Improvement of Directed Diffusion Routing Protocol in Wireless Sensor Network Security

Rumpa Das Gupta[1], Khaleque Md Aashiq Kamal[2], Ananya Nag[3]

[1]*Department of Computer Science & Engineering, Faculty of Engineering, Premier University, Bangladesh*

[2] *Department of Computer Science & Engineering, Faculty of Engineering, Premier University, Bangladesh*

[3] *Department of Electrical & Electronic Engineering, Faculty of Engineering, Premier University, Bangladesh*

Email address: rumpa_0504032@yahoo.com, aashiqkamal@gmail.com, ananya.nag79@gmail.com

Abstract— **Wireless sensor networks usually operate on unattended mode in hostile environments so the sensitive data should be protected using some sort of cryptography. Symmetric key scheme is appropriate cryptography for wireless sensor networks due to its low energy consumption but most of them cannot provide sufficient security level as public key approach does. In this work, we propose a new security scheme that overcomes the limitations presented in both public and symmetric key algorithm. We propose a new method that uses both public key cryptography and symmetric key cryptography in the encryption/decryption process. We have calculated the computational and communication overheads in terms of energy consumption in the new scheme for directed diffusion protocol. The results have shown that the proposed scheme is scalable and a strong competitor to pure symmetric key schemes in terms of energy .But it maintains all security levels provided by public key schemes. The proposed scheme is most suitable for wireless sensor networks that incorporate data centric routing protocols.**

Keywords— **Wireless Sensor Network, Directed Diffusion Routing Protocol, Symmetric Cryptosystem, Asymmetric Cryptosystem, Elliptic curve cryptography**

I. INTRODUCTION

A wireless sensor network is simply defined as a large collection of sensor nodes, each equipped with its own sensor, processor and radio transceiver. Due to the lack of tamper-resistant packaging and the insecure nature of wireless communication channels, these networks are vulnerable to internal and external attacks. This paper focuses on the aspects of providing secure communications in WSNs using a new cryptography technique. The objective of this work is to propose a feasible encryption/decryption technique those suites the limited resources of a sensor node while maintaining strong cryptography mechanism. When sensor networks are deployed in a hostile environment, Security becomes extremely important because networks are subject different types of malicious attacks [3]. The main challenge in sensory networks is that how to set up secret keys between communicating nodes. This problem is known as the key agreement problem which has been handled via two security mechanisms: Public Key Cryptography (PKC) and Symmetric Key Cryptography (SKC) [5],[6].Security experts favor the use of PKC.Whenever it is applicable because it provides all security services for the system under consideration including confidentiality, integrity, authentication, and non repudiation [7] . In addition, because of its asymmetry property, sensors do not need to carry the pre-distributed keys. The public key and symmetric key approaches maintain all features of the Directed Diffusion (DD) protocol presented in [4]. However, both of them have dangerous drawback that affects both the security level accomplished and of the protocol in terms of limited resources. For this reason an approach is proposed which consider that maintaining in-network processing feature of DD, accomplishing sufficient security level, and providing protocol efficiency in terms of energy and memory requirements. The main drawback of PKC is that it suffers from high computational complexity and communication overhead. The SKC, on the other hand, is very attractive for sensor networks due to their energy and memory efficiency. The main idea In SKC techniques is that the secret keys are pre-distributed among sensors before their deployment [4].Due to memory limitation of sensor nodes, perfect security satisfaction has not been achieved yet [8],[9]. This work proposes a new approach that combines the use of both PKC and SKC schemes in wireless sensor networks. The validity of the new security method is implemented using Directed Diffusion routing protocols proposed in [4]. Directed Diffusion routing protocol has been developed and has become a breakthrough in data-centric routing [2],[12]. In data-centric routing, the sink sends queries to certain regions and waits for data from the sensors located in the selected regions.

II. DIRECTED DIFFUSION ROUTING PROTOCOL

The key idea in DD is to get rid of unnecessary operations of network layer routing in order to save energy. The DD protocol bases its operations and communications on named data. The sink requests data by sending an interest for named data which is broadcast through its neighbors. Each sensor node receiving the interest can do in-network data aggregation and caching the interest for later use. The interest entry also contains several gradient fields where a gradient can be used to determine a reply link to a neighbor from which the interest was received. Using interest and gradients, paths can be established between sink and sources. Several paths can be established so that one of them is selected by the reinforcement process [4].When a sensor node in the specified region receives an interest; it activates its sensors to begin collecting information. This information returns along the reverse path of interest propagation. Intermediate nodes might aggregate the data by combining reports from several sensors. It is important to mention here that not all fields in the packets (interests and replies) are needed for aggregation at intermediate nodes, while the sink and the source must see the whole packet. I give the emphasis about this point in the implementation of the new security scheme.

