An Efficient Security Scheme in WSN using Three Tier Security Architecture

[#1]Amol Abhiman Magar, Pursuing M.Tech,
[#2]B.S.Sonawane, Asst. Professor,
Dept of CST
MIT COE, Aurangabad (MS), India
Email: amolmagar@outlook.com

Abstract— The main issue while setting up the WSN network for node communication is security. This report describes the efficient mechanism for achieving the security between communications of nodes by creating three tier security architecture. This system implements three tier architecture with the use of two polynomial pools having sensor nodes, mobile sinks and some access points that are also sensor nodes, to get better security. Two pools are mobile polynomial pool and static polynomial pool. Mobile sinks and access point carries keys from common mobile polynomial pool were as, access points and sensor nodes carries keys from common static polynomial pool. The main aspect of the three tier architecture Authentication is Communication gets established from mobile sink to access point then from access point to sensor node, that is achieved by pairwise key predistribution methods and authentication of the nodes with the use of polynomial keys and Pailier cryptosysem algorithm. Here, Mobile sink replication attack is implemented against the network. The malicious node, it is blocked. If it wants to communicate within the network then it needs to capture large no of keys from both the pools for authentication. But as the sufficient and valid keys are not available with it, it cannot communicate with the other nodes in the network.

Keywords— Wireless Sensor Network, Pailier Cryptosystem Algorithm, Pairwise Predistribution Key Scheme, Mobile Sink, Common Mobile Polynomial, Common Static Polynomial, Mobile replication attack.

Introduction

In electronic technology recent advanced have paved the way for the development of a new generation of wireless sensor networks (WSNs) consisting of a large number of low-power, low-cost sensor nodes that communicate wirelessly. Such sensor networks can be used in a wide range of applications, such as, military sensing and tracking, health monitoring, data acquisition in hazardous environments and habitat is monitoring [9]. The sensed data often need to be sent back to the base station for analysis. However, when the sensing field is too far from the base station, transmitting the data over long distances using multi hop may weaken the security strength (e.g., some intermediate may modify the data passing by, capturing sensor nodes, launching a wormhole attack, a Sybil attack, selective forwarding, sinkhole, and increasing the energy consumption at nodes near the base station, reducing the lifetime of the network. Therefore, mobile sinks (MSs) (or mobile soldiers, mobile sensor nodes) are essential components in the operation of many sensor network applications. The Proposed system {Security in WSN using polynomial pool based mechanism} "is the combination of two pools. This scheme uses two separate polynomial pools: the mobile polynomial pool and the static polynomial pool. Polynomials from the mobile polynomial pool are used to establish the authentication between mobile sinks and stationary access nodes, which will enable these mobile sinks to access the sensor network for data gathering. Thus, an attacker would need to compromise at least a single polynomial from the mobile pool to gain access to the network for the sensor's data gathering. Polynomials from the static polynomial pool are used to ascertain the authentication and keys setup between the sensor nodes and stationary access node.

Related Work

An Wireless sensor networks are one of the first real world examples of pervasive computing, the notion that small, smart and cheap sensing and computing devices will eventually permeate the environment [2]. Wireless sensor network (WSN) consists of a large number of ultra small sensor nodes. Each sensor node is an autonomous battery operated device with data processing capabilities, integrated sensors, limited memory and a short range radio communication capability. In application scenarios sensor nodes are randomly deployed over a region and collect data.

Wireless Sensor Networks are deployed for a wide variety of applications like military tracking, monitoring of environment, smart environments, patient tracking, etc. [1]. Security is extremely important when sensor nodes are deployed in hostile environments because they may be exchanging valuable or critical information about the environment and an adversary can use this information to his advantage or inject malicious information into the network. Apart from physical capture a malicious user can easily tap into the wireless communication and listen to the traffic, inject misleading data into the network or impersonate as a node of the network. To provide security, encrypted and authenticated communication is required. Active research is being pursued for efficient setup of secure keys in wireless sensor networks. Setting up of keys for secure communication is a part of the Key Management problem.

In general network environments there are three types of key agreement schemes: trusted server scheme, self enforced scheme and pre-distribution scheme. The trusted server scheme has a trusted server between two nodes to negotiate a shared key between the nodes. This scheme is not feasible in sensor networks because there is no central server in most WSN. Self enforcing scheme uses public key algorithms such as Diffie-Hellman key agreement or RSA. Pre-distribution scheme uses secret keys to establish pairwise keys after they are deployed.

In the new security framework , a small fraction of the preselected sensor nodes (see Fig. 1), called the stationary access nodes, act as authentication access points to the network, to trigger the sensor nodes to transmit their aggregated data to mobile sinks.

