DOI: http://dx.doi.org/10.26483/ijarcs.v9i5.6304

ISSN No. 0976-5697

Volume 9, No. 5, September-October 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# DATA AGGREGATION TECHNIQUES AND SECURITY IN DISTRIBUTED SENSOR NETWORKS

Dr. A .Banumathi,
Department Of Computer Science,
Government Arts College Karur. India

M. Durgadevi,
Department Of Computer Science,
Government Arts College Karur. India

***Abstract:*** Data aggregation is attracting much attention from researchers as efficient way to reduce the huge volume of data generated in wireless sensor networks by eliminating the redundancy among sensing data. Existing system integrated an efficient data aggregation technique for clustering-based periodic wireless sensor networks Further to a local aggregation at sensor node level, our technique allows cluster-head to eliminate redundant data sets generated by neighboring nodes by applying three data aggregation methods such as Vector similarity function, Jaccard function, Euclidean and Cosine distance (K-Mean) functions using to analyze the sensor data performances according to the energy consumption, data latency and accuracy. It's not providing the security for the data we introduce the security to data. To rectify these problems, this work focuses on efficient CDAMA protocol to obtain the additive encryption model and a novel key management technique to support large plaintext space. The paper also extends the aggregation protocol to obtain the secure aggregate of time-series data. It shows that the proposed protocols are faster than existing solutions, and it has much security communication overhead. In addition, proposes system a new Concealed Data Aggregation Scheme (CDAMA) which is homomorphism public encryption system based multi-application environment, extracts application-specific data from Aggregated Encrypted Ciphertexts and degrades the damage from unauthorized aggregations process.

***Keywords***– WSN, Data Aggregation, Similarity Function, KNN, PFF, CDAMA, AEC

## I.INTRODUCTION

Wireless Sensor Networks (WSNs) are rapidly emerging as an important new area in wireless and mobile computing research. Applications of WSNs are numerous and growing, and range from indoor deployment scenarios in the home and office to outdoor deployment scenarios in adversary's territory in a tactical battleground. For military environment, dispersal of WSNs into an adversary's territory enables the detection and tracking of enemy soldiers and vehicles. For home/office environments, indoor sensor networks offer the ability to monitor the health of the elderly and to detect intruders via a wireless home security system.

Sensor networks composed of small and cost effective sensing devices equipped with wireless radio transceiver for environment monitoring have become feasible. The key advantage of using these small devices to monitor the environment is that it does not require infrastructure such as electric mains for power supply and wired lines for Internet connections to collect data, nor need human interaction while deploying. These sensor nodes can monitor the environment by collecting information from their surroundings, and work cooperatively to send the data to a base station, or sink, for analysis. The main goal of data aggregation algorithms is to gather and aggregate data in an energy efficient manner so that network lifetime is enhanced. Wireless sensor networks (WSN) offer an increasingly attractive method of data gathering in distributed system architectures and dynamic access via wireless connectivity.

Wireless Sensor networks are deployed to sense, monitor, and report events of interest in a wide range of applications. The security monitoring networks consist of energy constrained nodes that are expected to operate over an extended period of time, making energy efficient monitoring an important feature for unattended networks. When wireless sensor networks are deployed in untrustworthy environments, protecting the privacy of the three parameters that can be attributed to an event-triggered transmission becomes an important security feature in the design of wireless sensor networks. While transmitting the "description" of a sensed event in a private manner can be achieved information of reported events cannot be achieved cryptographic means. Encrypting a message before transmission, for instance, can hide the context of the message from unauthorized observers, but the mere existence of the ciphertext is indicative of information transmission. The source anonymity problem in wireless sensor networks is the problem of studying techniques that provide time and location privacy for events reported by wireless sensor nodes.

In system model are designed to transmit information only when a relevant event is detected. The following in **Fig. 1.**1, describe the locations of the combat vehicle at different time intervals can be revealed to an adversary observing nodes transmissions. There are three parameters that can be associated with an event detected and reported by a sensor node: the description of the event, the time of the event, and the location of the event.

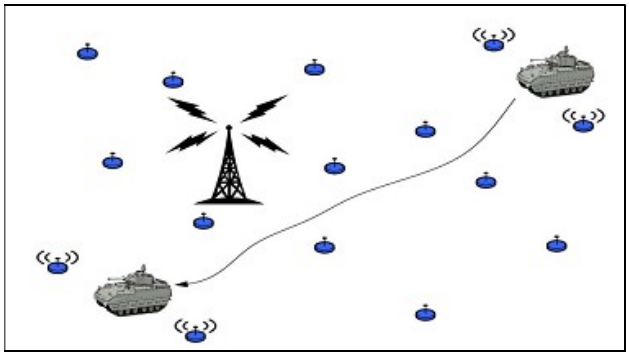


Figure 1.1Sensor network deployed in a battlefield

Data aggregation is a process of aggregating the sensor data using aggregation approaches. The general data aggregation algorithm works as shown in the below figure. The algorithm uses the sensor data from the sensor node and then aggregates the data by using some aggregation algorithms such as centralized approach, LEACH(low energy adaptive clustering hierarchy),TAG (Tiny Aggregation) etc. This aggregated data is transfer to the sink node by selecting the efficient path. There are many types of aggregation techniques are present some of them are listed below.