III. Security Requirements in WSNs

The process by which public key and symmetric key cryptography schemes should be selected is based on the following criteria:

- Energy: how much energy is required to execute the encryption /decryption functions?
- Program memory: the memory required to store the encryption /decryption program
- Temporary memory: the required RAM size or number of registers required temporarily when the encryption/decryption code is being executed
- Execution time: the time required to execute the encryption/decryption code.

Since proposed method suggests using a combination of two algorithms; a public key based algorithm and a symmetric key based algorithm, here show how use the above criteria in selecting these algorithms.

A. *Selecting the Public Key Algorithm*

Elliptic Curve Cryptography (ECC) is more efficient than Ron Rivest, Adi Shamir and Leonard Adleman (RSA) algorithm in terms of memory requirements because it requires much lower key size than RSA to achieve the same security level [8], [10].ECC with 160-bit keys provides the currently accepted security level, and is equivalent in strength to RSA with 1024 (RSA-1024). As a result, we choose the modified ECC proposed in [10] because it is better than RSA in terms of both memory requirements and execution time. In the below we analyze energy consumptions and memory requirements for ECC. ECC with 160-bit key size and 1024-bit message size in [10] shows that on 8-bit ATMEL microprocessor with 8 MHz clock rate is 0.81s. the execution time is defined as:

Execution Time =IC *CPI *CCT………… (1)

Where, IC is the Instruction Count, and the CPI is the Clock Rate per Instruction. From equation (1):

IC= Execution time/ (CPI*CCT)…………... (2)

Since CCT=1/processor frequency (for 8 MHz, CCT= 125 ns) and the average CPI=1.3, IC can be calculated as:

IC=0.81/ (1.3*125*10^-9)=4984615……..…(3)

Given that each instruction represents one unit of energy consumption then ECC Computation Energy Consumption,

EC (ECC)=4984615 energy units……………...(4)

B. *Selecting the Symmetric Key Algorithm*

RC5 with 64-bit key and 64-bit block size is the best algorithm in terms of execution time on all the microprocessor architectures [6]. Based on results in [7], the symmetric key encryption/decryption algorithm (RC5) with a block size of 64 bits and a key size of 64 bits accomplishes the same security level as the ECC discussed in the previous case. Operations of RC5 consist of only XOR, Add, and rotation operations [7], [9], we can conclude that the CPI of RC5 is 1 cycle on Atmega128 [7]. The work by Ganesan et al. in [7] showed that the execution time of RC5 on Atmega 128 8-bit 16 MHz microcontroller is 0.002823s. Thus the total amount of RC5 Computation Energy Consumption, is given by

Execution Time =IC *CPI *CCT

IC=Execution time/(CPI*CCT) ………………(5)

Since CCT=1/processor frequency (for 16 MHz, CCT= 6.25*10^-8s), *IC* can be calculated as:

IC=0.002823/(1*6.25*10^-8)=45168 ..……. (6)

So,EC(RC5)=45168energyunits …………….(7)

For hashing function Secure Hashing Algorithm is well known (SHA-1) [6], [8]. SHA-1 is also a one-way hash function that produces a 160-bit output. The operations constitute XOR, AND, OR, NOT and rotation thus the CPI is 1 cycle when running on Atmega128 [7]. The total execution time of SHA-1 using a 512-bit message is 0.007777s. Thus, using the same logic in SHA-1 energy consumption can be calculated as:

EC(SHA-1)= 0 .007777 *16* 10^6=124432 energy units ………………………….(8)

IV. Implementing Secured Directed Diffusion

DD protocol handle different security schemes by following ways. Here, discuss about the implementation of DD using ECC public key, RC5 symmetric key, and the proposed key schemes. For each implementation, here derive the energy consumption. In DD protocol an interest travels between three different types of nodes; the sink node, the intermediate nodes, and the source node within the node. We assume that a node uses the first radio model for sending and receiving data [3], [11]. According to that model, the total amount of energy to transmit and receive a message containing m bits is given here

$E_{TX}(m,d) = E_{elec} * m + \in_{amp} * m * d * d$ ……..………(9)

$E_{rX}(m,d) = E_{elec} * m$ ……………………………..(10)

Where, E_{elec} is the energy consumed in transmitting or receiving one bit, and is the energy consumed in amplification process.

