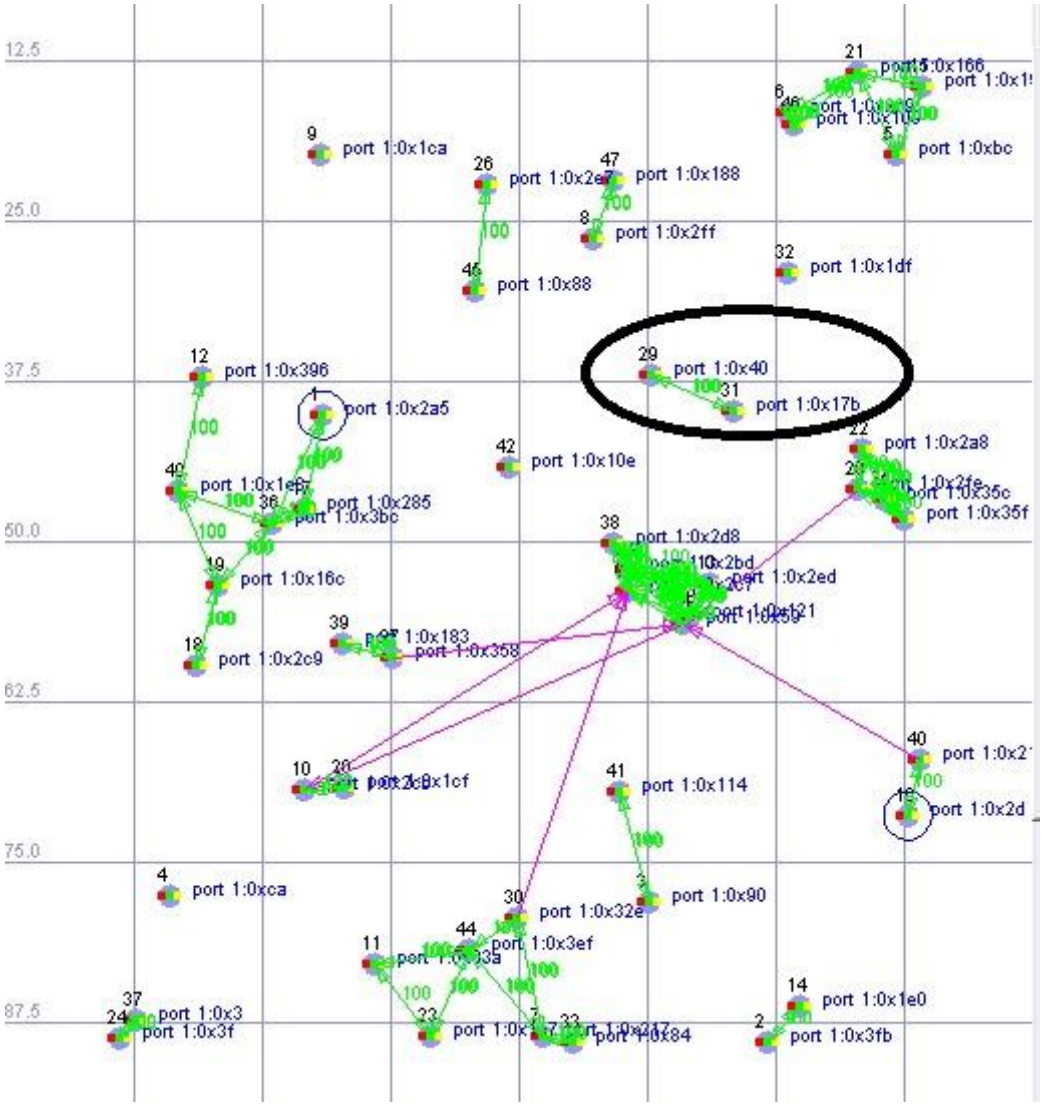
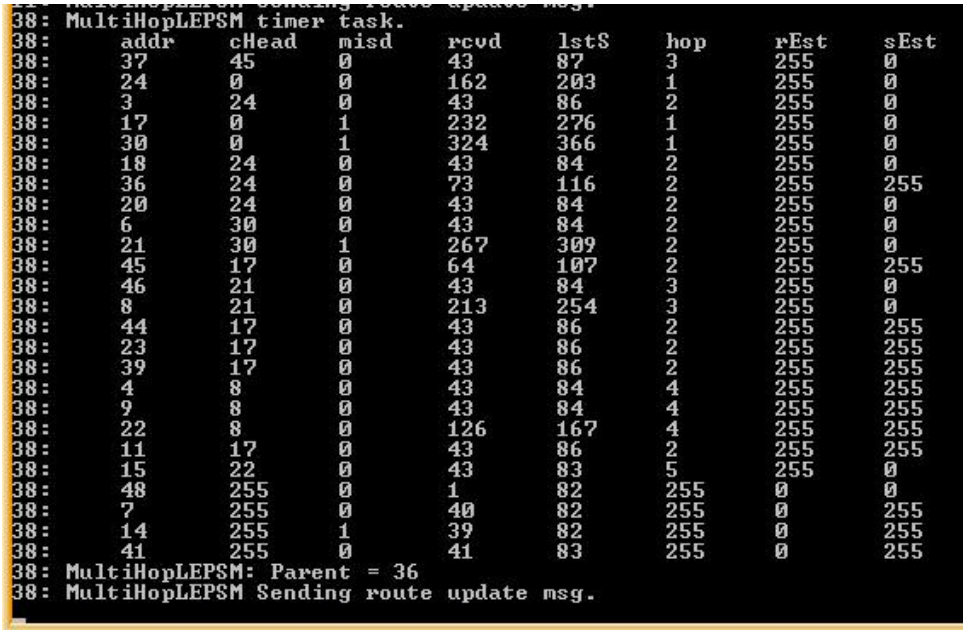


