INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2022, VOL. 68, NO. 2, PP. 329-336
Manuscript received May 27, 2021; revised May, 2022. DOI: 10.24425/ijet.2022.139886

Secure Data Aggregation in Wireless Sensor Network using Chinese Remainder Theorem

Sanu Thomas, and Thomaskutty Mathew

***Abstract*—A new method of lossless Secure Data Aggregation for Wireless Sensor Network is presented. Secure Data Aggregation is achieved using the popular Chinese Remainder theorem. Here, an 'Augmented Chinese Remainder System' is introduced that incorporates additional features to enforce a higher level of security to the aggregated data. The scheme provides inbuilt signature verification and eliminates the need for separate data validation algorithms. The method achieves data integrity and authentication simultaneously in addition to lossless data aggregation for the data forwarded from the Cluster Head to the Base Station. The aggregate contains the entire individual data from sensors in the encrypted form and the receiver de-aggregates it to get the original data in full without any loss. The Augmented Chinese Remainder System can be extended to secure Multi-level Data Aggregation for WSN.**

***Keywords*—Chinese Remainder Theorem; Authenticated Aggregation; Augmented Chinese Remainder System; TDMA Schedule; Uplink Path; Relay Nodes**

I. Introduction

DATA aggregation is the process of combining and summarizing data from sensors so as to reduce the data traffic in WSN [1]. Substantially higher volume of raw data from sensors to sink causes congestion, packet loss, additional energy consumption and increased communication overhead. In many applications where data redundancy is prevalent, raw data size can be reduced without much dilution in the information content using Data Aggregation (DA). Multiple data from different sensors are meaningfully combined to form the aggregate. DA is very much useful when there is data redundancy in spatial and time domain. In certain category of applications, the entire data from sensors is not needed at the BS. For example, in accidental Fire detection, only the maximum temperature is relevant. For slowly varying temporal parameters, the average value is sufficient. Therefore, instead of full data, the application may need only a set of filtered data for its satisfactory working. Then DA is a natural choice for this type of applications. Some of the common aggregates are sum, average, maximum, minimum and median values [2]. If aggregation takes place at intermediate nodes, it is referred as in-network aggregation.

A. Secure Data Aggregation

A WSN working in hostile environment requires protection from various types of attacks to achieve data confidentiality, data integrity and source authentication. When DA is employed, security is more critical, because the aggregated data is concentrated at aggregators. Therefore, aggregated data are more vulnerable to internal and external attacks compared to the non-aggregated data. Therefore, authentication and privacy of data has to be given higher priority when DA is adopted [1]. In Secure Data Aggregation, security and aggregation are integrated such that the aggregation process provides security automatically.

B. Chinese Remainder Theorem for Secure Data Aggregation

Several standard methods are available for "secure data aggregation" in WSNs. Elliptical Curve Cryptographic (ECC) techniques are used by several authors [3-6]. Use of Bilinear Pairings in loss-free aggregation is proposed by [7-10]. Cryptographic aggregation using Symmetric Keys are described [11-14] to decrease the computational overhead associated with ECC. In a few schemes, Filter based techniques are used to prevent the adversarial attacks [15-17]. A couple of researchers have adopted Synopsis Diffusion method [18,19] for data aggregation. In [20], Merkle Hash Tree structure is adopted to secure aggregated data in Sensor Networks. Chinese Remainder Theorem is applied [21] for the verification of the integrity of the aggregated data. However, the research article [21] uses only the sum of data as the aggregation candidate.

In this paper, the principle of Chinese Remainder Theorem (CRT) [22-25] is used to aggregate the data. The CRT approach aggregates the entire data and the full data is recovered after de-aggregation. Thus, it provides lossless aggregation. The CRT method provides data encryption and authentication in addition to DA. This is the main contribution of this work.

CRT is an ancient theorem (third century AD) in number theory. It has been applied in RSA algorithm, Threshold Cryptography, Homomorphic Encryption and Residue number System [22-26]. In this work, CRT is applied for encryption as well as authentication of aggregated data in WSNs.