According to [3] it is well known that the energy required to send 1 bit is equivalent to the energy to perform 1000 computations, with each computation equal to one instruction. Therefore, the energy required to execute one instruction,E_c, is

E_c = E_{TX} (1, d)/100 …………………………….. (11)

A. Implementing Public Key Algorithm

Figure 1 shows the implementation of the public key algorithm in the source node. A message in DD is issued by Eelec the source node and it encrypts the data packet using the ECC public key algorithm. Note that the packet is encrypted twice to achieve three security services: confidentiality, integrity, and authenticity [12].The total amount of energy consumption at the source node, Esn, to encrypt an m-bit packet using ECC, and then to send it, is given by Equation 12:

$E_{sn}(Ecc) = E_{tx}(m, d) + 2 * E_{enc-dec}(ECC)$………… (12)

$E_{sn}(Ecc) = [(E_{elec} * m + \in_{amp} * m * d * d) + (2 * 4984615 * (E_{elec} + \in_{amp} * d * d)/1000)]$ …………………………………….. (13)

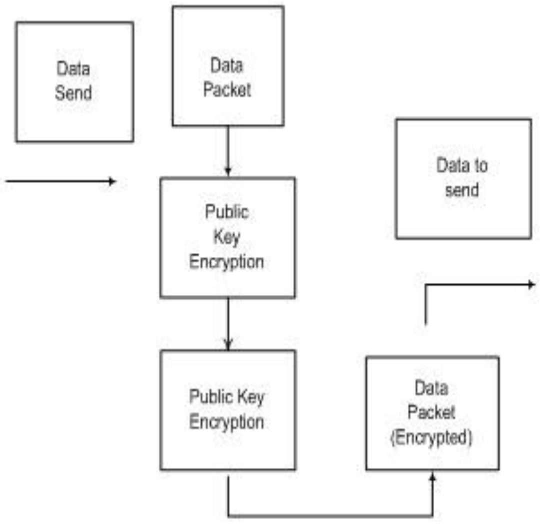

Fig. 1 Public Scheme executed by source node

When a packet reaches to an intermediate node, the node will consume a total amount of energy, E_{in}(ECC) for receiving, decrypting, encrypting and then for sending the packet, is as follows:

$E_{in}(Ecc) = [2 * (E_{elec} * m) + (\in_{amp} * m * d * d) + \{4 * 4984615 * (E_{elec} + \in_{amp} * d * d)/1000)\}]$………….. (14)

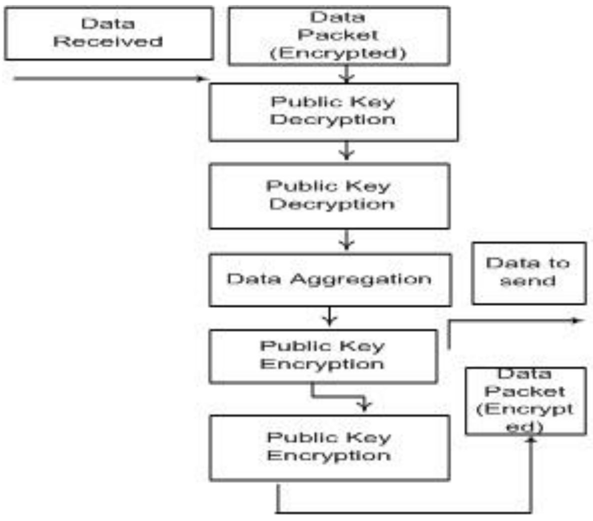

Fig. 2 Public scheme executed by each intermediate node

The sink node consumes a total amount of energy, E_{snk} (ECC), for only receiving and decrypting the packet, as follows:

$E_{snk}(Ecc) = E_{rx}(m) + 2 * E_{enc-dec}(ECC)$…………(15)

$E_{snk}(Ecc) = [(E_{elec} * m) + \{2 * 4984615 * (E_{elec} + \in_{amp} * d * d)/1000\}]$ .. ……………….(16)

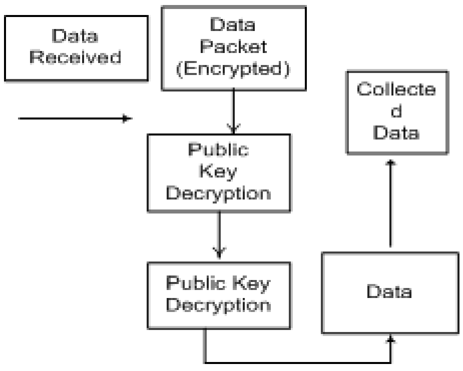

Fig. 3 Public Scheme executed by source node

Therefore, the overall energy consumption of sending a packet (data or interest) from the sink node to the source node or vice versa is,

$E_{s-snk}(Ecc) = E_{sn}(Ecc) + I * E_{in}(Ecc) + E_{snk}(Ecc)$ …………………………..(17)

Here I=intermediate nodes.