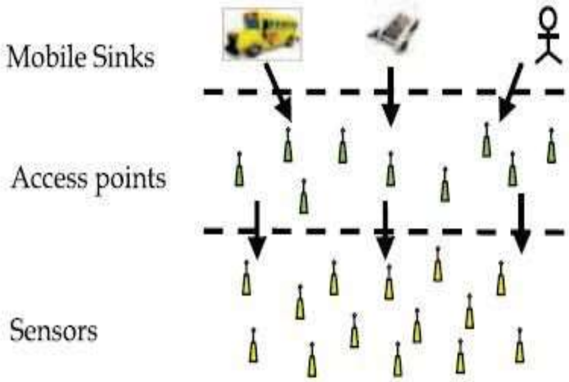


Fig 1: Three Tier Architecture of wireless sensor network.

A mobile sink sends data request messages to the sensor nodes via a stationary access node. These data request messages from the mobile sink will initiate the stationary access node to trigger sensor nodes, which transmit their data to the requested mobile sink. The scheme uses two separate polynomial pools: the mobile polynomial pool and the static polynomial pool. Using two separate key pools and having few sensor nodes that carry keys from the mobile key pool will make it more difficult for the attacker to launch a mobile sink replication attack on the sensor network by capturing only a few arbitrary sensor nodes. Rather, the attacker would also have to capture sensor nodes that carry keys from the Mobile key pool. Keys from the mobile key pool are used mainly for mobile sink authentication, and thus, to gain access to the network for data gathering. Although the above security approach makes the network more resilient to mobile sink replication attacks compared to the single polynomial pool-based key predistribution scheme [3].

Pairwise key establishment is another important fundamental security service. It enables sensor nodes to communicate securely with each other using cryptographic techniques.

The main problem is to establish a secure key shared between two communicating sensor nodes. However, due to the resource constraints on sensor nodes, it is not feasible for them to use traditional pairwise key establishment techniques such as public key cryptography and key distribution center (KDC).

Eschenauer and Gligor proposed a probabilistic key pre-distribution scheme recently for pairwise key establishment. The main idea is to let each sensor node randomly pick a set of keys from a key pool before the deployment so that any two sensor nodes have a certain probability to share at least one common key. Chan et al. further extended this idea and developed two key pre-distribution techniques: a q-composite key pre-distribution scheme and a random pairwise keys scheme. The q-composite key pre-distribution also uses a key pool but requires that two nodes compute a pairwise key from at least q predistributed keys that they share. The random pairwise keys scheme randomly picks pairs of sensor nodes and assigns each pair a unique random key. Both schemes improve the security over the basic probabilistic key pre-distribution scheme.However, the pairwise key establishment problem is still not fully solved.

For the basic probabilistic and the q-composite key pre-distribution schemes, as the number of compromised nodes increases, the fraction of affected pairwise keys increases quickly. As a result, a small number of compromised nodes may disclose a large fraction of pairwise keys and also achieves significant security under small scale attacks at the cost of greater vulnerability to large scale attacks.

The problem of authentication and pair wise key establishment in sensor networks with MSs is still not solved in the face of mobile sink replication attacks. For the basic probabilistic and q-composite key pre distribution schemes, an attacker can easily obtain a large number of keys by capturing a small fraction of the network sensor nodes, making it possible for the attacker to take control of the entire network by deploying a replicated mobile sink, preloaded with some compromised keys to authenticate and then initiate data communication with any sensor node.

There is a tradeoff to be made between security and vulnerability that has to be considered based on the sensor network size and application.

The proposed scheme uses two separate polynomial pools: the mobile polynomial pool and the static polynomial pool [7]. Polynomials from the mobile polynomial pool are used to establish the authentication between mobile sinks and stationary access nodes, which will enable these mobile sinks to access the sensor network for data gathering. Thus, an attacker would need to compromise at least a single polynomial from the mobile pool to gain access to the network for the sensor's data gathering. Polynomials from the static polynomial pool are used to ascertain the authentication and keys setup between the sensor nodes and stationary access nodes. Prior to deployment, each mobile sink randomly picks a subset of polynomials from the mobile polynomial pool. In this scheme, to improve the network resilience to mobile sink replication attack as compared to the single polynomial pool based approach; intend to minimize the probability of a mobile polynomial being compromised if Rc sensor nodes are captured. As an adversary can use the captured mobile polynomial to launch a mobile sink replication attack, achieve this by having a small fraction of randomly selected sensor nodes carry a polynomial from the mobile polynomial pool. These preselected sensor nodes are called the stationary access nodes. They act as authentication access points for the network and trigger sensor nodes to transmit their aggregated data to the mobile sinks.