II. Wireless Sensor Network Model and Assumptions

The basic layout of a single cluster WSN is shown in Fig. 1. The Sensors Nodes (SNs) numbered 1 to N send their data to the Cluster Head (CH). The received data is aggregated by the CH and then forwarded to the Base Station (BS) through one or more intermediate Relay Nodes (RNs). The CH is assumed to be a powerful device with high computational capacity and it is capable of aggregating data from a large

Authors are with School of Technology and Applied Science, Pullarikkunnu Campus, Mallooseery, Kottayam, Kerala, India (e-mail: thomas.sanu@gmail.com).

number of sensor nodes. The physical distance between any sensor and the CH is within the communication range of the sensor so that a single hop communication can be used between the CH and the associated SN and vice-versa. In the present scenario, it is assumed that the BS is situated at a longer distance from the CH. Therefore, intermediate Relay Nodes are introduced to communicate between the CH and the BS resulting in multi-hop communication between the CH and the BS through the RNs. In Fig. 1, two RN's are shown for easy visualization. In practice, it could be one or more depending on the distance between CH and the BS. In Fig 1, the up-link path is formed along CH→RN_1→RN_2→BS and the down-link path is formed along BS→RN_2→RN_1→CH.

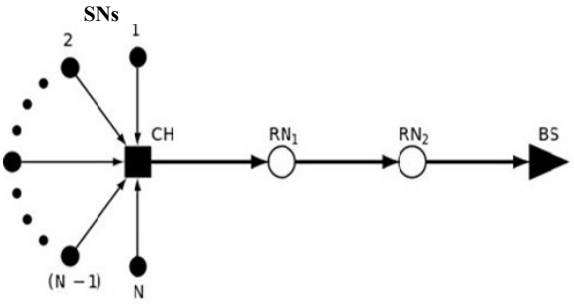

Fig. 1. Basic Layout of a Clustered WSN

During the initialization of the WSN, the BS assigns N distinct ids designated as, b_1, b_2, …,b_N for the respective sensor nodes. The bi's are also the addresses of the corresponding sensor nodes. The values bi 's for i = 1 to N are sent to the CH by the BS through a secure channel. The b_i 's are distinct, relatively large, prime numbers in the range of p_{min} to p_{max}. The need for the primality will be explained shortly. The pmin and the pmax are the designer's choice. The id or the address of node i is taken as bi itself. The b_i's are chosen as nonrepeating primes and hence they are evidently pairwise co-primes. Here, b_i's are the secret keys of the corresponding sensor nodes. The secret keys b_i's are represented by a row vector B as,

$$\mathbf{B} = [b_1, b_2, \ldots, b_N] \quad (1)$$

Row vector **B** is sent securely by the BS to the CH and the CH uses this **B** for secure aggregation/encryption of the data received from the sensor nodes. **B** is kept secret from relay nodes RN_1, RN_2 and the sensor nodes.

In this scheme, the sensors are mainly engaged in sensing the environmental parameters like temperature, humidity, air pollution etc. After sensing the data, the sensors transmit their data at periodic intervals to the CH based on TDMA schedule. The basic Time Slot allotment for the TDMA cycle is shown in Fig. 2. In Fig. 2, the first N time slots, denoted as $DCTS_1$, $DCTS_2$, …., $DCTS_N$ are the Data Collection Time Slots (DCTS) allotted for data collection. During these time slots the sensors send their data to the CH.

In Fig. 2, during the Data Aggregation Time Slot (DATS), the CH aggregates the data received from the sensors and during the Aggregate Forward Time Slot (AFTS), the CH transmits the computed aggregate to the BS.

Data are sensed and sent by sensors in each TDMA cycle to the CH as described in Figure 2. We designate the data value from sensor i as a_i, where the range of i is from 1 to N. The collection of a_i's is given by the row vector **A** as,

$$\mathbf{A} = [a_1, a_2, \ldots, a_N] \quad (2)$$

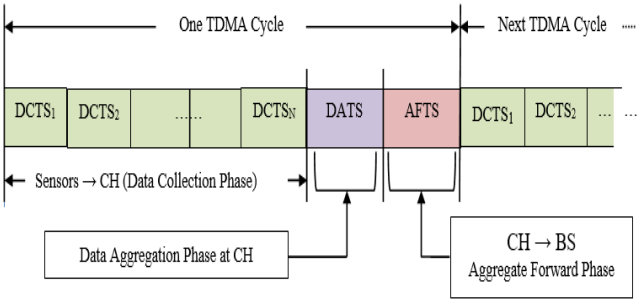

Fig. 2. TDMA Time Slot Allotment for DA using CRT

Here, the data value a_i's are treated as positive integers. In those cases where the actual data values from a sensor are fractional, the entire data stream is converted into an integer stream with a suitable scale-up factor. For example, the original data value 23.45 can be expressed as 2345 with 100 as the scale-up factor.