B. Implementing Symmetric Key Algorithm

Now energy calculation is done when the DD protocol implements the symmetric key for encryption/decryption process. We start with the source node. The total amount of energy consumption at the source node, E_{sn},(RC5) to encrypt and to send an m-bit packet using RC5 and SHA-1 is given by here

$E_{sn}(RC5) = E_{Tx}(m,d) + 2 * E_{enc-dec}(RC5) + E(SHA)E_{sn}(RC5) = [(E_{elec} * (m+h)) + \{\epsilon_{amp} * (m+h) * d * d) + \{2 * \left(\frac{m}{128}\right) * 45618 * (E_{elec} + \epsilon_{amp} * d * d)/1000\} + \{(m+65)/512) * 124432 * (E_{elec} + \epsilon_{amp} * d * d)/1000\}]$ ……………………………..(18)

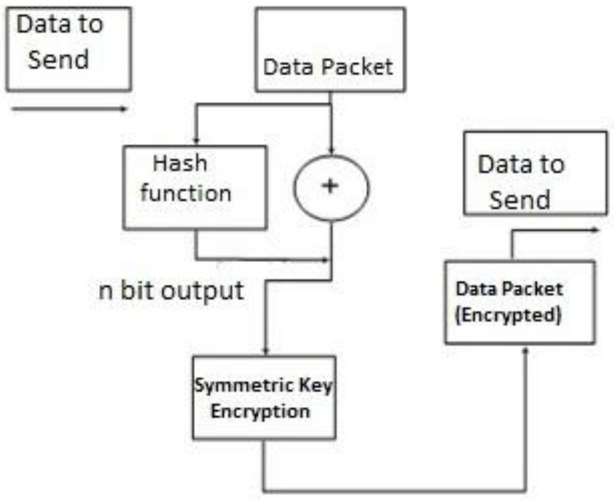

Fig. 4 Symmetric scheme executed by source node

Note that the key size and block size RC5 using a 60 bits long each (a total of 128 bits), and SHA-1 works on 512-bit block of the message (m). For the intermediate nodes, Figure 5 shows that each node will consume a total energy, E_{in}(RC5) on receiving m+h bits for encryption/decryption, hashing, and then will transmit m+h bits in here as,

$E_{in}(RC5) = E_{rx}(m+h) + 2 * E_{enc-dec}(RC5) + 2 * E(SHA) + E_{tx}(m,d)$ ……………………………….(19)

$E_{in}(RC5) = [(E_{elec} * (m+h)) + \left\{2 * \left(\frac{m}{128}\right) * 45618 * \frac{E_{elec} + \epsilon_{amp} * d * d}{1000}\right) + \{(m+65)/512) * 124432 * (E_{elec} + \epsilon_{amp} * d * d)/1000\} + \{E_{elec} * (m+h) + \epsilon_{amp} * (m+h) * d * d$ ……………………………………….…(20)

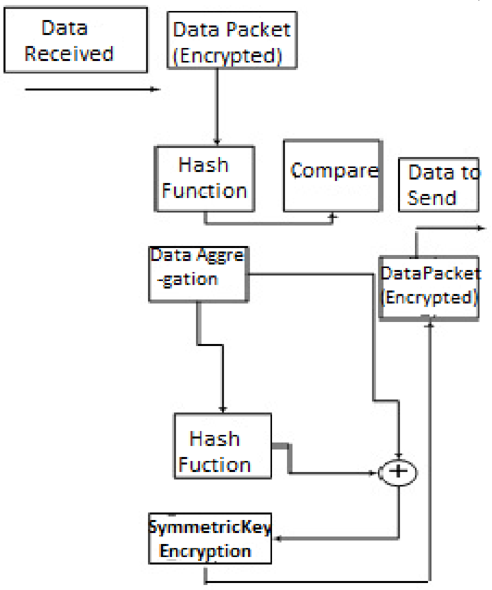

Fig. 5 Symmetric scheme executed by each intermediate node

The operations at the sink node for the symmetric key algorithm are shown in Figure 6. The sink node consumes a total amount of energy, $E_{snk}(RC5)$ for only receiving, decrypting, and hashing a packet of m+h bits, is given here:

$E_{snk}(RC5) = E_{rx}(m+h) + E_{enc-dec}(RC5) + E(SHA)$ ………………………………….(21)