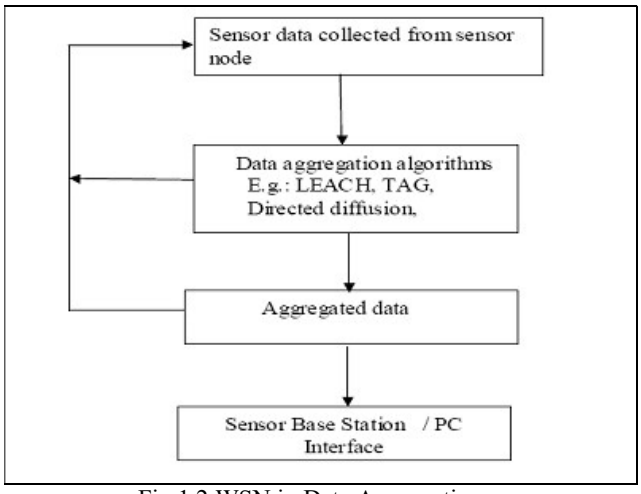



Fig 1.2 WSN in Data Aggregation

The figure explain about the overall data aggregation process. The **Fig 1.2** describes a Data aggregation for Wireless sensor networks. In this fig first phase sensor data collected from sensor node using LEACH, TAG protocol. The next phase describe a aggregate data from cluster head (CH) and send the aggregate data into base station.

## II. LITERATURE REVIEW

In this paper to attack the "SELECT *" problem for sensor networks [1]. To a robust approximate technique called Ken that uses replicated dynamic probabilistic models to minimize communication from sensor nodes to the network's PC base station. In addition to data aggregate, K-Mean is well suited to anomaly and event-detection applications. A key challenge in this work is to intelligently exploit spatial correlations across sensor nodes without imposing undue sensor-to-sensor communication burdens to maintain the models. Using traces from two real-world sensor network deployments, to demonstrate that relatively simple models can provide significant communication (and hence energy) savings without undue sacrifice in result quality or frequency. Choosing optimally among even our simple models is NP hard, but our experiments show that a greedy heuristic performs nearly as well as an exhaustive algorithm.

In this paper [3] declarative queries are proving to be an attractive paradigm for interacting with networks of wireless sensors [2]. The metaphor that "the sensornet is a database" is problematic, however, because sensors do not exhaustively represent the data in the real world. In order to map the raw sensor readings onto physical reality, a model of that reality is required to complement the readings. To enrich interactive sensor querying with statistical K-mean modeling techniques has been used.

In this paper [4] event-driven sensor networks operate under an idle or light load and then suddenly become active in response to a detected or monitored event. The transport of event impulses is likely to lead to varying degrees of congestion in the network depending on the sensing application. It is during these periods of event impulses that the likelihood of congestion is greatest and the information in transit of most importance to users. To address this challenge an energy efficient congestion control scheme for sensor networks called CODA (COngestion Detection and Avoidance) that comprises three mechanisms [4]:

- K-mean-receiver-based congestion detection;
- open-loop hop-by-hop back pressure; and
- Closed-loop multi-source regulation.

In this paper [4] describe a wireless sensor networks and created new opportunities for data aggregate in a variety of scenarios, such as environmental and industrial, where they expect data to be temporally and spatially correlated. Researchers may want to continuously collect all sensor data from the network for later analysis. Suppression, both temporal and spatial, provides opportunities for reducing the energy cost of sensor data aggregate. To demonstrate how both types can be combined for maximal benefit. They frame the problem as one of monitoring node and edge constraints. A monitored node triggers a report if its value changes [4].

In this paper, [5] they introduce a best-effort synchronization scheduling policy that exploits cooperation between data sources and the cache. And also propose an implementation of proposed policy that incurs low communication overhead even in environments with very large numbers of sources. Their algorithm is adaptive to wide fluctuations in available resources and data update rates. Through experimental simulation over synthetic and real-world data, demonstrate the effectiveness of data aggregate algorithm, and we quantify the significant decrease in divergence achievable with source cooperation.

## III. PROBLEM FORMULATION

Aggregation statistics need to be periodically computed from a stream of data contributed by mobile users [1], to identify some phenomena or track some important patterns in many scenarios. For example, the average amounts of daily exercise (which can be measured by motion sensors [2]) that people do can be used to infer public health conditions.

The average or maximum level of air pollution and pollen concentration in an area may be useful for people to plan their outdoor activities. Other statistics of interests include the lowest gasoline price in a city, the highest moving speed of road traffic during rush hour, and so on.

Although aggregation statistics computed from time-series data are very useful, in many scenarios, the data from users are privacy-sensitive, and users do not trust any single third-party aggregator to see their data values. For instance, to monitor the propagation of a new the aggregator will count the number of users infected by this flu. However, a user may not want to directly provide her true status ("1" if being infected and "0" otherwise) if she is not sure whether the information will be abused by the aggregator. Accordingly, systems that collect users' true data values and compute aggregate statistics over them may not meet users' privacy requirement [1]. Thus, an important challenge is how to protect the users' privacy in mobile sensing, especially when the aggregator is untrusted.

