Energy Aware Load Balancing and Secure Continuous Aggregation for Wireless Sensor Networks

Dr.T.Abirami
Assistant Professor (SRG),
Department of IT, Kongu Engineering College,
Perundurai

D.Evangeline
Assistant Professor (SRG),
Department of IT, Kongu Engineering College,
Perundurai

ABSTRACT

Continuous aggregation is required in sensor applications to obtain the temporal variation information of aggregates. It helps the users to understand how the environment changes over time. A Wireless Sensor Network (WSN) is a energy and security constraint network. Clustering is used for load balance to extend the lifetime of a sensor network by reducing energy consumption .In the existing secure aggregation scheme, once the cluster heads are attacked by malicious attacker, compromised nodes in the network will forge false values as the aggregation results of other nodes and it does not balance the load among clusters so it provide less throughput. Tricking the base station into accepting false aggregation results in networks is envisioned to be economic solutions to many important applications. Energy efficient load balancing algorithm is proposed to balance the load among the clusterby using some backup nodes. This approach will increase the network lifetime, high throughput and avoid overload by distributing work among identical type of sensor nodes with energy and security efficient routes. In the continuous aggregation, the adversary could manipulate a series of aggregation results through compromised nodes to fabricate false temporal variation patterns of the aggregates. To make the aggregates more effective the data packets are transmitted in a secure manner by using the digital signature cryptosystem .

Keywords

wireless sensor networks, load balancing, backup nodes,continuous data aggregation, digital signature

1. INTRODUCTION

Wireless Sensor Networks (WSN) are widely used for environmental and security monitoring. Small wireless sensors have the capacity to sense, compute, store and transmit; it integrates with each other to form a network. In a WSN, the sensors monitor the immediate surroundings, and the data is transmitted to a well-equipped node called the Sink. Patterns in the data are analyzed offline, but this results in transmitting a large amount of data through the network leading to communication overhead. Protocols can reduce transmitted power in two ways. First where nodes can emit to short distances such as data sinks or cluster nodes. The cluster node can then send the data over a larger distance preserving the power of the smaller nodes. The second is by reducing the number of bits (amount of data) sent across the wireless network applications of wireless sensor networks (WSNs), the aggregations of sensed data, such as sum, average, and predicate count, are very important for the users to get summarization information about the monitored area. Instead of collecting all sensor data and computing aggregation results at the base station (BS), in network aggregation allows sensor readings to be aggregated by intermediate nodes, which efficiently reduces the communication overhead.In this paper, we consider the security of continuous in network aggregation in WSNs. In many WSN applications the users often need the temporal variation information in a series of aggregation results rather than an individual aggregation result. For a continuous aggregation query, a time interval, called epoch, is specified and the aggregation is evaluated in every epoch. The duration of every epoch specifies the amount of time sensor nodes wait before acquiring and transmitting each successive sample. Because of the importance of temporal variation information of aggregation results, we focus on the attack against continuous in-network aggregation that the adversaries attempt to distort the real temporal variation pattern of the aggregate by disrupting a series of successive aggregation results.

2. METHODS AND IMPLEMENTATION

The proposed approach assume network with the sensor nodes having different energy levels and processing power. Some high computing nodes are deployed nearby each other. All the nodes with high initial energy level and processing power are selected. Some nodes from the set are selected as cluster head (CH) according to their location. Each CH defines its communication range in terms of power level to form cluster. Some nodes with comparable energy and processing power in the CH range are asked to go to sleep and information about those nodes is maintained with the CH. All the cluster members will send the sensed data to the CH. The CH will send the aggregated data to the Base Station directly or by using some intermediate CH.

2.1 Implementation of load balancing algorithm

Algorithm 1: Load Balancing in WSN

```
Require: Initialize N Nodes with
Require: L <= Number of Work Load Processes (l1,l2,l3...ln=L)

while l <= L do
  while i <= N do
if Type of node i belongs to a group==Process type of l then
Allocate this process l to Node i.
  else
Allocate load to free node which can belongs to group.
      end if
    end while
end while
```

In this algorithm CH sends membership request message to all the nodes in its range and request to replywith their current energy status. The nodes with high residual energy will be identified and they aremade to sleep by using the equation 2.1. They become the backup nodes. The nodes which are not in the range of cluster head, will try to join the cluster by sendingthe message to the nearest cluster member.The residual energy (RE) of each node (Ni) is calculated by using formula.

$$\mathbf{RE = Ei - (Etx + Erx + Ea)} \qquad \mathbf{(2.1)}$$

Where,

- RE is the Residual Energy.
- Ei is an Initial Energy of Each node.
- Etx is an Energy utilized at the time of transmission.
- Erx is an Energy utilized at the time of data reception
- Ea is an Energy required to keep the node active.