$E_{snk}(RC5) = [(E_{elec} * (m+h)) + \left\{\left(\frac{m}{128}\right) * 45618 * \frac{E_{elec} + \epsilon_{amp} * d * d}{1000}\right) + \{(m+65)/512) * 124432 * (E_{elec} + \epsilon_{amp} * d * d)/1000\}$……………………………….. (22)

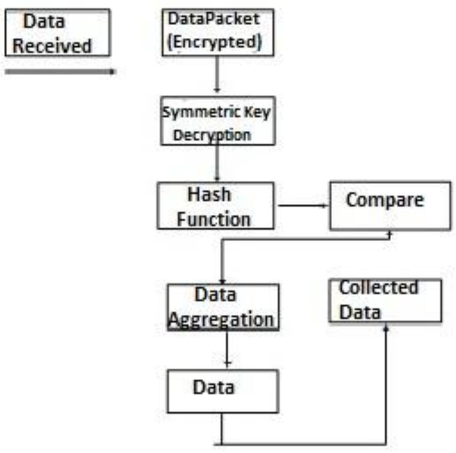

Fig. 6 Symmetric scheme executed by sink node

So, overall energy consumption of sending a packet (data or interest) from the sink node to the source node or vice versa, $E_{s\text{-}snk}(RC5)$, can be given here:
$E_{s\text{-}snk}(RC5)=E_{sn}(RC5)+I*E_{in}(RC5)+E_{snk}(RC5)$ …(25)

where I=numer of intermediate node

C. Implementation of Proposed Key Algorithm

The proposed schemes may use the public key and the symmetric key which was implemented in the node.
Here start with the source node which performs the operations shown in Figure 7. The total amount of energy consumption at the source node using the Proposed scheme, E_{sn},(P), to encrypt an m-bit packet and then to send it, is given by:

$E_{sn}(p) = E_{tx}(m+h,d) + E_{enc-dec}(RC5 + ECC) + E(SHA)$ …………………………………………….(26)

$E_{sn}(p) = E_{elec} * (m+h) + \epsilon_{amp} * (m+h) * d * d + (45618 + 4984615) * (E_{elec} + \epsilon_{amp} * d * d)/1000 + \{\frac{m+65}{512} * 124432 * (E_{elec} + \epsilon_{amp} * d * d)\}$ ……….(27)