A mobile sink sends data request messages to the sensor nodes via a stationary access node. The mobile sink's data request messages will initiate the stationary access node to trigger sensor nodes to transmit their aggregated data to the requested sink. Each stationary access node may share a mobile polynomial with a mobile sink. All sensor nodes, including the stationary access nodes, randomly select a subset of polynomials from the static polynomial pool. The advantage of using separate pools is that mobile sink authentication is independent of the key distribution scheme used to connect the sensor network. This scheme is divided into two stages: static and mobile polynomial predistribution and key discovery between a mobile sink and a sensor node [8].

Static and mobile polynomial predistribution:

Stage 1 is performed before the nodes are deployed. A mobile polynomial pool |M| of size |M |and a static polynomial pool S of size |S| are generated along with the polynomial identifiers. All mobile sinks and stationary access nodes are randomly given Km and one polynomial (Km > 1) from M. The number of mobile polynomials in every mobile sink is more than the number of mobile polynomials in every stationary access node. This assures that a mobile node shares a common mobile polynomial with a stationary access node with high probability and reduces the number of compromised mobile polynomials when the stationary access nodes are captured. All sensor nodes and the preselected stationary access nodes randomly pick a subset of Ks and Ks -1 polynomials from S. Fig. 2 show the key discovery between the mobile node and stationary node.

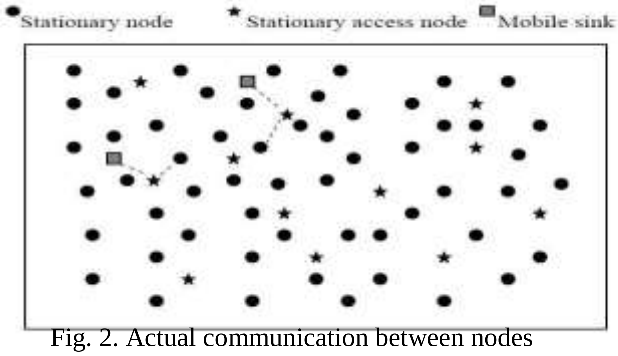


Fig. 2. Actual communication between nodes

Key discovery between mobile node and stationary node:

To establish a direct pairwise key between sensor node u and mobile sink v, a sensor node u needs to find a stationary access node a in its neighborhood, such that, node a can establish pairwise keys with both mobile sink v and sensor node u. In other words, a stationary access node needs to establish pairwise keys with both the mobile sink and the sensor node. It has to find a common mobile polynomial with the mobile sink and a common static polynomial with the sensor node. To discover a common mobile/static polynomial, a sensor node i may broadcast a list of polynomial IDs, or alternatively, an encryption list α, EKv(α), v = 1, . . . ,|Ksi|, where Kv is a potential pairwise key and the other node may have as suggested in [4] and [5].

When a direct secure path is established between nodes u and v, mobile sink v sends the pairwise key Kc to node a in a message encrypted and authenticated with the shared pairwise key Kv,a between v and a. If node a receives the above message and it shares a pairwise key with u, it sends the pairwise key Kc to node u in a message encrypted and authenticated with pairwise key Ka,u between a and u.

If the direct key establishment fails, the mobile sink and the sensor node will have to establish a pairwise key with the help of other sensor nodes. To establish a pairwise key with mobile sink v, a sensor node u has to find a stationary access node a in its neighborhood such that node a can establish a pairwise key with both nodes u and v. if node establishes a pairwise key with only node v and not with u. As the probability is high that the access node a can discover a common mobile polynomial with node v, sensor node u needs to find an intermediate sensor node i along the path u-- i -- a --v, such that intermediate node i can establish a direct pair wise key with node a.

The use of mobile sinks in WSN introduces a new security challenge. wireless sensor network with one mobile sink and a base station. Sensor nodes store the generated data in their buffers. The mobile sink traverses the network using random walk, periodically transmitting beacon signals. Sensor nodes that hear the mobile sink's beacon transmission begin transferring their aggregated data to the mobile sink. Since the mobile sink's beacon signal received by sensor nodes is not authenticated, an adversary can attack the network by placing a malicious mobile sink.

Problem Description

1) Algorithm and Analysis

Paillier Cryptography Algorithm:

The Paillier encryption scheme is composed of key generation, encryption, and decryption algorithms as follows [6]:

Key Generation: Choose two large prime numbers p and q randomly and independently of each other, such that gcd(pq, (p-1)(q-1)) = 1

This property is assured if both primes are of equal length.

