A Dynamic Cryptographic Algorithm To Provide Nodal Level Security In Wireless Sensor Network

Nivedita Mukherjee
Department of Information Technology
National Institute of Technology
Durgapur-713029,West Bengal,India
Email: nivedita211@gmail.com

***Abstract*—Wireless sensor networks (WSNs) continue to evolve as one of the most exciting and challenging research areas. There are many applications of wireless sensor networks that collect and disseminate sensitive and important information. For successful operation of many of the sensor node applications, it is necessary to maintain the privacy and security of the data stored in the sensor nodes and of the transmitted data. The security models and protocols used in the wired and other networks such as ad-hoc networks are not suitable for WSNs due to their severe resource constrictions. Thus there is a lack of an agreeable and most effective way of securing the information. In this paper a unique, dynamic cryptographic algorithm to provide security to the wireless sensor network by securing the individual nodes of the network, has been proposed.**

I. INTRODUCTION

One of the greatest challenges in WSNs is secure communication which is characterized by the following characteristics:-

- **Authentication**: Receiver node should be able to verify the ID of the sender node and thus be able to verify the validity and genuineness of the received data.

- **Integrity**:- Integrity means that no data falsefication takes place during transmission.

- **Confidentiality**:- Data must be protected from being captured by adversaries.
  Due to the stringent resource and performance constrictions none of the currently available security protocols are completely agreeable. So a scheme which uses a unique dynamic encryption-decryption cycle, allowing the use of lightweight mathematics to prevent the individual nodes from being compromised by adversaries, is proposed. The use of lightweight mathematics for performing encryption-decryption saves energy. This scheme also allows the use of a single or a small set of network-wide shared keys, which are embedded into the nodes before their deployment so that the extra transmission cost incurred in setting up shared keys in the network after the deployment of the nodes can be avoided as opposed to the most of the currently available security protocols. Transmission of data in WSNs take up 80% of the total power used in the network. Thus this scheme can save a substantial amount of power.

The rest of the paper is organized as follows. In section two we discuss the related works done so far in securing wireless sensor networks. Section three describes the proposed scheme in detail accompanied with calculations. Section four describes the details associated with implementation of the scheme. Section five shows simulation details.In section six an analysis of the algorithm of this scheme has been shown. In section seven conclusion of the paper is given and in section eight the future scope of this scheme is highlighted.

II. RELATED WORK

Security in WSN has not been researched extensively. The currently available security protocols use the key management techniques to establish secure communication between two or more sensor nodes and also between the sensor node and the base-station. A brief review of the popular key-management schemes used in WSNs [1] and their drawbacks is given in this section.

- **$\mathbf{Single\ network - wide\ key}$** : This scheme uses a single network-wide key for data transmission. It is the simplest technique from the point of view of energy and memory consumption and avoidance of complex protocols.
  Main drawback is that if a single node gets compromised, the entire network also gets compromised.

- **$\mathbf{Pairwise\ key\ establishment\ scheme}$** : This scheme requires each of the nodes to establish unique keys with all the other nodes which puts an additional communication overhead on the nodes. Also the nodes are required to maintain all the keys in their memory which is expensive from the memory point of view. These overheads limit the sensor nodes mobility as well as scalability.

- **$\mathbf{Trusted\ base - station}$** : This scheme uses the centralized key distribution centre (KDC) approach. The drawbacks are that it is not scalable and the base station

978-0-7695-3942-3/10 $26.00 © 2010 IEEE
DOI 10.1109/CICC-ITOE.2010.56

becomes the target of attacks.

- $\mathbf{Authentication : \mu TESLA :}$ Perrig et al.[2] presented a security protocol suite, optimized for WSNs, called SPINS which is built upon two secure building blocks: SNEP (offers data confidentiality, authentication, integrity and freshness) and μ TESLA (offers broadcast data authentication). Though this schme is less complex and provides strong security with low communication cost, it includes μ TESLA's overhead from releasing keys after a certain delay resulting in message delay, which may impede the communication of real-time critical messages.