Additionally, if the actual data stream from a sensor has negative members, then a fixed positive offset is added to the entire data stream to make it positive throughout. These pre-defined offsets and scale factors are properly chosen by the designer, depending on the type and range of the sensor data.

For example, if the data is -3.46, it is scaled-up by 100 to -346 and then an offset say 500 is added to -346 to get 154. On receiving the data, the BS subtracts the offset 500 first, and then scales down by 100 to get back the original data -3.46.

This scaling and offsetting can be implemented at the CH or at sensors themselves. These offsets and scale factors are made available to the BS at the time of initialization. Consequently, the elements of **A** of (2) are positive integers.

The integer constraint on a_i's is due to the fact that the Chinese Remainder Theorem (CRT) operates only with positive integers. An additional restriction to be satisfied while using the CRT [27] is,

$$a_i < b_i \quad (3)$$

for all *i*'s in the range 1 : *N*. The inequality given by (.3) should hold true during all TDMA cycles during the working life of the network. Hence, the secret key b_i has to be chosen higher than the possible maximum value of a_i during the operating life of node *i* for *i* = 1 : N.

III. Basic Lossless Data Aggregation at Cluster Head

During the Data Collection Phase of the TDMA cycle, the CH has collected all the data a_i's from the sensor nodes. The CH has the knowledge of the corresponding b_i 's for i = 1 : N.

Let the smallest solution of N simultaneous Chinese Remainder equations [27] be denoted by integer x as,

$$x = a_i \bmod b_i \quad (4)$$

for i = 1 : N. Using the standard Matlab function **mod**(…), Equation (4) can be rewritten as,

$$\text{mod}(x, b_i) = a_i \quad (5)$$

for all i's. Equation (5) can be compactly represented in terms of arrays **A** and **B** as,

$$\mathbf{A} = \text{mod}(x, \mathbf{B}) \quad (6)$$

The well-known CRT algorithm [28] can be used to determine x which satisfies (6), with input arrays **A** and **B.** Here, x is obtained as the output of function CRT(**A**, **B**) as,

$$x = \text{CRT}(\mathbf{A}, \mathbf{B}) \quad (7)$$

Function CRT uses extended Euclid theorem [29] to find x.

A. Chinese Remainder System

The system that uses CRT for data aggregation is referred as *Chinese Remainder System* (CRS). It is mathematically represented as,

$$\text{CRS} = \{x, \mathbf{A}, \mathbf{B}, N\} \quad (8)$$

CRS is basically a quadruple. In (8), x is the smallest dividend of the CRS which satisfies (6). For convenient reference, x is designated as the Chinese Dividend. Elements of array **B** are the divisors (moduli) which are pairwise co-prime. **A** is the array of corresponding collection of remainders and the size of **B** as well as **A** is N. It is assumed that **B** is selected to satisfy the inequality (3) for all possible values of **A**.

It can be noted that, in (8), **A** represents the set of data from the sensors, **B** represents the corresponding set of secret keys at the CH, x represents the aggregated data at the CH and N represents the number of sensor nodes of the cluster. **A** can vary in successive TDMA cycles, but **B** is a constant for a given session.

B. Solution of the CRS

Consider the CRS represented by (8). To determine x for a given **A** and **B**, using (7) is termed 'solving the CRS'. Let us consider the working of the basic WSN shown in Figure 3.1. At the end of the Data Collection Phase of the present TDMA cycle, the CH calculates x using (7). Generally, x is a large positive integer. The size of x depends mainly on the values of the array elements of **B**. This x is the aggregate of the data array **A**. Elements of array **A** are hidden within x. Given **x** and knowing **B**, the BS recovers data array A by applying (6). Therefore, array **B** acts as the secret key of the cryptosystem.