Compute

$$n = pq, \lambda = \text{lcm}\ (p\text{-}1)\ (q\text{-}1)$$

where lcm stands for the least common multiple.

Select random integer g where $g\epsilon Z*_{n2}$.

Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:

$$\mu = (L(g\hat{}\lambda\ (\text{mod}\ n^2)))^{-1} \text{ mod } n$$

where function L is defined as,

$$L(u)= u\text{-}1/n$$

Note that the notation a=b does not denote the modular multiplication of a times the modular multiplicative inverse of b, but rather the quotient of a divided by b.

Finally, the public (encryption) key is (n,g) and the private (decryption) key is (λ,μ).

If using p; q of equivalent length, a simpler variant of the above key generation steps would be to set

$$g = n + 1, \lambda = \varphi(n), \mu= \varphi(n)^{-1} \text{ mod } n$$

Where φ (n) = (p-1) (q-1).

Encryption: Let m be a message to be encrypted where $m \epsilon Z_n$.

Select random r where $r \epsilon Z_n^{*.}$ Compute cipher text as

$$c = g^m.r^n\ (\text{ mod } n^2)$$

Decryption: Let c be the ciphertext to decrypt, where $c \epsilon Z_{n2}^{*.}$

Compute the plaintext message as:

$$m = L\ (c\hat{}\ \lambda\ (\text{ mod } n^2))\ \mu\ (\text{mod } n).$$

As the original paper points out, decryption is “essentially one exponentiation modulo n^2”.

The Paillier encryption scheme exploits the fact that certain discrete logarithms can be computed easily. For the implementation purpose all the no.’s are used as BigInteger, so theres no limitations for no’s here.

Paillier Example: An example of the Paillier encryption scheme with small parameters is shown as follows. For ease of calculations, the example will choose small primes, to create a small n. Let

$$p = 7; q = 11$$

then

$$n = p.q = 7 .11 = 77$$

IJERGS

Next, an integer g must be selected from Z_{n2}^{*} , such that the order of g is a multiple of n in Z_{n2} . If we randomly choose the integer

$$g = 5652$$

Then all necessary properties, including the yet to be specified condition, are met, as the order of g is 2310 = 30.77 in Z_{n2} . Thus, the public key for the example will be