- $\mathbf{Public\ key\ schemes[3-6]}$:Public key encryption schemes such as RSA, ECC and TinyOS are computationally expensive.

- $\mathbf{Key\ predistribution\ schemes :}$ In a key predistribution scheme, some keys are preloaded into each sensor node before deployment. After deployment, sensor nodes undergo a discovery process to set up shared keys for secure communication. This key discovery and establishment process includes communication overhead and reduces the mobility of the network. Also since in WSNs the power-constrained sensor nodes have a high probability of becoming inactive, newer paths for data transmission have to be discovered and established by all the sensor nodes that were dependent on the node that became inactive. As this situation may arise frequently, a high comminication overhead is put upon the sensor nodes. Also the sensor nodes have to maintain a large pool of keys to ensure that it shares at least one of the keys with some other node to form a path with all the nodes in the network. This brings about a substantial amount of memory cost and thus reduces the scalability of the network. There are may different key predistribution protocols are[7-13].
- $\mathbf{Dynamic\ key\ management[14] :}$ Some of the advantages of using dynamic key management scheme are improved network survivability and better support for network growth. The problem lies in making this scheme secure and efficient.

- $\mathbf{Hirearchial\ key\ management :}$ In these schemes the network is divided into a hirearchy and flow of data is through the hierarchy. Sometimes clusters are formed corresponding to each level of hierarchy and a cluster-head is assigned to be in-charge of each cluster. Since cluster-heads require more resources than sensor nodes, the overall cost of the network may increase. Also the cluster-heads become the targets of the attackers. Forming a hierarchy through mutual communication between the nodes imposes communication overhead. Some of the protocols complying to this scheme are:- LEAP[15], Key management for heterogeneous sensor networks [3,16], Pairwise keys in heterogeneous sensor networks [3,16,17].

III. Proposed scheme

The main motivation behind the idea of a dynamic cryptographic algorithm to provide nodal level security in a WSN is to make the vital data and information stored in each of the nodes secure enough to allow the use of simpler and more energy efficient protocols for data transmission and communication over the network. As discussed in the subsequent sections, this method helps in avoiding the drawbacks, complexity and the cost involved in the above mentioned protocols while providing an acceptable level of security both for data transmission and data storage at the sensor nodes.

In the given proposal the data present in each of the nodes has been divided into two parts:-

- $\mathbf{Hard-coded\ information :}$ The codes of the programs and other data embedded into the nodes before deployment, which are used to process the sensed information and to perform other operation related to data transmission, security and control. These hard coded information are mutable.

- $\mathbf{Sensed\ information :}$ The information gathered by the sensor node from the environment it is monitoring.

A. Concept

The proposed concept is as follows:-

- All the nodes will be embedded with a single network-wide shared key, say K_1 (K_1 comes under the category of hard-coded information). K_1 will be used for message transmission using a symmetric encryption protocol such as HIGHT[18] or PRESENT[19] or any other light-weight cryptographic mathod designed for WSNs.
- All nodes will also contain the sensed information from the environment they are monitoring. Since the sensor node does not process the sensed data itself, so we can encrypt this data as soon as it is gathered from the environment using the key K_1 and the cryptographic protocol used in message transmission.
- Each node will contain a second key K_2.
- K_2 will be used for encrypting the sensed data and the network-wide shared key K_1, using simple logical, invertible operation(s) such as XOR so that computation does not consume much energy. We can also use the modified, lightweight, version of some well known protocols such as HEIGHT[18] or PRESENT[19].
- A resourceful attacker such as an attacker equipped with a laptop can easily guess the key K_2 thereby exposing K_1, thus gaining access to
the sensed information that was encrypted using K_1 and

thus the entire network gets compromised since K_1 is the only network-wide shared key used

for message transmission among the different nodes and also among a node and a base station. So to address this security threat K_2 is made dynamic

by changing its value periodically. So that by the time the attacker comes close to guessing the value of K_2, it gets changes as a result rendering

K_2 unsure for the attacker. But before changing the value of K_2 the data and K_1 are decrypted, both of which were encrypted by the old value of K_2, and

again are encrypted with the new value of K_2 after changing it. To ensure that the data and K_1 are safe during th period for which they are

left exposed i.e the time required to decrypt and re-encrypt them, which is negligible, a safe time gap is maintained. The details of calculating this safe time gap will be shown later in this section.

