Confidential Data Aggregation in Wireless Sensor Networks Using Coding Theory

Kanusu Srinivasa Rao
Dept.of Computer Science
Yogi Vemana University
Andhra Pradesh, India.
kanususrinivas@yahoo.co.in

Ratnakumari Challa
Dept. of CSE
APIIIT (RGUKT)
Andhra Pradesh, India
Ratnamala3784@gmail.com

***Abstract*—** Wireless sensor networks are recently receiving substantial attention due to their unlimited potency. The data aggregation scheme provides better security as cluster head perform aggregation on cipher text directly without decryption, accordingly transmission overhead is reduced. We propose the aggregation scheme based on coding theory. McEliece public key encryption based on coding is providing the best alternate for cryptosystem. They leverage error correcting codes as a mechanism for encryption. Different from RSA and ELGAMAL, quantum computer cannot break the McEliece public key cryptosystem and here encryption and decryption operations are more efficient and even secure against chosen cipher text attacks**.**

Keywords- *wireless sensor networks, public key cryptosystem, cluster head*

*****

I. INTRODUCTION

Wireless sensor networks (WSNs) are ad-hoc networks having tiny devices. They comprise of large number of nodes. The ad-hoc feature of network represents that sensors are used in the network construction, they involve in sending their data and also receiving data from other sensors.

Aggregation in WSN

Aggregation techniques help in battery power saving and used to reduce the amount of data that is transmitted between them, like average and/or variance of the temperature or humidity within an area. The general way that is used is to send data to special node called aggregators. Data is then distilled by aggregators before sending to nodes, by performing arithmetic operations to data. Additive aggregation is the process all the values of sensors are summed up which it received from all children. Hop-by-hop basis encryption can be done when sensors are trusted. The Three types of nodes base station, aggregator and leaf node perform their functions to support aggregation. The data is gathered from different nodes and aggregate operations are performed like MAX, MIN, SUM, AVERAGE, HISTOGRAM, etc. These aggregation operations are performed by cluster head, the divided network grouped into clusters. This aggregation operation reduces communication overhead and increases lifetime of network. An aggregator can openly execute addition operations on encrypted binary data. Addition in homomorphic public-key encryption used in a aggregation scheme is proposed [1]. But here authenticity of data is not founded. It requires protection of data from neighbours why because they know aggregated data and key therefore it is a challenging task

II.RELATED WORK

Privacy homomorphic property of encryption algorithm EPSDA [2] it solves the problem by performing aggregation on encrypted data. By using this protocol we can prevent replay attack and accuracy of data can be achieved by reducing number of transmissions. However this scheme increases overhead as it recovers sensing data other than aggregated result.

The scheme based on a commit and attest paradigm is introduced by Yang et al. [1].the commit phase handles by having a aggregator head at cluster, while measurements are committed by nodes. To detect and prevent cheating, secure aggregation tree (SAT)[3] is proposed, Aggregation tree handles cheating by using topological constraints. The paper proposes the symmetric homomorphic scheme texts through its additive property, thus achieving data confidentiality

III. MATERIALS AND METHODS

McEliece cryptosystem and its security:

Let G be a k × n generator matrix of a code C, for which there is an efficient algorithm Dec C that can decode any codeword with up to t errors. Let S be a random non-singular matrix of order k × k , and let P be a random permutation matrix of order n× n.

(Generalized) McEliece cryptosystem (MECS) is defined as follows:

Secret key: (Dec C, S, P)

Public key: G_ = S ・ G ・ P

Encryption: Let m be a k-bit message, and let e be a random n-bit vector with wH (e) ≤ t. Then c = m ・ G_ + e is a cipher text.

Decryption: Decryption is given by the following algorithm:

1: c_ ← c ・ P−1

2: m_ ← Dec C (c_)

3: m ← m_ ・ S−1

Public key of MECS is a generator matrix G_ of some linear code C_, which is permutation equivalent to C. MECS hides a message by transmitting a codeword of C_ given by mG_ = m ・S・G・P, further masked by an artificial error vector e. The weight of the error vector is not changed under codeword permutation.

Thus the legitimate recipient can reverse the secret permutation P and decode the resulting codeword of C. Finally, he can recover the original message by inverting the linear transform given by S. The schemes which provides such kind of security would be interesting for at least the following two reasons.

1) Quantum computers bring the need of having this kind of assumption that is Alternative Security Assumption.
2) Efficient Decryption Operation.