$$(n, g) = (77, 5652)$$

To encrypt a message

$$m = 42$$

Where m ϵ Z_n , choose a random

$$r = 23$$

Where r is a nonzero integer, rϵZ_n . Compute

$$c = g^m\, r^{\,n} \pmod{n^2}$$

$$= 5652^{42}.23^{77}(\text{mod } 5929)$$

$$= 4624 \text{ (mod } 5929)$$

To decrypt the cipher text c, compute

$$\lambda = \text{LCM}\ (6, 10) = 30$$

Define L (u) = (u-1)/n, compute

$$k = L\ (\ g\hat{}\ \lambda\ (\text{ mod } n^2\ ))$$

$$= L\ (565230\ (\text{mod } 5929))$$

$$= L\ (3928)$$

$$= (3928\text{-}1)\ /\ 77$$

$$= 3927\ /\ 77 = 51$$

Compute the inverse of k,

$$\mu = k^{-1} \text{ (mod n)}$$

$$= 51^{-1} = 74 \text{ (mod 77)}$$

Compute

$$m = L\ (c\hat{}\lambda \text{ mod } n^2).\mu \text{ (mod n)}$$

$$= L\ (462430\ (\text{mod } 5929)).74 \text{ (mod 77)}$$

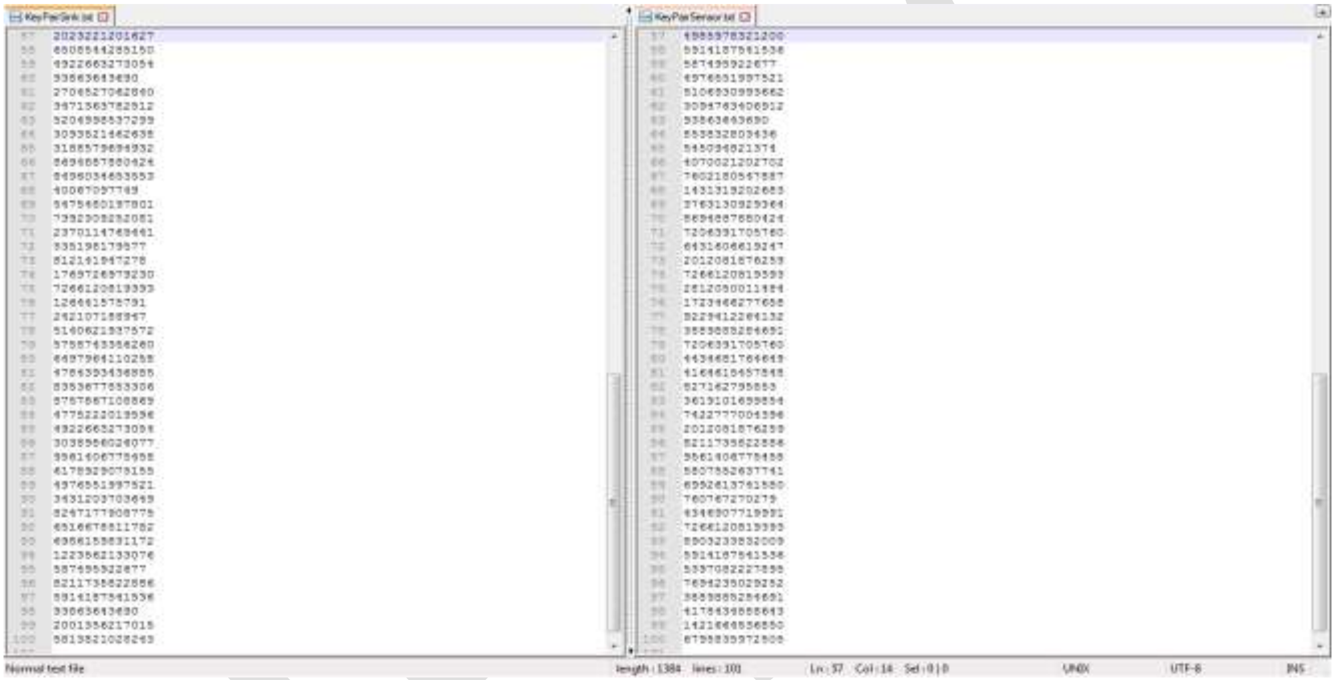
$$= L\ (4852).\ 74 \text{ (mod 77)}$$

$$= 42$$

IJERGS

Result and Discussion

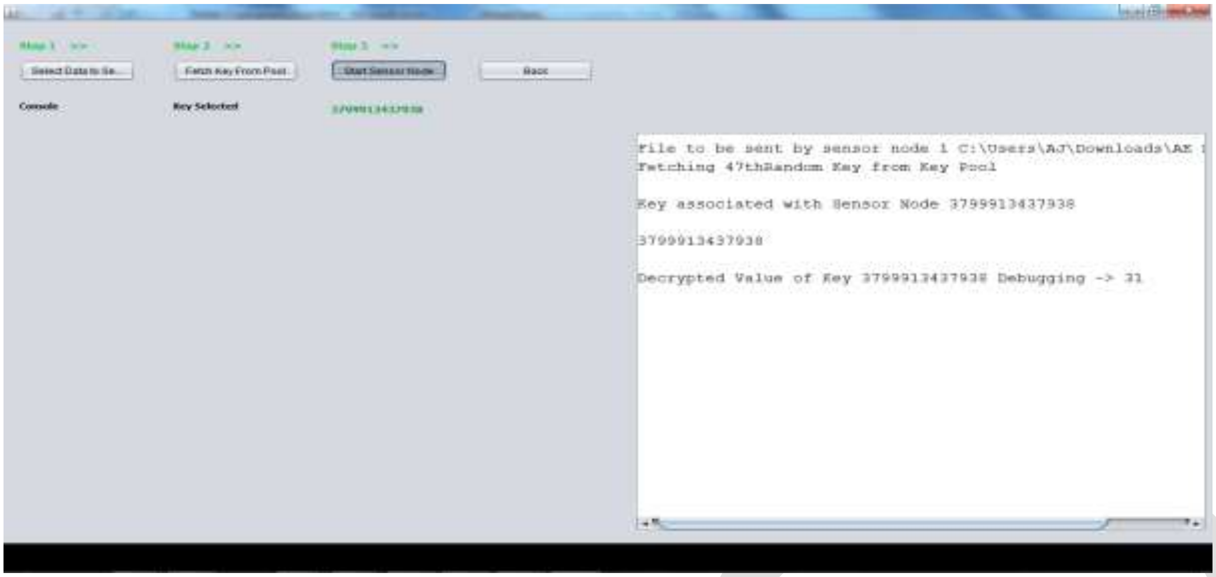
The Pailier Cryptosystem, with the pair wise key distribution scheme i.e. polynomial pool based key scheme used for the implementation of this project. For key generation means separately created mobile polynomial pool and static polynomial pool using above algorithm, so here main thing which the data transaction which done by encryption and decryption of the keys of mobile sink and sensor node, via the stationary access nodes, whenever any mobile sink needed the data, as it gets that request from particular base station, so mobile sink collects the data from sensor, but direct communication is not possible as this is three tier communication which is added for security purpose, so in this proposed scheme implementation contains mobile sink as a node which has factors as IP address of that sensor from which it needs the data, also it can save that file according to its specification, key fetching happens in this phase as key which is encrypted by sensor that should be decrypted by mobile sink using its keys, then only communication will be successful, also sensor contains all the data, so as it gets any request for the particular data, it select that data from its memory and start the fetching of the keys, i.e encryption of keys, then it forwards the data and wait for the particular stationary access node or mobile sink to catch the data, the mobile sink whose flag is on, and who has the ability to decrypt the encrypted data, he receives the data.



```
2023221201627
6508544285150
4922663273054
33863643690
2706527062860
3471363782812
5204998537299
3093821462638
3188579694932
8696887880424
8498034653553
40087097749
5475460197801
7392308252081
2370114769461
335198179577
812141947278
1769726979230
7266120819393
126661575791
242107188947
5140621937572
5755743356260
6487964110255
4784393436885
8353677653306
5757887108869
4775222013996
4922663273054
3038956026077
5561406775455
6178929079155
4976551997521
3431203703649
8247177908775
6516678811782
6986159831172
1223862133076
587495922677
8211738622886
5914187541536
33863643690
2001356217015
5813821026263
```

```
4985978321200
5914187541536
587495922677
4976551997521
5106930993662
3094763406912
33863643690
553832803436
545094821374
4070021202702
7602180567587
1431319202683
3763130929364
8696887880424
7206391705760
6431606619247
2012081876259
7266120819393
2812050011484
1723466277658
8229412264132
3553688284691
7206391705760
4434481764649
4166615457548
527162795853
3619101699854
7422777004396
2012081876259
8211738622886
5561406775455
5807882637741
6952613761550
760767270279
4346907719991
7266120819393
8903233832009
5914187541536
5337082227898
7694235029252
3553688284691
4178434888643
1421664536850
6799835972808
```