Figure 7: Topology

The pink line in the figure 7 indicates the cluster head communication. The green line indicates the best link quality and cluster members of each cluster. The routing table shows the cluster head, hop count, received, missed packets, sequence number and two way link estimation values for each node. The figure 8 shows the routing table for the 18th node and it gets updated each time the timer fires.



```
38: MultiHopLEPSM timer task.
38:      addr     cHead    misd     rcvd     lstS     hop      rEst     sEst
38:      37       45       0        43       87       3        255      0
38:      24       0        0        162      203      1        255      0
38:      3        24       0        43       86       2        255      0
38:      17       0        1        232      276      1        255      0
38:      30       0        1        324      366      1        255      0
38:      18       24       0        43       84       2        255      0
38:      36       24       0        73       116      2        255      255
38:      20       24       0        43       84       2        255      0
38:      6        30       0        43       84       2        255      0
38:      21       30       1        267      309      2        255      0
38:      45       17       0        64       107      2        255      255
38:      46       21       0        43       84       3        255      0
38:      8        21       0        213      254      3        255      0
38:      44       17       0        43       86       2        255      255
38:      23       17       0        43       86       2        255      255
38:      39       17       0        43       86       2        255      255
38:      4        8        0        43       84       4        255      255
38:      9        8        0        43       84       4        255      255
38:      22       8        0        126      167      4        255      255
38:      11       17       0        43       86       2        255      255
38:      15       22       0        43       83       5        255      0
38:      48       255      0        1        82       255      0        0
38:      7        255      0        40       82       255      0        255
38:      14       255      1        39       82       255      0        255
38:      41       255      0        41       83       255      0        255
38: MultiHopLEPSM: Parent = 36
38: MultiHopLEPSM Sending route update msg.
```

Figure 8: Routing Table

Ones clusters are formed shared secret key is generated for each nodes with the base station using ECC and are exchanged using ECDH algorithm. Figure 9 shows the shared secret key generation for node 17 and the size of the key is 160 bits.

```
17: Generating shared secret,
17: with my private key:
02 b6 8e f3 00 9a db 50 4f 77 f6 f0 ba 9d 53 d7 f2 13 0c e4 d9
17: and  public key:
x:
02 59 fd 51 ce 64 99 3c 3d 05 27 0c 08 87 52 64 49 59 67 00 69
y:
07 69 3d 75 f0 15 96 b1 64 05 20 ff 76 24 9b 8e 8b 12 c6 af 1b
17: OUR SHARED SECRET:
x:
05 a1 06 39 49 62 fe a7 55 31 ed 4c 7f 3c 34 f3 d8 ae 52 02 8a
y:
00 ca cb 2b 68 f2 55 09 85 d3 5b c6 7d 1b 09 6c a8 80 85 80 39
```

Figure 9: ECDH Shared Secret Key Generation

The packets being send to the base station can be viewed through the listener port as shown in figure 10. The packet to be sent to the base station contains all header details along with the payload ie, the sensed readings. For example the readings of

node 29 and 31 are 0x40 and 0x17B respectively as shown in figure 10. In figure 11 the packet contains payload (marked in red) along with its node id in hexadecimal value (marked in blue) and other details. The packet format is given by,

- Destination address (2 bytes)
- Active Message handler ID (1 byte)
- Group ID (1 byte)
- Message length (1 byte)
- Payload (up to 29 bytes):
  - Source mote ID (2 bytes)
  - Sample counter (2 bytes)
  - Channel (2 bytes)
  - Data readings (10 readings of 2 bytes each)