The theoretical calculation for finding out the ideal time period after which K_2 should be changed is shown in the followins subsections.

B. Time period calcualation

Let K_2 be a 32-bit value.
Number of possible values of K_2 = N = 2^{32}*$(1 \div 2)$ = 4.29*10^9*$(1 \div 2)$
N also denotes the number of guesses required by the attacker to guess the value of K_1 using brute force method.
Therefore the probability that the attacker guesses K_2 = G = $1 \div (4.29 * 10^9 * (1 \div 2))$. Thus it is seen that the probability of guessing the value of K_2 is very small.
Hence, if the attacker takes time 't' to perform one guess operation, then the time 'T' required to perform N guesses = 4.29*10^9*$(1 \div 2)$*t.
Thus, the ideal time period after which K_2 should be changed is T. But to maintain the safe gap as mentioned in the pervious section to keep the data and K_1 safe during the negligible period when K_2 is changed and the data and K_1 are re-encrypted, over which it if left un-encrypted, the time period of changing K_2 will practically be T-Δ, where Δ is the time taken to change K_2 and re-encrypt data and K_1.

C. Energy consumption calculation

The simulation was done on a theoretical basis. Since the entire simulation was done on a Unix system running on Intel's Core Duo, a 32-bit architecture, the assembly language generated by the simulation was converted to Atmel ATMEGA 128L's[23] instruction set.
The following are the calculations involved in theoretical evaluation:-// Clock-cycles (C) = 420;
The frequency, which is nothing but the number of clock-cycles per second, of ATMEGA 128L is 16MHZ.
Therefore time required by 420 clock cycles = $420 \div 16 * 10^6$ sec = $26.25 * 10^{-6}$ sec = 26.25 $\mu sec.$

Supply voltage of ATMEGA 128L 16MHZ is 4.5-5.5 V.

Considering, Voltage (V)=4.5 V,

Current (I)=1 mA (which is quite usual for sensor nodes),

Using the relation P (power) = V*I

Value obtained is, P=4.5*1*10^{-3} W

Also, P=$E \div t$. Here E is energy.

Thus, result obtained is,
E = P*t = 4.5*10^{-3}*26.25*10^{-6} J = 118.125*10^{-9} J = 0.118125 μJ

IV. Details of implementation

As stated earlier, energy is a major concern in WSNs. The sensor nodes are generally powered by Nickel-Cadmium or Lithim-ion betteries. So care must be taken that the operations performed at the nodes are not computationally heavy thereby consuming more energy of the nodes.Else the nodes will not last for a substantial amount of time. This is the main reason that the conventional cryptographic methods such as RSA and other asymmetric cryptographic techniques cannot be used for data security at the nodes or for transmission of messages.
So keeping this in mind the algorithm and some of its vital components have been designed to be energy efficient. The implementation details are shown below.