C. Aggregation of Data array A into Aggregate x at CH

The CH calculates integer x using (7) in the time slot DATS (see Fig. 2). Here x is the 'ciphertext' which is the encrypted aggregate of data array **A**. Thus (7) represents the process of encrypted aggregation of data array **A** using the key array **B**. After aggregation in the time slot DATS, of the present TDMA cycle, the CH forwards the computed aggregate x to the BS in the time slot AFTS. The transmitted aggregate x reaches the BS via the uplink path CH→ RN1 → RN2 →BS.

D. Decryption of Aggregate x at BS

The BS knows **B** and it recovers **A** by decrypting x using (6). Only those entities who know **B** can recover **A** by decrypting x. In this scheme, RN1 and RN2 have no access to **B**. Therefore, they cannot decode x.

Example 1**:** In this demonstrative example, we have taken the number of sensors, N = 5, input data array **A** = [101, 120, 111, 102, 108] and the secret key array **B** = [107, 127, 113, 103, 109]. Here, the elements of **B** are prime numbers and hence, they are mutually co-prime. Also, it is ensured that (3) is satisfied while selecting B. On using (7), x is found to be 8454829159. CH sends x to BS. Input data array **A** can be recovered at BS using (6) as **A** = mod(8454829159, [107, 127, 113, 103, 109]) = [101, 120, 111, 102, 108]

E. Deficiency of Simple Aggregation

Consider the special case when all the N data elements of array **A** are same to say r as

$$\mathbf{A} = [r, r, ..., r]$$

Under constraint (3), we have $r < b_i$ for i = 1 to N. Then, from (4), we see that x is r itself.

Now x no longer encrypts the data but reveals the data directly. To overcome this problem, we introduce additional diversifying elements to the CRS which also provide data verification and authentication. This will be explained in section IV.

IV. SECURED DATA AGGREGATION USING AUGMENTED CHINESE REMAINDER SYSTEM

Let **A**(1), **A**(2),…,**A**(i),..., **A**(K) be the K number of data sequences to be transmitted from the CH to the BS in successive TDMA cycles in a given session.

In the Augmented Chinese Remainder System (ACRS), the CH appends three extra elements to the data array **A** and three corresponding divisor elements (keys) to array **B** as follows.

Three verification parameters designated as g(1), g(2) and h(2) are appended to array **A**(1) to get the augmented array as,

$$\mathbf{C}(1) = [\mathbf{A}(1), g(1), g(2), h(2)] \quad (9)$$

Initially, g(1) is either sent from BS to CH before the commencement of the 1st TDMA cycle through a secured channel or it is stored in the CH at time of installation. The values of g(2) and h(2) are selected by the CH itself. Here, **C**(1) is the augmented sequence (array) of A(1) for the first packet.

Similarly, three dissimilar divisor elements h(1), bb_1 and bb_2 are appended to sequence B to get the augmented array **D**(1) as,

$$\mathbf{D}(1) = [\mathbf{B}, h(1), bb_1, bb_2] \quad (10)$$

Here also, initially, **B**, h(1), bb1 and bb2 are either sent from BS to CH before the commencement of the 1st TDMA cycle

through a secured channel or they are stored in the CH at time of installation.

Since three new elements have been appended to get **C**(1) and **D**(1), their new size is (N+3). The additional parameters g(1) and h(1) embedded in the aggregated data, sent from the CH to the BH, are used to provide signature verification and data integrity for the first packet, whereas the signature parameters g(2) and h(2) embedded in the aggregated data sent in the first packet are used to provide valid signature verification and data integrity for the second packet to be sent in the second TDMA cycle. The sole purpose for sending g(2) and h(2) in advance with respect to the second packet is for enhancing the security. This aspect is explained in section V.

As explained earlier, the values of the secret divisor keys, [**B**, h(1), bb_1, bb_2] and the parameter g(1) are the designers choice depending on the application. BS calculates these values and sends them to CH through secured channel. They are not known to the intermediate relay nodes. On the other hand, the signature parameters g(2) and h(2) are properly chosen by the CH. The parameters g(1), g(2) and h(2) are embedded in the first packet and sent to the BS. The augmented Chinese Remainder System for the first packet of the session is represented as,

$$\text{ACRS}(1) = \{w(1), \mathbf{C}(1), \mathbf{D}(1), N + 3\} \quad (11)$$

Here, w(1) is the dividend (the new data aggregate) of the ACRS(1) where C(1) and D(1) are given by (9) and (10) respectively. The aggregate w(1) is given by,

$$w(1) = \text{CRT}\{\mathbf{C}(1), \mathbf{D}(1)\} \quad (12)$$

A. Aggregation and signature embedding (Encoding) at CH using ACRS

The TDMA cycle for Aggregation and signature embedding for ACRS is shown in Fig 3.

