A Chinese Remainder Theorem Based Key Management Algorithm for Hierarchical Wireless Sensor Network

Pranave Kumar Bhaskar and Alwyn R. Pais

Department of Computer Science and Engineering, NITK Surathkal, India
pkbls.10@gmail.com
alwyn@nitk.ac.in

Abstract. Wireless Sensor Networks (WSN) are network of sensors having low computation, storage and battery power. Hierarchical WSN are heterogeneous network of sensors having different capabilities which form a hierarchy to achieve energy efficiency. Key management algorithms are center of the security protocols in WSN. It involves key pre distribution, shared key discovery, key revocation, and refreshing. Due to resource constraints in WSN achieving a perfect key management scheme has been quite challenging. In this paper a new key management scheme for Hierarchical WSN based on Chinese Remainder Theorem has been proposed. An experimental setup is created to evaluate this scheme. The results indicate that it establishes the key with minimum computation, communication, storage cost at each node, also it is scalable and resilient to different attacks.

1 Introduction

Since the evolution of practical cryptography, key management has been subject of attention. This is mainly because prior to any secure communication, encryption/decryption key must be obtained. Key exchange generally uses public key cryptography, however for a WSN it becomes infeasible, for want of resources. Thus a key management scheme is needed. In this paper a key management algorithm for hierarchical sensor network based on Chinese remainder theorem is proposed.

The organization of this paper is as follows. Section 1 introduces the topic in general. In Section 2 popular existing schemes for sensor network key management are discussed. At the end of this section their respective pros and cons are analyzed. Section 3 discusses the architecture of sensor network and proposed scheme in detail. In Section 4 experimental setup and simulation parameters are explained in brief. In Section 5 the result of the experiments are presented with a detailed discussion on these results and finally an analysis is made. Section 6 provides conclusion and scope for future work.

2 Existing Schemes

This section gives a brief account of different popular schemes for key management with their pros & cons. There are many key management algorithms proposed in literature for WSN [6]. Table 1 presents a comparative analysis of the schemes based

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 311–317, 2015.

on different parameters such as scalability, resilience, process load, communication load and storage load. From Table 1 it is seen that all these algorithms have their respective limitations. Some algorithms provide connectivity but require either heavy computation [3] or they have large storage and communication requirements [2], [4]. Some algorithms do provide key distribution without these shortcomings but they have their own requirements like prior deployment knowledge [5]. The hybrid schemes have other issues such as scalability and lack of resilience to common attacks.

Table 1. Comparison of different key management algorithms for WSN

Protocol	**Theory**	**Resi-lience**	**Process Load**	**Comm. Load**	**Storage Load**
Pure Probabilistic [1]	Random Graph	Medium	Medium	Medium	High
Q Composite [2]	Random Graph	Good	Medium	High	High
Polynomial based [3]	t-degree polynomial	Good	High	Medium	High
Matrix based [4]	Symmetric Matrix	Good	Medium	Medium	High
Deployment Info based [5]	Random Graph	Excellent	Medium	Medium	Medium

Thus there is a need for a novel key management scheme which overcomes the above discussed limitations. A scheme for key management in HSN based on CRT was presented in [7], which discussed the theoretical idea. In this paper this idea is extended and evaluated to make it practical in a real sensor network environment.

3 Proposed Scheme

3.1 Architecture

The sensor network architecture for which the key management algorithm is proposed is HSN (Hierarchical Sensor Network). A HSN is organized into groups called clusters with a CH (cluster head). All communication to base station (BS) happens via this CH. CH generally have larger computation power and memory. Individual sensor nodes in a cluster are responsible to accumulate the sensing data and send it at regular time interval to the CH. Each of these clusters has a group key (GK) which is used for all communication within the cluster. The CH sends these sensed data to BS on request basis. The communication between CH and BS is encrypted via a key that is exclusively shared by each CH and BS. This key is called BGK. Typical architecture of HSN is as shown in Fig. 1.