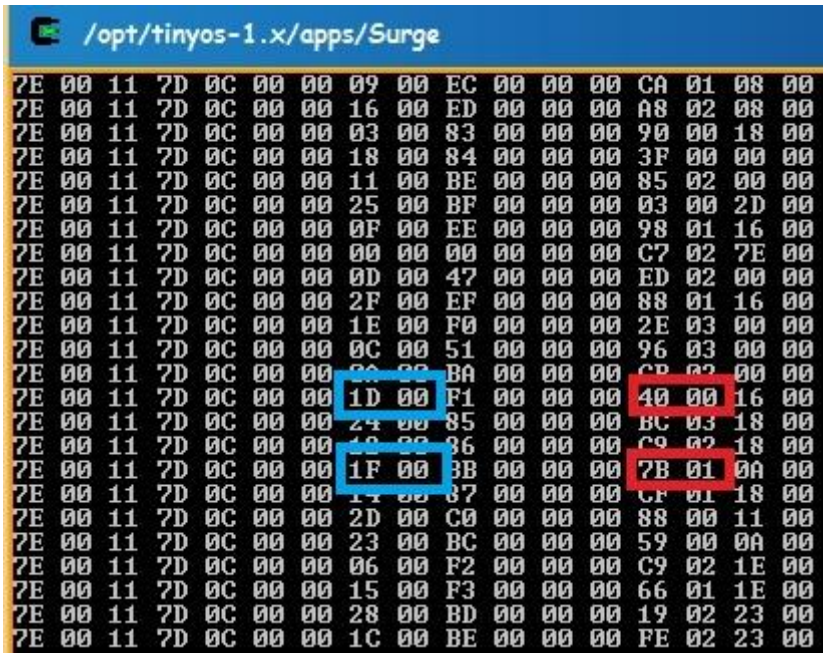


Figure 10: Listener port

The key generated using ECDH is passed to MD5 to permute it to 128 bit key to use it in AES for encryption of the payload. Figure 11 shows the generation of message digest, encrypted data that is being forwarded to the base station and the time taken for encryption.

```
Packet
7E 00 11 7D 0C 00 00 00 00 00 00 00 00 10 01 7E 00
PAYLOAD
0110
MD5: 2a66acbc1c39026b5d70457bb71b142b
Cipher Text generated using AES is : Bx+UgqP0KZo3pFf1+dWMCw
start Time: 20213182838042
stop Time: 20213182906207
Time taken for encryption: 68165 ns


Packet
7E 00 11 7D 0C 00 00 05 00 3F 04 00 00 6F 02 00 00
PAYLOAD
026F
MD5: 3a5d40548bf30afc6af21f0fb450c63b
Cipher Text generated using AES is : S5QTEAoGfsPwkpH4PYbrSw
start Time: 20213183422474
stop Time: 20213183490359
Time taken for encryption: 67885 ns
```

Figure 11: Encrypted Data

With BER introduced through simulating the environment in lossy mode [10] packet loss occurs. The percentage of packet loss in both the ideal and lossy mode for the entire network is given in the figure 12 and the percentage of packet loss for the node in the network in both the modes is given in figure 13.