Figure 3 is an extension of Figure 2 with an additional time slot for appending the signature parameters g(1), g(2) and h(2). During the data collection phase of the of the first TDMA cycle, the CH has received all the data values from the sensors and it forms **A**(1). The CH also appends g(1), g(2) and h(2) to **A**(1) in the time slot ASPTS (Append Signature Parameters Time Slot) to get C(1) as given by (9). The CH already knows D(1) specified by (10). Then, it calculates w(1) using (12) in the time slot DATS (Data Aggregation Time Slot). Now w(1) is the augmented aggregate of C(1) where g(1) acts as the ***signature*** of the CH.

In ACSR(1), scalar w(1) is the aggregate of the data of the first TDMA cycle. The CH sends this w(1) to the BS as the first packet of the session, in time slot AFTS (Aggregate Forward Time Slot), through the uplink path CH→ RN1 → RN2 →BS.

The signature parameters g(2) and h(2) are embedded (hidden) in w(1) to be used as the signature parameters for the second packet corresponding to the second TDMA cycle.

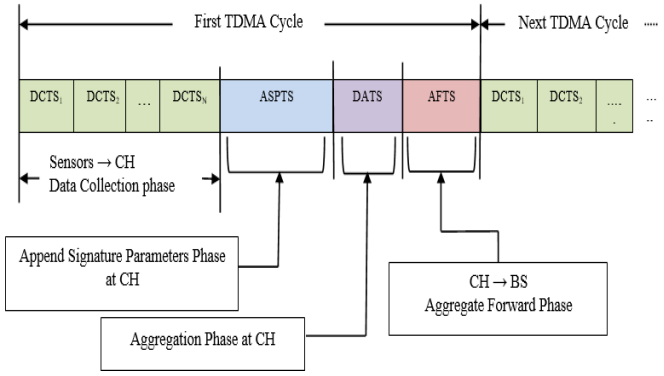

Fig. 3. TDMA Time Slot Allotment for ACRS

B. Redressal of the deficiency of the simple CRS

Additional dissimilar signature parameters, g(1), g(2) and h(2) included in C(1) [Eqn. (9)], also overcome the shortcoming of the CRS when all the data elements are same as discussed in section III E. The augmented array C(1) will now contain dissimilar elements due to the presence of g(1), g(2) and h(2), even when A(1) is an equal element array. Therefore, the ACRS encryption hides the data elements perfectly. This is demonstrated in Example 2 in section VI, where, **A**(1) = [100, 100, 100] and array **C**(1) = [**A**(1), g(1), g(2), h(2)] = [100, 100, 100, 102, 130, 139].

C. Decoding of the Augmented Aggregate at the BS

On receiving the aggregate w(1), the BS decodes it as,

$$C(1) = \text{mod}\{w(1), D(1)\} \quad (13)$$

We know that the $(N+1)^{th}$ element of C(1) is g(1) and that of D is h(1). Therefore,

$$g(1) = \text{mod}\{w(1), h(1)\} \quad (14)$$

From (9), (10) and (12), the original full data array A(1) is de-aggregated at BS by decrypting W(1) as,

$$\mathbf{A}(1) = \text{mod}\{w(1), \mathbf{B}\} \quad (15)$$

D. Verification and Decryption of w(1) at the BS

In this scheme, RN1 and RN2 have no access to the augmented secret key array **D**(1). Hence, they cannot decode w(1). The BS, on receiving w(1), calculates $g(1)_{cal}$ as,

$$g(1)_{\mathbf{cal}} = \text{mod}\{w(1), h(1)\} \quad (16)$$

The BS compares $g(1)_{cal}$ with g(1) which is already known to the BS.