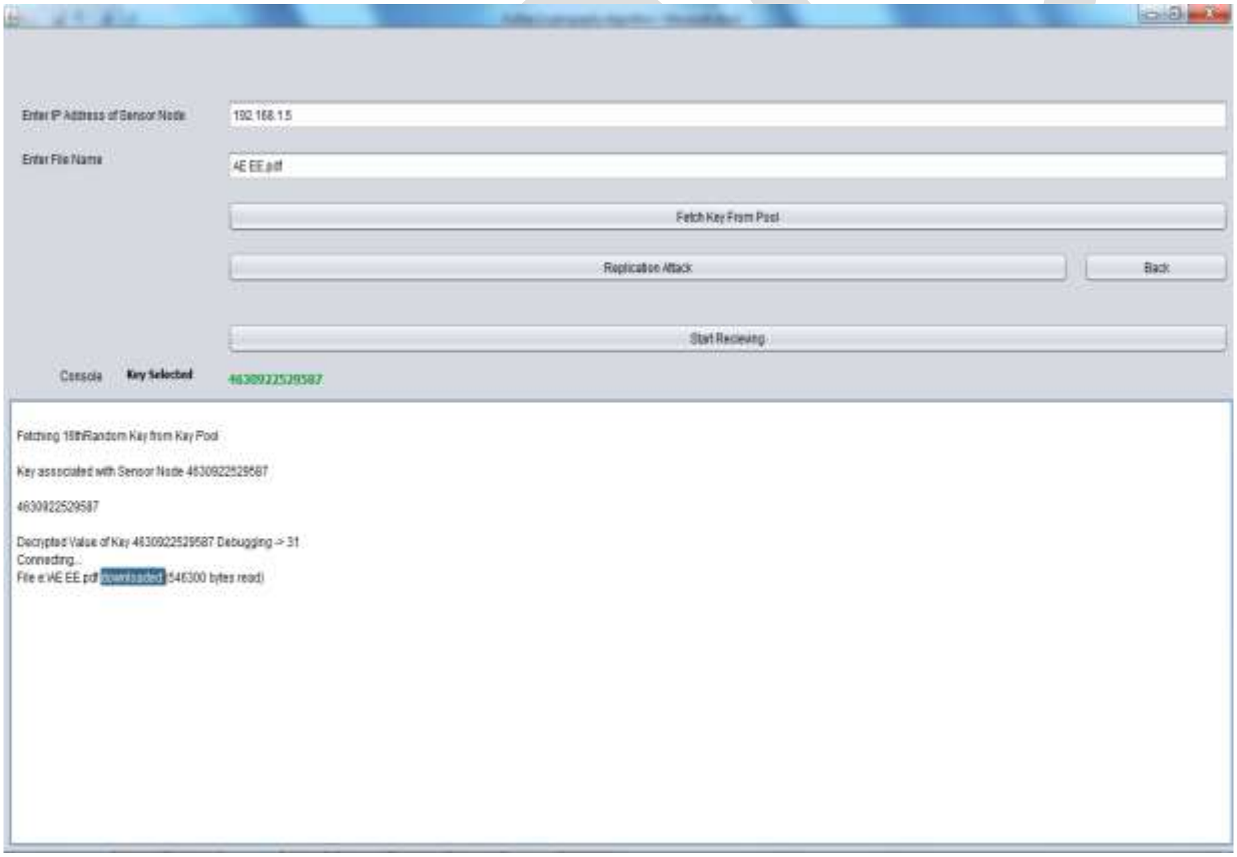
Fig 3: Key generation in mobile sink polynomial pool and static polynomial pool