Most previous works on sensor data aggregation assume a trusted aggregator, and hence cannot protect user privacy against an untrusted aggregator in mobile sensing applications. Several recent works [3] consider the aggregation of time-series data in the presence of an untrusted aggregator. To protect user privacy, they design encryption schemes in which the aggregator can only decrypt the sum of all users' data but nothing else.

Its computation overhead is too high for an aggregator to run real-time monitoring applications with short aggregation intervals and to collect multiple aggregate statistics simultaneously. Moreover, none of these existing schemes considers the Min aggregate (i.e., the minimum value) of time-series data, which is also important in many mobile sensing applications. This thesis proposes a new protocol for mobile sensing to obtain the sum aggregate of time-series data in the presence of an untrusted aggregator. The protocol employs an additive homomorphic encryption and a novel key management scheme to ensure that the aggregator can only obtain the sum of all users' data, without knowing individual user's data or intermediate result.

## IV. TECHNIQUES

The wireless sensor network, data aggregation scheme that reduces a large amount of transmission is the most sensor technique. In proposed algorithm, encryptions have been applied to conceal communication during aggregation such that enciphered data can be aggregated algebraically without decryption. Since aggregators collect data without decryption, adversaries are not able to forge aggregated results by compromising them. The one aggregator send the data to another aggregator with encryption process, the another aggregator receive the data with decryption process they can only get the original aggregated data, otherwise could not receive the original data.

However, these schemes are satisfy

- Multi-application environments.
- CDAMA schemes become secure in case sensor nodes are compromised.
- CDAMA, schemes do will be providing secure counting
- CDAMA do not may suffer unauthorized aggregation attacks.

In addition propose a new Concealed Data Aggregation Scheme extended (CDAMA) from public group encryption system. The proposed scheme proves the robustness, efficiency and also conducted the comprehensive analyses and comparisons in the end.

- CDAMA Scheme is applied in network environment.
- CDAMA Scheme is applied in database service environment.
- More security is applied for client data.
- It can be applied both in wireless sensor network environment and cloud data environment.
- Group of nodes can communicate with a single node in secure manner.

**A) Jaccard Similarity Function**

In this module, the aggregation process uses the similarity functions at CH (Cluster Head) level, such as the Jaccard function, to search the similarities between data sets. The Jaccard similarity function returns a value in [0; 1] where a higher value indicates that the sets share more similarities. Thus we can treat pairs of sets with high Jaccard similarity value as near duplicate to reduce the size of final data sets transmitted from the CH to the sink. A Jaccard similarity functions between two sets Mi' and Mj', generated respectively by the sensors Si and Sj. Definition: (Similar function): Define the Similar function between two measurements captured by the same sensor node Si at a period p as:

Where mij and mik □ Mi and ε is a threshold determined by the application. Furthermore, two measures are similar if and only if their similar function is equal to 1.

$$\text{Similar}(m_{ii}, m_{ik}) = {}_1\{^{0} \quad \text{if } \| m_{ii} - m_{ik} \| \leq \varepsilon, \text{ otherwise}$$

**B) Cosine Distance**

Cosine distance is a measure of dissimilarity between two vectors that measures the cosine of the angle between them. This kind of dissimilarity has been used widely in many aspects, such as the anomaly detection in web documents and medical diagnosis [45]. Depending on the angle between the vectors, the resulting dissimilarity ranges from −1meaningexactly the opposite, to1meaning exactly the same. The Cosine distance (Cd) between two sets MiandMj,before applying local aggregation, is given by Thus, Mi and Mj are redundant if Cd (Mi,Mj) ≤td. Then, to adapt the Cosine distance to the measures' weights in M0iandM0jas follows.

$$Cd(Mi,Mj) = 1- P\tau k=1(mik \times mjk) /$$

**C) Prefix Frequency Filtering (PFF)**

The PFF technique is used to find similar sets. In order to prevent CH from enumerating and comparing every pair of sets which has a O(n2) number of comparisons, the prefix frequency filtering (PFF) technique is used in this module. In the candidate pairs' generation step, the CH searches the candidates (which may or may not be similar) sets for every data set. This step is based on the intuition that if all sets of measures are sorted by a global ordering, some fragments of them must share several common tokens with each other in order to meet the Jaccard threshold similarity ($t_J$). A Jaccard similarity function between two sets M'i and M'j, generated respectively by the sensors Si and Sj , can be formulated as follows:

$$J(M'i,M'j) \geq tJ \leftrightarrow |M0i \cap sM0j| \geq \alpha = 2 \times tJ \times \tau / 1 + tJ$$

Where $t_{jis}$ the Jaccard threshold defined by the application itself and "$\cap_s$" is defined as:

Definition: Consider two sets of measurements M'i and M'j , then we define the overlap, ∩s, between them as: M'i ∩s M'j = { ( mi, m'j ) < I × M 0 j with weight wgt min ( m 0 I ,m 0 j ) such that Similar (m'i ,m' j ) = 1} ; where wgtmin (m'i,m'j) =min (wgt(m'i),wgt(m'j)), the minimum value of the weight of m'i and m'j.