IV. THE IDEA OF SCHEME

Here we give detailed description of the scheme we are going to use. Main theme of error correcting code is to transfer message protectively over noisy channel. To survive this purpose, a code C uses two algorithms, Encode and Decode. Firstly, Encode technique maps a message m to Fk into a codeword w to Fn. Here, F is a finite field and k < n. After transmitting w, some of the positions get neutered and gets code word w0 2 Fn. However number of bad positions that is altered positions does not exceed given limit then we can decode it and recover it as m, and those unaltered positions are called as good positions. The general approach we follow every time is to encode plaintext m 2 Fk which is encoded to codeword w2c and then we add errors to w so that we get cipher text c. Here we assume that secret key helps to identify bad which makes decoding task easy. Here we follow a principle which is a extension of scheme proposed by Kiayias and Yung [9-11]. Encryption consists of two steps:

(i) Plaintext m 2 F is encoded into an error-free codeword w.
(ii) At some fixed positions some errors are entered which are provided by secret key to get an erroneous codeword c.
(iii) At this end, we make use of the fact that for linear codes, the sum (being the vector sum) of two codeword's yields a codeword again.

Here the encoding process involves like adding two code words is same as adding two messages prior to encoding. This altogether allows for tapping the additive structure of linear codes [12] in a natural way to achieve encryption.

V. DISCUSSION

Suppose we choose n=1024=2^10, t=0 then there will be about 10^149 possible goppa polynomials and an astronomical number of choices for S and P. The dimension of the code will be about k=1024.50.10=524.Hence, a brute force approach to decoding based on comparing x to each codeword has a work factor of about 2^524=10^158: and a brute force approach based on cosset leaders has a work factor of about 2^500=10^151.A most assuring attack is to select k of the n coordinates arbitrarily in trust that no values of k are in error and u is calculated based on this. The probability of no error, however, is about

$$\left(1-\frac{t}{n}\right)^{k}$$

and work involved in solving the k simultaneous equations in k unknown in about k^3. Hence, before finding u using this attack the expected work factor of

$$k^{3}.(1-\frac{t}{n})^{-k}$$

For n=1024, k=524, t=50 this is about 10^19 approximately equal to 2^65. This analysis helps to suggest that public key system is quite secure.

VI. CONCLUSION

Reducing energy consumption is a compulsory objective in the design of any communication protocol for Wireless Sensor Networks. Most of this energy can be saved through data aggregation, given that most of the sensed information is redundant due to geographically collocated sensors. However, a second compulsory design objective of any communication Protocol for WSNs is security. Unfortunately, while aggregation eliminates redundancy and hence saves energy, it makes data integrity verification more complicated. The proposed approach uses homomorphic encryption where to achieve data confidentiality while allowing in-network aggregation. We have used McEliece public key cryptosystem to provide integrity of the aggregate. The performance evaluation shows that the proposed scheme is feasible for large WSNs where additive homomorphism is fulfilled. The future research directions are drawn to support multiplicative homomorphism to complete aggregation operation.

VII. REFERENCES

[1]. E. Mykletun, J. Girao, and D. Westhoff, "Public key based Crypto schemes for Data Concealment in Wireless sensor Networks",
[2]. Joyce Jose, M. Princy, and Josna Jose, "EPSDA: Energy Efficient Privacy preserving Secure Data Aggregation for Wireless Sensor Networks", International Journal of Security and Its Applications, vol.7, No. 4, July, 2013 Proc. IEEE Int'l Conf. Comm., vol.5, pp.2288-2295, June 2006
[3]. Y. Yang, Wang, S. Zhu, and G. Cao, "SDAP: A secure hop-by-hop data aggregation protocol for sensor networks", in MobiHoc, 2006
[4]. Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. Accepted to EUROCRYPT'11, 2011.
[5]. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, Public Key Cryptography, volume 6056 of Lecture Notes in Computer Science, pages 420–443. Springer, 2010.
[6]. Dan Boneh and David Mandell Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. Cryptology ePrint Archive, Report 2010/453, 2010. Accepted to PKC'11.
[7]. Dan Boneh and David Mandell Freeman. Homomorphic signatures for polynomial functions. Cryptology ePrint Archive, Report 2011/018, 2011. Accepted to EUROCRYPT
[8]. Damien Stehl´e and Ron Steinfeld. Faster fully homomorphic encryption. In Masayuki Abe,editor, ASIACRYPT, volume 6477 of Lecture Notes in Computer Science, pages 377–394. Springer, 2010.
[9]. A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes with napplications. Electronic Colloquium on Computational Complexity (ECCC), 2002.
[10]. A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes. Cryptology ePrint Archive, Report 2007/153, 2007.
[11]. A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of reed-solomon codes. IEEE Transactions on Information Theory, 54(6):2752–2769, 2008.
[12]. ZHAO Cheng-cheng, YANG Ya-tao LI Zi-chen. The Homomorphic Properties of McEliece Public-key Cryptosystem, 2012 Fourth International Conference on Multimedia Information Networking and Security