A. Algorithm for encrypting the sensed data and K_1

The K_2 which is generated using an RNG designed using the system time,shown in the algorithm within the while loop, is used to encrypt the sensed data and the network-wide shared key K_1 using a simple XOR operation. XOR is also used in the simulation as the encryption algorithm for data transmission in the network. So the data is encrypted by K_1 using XOR operation.
Since one of the goals of this cryptographic method is to keep the computation as minimum as possible and also since an easily invertible operation is needed, XOR has been chosen as the encryption operation. Though a single XOR operation by itself is not cryptographically secure but with the periodically changing K_2 it becomes cryptographically robust.
The algorithm used for encrypting the data and K_1 using K_2 is shown below.

```
$long$ $k_2$ = 25, $data$ = 1768212, $k_1$ = 74589231;
Encrypt_By_Network_Key() {

      $data$=$data \wedge data$;
}

Decrypt() {
        $data$=$data\wedge data$;
        $k_1$=$k_1\wedge k_2$;
}
Encrypt() {
        $data$=$data\wedge k_2$;
        $k_1$=$k_1\wedge k_2$;
}
Dynamic_Algorithm() {
        $struct\ timeval\ start, end$;
        $long\ seconds, useconds, nseconds, nuseconds, seed$ = 31825329;
        Encrypt_By_Network_Key();
        Encrypt();
        $while(1)$
        {
                $gettimeofday(\&start, NULL)$;
                $usleep(50000)$;
                $gettimeofday(\$end, NULL)$;
                $Decrypt()$;
                $nseconds$=$end.tv\_sec \wedge start.tv\_sec$;
                $nuseconds$=$end.tv\_usec \wedge start.tv\_usec$;
                $nseconds$=$\neg nseconds$;
                $k$=$\neg(\neg nuseconds \wedge nseconds)\wedge\neg$;
                $k_2$=$k_2$<<8;
                $seed$=$k$;
                Encrypt();
        }
}
```

Since this algorithm was implemented using a C program in Linux, it uses the data structure ”struct timeval” associated with the function ”gettimeofday” . The interval at which the system time is obtained is 50000 microseconds and during this interval the system is sent to sleep state to conserve energy.As seen in the algorithm, mostly logical operations are used to obtain the random number so that the computation is light to use less energy.Besides using the system time for producing the random number saves some complex computation involved in conventional random number generators.

In the algorithm K_2 has been taken as a 32-bit value.
Deva Seetharam [20] had devised a random number generator (RNG) based on the system clock of the sensor node, but it resets the system clock which might interfere with the other time dependent processes, but the RNG described above does not reset the clock time and thus, does not interfere with the other processes.

Using a system time dependent RNG has the following advantages:-

- Saves computation expenses of difficult mathematical operations for resource constrained sensor nodes, by using readily available time related data.
- Difficult to guess as the attacker would require to synchronize its clock to the individual nodes' system time and time synchronization in WSNs is itself a serious problem.

For testing the RNG 600 random numbers were generated on each run and it was run 100 times. These random numbers were put through some well-known RNG test such as the ENT test suite and the RGB tests, the results were satisfactory and as follows.

Test names	Obtained results
Chi-Square Test	Chi square distribution for 48104 samples is 906.64, and randomly would exceed this value 0.01% of the times.The ideal value depends on the distribution and a percentage value of between 10% and 90% the sequence is truely random.
Optimum compression	Optimum compression would reduce the size of this 48104 bit file by 0.001%.Optimum compression would reduce the size of this 48104 bit file by 0%.
Entropy	0.986361 bits per bit.Optimum value is 0.997578375 bits per bit.
Arithmetic Mean of data bits	0.4314 .Optimum value 0.5
Serial correlation coefficient	0.0015538.Optimum value is 0.0(uncorrelated)

- **RGB Timing test** : − Average time per rand = 7.936890e+02 nsec Rands per second = 1.259939e+06 The test shows that the designed RNG has a throughput of about 1.26 million random numbers per second when tested with a time interval of 50000 μ sec which is an acceptable value for the purposed scheme.Also this value can be varied by varying the time interval thus giving more flexibility of implementation.

- **RGB Bit Persistance Test** : − This RNG also passes the RGB bit persistence test.
  Cumulative mask = 0 = 00000000000000000000000000000000
  random mask = 4294967295 = 11111111111111111111111111111111

random max = 4294967295 = 11111111111111111111111111111111

V. Simulation

For simulating the above scheme, socket programming was used on a client-server architecture.Though the use of the client-server architecture might seem controversial but the simulation does not depend on the actual architecture as it was not done at the network level. Simulation was done at the nodal level to check the robustness of the proposed scheme in protecting a single node from an attacker.