**D).K-Means Adapted To Variance**

Adopted variance between measurements in the data sets is another way of finding nodes that generate redundant data sets and it is implemented based on the k-means algorithm adopted by the Anova model. This model is used to identify if the variance (R) between measures in a group of data sets is significant or not. R can be calculated in different manners depending on the statistic tests proposed in the Anova model. This process is utilized here in order to detect all pairs of nodes with identical behavior which generate redundant data logs or sets.

**F) CDAMA Scheme**

One building block of the solution is the additive homomorphic encryption scheme. This scheme works as follows:

**Encryption:**

- Represent message m as an integer within range [0, M-1], where M is a large integer.
- Let k be a randomly generated key.
- Output ciphertext c ( m + h($f_k$r))) mod M, where $f_k$ is a pseudorandom function (PRF) that uses k as a parameter, h is a hash function and r is the sample value for the message.

**Decryption:**

Output plaintext m = (c - h($f_k$r))) mod M.

Secret distribution: The key dealer generates nc random and different secrets s1; ... ;Snc. Let S denote the set composed of all the secrets. The key dealer divides these secrets into n random disjoint subsets, with c secrets in each subset. For convenience, we call these subsets additive subsets. Let Si denote the ith additive subset. Clearly. S= ∑ ni=1 Si.

Out of the nc secrets, the key dealer randomly selects q secrets and assigns them to the aggregator. Let ^S denote the set of secrets assigned to the aggregator. The key dealer divides the remaining nc - q secrets evenly into n random disjoint subsets. The key □□dealer assigns the secrets in the additive subset Si and subtractive subset Si to user i.

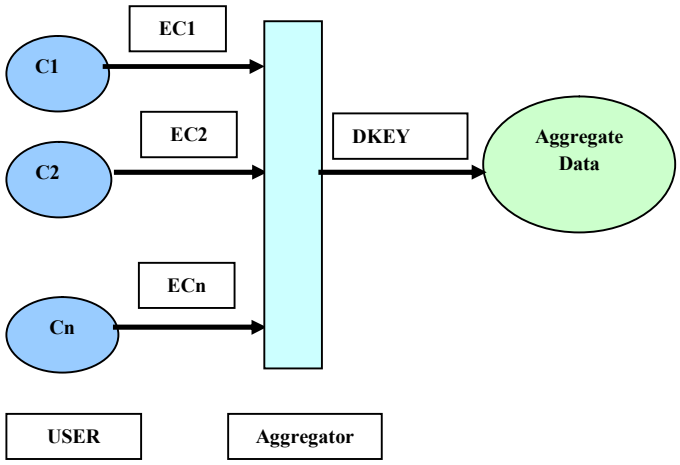
Encryption key generation. In time period t > IN, user I generates its encryption key by computing

$$K_i = (\Sigma_{s' \in Si} h(fs'(t)) - \Sigma h(fs'(t))) \bmod M \quad (4)$$

Decryption key generation. In time period t < IN, the aggregator generates the decryption key by computing

$$k_0 = ( \Sigma h(fs'(t))) \bmod M \quad (5)$$

The Min aggregate is defined as the minimum value of the users' data. This section presents a protocol that employs the Sum aggregate to get Min.




Setup. The key dealer assigns a set of secret values (secrets for short) to each user and the aggregator.

Enc. In each time period, user i generates encryption key ki using the secrets that it is assigned. It encrypts its data xi by computing

$$ci = (ki + xi) \bmod M \quad (1)$$

where $M = 2^{\square \log_2 (n \square \square) \square}$. Then, it sends the ciphertext ci to the aggregator.

*AggrDec.* In each time period, the aggregator generates decryption key k0 using the secrets that it is assigned, and decrypts the sum aggregate

$$S = \sum_{i=1}^{n} xi \text{ by computing}$$

$$S = ( \sum_{i=1}^{n} ci - k0) \bmod M \quad (2)$$

The keys are generated using a PRF family and a length-matching hash function (see later). According to the aggregator can get the correct sum so long as the following equation holds:

$$k_0 = ( \sum_{i=1}^{n} ki) \bmod M \quad (3)$$

In this protocol, the setup phase only runs once. After the setup phase, the key dealer does not need to distribute secrets to the users and the aggregator again. In

addition, the users and the aggregator do not have to synchronize their key generations with communications in every time period. These restrictions make it challenging for the users and the aggregator to generate their keys such that (3) holds in every time period and the encryption (decryption) key used by each user (the aggregator) cannot be learned by any other party besides the key dealer.

To proposed a construction for key generations that preserves the privacy of each user and the Sum aggregate efficiently. Before presenting this construction, it first discusses a straw-man construction which is very efficient for the users but not efficient for the aggregator. Then, it extends to straw-man scheme to derive this construction. Both constructions include three processes, which are secret setup, encryption key generation, and decryption key generation. They proceed in the Setup phase, Enc phase, and Aggraded phase of the aggregation protocol, respectively.