Again, the same criterion of finding the minimum distance cluster member is applied to find the appropriate cluster

2.2 Implementation of Energy Efficient Load Balancing

Algorithm 2: Energy Efficient Load Balancing in WSN

Require: Initialize N Nodes with L; PG; LG; etc

Require: L <= Number of Work Load Processes(l1,l2,l3...ln=L)

Require: N <= Number of Nodes in the network(n1,n2,n3...nm=N)

Require: E <= Energy Level of each node in N(e1,e2,e3...en=E), here e1 for n1, e2 for n2...

```
while l <= L do
  while i <= N do
    if Type of node i belongs to a group == process type of l then
      while until find a node from i which consumes minimum energy e do
        Allocate this process l to Node i.
      end while
    else
      while Until find a node from i which consumes minimum energy e do
        Allocate this process l to Node i.
        Allocate load to free node which can belongs to group.
      end while
    end if
  end while
end while
```

With the load balancing algorithm, the implementation of an efficient energy strategy, so that the network life can be improved and increased with load balancing. In Algorithm 2, the process of assigning workload to nodes with energy efficient. In this network, L is Number of Work Load Processes called l1,l2,l3...ln and N is the Number of Nodes in the network called n1,n2,n3...nm and E is Energy Level of each node in N called e1,e2,e3...en, here e1 corresponds to n1, similarly e2 corresponds n2 and so on.

The cluster members send the sensed data to the CH in the allotted time using TDMA schedule. The non cluster members will send the sensed data to the cluster head through the intermediate cluster member. When the energy level of the CH will reach to the threshold value TL, the CH will activate one of the sleeping nodes and will make it CH. This information about the new CH will be sent to all the cluster member and other CH also. The old CH will become the general sensor node. The CH will sent the aggregated data to the base station directly or by using some intermediate CH.

2.3 Secure Data Aggregation

This module performs the aggregation function. The system performs continuous secure aggregation on every node by collecting a physical quantity of humidity and temperature from the surrounding environment and process the acquired data and transfer the collected data to a sink node or base station. The private and public keys are generated using the RSA algorithm. It is used to provide both secrecy and digital signatures. Its security is based on the intractability ofthe integer factorization problem. The major advantage of RSA is that it does not increase the size of the message. It may be used to provide privacy and authentication over communication links through digital signatures. The security has been analyzed using RSA Public key crypto system.

Initially it is assumed that all the sensor nodes have their unique public key during its deployment in the phenomena. During the route construction phase, the sink broadcasts Route Construction (RCON) packets to its neighbouring nodes. The neighbouring nodes receive the RCON packet. A neighbouring node updates RCON packet with its public key. It rebroadcast the RCON packet to its neighbouring nodes. Similarly all the nodes in the network update their routing table with their neighbouring node’s public key.

2.3.1 Key Generation

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q. For security purposes, the integers p and q should be chosen at random, and should be of similar bit length. Prime integers can be efficiently found using a primality test.

2. Compute n = p*q.
n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

3. Compute $\varphi(n) = \varphi(p)\,\varphi(q) = (p - 1)(q - 1)$ where φ is Euler's totient function. (3.3)

4. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$ e and φ (n) are prime.

5. Determine *d* as $d \equiv e^{-1} \pmod{\varphi(n)}$, *d* is the multiplicative inverse of *e* (modulo φ(*n*)).

The public key consists of the modulus n and the public exponent e. The private key consists of the modulus n and the private exponent d, which must be kept secret. p, q and φ (n) must also be kept secret because they can be used to calculate d.