A. Victim Sensor-node

A server-script including the proposed algorithm has been designed to act as the sensor node. This script was implemented using scocket programming in C.

B. Adversary and Attack

A client-side script, using C socket programming, has been designed to act as the attacker.
In this simulation the data is first being encrypted by k_1 immediately after being sensed into the sensor node and then both the data and k_1 are encrypted by k_2. This encrypted data is being continuously sent to the attacker (Client-Program), which then tries to decrypt the data by using brute force method. This model is similar to a remote attacker having access to the internal data present in a node. The data is sent after the interval T(time interval between the n^{th} encryption and the $(n+1)^{th}$ encryption).

C. Result

The simulation was run for two days at a stretch, with the attacker not being able to decipher the data even once as the key k_2 kept changing. The key size can be increased to increase T, e.g : an attacker, able to process 10^6 decryptions per second takes 10.01 hrs to guess the correct key[22].

VI. Analysis of the Dynamic Cryptographic Algorithm

A. Space complexity of the algorithm

The space complexity of the proposed algorithm is calculated to be approimately 74 bytes which is small enough to be implemented in a sensor node.This also leaves a scope to increase the key sizes to make it more robust against cryptanalysis.

B. Time complexity of the algorithm

The time complexity of the algorithm is calculated as follows:-
4*$O(32)$+19*$O(32)$+10*$O(64)$+T*10*$O(64)$+T*11*$O(32)$
Thus for one round of the algorithm as T is small, the time complexity becomes constant.

VII. Conclusion

Based upon the thorough theoretical analysis it can be concluded that the proposed scheme is efficient both in terms of energy and memory. It provides a high degree of security against brute force attack which is the most common type of cryptanalytic attack.Also it reduces the overall energy consumption in the wireless sensor network by making the nodes secure enough to use a single (or small pool of) network-wide shared key(s) by ensuring that the nodes cannot be compromised by a remote attacker by simply decrypting the vital information present in a node using brute force attack. Thus it greately reduces the communication required in the complex keying schemes used in key-management techniques as discussed in section I and as 80% of the overall energy of a node is used up in communication, this scheme can help in prolonging the longevity of the nodes in a sensor network. Also by reducing the mutual dependency for security amongst the nodes(contrary to many popular key-management schemes), it allows greater mobility and also reduces the problem of mutual re-keying relevant in the pairwise keying schemes due to the deactivation of the power constrained sensor nodes.Thus it allows greater scalability.

VIII. Future scope

The proposed scheme needs to be evaluated by implementing it in the sensor nodes of a WSN. Here, for the encryption of the data and k_1, using k_2, a single round of XOR operation has been used which may not provide sufficient protection against more advanced cryptanalytic attacks. So some lightweight, invertible cryptographic methods need to be devised. For example cryptographic algorithms such as HIGHT[18],PRESENT[19] can be simplified to a great extent to provide high degree of security against many cryptanalytic attacks using this scheme. Also, for simplicity, this scheme has been tested with a 32-bit key, but as memory is becoming cheaper, longer keys such as 128-bit or more can be used, by doing do the time period T can be increased, hence decreasing the frequency of the encryption-decryption cycle and allowing longer sleep period for the nodes. Improving the RNG used in this paper is another avenue to increase the effectiveness of this scheme. Hardware RNGs based on atmospheric noise and thermal noise or on the data sensed by the nodes hold promise to make good source for random numbers.This is a unique scheme which has the potential to provide complete security to the nodes and also to the entire network at a very low cost of energy.

References

[1] .A Survey of Key Management Schemes in Wireless Sensor Networks Yang Xiao,Venkata Krishna Rayi,Bo Sun, Xiaojiang Du,Fei Hu e,Michael Galloway

[2].A Perrig. Et al., SPINS : security protocol for sensor networks, Proceedings of ACM MOBICOM (2001).