CDAMA is constructed on a cyclic group of elliptic curve points. Precisely, these points form an algebraic group, where the identity element of the group is the infinite point, ∞. Notation ord(P ) denotes the order of a point P. Supposing ord(P ) = q, it indicates that q is the minimum integer that satisfies q * P = ∞. In the KEYGEN function, the order of E is equivalent to the number of points in E. Instead of relying on a trusted key dealer, our protocol can be easily adapted to work with an honest-but-curious key dealer that does not collude with the aggregator. An honest-but-curious key dealer correctly follows the protocol steps, but wants to get users' data values from the transcript of messages in the protocol.

To provide privacy under this model, the protocol adds one more encryption and decryption to the data that each user submits to the aggregator. More specifically, each user encrypts its data using the secrets assigned by the key dealer to derive an intermediate result z, encrypts z with a key pre-shared with the aggregator, and then sends the obtained ciphertext to the aggregator. The aggregator first uses the pre-shared key to decrypt each user's intermediate result z, and then decrypts the noisy sum with the secrets received from the key dealer. The key dealer cannot obtain the intermediate result z of any user, as long as it does not collude with the aggregator. Hence, it cannot get any user's data value.

The honest-but-curious model is realistic because it can be enforced with trusted hardware. In practice, certificate authorities such as VeriSign (which already provides key management services) may serve as the key dealer. Since these authorities usually undergo extensive audits, collusion with the aggregator can be mitigated. More aggregate statistics. In the basic aggregation scheme for Min presented above, the aggregator can actually get the number of times that each possible data value appears, and derive the accurate distribution of the users' data in the plaintext space

Differential privacy for Sum: Differential privacy provides strong privacy guarantee for users such that a user's participation in the system only leaks negligible information about the user. Our protocol for Sum can be adapted to provide computational differential privacy.

CDAMA is constructed on a cyclic group of elliptic curve points. Precisely, these points form an algebraic group, where the identity element of the group is the infinite point, ∞. Notation ord (P) denotes the order of a point P. Supposing ord (P) = q, it indicates that q is the minimum integer that satisfies q * P = ∞. In the KEYGEN function, the order of E is equivalent to the number of points in E.

**KEYGEN**($\tau$): generate public-private key pairs for group $G_A$ and $G_B$
1. Based on security parameter $\tau$, compute $(q_1, q_2, q_3, E)$, where
   $E$ is the set of elliptic curve points which form a cyclic group;
   ord($E$) = $n$, and $n = q_1 q_2 q_3$; $q_1$, $q_2$, $q_3$ are large primes;
   the bit lengths of $q_1$, $q_2$, and $q_3$ are the same, i.e., $|q_1| = |q_2| = |q_3|$.
2. Randomly pick up three generators, $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ such that ord($\mathcal{G}_1$) = ord($\mathcal{G}_2$) = ord($\mathcal{G}_3$) = $n$.
3. Compute point $\mathcal{H} = q_1 q_2 * \mathcal{G}_3$ ; ord($\mathcal{H}$) = $q_3$.
4. Select parameter $T$ as the maximum plaintext boundary where Pollard's $\lambda$ method is feasible;
   then compute $T_A = T_B = \lfloor \frac{T}{x} \rfloor$ where $x$ is the number of sensors in an application.
5. Compute $\mathcal{P} = q_2 q_3 * \mathcal{G}_1$, ord($\mathcal{P}$) = $q_1$;
   then output $G_A$'s group public key $PK_A$: $PK_A = (n, E, \mathcal{P}, \mathcal{H}, T_A)$.
6. Compute $\mathcal{Q} = q_1 q_3 * \mathcal{G}_2$ , ord($\mathcal{Q}$) = $q_2$;
   then output $G_B$'s group public key $PK_B$: $PK_B = (n, E, \mathcal{Q}, \mathcal{H}, T_B)$.
7. Output $G_A$'s group Private key $SK_A$ as $(q_2 q_3)$, and $G_B$'s group Private key $SK_B$ as $(q_1 q_3)$.

**ENC**($PK_A$, $M$): Message encryption in $G_A$
1. Check if message $M \in \{0, \cdots, T_A\}$.
2. Randomly select $R \in \{0, \cdots, n-1\}$.
3. Generate the resulting ciphertext $C$ as: $C = M * \mathcal{P} + R * \mathcal{H}$.
4. Return $C$.

**ENC**($PK_B$, $M$): Message encryption in $G_B$
1. Check if message $M \in \{0, \cdots, T_B\}$.
2. Randomly select $R \in \{0, \cdots, n-1\}$.
3. Generate the resulting ciphertext $C$ as: $C = M * \mathcal{Q} + R * \mathcal{H}$.
4. Return $C$.

**AGG**($C_1, C_2$): Message aggregation on two ciphertexts $C_1$ and $C_2$
1. Compute the aggregated ciphertext
   $C' = C_1 + C_2$; $C' = (\sum M_i) * \mathcal{P} + (\sum M_j) * \mathcal{Q} + (\sum R_i) * \mathcal{H}$, where
   $\sum M_i$ represents the aggregated result of $G_A$,
   $\sum M_j$ represents the aggregated result of $G_B$,
   and $\sum R_i$ represents the aggregated randomness of both groups.
2. Return $C'$.