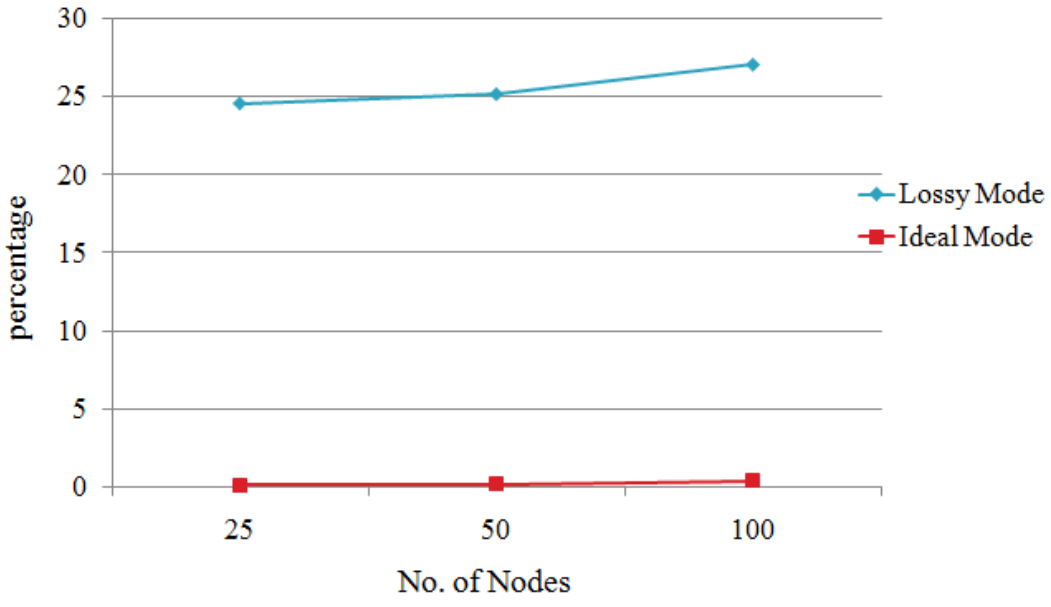


Figure 12: Comparison of packet loss for the entire network

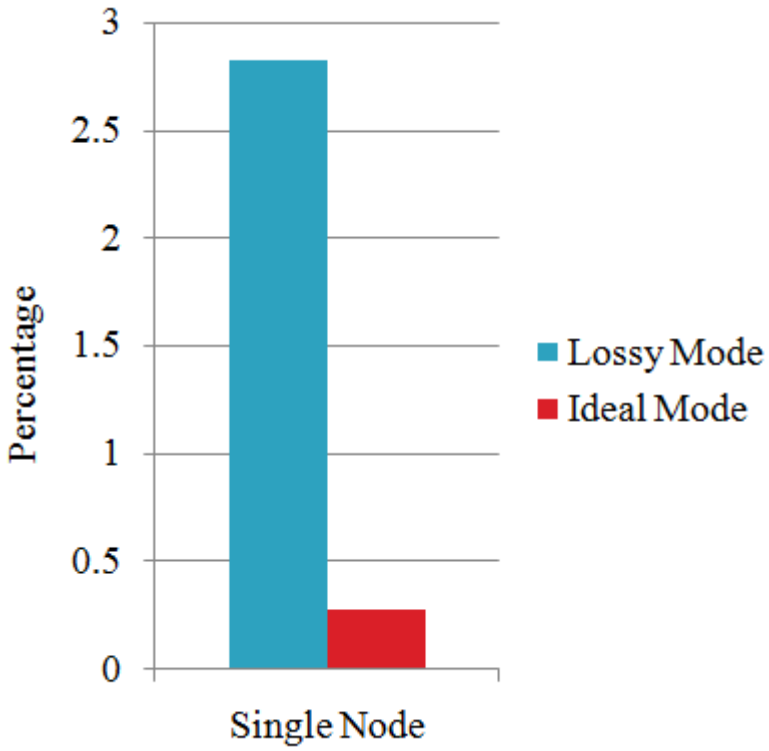


Figure 13: Average packet loss for a single node

5. Conclusion and Future Work

Hybrid key management involving combination of both symmetric and asymmetric key is simulated using TOSSIM simulator for secure data transmission. Parameters like encryption time, average packet loss are calculated. Ecc algorithm used for key generation is involved only in the initial stage; its computation complexity is a bit high even then it is required to provide better security. It is observed that hybrid key management scheme is feasible to employ in real time implementation. Cluster based communication will increase the network lifetime. Ecc algorithm provides better security with smaller key size than other asymmetric algorithms. In future dynamic clusters could be used i.e. cluster head selection could be dynamic and could be implemented in mobile nodes.

References

1. http://www.cs.wustl.edu/~jain/cse56706/ftp/encryption_perf/
2. Jian-wei, Jiang. "Research on key management scheme for WSN based on elliptic curve cryptosystem." Networked Digital Technologies, 2009. NDT'09. First International Conference on. IEEE, 2009.
3. Karlof, Chris, Naveen Sastry, and David Wagner. "TinySec: a link layer security architecture for wireless sensor networks." Proceedings of the 2nd

international conference on Embedded networked sensor systems. ACM, 2004.

4. Lauter K, "The advantages of elliptic curve cryptography for wireless security." IEEE Wireless Communications, vol. 11, pp. 62-67, 200463, 2004.
5. Levis, Philip, et al. "TOSSIM: Accurate and scalable simulation of entire TinyOS applications." Proceedings of the 1st international conference on Embedded networked sensor systems. ACM, 2003.
6. Lukosius, Arturas. "Context Routing in Wireless Sensor Networks."Communication Networks, master project, University of Bremen (2006).
7. Nath, Rudradeep. "A TOSSIM based implementation and analysis of collection tree protocol in wireless sensor networks." International Conference on Communications and Signal Processing (ICCSP), IEEE, 2013.
8. Shah, Pritam Gajkumar, Xu Huang, and Dharmendra Sharma. "Sliding window method with flexible window size for scalar multiplication on wireless sensor network nodes." Wireless Communication and Sensor Computing, 2010. ICWCSC 2010. International Conference on. IEEE, 2010.
9. Smith, Temple F., and Michael S. Waterman. "Identification of common molecular subsequences." Journal of molecular biology 147.1 (1981): 195-197.
10. Srivastava Ankit, Revathi Venkataraman, "AES-128 Performance in TinyOS with CBC Algorithm" International Journal of Engineering Research and Development, June 2013,
11. Wang, Haodong, et al. "Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control."Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on. IEEE, 2008.
12. Xiao, Yang, et al. "A survey of key management schemes in wireless sensor networks." Computer communications 30.11 (2007): 2314-2341.