[3].X. Du, M. Guizani, Y. Xiao, S. Ci, H.H. Chen, A routingdriven elliptic curve cryptography based key management scheme for heterogeneous sensor networks, in: IEEE Transactions on Wireless Communications, accepted for publication (to appear).

[4].D. Malan, M. Welsh, M.D. Smith, A publickey infrastructure for key distribution in TinyOS based on elliptic curve cryptography, in: Proceedings of 1st IEEE International Conference Communications and Networks (SECON), Santa Clara, CA, October 2004.

[5].N. Gura, A. Patel, A. Wander, H. Eberle, S.C. Shantz, Comparing elliptic curve cryptography and RSA on 8bit CPUs, in:Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston, Massachusetts, August 2004.

[6].A.S. Wander, N. Gura, H. Eberle et al., Energy analysis of publickey cryptography for wireless sensor networks, in: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications (PERCOM), 2005.

[7].Eschenauer, V.D. Gligor, A key management scheme for distributed sensor networks, in: Proceedings of the 9th ACM Conference on Computer and Communication Security

[8].H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: Proceedings of the 2003 IEEE Symposium on Security and Privacy, May 1114,pp. 197213.

[9].D.Liu, P. Ning, Establishing pirwise keys in distributed sensor networks, Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03) (2003),pp. 5261.

[10].P. Ning, R. Li, D. Liu, establishing pairwise keys in distributed sensor networks, ACM Transactions on Information and System Security 8(1) (2005), pp. 4177.

[11].W. Du, I. Deng, Y.S Han, S.Chen, P.K.Varshney, A key management scheme for wireless sensor networks using deployment knowledge, in: Proceedings of IEEE INFOCOM 2004.

[12].F. Anjum, Location dependent key management using random keypredistribution in sensor networks, in: Proceedings of WiSe06.

[13].M.F. Younis, K. Ghumman, M. Eltoweissy, Locationaware combinatorial key management scheme for clustered sensor networks, IEEE Transactions on Parallel and Distributed Systems 17 (8) (2006), pp. 865882.

[14].M. Eltoweissy, M. Moharrum, R. Mukkamala, Dynamic key management in sensor networks, IEEE Communications Magazine 44 (4) (2006), pp. 122130.

[15].S.Zhu, S.Setia, S.Jajodia, LEAP: efficient security mechanisms for largescale distributed sensor networks , in: Proceedings of The 10th ACM Conference on Computer and Communications Security (CCS '03), Washington D.C., October, 2003.

[16].X. Du, Y. Xiao, M. Guizani, H.H. Chen, An Effective Key Management Scheme for Heterogeneous Sensor Networks, Ad Hoc Networks, Elsevier, vol. 5, issue 1, January 2007, pp. 2434.

[17].P. Traynor, H. Choi, G. Cao, S. Zhu, T. Porta, Establishing pairwise keys in heterogeneous sensor networks, in: Proceedings of IEEE INFOCOM 06.

[18]. D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.S; Koo,C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. HIGHT: A New Block Cipher Suitable for LowResource Device, In L. Goubin and M. Matsui, editors, Proceedings of CHES 2006, LNCS, volume 4249, pp. 4659, SpringerVerlag, 2006.

[19].A. Bogdanov et al., PRESENT: An UltraLightweight Block Cipher, Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES 07), LNCS 4727, Springer, 2007, pp. 450466.

[20].Deva Seetharam and Sokwoo Rhee, Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN04), pp. 0104.

[21].J.Walker. Http://www.fourmilab.ch/random.

[22].Cryptography and Network Security Princiles and Practice 4th Edition , William Satllings, pp66,table22

[23].ATMEL 128L instruction set.

ACKNOWLEDGMENT

I humbly thank Prof.L.M.Patnaik (Vice-Chancellor of Defence Institute of Technology (Pune),Maharashtra,India) for lending his invaluable expertise by refereeing this project.

Also special thanks to Dr.Seokhi Hong, Dr.Donghoon Lee, Dr.Bonseok Koo and the entire team of HEIGHT[18] for giving access to valuable information and technical expertise.