**DEC**($SK_A$, $C$): Message decryption on $C$ for group $G_A$
1. Compute $M = \sum M_i = \log_{\tilde{\mathcal{P}}}(q_2 q_3 * C)$ where $\tilde{\mathcal{P}} = q_2 q_3 * \mathcal{P}$
2. Return $M$.

**DEC**($SK_B$, $C$): Message decryption on $C$ for group $G_B$
1. Compute $M = \sum M_j = \log_{\tilde{\mathcal{Q}}}(q_1 q_3 * C)$ where $\tilde{\mathcal{Q}} = q_1 q_3 * \mathcal{Q}$.
2. Return $M$.

**SUM PROTOCOL**

Sum protocol, model is 'n', 'c' values are given. The product of n*c is calculated. Then the data prepared by the nodes are given as comma separated values. These become x1, x2, xn. Then key1, key2, key n and key0 are calculated. The cipher values are calculated and then cipher sum is created. These values are given to aggregator node. That node find out the data sum from the given cipher sum.

The max aggregation functionality is shown in the above figure. The 'n', 'c' values are given to the system via the appropriate fields provided in the experimental setup. The product of n*c is calculated. Then the data prepared by the nodes are given as comma separated values. These become x1, x2, … xn. Then key1, key2, keyn and key0 are calculated. The cipher values are calculated and then cipher sum is created. These values are given to aggregator node. That node find out the maximum value from the given cipher sum

**MIN AGGREGATE SCHEME**

This scheme gets the Min aggregate of each time period using parallel Sum aggregates in the same time period. After aggregated the data and delivered to aggregator. The Min aggregate is defined as the minimum value of the users' data.

**MAX AGGREGATE SCHEME**

Instead of relying on a trusted key dealer, our protocol can be easily adapted to work with an honest-but-curious key dealer that does not collude with the aggregator. An honest-but-curious key dealer correctly follows the protocol steps, but wants to get users' data values from the transcript of messages in the protocol.

**SMM PROTOCOL**

**INPUT: Aggregator Node AN, Data Collector Nodes DCN, Collected Data CD, Message Upper Value M (Nodes will collect data from 0 to M value)**

**OUTPUT: MAX VALUE, Aggregator Node AN, Data Collector Nodes DCN, Collected Data CD AggregatedSum**

1. Distribute Key K0 to AN by key Generator.
2. Distribute K1, K2, … , Kn to DCN one by one.
3. CD = Data Collection by DCN.
4. CipherSum = 0
5. For each N in DCN
6. Data = Collected Data of N (in CD)
7. CipherValue = Hash Value(Data) + Kn [Kn=Key of Node n]
8. CipherSum = CipherSum + CipherValue
9. Next
10. Aggregated Cipher Sum propagated to AN
11. Aggregated = 0
12. AggregatedSum = CipherSum – K0
13. return AggregatedSum
14. Prepare a Matrix RM with rows = Nodes Count in DCN and columns = M
15. Set a Vector SumValue[0 to M]
16. For each N in DCN
17. Data= Data of N in CD
18. RM[N,Data]=1 [Assign Data$^{th}$ column of row N to 1]
19. RM[N,Data]=0 [Assign all other column of row N to 0]
20. Next
21. For I = 0 to DCN
22. For J = 0 to M
23. SumValue[J] = SumValue[J] + RM[I,J]
24. Next Next
25. Propagate SumValue to AN
26. BigIndex = 0 For I = M To 0
27. if (SumValue[I] > 0) BigIndex = I Break end if Next
28. Return BigIndex [This is the minimum value MAX VALUE collected in any of the nodes]

The honest-but-curious model is realistic because it can be enforced with trusted hardware. In practice, certificate authorities such as VeriSign (which already provides key management services) may serve as the key dealer. Since these authorities usually undergo extensive audits, collusion with the aggregator can be mitigated. More aggregate statistics. In the basic aggregation scheme for Min presented above, the aggregator can actually get the number of times that each possible data value appears, and derive the accurate distribution of the user's data in the plaintext **Differential privacy for Sum**. Differential privacy provides strong privacy guarantee for users such that a user's participation in the system only leaks negligible information about the user. Our protocol for Sum can be adapted to provide computational differential privacy.

**V. EXPERIMENTAL RESULTS**

The following **Fig 5.1** describes experimental result for Number of data Aggregation J-K-PFF& SMM and CDAMA aggregation data scheme system. The table contains number of cluster group, cluster size and number of aggregated data and average aggregated details are shown.