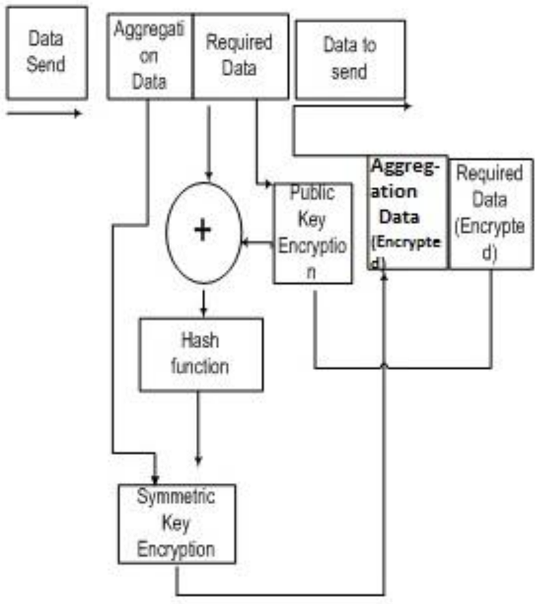

Fig. 7 Proposed scheme executed by source node

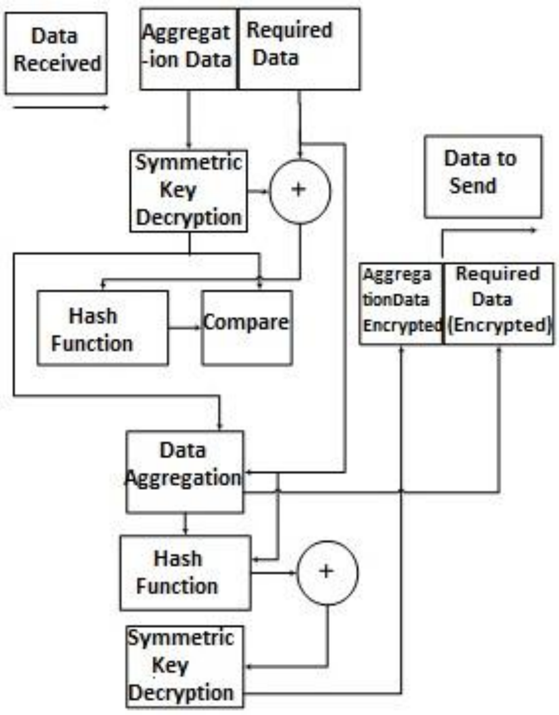

Fig. 8 Proposed scheme steps executed by each intermediate node

For intermediate nodes, each node does not need to encrypt the part of the packet that is encrypted by the source/sink node using the public key, it rather needs to decrypt and encrypt the aggregation data using the symmetric key algorithm *(RC5)* and *SHA-1*, as shown in Figure 8. This is because the following fact about public key algorithm. Given two messages M1 and M2, if M1=M2 and the encryption and decryption keys are the same, then the cipher of both M1 and M2 are equal, i.e E_k[M1]=E_k[M2]=C. so the node will only check if the encrypted data is already existed in the data cache.

The energy consumption of the each intermediate node:

$E_{in}(p) = E_{rx}(m+h,d) + 2 * E_{enc-dec}(RC5) + 2 * E(SHA) + E_{tx}(m+h,d)$ ....(28)

$E_{in}(p) = E_{elec} * (m+h) + \{2 * 45618 * (E_{elec} + \epsilon_{amp} * d * d)/1000\} + \{2 * 124432 * \frac{m+65}{512}\} * (Eelec + \epsilon_{amp} * d * d)/1000\} + \{E_{elec} * (m+h) + (E_{elec} + \epsilon_{amp} * d * d)\}/1000\}$ ....(29)

The sink node for the Hybrid key algorithm is shown in Figure 9. The sink node consumes a total amount of energy, *Esnk(P)*,on receiving *m+h* bits, decrypting the required data portion using the ECC public key algorithm,decrypting the aggregation portion using RC5 symmetric key algorithm, and hashing the data using SHA-1, as follows:

$E_{snk}(p) = E_{rx}(m+h) + E_{enc-dec}(RC5 + ECC) + E(SHA)$ ....(30)

$E_{snk}(p) = E_{elec} * (m+h) + \{(45618 + 4984615) * (E_{elec} + \epsilon_{amp} * d * d)/1000\} + \{2 * 124432 * \frac{m+65}{512}\} * (Eelec + \epsilon_{amp} * d * d)/1000\}$

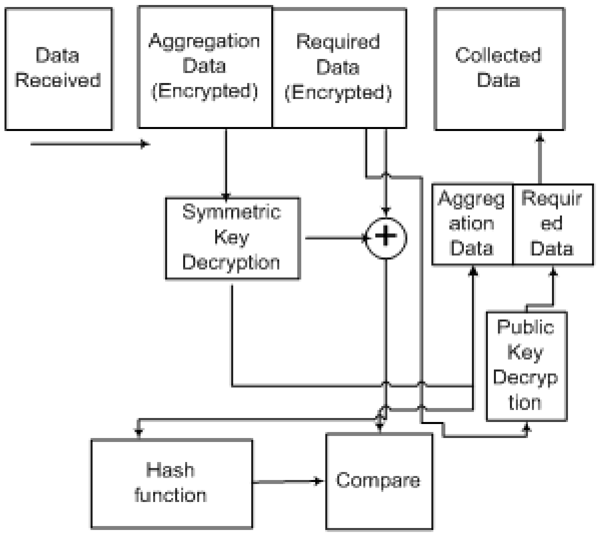

Fig. 9 Proposed scheme executed by sink node

The overall energy consumption of sending data or interest message from the sink node to source node, or vice versa, using the proposed key algorithm, $E_{s\text{-}snk}(p)$can be calculated by

$E_{ssnk}(p)=E_{sn}(p)+I*E_{in}(p)+E_{snk}(p)$ …………(32)

I=No.of intermediate nodes.

V. Results and Discussion

In this section, we present and discuss results obtained from the energy analysis made in the previous sections. To provide a valid and fair comparison assume the three security schemes we considered in the above sections are executed on Atmega 128 ,16MHz, 8-bit architecture AVR instruction set, this microprocessor is widely used in many today's sensor nodes. Also, assume that the sensor network consists of n nodes with I (I=10) intermediate nodes between the sink and any source node, and the distance between any two neighboring nodes is 1 meter. The number of intermediate nodes indicates how large the network is. Furthermore, the node uses the first order model for radio transmission (Equations 9 and 10) with transmitter electronics=receiver electronics=E_{elec}=50 nJ/bit, and the transmitter amplifier amp($€_{amp}$)=100 pJ/bit/m2. These parameters are consistent with many related works [10], [11].