3799913437938

```
File to be sent by sensor node 1 C:\Users\AJ\Downloads\AE 
Fetching 47thRandom Key from Key Pool

Key associated with Sensor Node 3799913437938

3799913437938

Decrypted Value of Key 3799913437938 Debugging -> 31
```

Fig 4: Sensor node fetching the key from static polynomial pool and ready to send the data to node


Enter IP Address of Sensor Node: 192.168.1.5

Enter File Name: AE EE.pdf

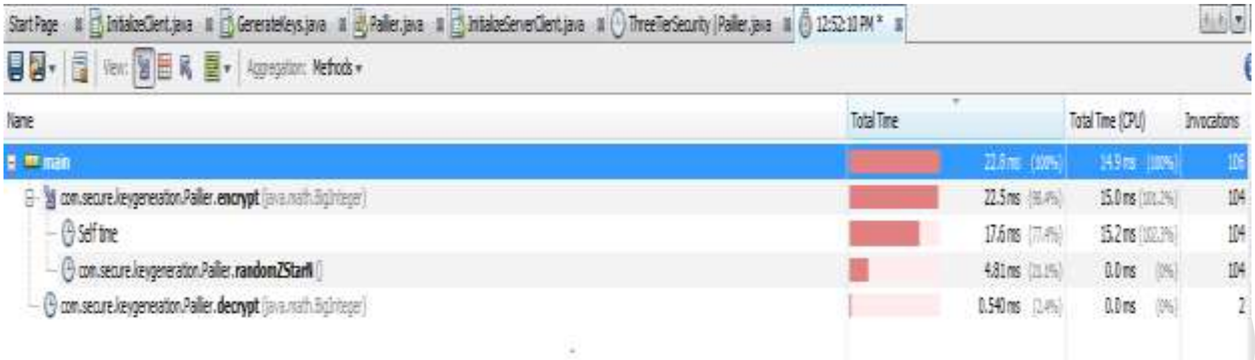
4630922529587

```
Fetching 18thRandom Key from Key Pool

Key associated with Sensor Node 4630922529587

4630922529587

Decrypted Value of Key 4630922529587 Debugging -> 31
Connecting...
File e:\AE EE.pdf downloaded (546300 bytes read)
```

Fig 5: Mobile sink whose flag is on, ready to receive the data, by fetching the key from pool.

The mobile sink in the implementation contains a particular access, so it can check whether there is any mobile replication attack is there or not, as anyone sends the data, and it checks, is this the attack or not, first of all checks whether key matches or not, if not then it's not a valid node, and it will stop the particular transaction here.

Also, the results are calculated on the basis of key generation, encryption, decryption timing, as compared to other cryptosystems, its timing is very much less, mostly it's in milliseconds.



Name	Total Time	Total Time (CPU)	Invocations
main	22.6 ms (100%)	14.9 ms (100%)	106
com.secure.keygeneration.Pailler.**encrypt** (java.math.BigInteger)	22.5 ms (99.4%)	15.0 ms (101.2%)	104
Self time	17.6 ms (77.4%)	15.2 ms (102.3%)	104
com.secure.keygeneration.Pailler.**randomZStarN** ()	4.81 ms (21.1%)	0.0 ms (0%)	104
com.secure.keygeneration.Pailler.**decrypt** (java.math.BigInteger)	0.540 ms (2.4%)	0.0 ms (0%)	2

Fig 6: Computation time required for key generation, encryption & decryption.