**Table 5.1 Comparison between Different Aggregate Schemes**

| S.NO | NO.OF CLUSTER | NO.OF AGGREGATION DATA | | J-K-PFF & SMM Scheme (%) | CDAMA Scheme (%) |
|---|---|---|---|---|---|
| | | J- K-PFF & SMM Scheme | CDAMA Scheme | | |
| 1 | GA | 558 | 580 | 69.75 | 72.5 |
| 2 | GB | 574 | 597 | 71.75 | 74.62 |
| 3 | GC | 570 | 578 | 71.25 | 72.25 |
| 4 | GD | 542 | 557 | 67.75 | 69.62 |
| 5 | GE | 566 | 579 | 70.75 | 72.37 |
| 6 | GF | 563 | 569 | 70.375 | 71.12 |
| 7 | GG | 558 | 580 | 69.75 | 72.5 |
| 8 | GH | 574 | 597 | 71.75 | 74.62 |

The table discuss about the details of comparison of different aggregate scheme averages. The existing schemes gives only the minimum aggregated data, so in this process CDAMA provide maximum aggregated data and also giving the security for data. Finally the CDAMA Scheme is better than other schemes.

The following **Fig 5.2** describes experimental result for Average number of aggregate J-K-PFF& SMM and CDAMA aggregation data scheme system. The table contains number of cluster group, cluster size and number of aggregated data and average aggregated details are shown.

For communication cost per application, the communication cost is measured as the size of a ciphertext over the number of applications whose messages can be encrypted in the ciphertext (Table 4.2). As we can see, per application cost of CDAMA is decreased with the value of k. For instance, when there are four applications, the size of ciphertexts in TinyPEDS is 328 * 4 = 1312.

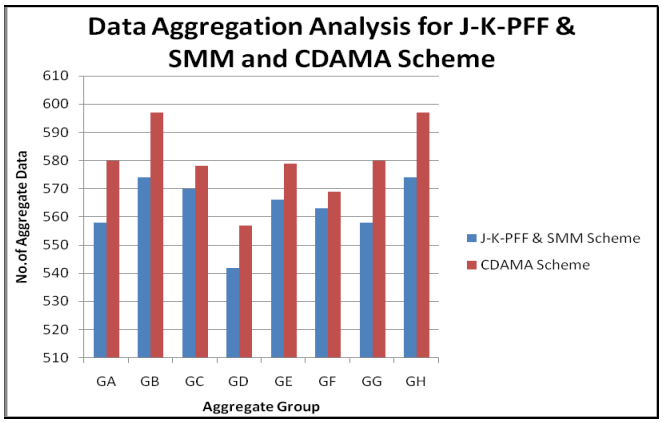



Fig 5.1 Data Aggregate J-K-PFF& SMM and CDAMA Scheme

The following **Fig 6.5** describes experimental result for Average number of aggregate J-K-PFF& SMM and CDAMA aggregation data scheme system. The table contains number of cluster group, cluster size and number of aggregated data and average aggregated details are shown

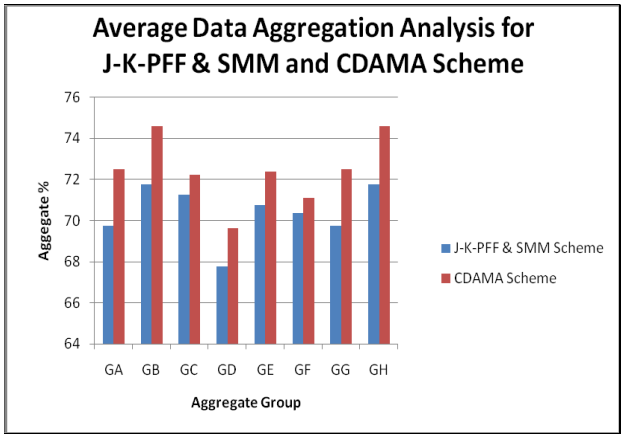



Fig 5.2 New Aggregate J-K-PFF& SMM and CDAMA Scheme

In the above Fig 5.2, database as aggregation service paradigm is described. In CDAMA model, a client stores her database on an un-trusted service provider. Therefore, the client has to secure their database through J-K-PFF schemes because J-K-PFF schemes keep utilizable properties than standard ciphers. Based on J-K-PFF schemes, the provider can conduct aggregation queries without decryption. The most important of all is that, do not have to consider the computation cost and the impact of compromising secret keys (i.e., compromising a client in SMM model is harder than compromising a sensor).

- Data aggregation is more secure in Sum, Min, Max Aggregates as well as in CDAMA.
- Through CDAMA, the ciphertexts from distinct applications can be aggregated, but not mixed.
- For a single-application environment, CDAMA is still more secure than other CDA schemes.
- When compromising attacks occur in WSNs, CDAMA mitigates the impact and reduces the damage to an acceptable condition.
- CDAMA is the first CDMA scheme that supports secure counting.
- The base station would know the exact number of messages aggregated, making selective or repeated aggregation attacks infeasible.
- The performance evaluation shows that CDAMA is applicable on WSNs while the number of groups or applications is not large.
- Using the Database as a Service Model, the provider can conduct aggregation queries without decryption.