In this work, using a message size of 1024 bits, a basic block size of 1024 bits for the SHA-1 hashing function with output size, h of 160 bits, the basic block size in the RC5 symmetric key algorithm is 64 bits. Here show and compare the energy consumptions at the sink, source, and intermediate nodes for all security schemes under consideration assuming a message of size 1024 bits is traveling from the sink node to a source node (or vice versa) through I intermediate nodes.

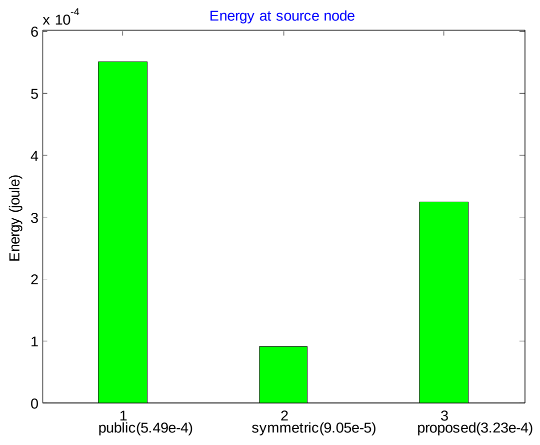

Fig. 10 Energy consumption at source node

A. Public Key Scheme

The energy consumption of the public key scheme, ECC, exceeds all of the symmetric and proposed schemes. The amount of energy consumptions in Figures 10 through 13 shows that source node using ECC consumes energy 1.5 times more than that of the proposed scheme and 5 times more than that of RC5.From, figure 11 it says that the intermediate node applying ECC consumes more than 6 times as it does in RC5, and more than 7 times as in the proposed scheme. Additionally, the sink node consumes almost the same energy as the source node does for ECC algorithm. The energy consumption of the public key scheme is high and it was expected because the cost of running its code is very high compared to other schemes.

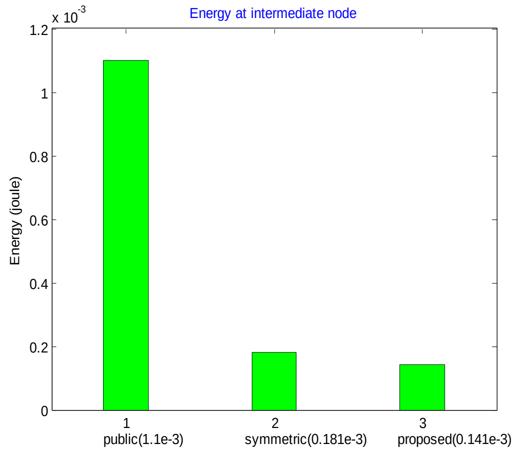

Fig. 11 Energy consumption at intermediate node

B. Public Key Scheme

The energy consumptions depicted in Figure 11 show that source node using RC5 saves about 72% of the energy consumed by the proposed scheme and 83% of the energy consumed by ECC. Figure 12 indicates that the intermediate node consumes additional 22% of the energy consumed by the intermediate node using the proposed scheme, and it saves more than 83% of the energy consumed by the intermediate node using ECC. Additionally, the sink node using RC5 saves about 72% of the energy consumed by the proposed scheme and 82% of the energy consumed by ECC.

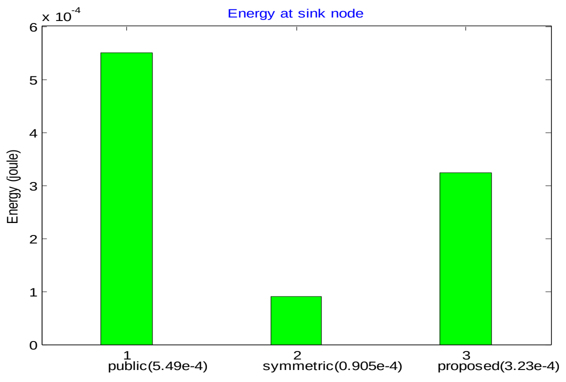

Fig. 12 Energy consumption at sink node

C. Proposed Scheme

Like the symmetric key scheme, the proposed scheme reduces the energy consumption through maximizing the use of RC5 over ECC. Thus we expect its performance to be close to the symmetric key scheme. The energy consumptions in figure 10 show that source node in the proposed scheme saves about 33% of the energy consumed by the public key scheme, but it consumes 3 times the energy consumed by the symmetric scheme. From figure 11 ,we have the intermediate node in this scheme will save more than 86% of energy consumed by the intermediate node using the public scheme, and it saves more than 15% of the energy consumed by the intermediate node in the symmetric key scheme. The sink node saves about 33% of the energy consumed by the public scheme, but it consumes 3 times more than the energy consumed when using the symmetric key scheme. This is because the sink node has to decrypt both the aggregation data and the collected data which means that the symmetric, public, and hash functions should be executed. As a result, the overall network energy in the proposed scheme is equal to that of the symmetric key scheme as it is indicated in Figure 13 but with additional level of security. Therefore, the proposed scheme is a suitable algorithm for WSN.