2.3.2 Encryption of Digital Signature Using RSA Algorithm

The source node will generate the digital signature d_{sign} by encrypting the message with its private key d. The source node forwards d_{sign} with data M, (d_{sign}, M) to its neighbouring node through the path it takes to reach sinkas shown in the Fig 2.1

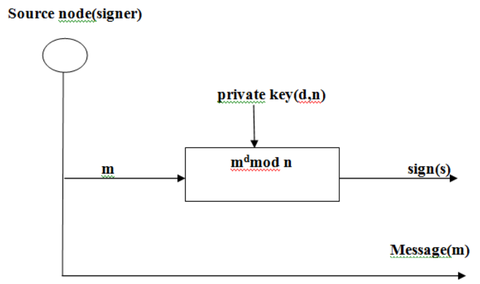

Fig 2.1 Encryption of digital signature using RSA

2.3.3 Verification of digital signature

A neighbouring node on reception of signature and message verifies the digital signature by comparing decrypted value of d_{sign} mod n using sender's public keys as shown in the Fig 2.2. If the generated message by the receiver and the decrypted message of digital signature $d_{sign}$is equal, then the receiver accepts the data, otherwise rejects the data and informs the sender that the data is altered through by generating route error packet. This process is repeated in every hop of the node between source and destination.

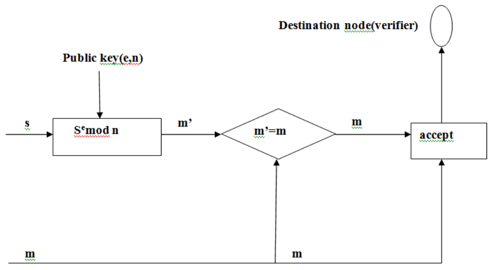

Fig 2.2 Verification of digital signature using RSA

3. PERFORMANCE OF AGGREGATION VERIFICATION

Nodes Vs Delay

The time taken for a packet to transmit from source to destination is end to end delay. The delay for proposed is reduced by 2.78% than existing work. The end to end delay variation is shown in Fig 3.1.

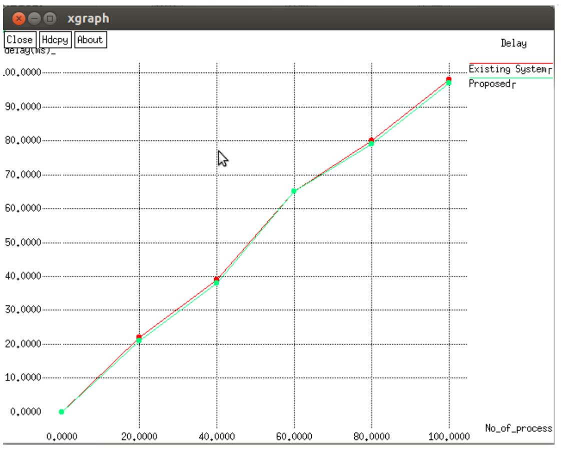

Fig 3.1 Nodes Vs Delay

Nodes Vs Energy

Energy is defined as the amount of energy consumed in a network. It is usually measured in Joules. In the Fig 3.2 energy consumption is minimized by 3.28% than existing work.

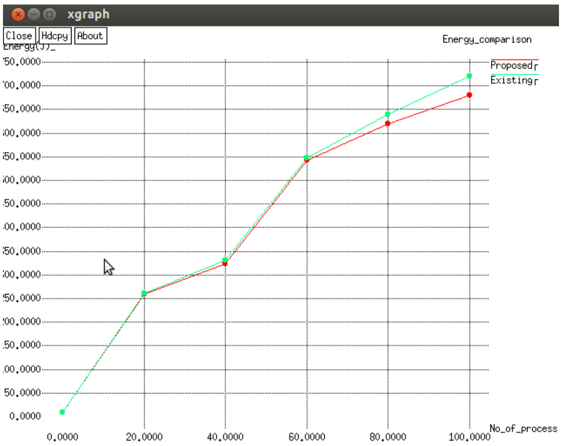

Fig 3.2 Nodes Vs Energy

Nodes Vs Throughput

Throughput is usually measure in bit per second, and sometimes in data packets per second. Throughput of proposed work gives 2.88% better performance than existing work. Throughput variation is shown in Fig 3.3