## VI. CONCLUSION

This paper studied Sum aggregation protocol in WSN environment and it also proposed Min and Max aggregate of time-series data. This paper also studied CDAMA scheme for a multi-application environment, which is the first scheme. Through CDAMA, the cipher texts from distinct applications can be aggregated, but not mixed. For a single-application environment, CDAMA is still more secure than other CDA schemes. When compromising attacks occur in WSNs, CDAMA mitigates the impact and reduces the damage to an acceptable condition. Besides the above applications, CDAMA is the first CDA scheme that supports secure counting. The base station would know the exact number of messages aggregated, making selective or repeated aggregation attacks infeasible. Finally, the performance evaluation shows that CDAMA is applicable on WSNs while the number of groups or applications is not large. In addition, it applied CDAMA to realize aggregation query in Database-As-a-Service (DAS) model. In DAS model, a client stores her database on an untrusted service provider.

## VII. FUTURE ENHANCEMENTS

It is believed that almost all the system objectives that have been planned at the commencement of the software development have been net with and the implementation process of the thesis is completed. A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order. In future, the concepts can be applied and tested in real wireless network environments. In addition wireless sensor network is data aggregation process implement by the tube search algorithm applied aggregation efficiency is improved. Data aggregation process implement by real time environment applied for AI algorithms.

## VIII. REFERENCES

[1] S. Cheng, Z. Cai, J. Li, and X. Fang, "Drawing dominant dataset from big sensory data in wireless sensor networks," 2015 IEEE Conference Computer Communications (INFOCOM), pp. 531–539, 2015

[2] K. R. Bhakare, R. Krishna, and S. Bhakare, "An energy-efficient grid based clustering topology for a wireless sensor network," International Journal of Computer Applications, vol. 39, no. 14, 2012

[3] M. Shanmukhi and O. Ramanaiah, "Cluster-based comb-needle model for energy-efficient data aggregation in wireless sensor networks,"

Applications and Innovations in Mobile Computing (AIMoC), pp. 42–47,2015

[4] Y. Lu, I. Comsa, P. Kuonen, and B. Hirsbrunner, "Dynamic data aggregation protocol based on multiple objective tree in wireless sensornetworks," Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), IEEE, pp. 1–7, 2015.

[5] J. Li, S. Cheng, Y. Li, and Z. Cai, "Approximate holistic aggregation in wireless sensor networks," Proceeding 35th IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 740–741, 2015.

[6] J. Hicks, N. Ramanathan, D. Kim,M.Monibi, J. Selsky, M. Hansen, and D. Estrin, "AndWellness: An Open Mobile System for Activity and Experience Sampling," Proc. Wireless Health, pp. 34-43, 2010.

[7] N.D. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell, "Bewell: A Smartphone Application to Monitor, Model and Promote Well-being," Proc. Fifth Int'l ICST Conf. Pervasive Computing Technologies for Healthcare, 2011.

[8] V. Rastogi and S. Nath, "Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2010.

[9] P.-A. Fouque, G. Poupard, and J. Stern, "Sharing Decryption in the Context of Voting or Lotteries," Proc. Fourth Int'l Conf. Financial Cryptography (FC '00), pp. 90-104, 2000.

[10] E.G. Rieffel, J. Biehl, W. van Melle, and A.J. Lee, "Secured Histories: Computing Group Statistics on Encrypted Data While Preserving Individual Privacy," http://arxiv.org/abs/1012.2152, 2010.

[11] E. Shi, T.-H.H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-Preserving Aggregation of Time-Series Data," Proc. Network and Distributed System Security Symp. (NDSS '11), 2011.

[12] S.B. Eisenman, E. Miluzzo, N.D. Lane, R.A. Peterson, G.-S. Ahn, and A.T. Campbell, "The Bikenet Mobile Sensing System for Cyclist Experience Mapping," Proc. ACM Fifth Int'l Conf. Embedded Networked Sensor Systems (SenSys '07), pp. 87-101, 2007.

[13] R. Norris, D. Carroll, and R. Cochrane, "The Effects of Physical Activity and Exercise Training on Psychological Stress and Well-being in an Adolescent Population", Journal of Psychosomatic Research, vol. 36, no. 1, pp. 55–65, 1992.

[14] K.R. Fox, "The Influence of Physical Activity on Mental Well-being", Public Health Nutrition, vol. 2, no. 3a, pp. 411–418, 1999.

[15] L.K. George, D.G. Blazer, D.C. Hughes, and N. Fowler, "Social Support and the Outcome of Major Depression", The British Journal of Psychiatry, vol. 154, no. 4, pp. 478, 1989.

[16] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles Tanzeem Choudhury, and Andrew T. Campbell, "A Survey of Mobile Phone Sensing", Comm. Mag., vol. 48, pp. 140–150, September 2010.

[17] Pedro Ferreira, Pedro Sanches, Kristina H¨ o¨ ok, and Tove Jaensson, "License to Chill!: How to Empower Users to Cope with Stress", in Proc. of the 5th Nordic Conference on Human-computer Interaction, pp. 123–132, Lund, Sweden, Oct 20-22, 2008.

[18] K. Patrick, F. Raab, M.A. Adams, L. Dillon, M. Zabinski, C.L. Rock, W.G. Griswold, and G.J. Norman, "A Text Message–based Intervention for Weight Loss: Randomized Controlled Trial", Journal of Medical Internet Research, vol. 11, no. 1, 2009.