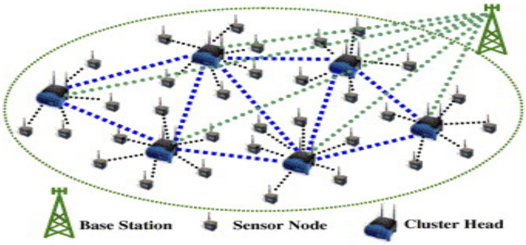

Fig. 1. Hierarchical Sensor Network Architecture

The total numbers of nodes considered in this experiment are 128, 256, 512 and 1024 i.e. total four setups. The number of nodes in each cluster is taken to be 8 and 12.

3.2 Scheme Details

This section explains establishment of group key and rekeying in HSN architecture using CRT. The best algorithm to solve CRT congruence takes m (log n)^3 operations, where m being total equations and n, bit size of keys. In pre-distribution phase each of the sensor node get their private key K_i from the BS's key pool, each of these keys are relatively prime to each other. The BS and CH maintains ID↔K pair in its database for each node. In running phase the cluster is formed by sending HELLO message by the CH, the sensor nodes in the proximity respond to this message and forms the cluster. Once the cluster is formed the CH deletes keying information of nodes not in its cluster. In each of these clusters, the CH now chooses a randomly generated group key GK and forms a congruence system as follows

$X \equiv a_1 \pmod{K_1}$

$X \equiv a_2 \pmod{K_2}$

:

$X \equiv a_n \pmod{K_n}$

Where $a_i = GK \oplus K_i$ and K_i is the secret key of sensor SN_i. The CH solves this congruence to find X. The CH then broadcast this X value to sensor nodes in its group. The sensor nodes will calculate the group key by formula $GK = (X \bmod K_i) \oplus K_i$.

XOR Overflow. While creating the congruence the residuals of congruence is calculated by XOR of node keys with group key i.e. $a_i = GK \oplus K_i$. These a_i some times are greater than K_i. So while creating the congruence instead of using a_i, use $a_i\% K_i$ (i.e. reminder of a_i divided by K_i). Viz. if $K_i= 17$ and $GK = 53$ then $a_i= GK \oplus K_i =36$. In this case the congruence equation becomes:

~~$X \equiv 36 \pmod{17}$~~ => $X \equiv 2 \pmod{17}$

So, while calculating GK at node the value of a_i is taken as 2 instead of 36. Thus the divisor value should be preserved for each congruence equation to get original a_i. For this purpose divisor value d_i is stored for each node and unicasted to individual nodes separately. The formula for calculating group key at node level now changes as follows:

$X \bmod K_i = a'_i$(This is not actual a_i)

$a_i= d_i*K_i + a'_i$

And finally $GK = a_i \oplus K_i$.

Key Selection. There are two stages in key selection. In first stage the key pool (KP) is selected from a set of strong primes N, in second phase individual node keys are chosen from this key pool. The key pool size depends on size of the network. Initially the key pool is selected randomly and then refreshed regularly. New keys are selected from N using following formula:

$$K_i\,(new) = N[K_i\,(old)*F \bmod |N|]$$

Where F is no. of refresh and |N| is size of set N. These key pools (KP) are stored in a 2D array of size nxn. Individual keys K_i for nodes are selected from the key pool based on following formula:

$$K_i = KP[q][r],$$

Where q = (A*22 + C) mod n and r = (B*22 + D) mod n and A, B, C and D are each decimal representation of 8 bit parts taken from four equal division of 32 bit ID.

4 Experimental Setup

As explained in the architecture section, there are four different setups considered for this experiment i.e. sensor networks having 128, 256, 512 and 1024 total nodes (CH+SN). The number of nodes in each cluster is taken to be 8 and 12 for each case. Different specifications for these nodes are used as simulation parameter that are listed in Table 2.

Table 2. Generic simulation parameters

Parameter name	Value
Number of sensor nodes	128/256/512/1024
Max nodes in cluster	8 /12
Key pool	20X20/30X30/40X40/60X60 with max key size 16 bit
Area size (A)	200 m x 200 m (for 128 nodes and accordingly)
Radio range in open air	200 m
Bandwidth	20kbps
Max Packet size	512 bits
Initial battery capacity	200 J (for Sensor Node), 4000J (for CH)
Min Simulation time	600 sec

5 Results and Analysis

The criteria for evaluating key management schemes include processing complexity (T_p), communication complexity (T_c), storage complexity (T_s), resilience, rekeying cost and scalability. Different results and their analysis w.r.t. these evaluation criteria are as follows:

Computation Cost (T_p). The time taken to calculate the CRT congruence is close to 3.2 and 5 μsec (Fig. 2) for clusters of size 8 and 12 respectively. The theoretical value of time consumed are 3 and 4.8 μsec respectively. These theoretical values are approximately same as obtained in the experiment.