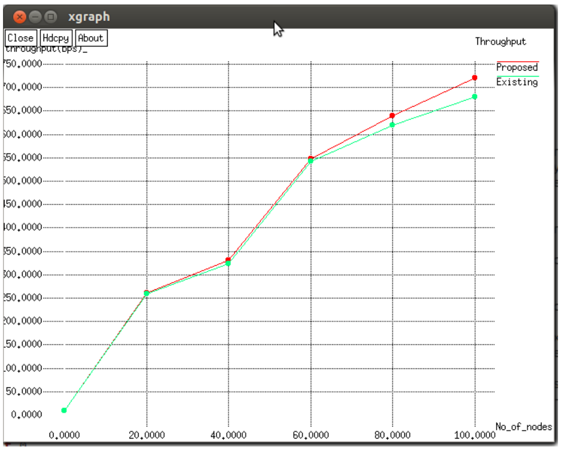

Fig 3.3 Nodes Vs Throughput

4. CONCLUSION

A Wireless Sensor Network (WSN) is a energy and security constraint network. Compared with the existing secure aggregation schemes, the proposed scheme attempting to apply efficient algorithm for load, energy and security such that network life can be increased with security. Energy effective load reconciliation in a WSN needs to spread workload across multiple sensor nodes therefore, energy effective load reconciliation will achieved and optimization of resource usage, maximize throughput, maximize network lifetime, and avoid overload by distributing work among identical type of sensor nodes with energy and security efficient routes.The support of RSA digital signature algorithm, provides high security, high throughput with reduced key size. So it provides a high secure and energy efficient solution for WSNs and it protect the aggregator node not being compromised by the attacker.

REFERENCES

[1] Lei S. and Li Y. (2014), ‘Secure Continuous Aggregatio in Wireless Sensor Networks’, proceeding IEEE Transaction on Parallel and Distributed Systems,pp. 265-266.

[2] SajidHussain D. and Abdul Matin W. (2013), ‘Hierarchical Cluster-based Routing in Wireless Sensor Networks’, proceeding on IEEE Transaction on Computing Systems (ICDCS), pp. 255-259.

[3] Ji S. and Cai Z. (2012), ‘Distributed Data Collection and Its Capacity in Asynchronous Wireless Sensor Networks’, IEEE INFOCOM, pp. 2113-2122.

[4 Ji S., Beyah R. and Cai Z. (2012), ‘Snapshot/Continuous Data Collection Capacity for Large-Scale Probabilistic Wireless Sensor Networks’, proceeding on IEEE INFOCOM, pp. 1035-1043.

[5] Cai Z., Ji S. and Bourgeois A G. (2012), ‘Optimal Distributed Data Collection for Asynchronous Cognitive Radio Networks’, IEEE Transaction on Distributed Computing Systems (ICDCS), pp. 245-254.

[6] Yang Y., Wang X., and Cao G. (2006), ‘SDAP: A Secure Hop-By- Hop Data Aggregation Protocol for Sensor Networks’, ACM Mobile Hoc, pp. 356-367.

[7] Chan H., Perrig A., and Song D. (2006), ‘Secure Hierarchical In-Network Aggregation in Sensor Networks’, ACM Conference Computer and Communication Security (CCS), pp. 278-287.

[8] Dr.T.Abirami, M.Meenalochini(2014) Secure Data Aggregation with False Temporal Pattern Identification for Wireless Sensor Networks,International Journal of Engineering and Advanced T echnology,pp.195-197

[9] Dr.T.Abirami,M.Meenalochini(2015) , Secure continuous aggregation with load balancing for wireless sensor networks, international journal of advanced research trends in engineering and technology,pp.72-76

[10] Liu D. (2003), ‘Establishing Pairwise Keys in Distributed Sensor Networks”, ACM Conference on Computer and Communication Security (CCS), pp. 52-61.

[11] Przydatek B., and Perrig A. (2003), ‘SIA: Secure Information Aggregation in Sensor Networks’, ACM Conference on Embedded Networked Sensor Systems, pp. 255-265.

[12] Madden S. (2002), ‘Tag: A Tiny Aggregation Service for Ad-Hoc Sensor Networks’, Operating Systems Design and Implementation (OSDI), pp.4192-4203.