S.No	Algorithm	Pack Size (KB)	Encryption Time(sec)	Decryption Time (sec)
1.	*AES*	153	3.0	1
	DES		1.6	1.1
	RSA		7.3	4.9
	Pallier Crypt.		1.2	0.7
2.	*AES*	118	3.2	1.2
	DES		1.7	1.2
	RSA		10.0	5.0
	Pallier Crypt.		1.1	0.5
3.	*AES*	868	4.0	1.8
	DES		2.0	1.2
	RSA		8.2	5.1
	Pallier Crypt.		1.5	0.9

Table 1: Difference in time taken in various systems for encryption and decryption.

ACKNOWLEDGMENT

This paper could not be written to its fullest without prof. B.S.Sonawane, who is my guide, as well as one who challenged and encouraged me throughout my time spent studying under him. He would have never accepted anything less than my best efforts, and for that, I thank him.

IJERGS

Conclusion

The Three-Tier security architecture overcomes the drawbacks of existing system and gives the better resilience against the attackers. As two separate pools are used for the purpose of authentication the attackers would not be able to capture the node information. Using two separate key pools and few stationary access nodes which carry polynomials from the mobile pool in the network, an attacker may fail to gather sensor data,

Larger the pool size, lower the probability of two pairs of nodes sharing the same key. The number of keys to be assigned to each sensor does not depend on the size of the WSN which improve the network connectivity.

Based on the two polynomial pool-based key predistribution scheme with the Pailier Cryptosystem algorithm substantially improved network resilience to mobile sink replication attacks as compared to the single polynomial pool-based key predistribution approach. Also it is not increasing the communication overhead.

Analysis indicates that key generation, encryption and decryption time is very much less as compared to other cryptosystems, also system becomes resilient against denial of service attack or brute force attack, as adversary has to guess the original prime no. which used for pailier algorithm, which highly difficult because of there's huge no. of prime no. available.

REFERENCES:

TABLE I. A. Rasheed and R. Mahapatra, "An Efficient Key Distribution Scheme for Establishing Pairwise Keys with a Mobile Sink in Distributed Sensor Networks," Proc. IEEE 27th Int'l Performance Computing and Comm. Conf. (IPCCC '08), pp. 264-270, Dec. 2008.

TABLE II. H. Chan, A. Perrig, and D. Song, "Key Distribution Techniques for Sensor Networks," Wireless Sensor Networks, pp. 277-303, Kluwer Academic, 2004.

[3] Keith M. Martin, Maura B. Paterson, and Douglas R. Stinson. Key predistribution for homogeneous wireless sensor networks with group deployment of nodes, 2008.

[4] Seyit Ahmet C¸ amtepe and B¨ulent Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. In ESORICS, pages 293–308, 2004.

[5] R. Kannan S.S. Iyengar R. Kalidindi and A. Durresi. Sub-grid based key vector assignment: A key pre- distribution scheme for distributed sensor networks. Journal of Pervasive Computing and Communications, 2(1):35–43, 2006.

[6] P. Paillier, D. Pointcheval, Efficient public-key cryptosystems provably secure against active adversaries, in Proceedings of Advances in Cryptology, ASIACRYPT'99, 1999, pp. 165–179

[7] A. Rasheed and R. Mahapatra, "An Efficient Key Distribution Scheme for Establishing Pairwise Keys with a Mobile Sink in Distributed Sensor Networks," Proc. IEEE 27th Int'l Performance Computing and Comm. Conf. (IPCCC '08), pp. 264-270, Dec. 2008.

[8] A. Rasheed and R. Mahapatra, "A Key Pre-Distribution Scheme for Heterogeneous Sensor Networks," Proc. Int'l Conf. Wireless Comm. and Mobile Computing Conf. (IWCMC '09), pp. 263-268,June 2009.

IJERGS

[9] A. Rasheed and R. Mahapatra, “Three-Tier security scheme in wireless sensor network with mobile sink,” IEEE Transaction on parallel and distributed system,vol-23,no.5,May-2012.

[10] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-Secure Key Distribution for Dynamic Conferences,”Proc. 12th Ann. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO ’92), pp. 471-486, 1993.

[11] L. Eschenauer and V.D. Gligor, “A Key-Management Scheme for Distributed Sensor Networks,” Proc. ACM Conf. Computer Comm. Security (CCS ’02), pp. 41-47, 2002.

IJERGS