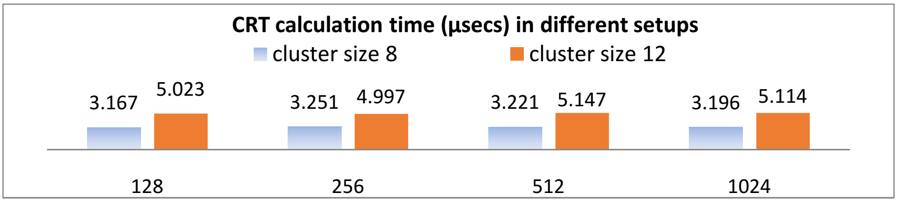

Fig. 2. Number of nodes Vs. CRT calculation time (μsec)

If the energy (battery power) consumed in CRT computation is considered, it is approximately 1.26and 2.02μJ (Fig. 3) for clusters of size 8 and 12 respectively. This is slightly greater than theoretical values which are 1.15 and 1.82 μJ respectively.

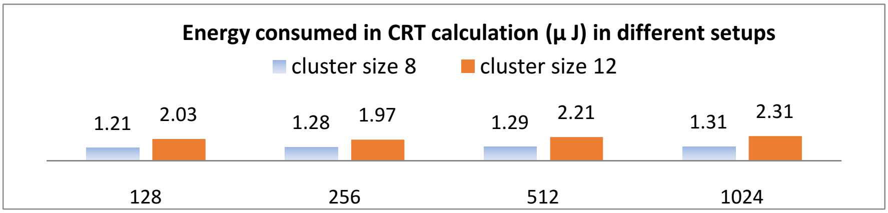

Fig. 3. Number of nodes Vs. Energy consumed in CRT calculation (μJ)

Communication Cost (T_c). In this scheme to establish group key the CH broadcasts the cluster key *X* (i.e. one transmit) and sensor nodes receive the cluster key.

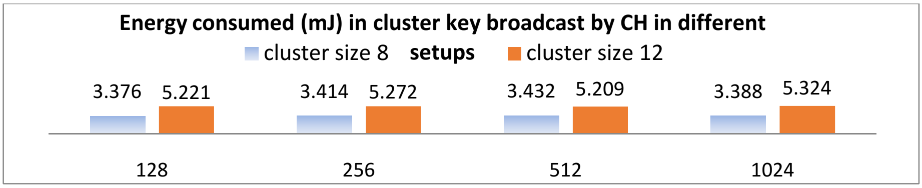

Fig. 4. Number of nodes Vs. Energy consumed (mJ) in cluster key broadcast by CH

The results from Fig. 4 indicate that the energy consumed at CHs are approximately 3.4mJ and 5.2mJ respectively for clusters of size 8 and 12. This is very close to theoretical values which are 3.2 and 5.0mJ. Similarly from Fig. 5, the energy consumed per node for cluster key receive is 58 and 90 μJ for clusters of size 8 and 12 respectively.

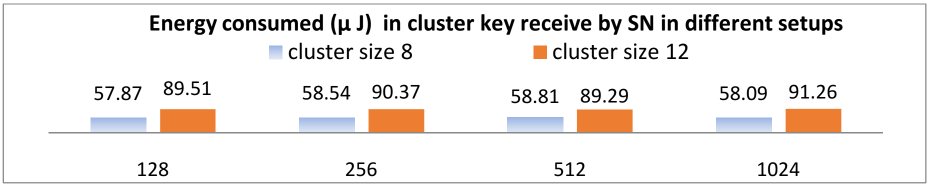

Fig. 5. Number of nodes Vs. Energy consumed (μJ) in cluster key receive by SN

Here energy consumed in communication is significantly higher than the energy consumed in computation (see prev. section); this also supports the fact that this scheme is computationally efficient.