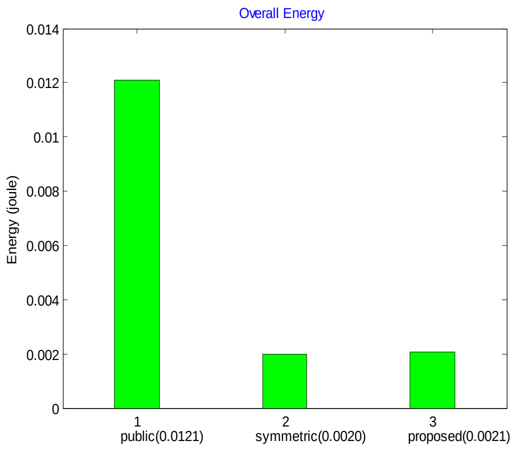

Fig. 13 Overall energy consumption

VI. Conclusion

In this work, we have presented a new method of applying cryptography techniques in Wireless Sensor Networks. The proposed scheme uses Public Key Cryptography and Symmetric Key Cryptography in very selective way. It maximizes the lifetime of sensor node's batteries through minimizing the use of Public Key Cryptography. We evaluated the energy consumption for the proposed scheme and made a comparison between Public Key Cryptography and Symmetric Key Cryptography schemes. Experimental results indicate that, the proposed scheme provides better performance when compared to pure Public Key Cryptography with energy saving ranges from 33 to 86 percents. Moreover, It competes with Symmetric Key Cryptography and provides slightly better energy saving. From the Experimental results, we also observe that the intruder able to detect message in case of Symmetric Key Algorithm. In the case of Symmetric Key Algorithm, data are encrypted by Public Key which is publically known. So, Security is relatively low with respect to Public Key Algorithm. But, in the case of Public Key Algorithm, Private Key never needs to be transmitted. Moreover, public-key systems can provide guarantee about the integrity and authentication, not only privacy. In our proposed method, main portion of data are encrypted by Public Key Algorithm. So, the security level is high with low energy. From the Experimental results, we also describe that the proposed approach is scalable and suitable for large Wireless Sensor Networks.

References

[1] K. Kifayat, M. Merabti, Q. Shi, D. Llewellyn-Jones, "Group Based Secure Communication for Large-Scale Wireless Sensor Networks", Journal of Journal of Information Assurance and Security, 2(2), pp. 139-147, 2007.

[2] Alessandro Sorniotti, Laurent Gomez, Konrad Wrona and Lorenzo Odorico, "Secure and Trusted in-network Data Processing in Wireless Sensor Networks: a
Survey", Journal of Journal of Information Assurance and Security, 2(3) (2007),

[3] M. AL-Rousan, Al-Ali, and A. Talaa, "TCRP: A Tree-Clustering Protocol for Tracking Mobile Targets In Wireless Sensor Networks," In Proceedings of the
International Symposium in Mechatronics, Jordan, pp. 1-5, 2008

[4] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, pp. 56-67, 2006.

[5] Yi Cheng, Dharma P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks" Ad Hoc Networks), pp. 35-48.

[6] W. Du, R. Wang, P. Ning: An efficient scheme for authenticating public keys in sensor networks. Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing), Urbana-Champaign, pp. 58-67, 2005.

[7] Y. Law, J. Doumen, and P. Hartel, "Survey and benchmark of Block Cipher for Wireless Sensor Neworks",ACM Transactions on Sensor Networks 2(1), pp. 65-93

[8] D. Liu, P. Ning, "Improving key pre-distribution with deployment knowledge in static sensor networks" ACM Transactions on Sensor Networks 1(2): 2005.

[9] G. Gaubatz, J. Kaps, and B. Sunar, "Public keys cryptography in sensor networks" In Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS), 2004.

[10] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz "Comparing elliptic curve cryptography and RSA on 8- bit CPUs", In Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems

[11] D. J. Malan, M. Welsh, and M. D. Smith,"A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography" In Proceedings of the IEEE International Conference on Sensor &Ad Hoc Communications

[12] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks", In Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 197–213, 2003.