Scalability. For the purpose of evaluating scalability and consistence of the network, simulation is carried for specified number of times (min 600 secs) and energy consumed per operation (i.e. per node addition/deletion) is noted.

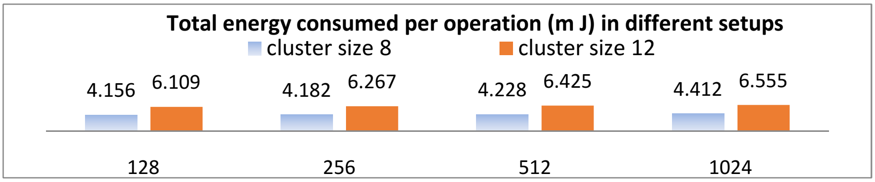

Fig. 6. Number of nodes Vs. Total energy consumed per operation (mJ)

By observing this energy consumed data (Fig. 6), we can made a conclusion that the energy consumed per operation is independent of size of the network and the energy consumed depends on cluster size and increases linearly with change in cluster size.

Rekeying Cost. In the above experimental setup total number of message exchange per operation (including broadcast and unicast messages) is also measured. This gives the rekey cost calculation.

Security Analysis. Here we discuss resilience of proposed algorithm to different attacks. The performance of the network in case of node removal/addition is already discussed in previous section. Other attacks and their effects are further discussed.

Brute Force Attack. This algorithm is designed to make brute force attacks very difficult. Suppose key pool size is P and cluster size is C then probability of compromise of a group key is C/P. In this setup maximum key pool size is 3600 and cluster size is 12, so the probability of a key compromise using brute force is 0.0033.

Node Capture Attack. If an adversary is able to compromise a node, the keying information is revoked from that node, and whole congruence is recalculated excluding that node to establish a new group key.

Collusion Attack. This scheme is full collusion resistant i.e. if an adversary is able to compromise k nodes he can't establish a GK with other nodes or get keying information of an uncompromised node.

Forward Secrecy. This scheme provides forward secrecy as the group key GK is chosen by CH at random and it has no relation with older keys. If size of key space out of which the GK is chosen is n and a perfect random number generator is used then probability of key reuse at next renewal is $\frac{1}{n}$.

Backward Secrecy. In this scheme if an adversary is able to get information of too many revoked nodes, he may be able to find a pattern and guess a future key. To avoid attack against backward secrecy, we refresh the key pool at regular intervals.

6 Conclusion and Scope for Future Work

This paper discusses a new key management technique for Hierarchical Sensor Network which is based on Chinese Remainder Theorem. This scheme provides key establishment in a cluster like environment with minimal computation, storage and communication cost. Experimental result also suggests that it is highly scalable and consistent. The resilience to different attacks was also analyzed and it can be concluded that it is protected from most common attacks that may happen in clustered architecture of HSN. Future work may include combining the CRT based scheme with distributed architecture in a hybrid scheme.

References

1. Eschenauer, L., Gligor, V.: A Key-Management Scheme for Distributed Sensor Networks. In: Proc. of ACM CCS (2002)
2. Chan, H., Perrig, A., Song, D.: Random Key Pre distribution Schemes for Sensor Networks. In: IEEE Symposium on Research in Security and Privacy (2003)
3. Liu, D., Ning, P.: Establishing Pairwise Keys in Distributed Sensor Networks. In: 10th ACM CCS, Washington D.C (2003)
4. Blom, R.: An optimal class of symmetric key generation systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)
5. Du, W., Deng, J., Han, Y., Chen, S., Varshney, P.: A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In: IEEE Infocom (2004)
6. Akyildiz, F., Su, W., Sankarasubramaniam, Y., Cyirci, E.: Wireless Sensor Networks: A Survey. Computer Networks 38(4), 393–422 (2002)
7. Bhaskar, P.K., Sahoo, S.: A Novel Key Establishment Scheme for Hierarchical Sensor Network based on Chinese Remainder Theorem. In: National Workshop on Cryptology, SITE, VIT University (2012)