If $g(1)_{cal}$ = g(1), then the BS is assured of the integrity of the received w(1). It means that w(1) has not been tampered during transmission and the data source is certainly from the CH. Thus, the condition $g(1)_{cal}$ = g(1) provides data integrity and signature verification (source authentication).

Once the verification is successful, the BS which knows the **B**, decrypts w(1) using (15) to recover original data **A**(1). In this way, ACRS(1) provides automatic data integrity and source authentication through signature verification for the first packet, thereby eliminates the need for a separate verification scheme. Here g(1) acts as the signature of the CH for aggregate w(1). Thus, g(1) and h(1) together form the signature parameters for the first packet. This signature verification and data reconstruction is demonstrated in Examples 2 and 3 of section VI.

On the other hand, if $g(1)_{cal} \neq g(1)$, it means the integrity of the received w(1) is lost. Then w(1) is discarded and the BS may request the CH for the retransmission of w(1). The failure of the signature verification is demonstrated in Example 3 of section VI.

E. Signature verification of the second and subsequent packets

The additional signature parameter set g(2) and h(2), embedded in w(1), has already been sent by the CH to the BS in AFTS of first TDMA cycle. The BS recovers g(2) and h(2) from the aggregate w(1) as,

$$[g(2), h(2)] = \text{mod}\{(w(1), bb_1, bb_2)\}$$

The parameters g(2) and h(2) are kept ready for the signature verification of the second packet.

Now in the second TDMA cycle, the CH forms the arrays C(2) and D(2) as**,**

$$\mathbf{C}(2) = [\mathbf{A}(2), g(2), g(3), h(3)]$$

$$\mathbf{D}(2) = [\mathbf{B}, h(2), bb_1, bb_2]$$

The aggregate w(2) for the second packet is calculated by the CH as,

$$w(2) = \text{CRT}\{\mathbf{C}(2), \mathbf{D}(2)\}$$

This w(2) is sent to the BS as the second packet in the second TDMA cycle.

When the BS receives w(2), the data recovery and the signature verifications for the second packet are similar as in the case of the first packet. For the signature verification of the second packet, BS use g(2) and h(2), similar to the procedure as explained in section IV D. Similarly, g(3) and h(3) are used for the signature verification of the third packet. This process is continued for the subsequent packets of the session. In general, for the i^{th} packet,

$$C(i) = [A(i), g(i), g(i+1), h(i+1)] \quad (17)$$

$$D(i) = [B, h(i), bb1, bb2] \quad (18)$$

$$w(i) = \text{CRT}\{\mathbf{C}(i), \mathbf{D}(i)\} \quad (19)$$

V. SECURITY OF ACRS

If the same signature parameters g's and h's are used for all the packets, sent from the CH to the BS, then an adversary may guess g's and h's by capturing the successive packets containing the aggregates. This malicious security attack is effectively prevented by the use of dissimilar signature parameters g(1), h(1), g(2), h(2) in successive packets.

For this purpose, we send the signature parameters g(2) and h(2) in advance, embedded in the aggregate of the first packet. These g(2) and h(2) are used for the signature verification for the second packet.

A. Selection of g and h

The signature parameters g(1), g(2),...etc. have to be distinct and different with good diversify so that hackers cannot break the secrecy of g's. Let K be the total number of packets in a session. For i = 1 to K, the h(i) selected should be greater than g(i) [Eqns. (17) and (18)] and should be a prime number to satisfy the requirements of CRT. The divisor sequence D(i) contains h(i) along with B, bb_1 and bb_2 as its co-members. Therefore, h(i) should be so chosen that it is distinct from its co-members. Thus **D**(i) should be a non-repeating sequence of prime numbers for i = 1 to K.

B. Selection of bb_1 and bb_2

In the ACRS scheme, bb_1 and bb_2 are the divisors for the remainders g(i) and h(i) respectively for i = 2 to K. Therefore, bb_1 and bb_2 have to be greater than g(i) and h(i) respectively for i = 2 to K. Additionally, bb_1 and bb_2 have to be primes and should be distinct from their co-members in array **D**(i)'s. Array **B**, parameters bb_1 and bb_2 remain same for all the packets in a given session.

C. Length of Secret Keys in ACRS

The Secret keys array **B**, h(i)'s, bb_1 and bb_2 used in ACRS are distinct prime numbers. If they are small and within certain fixed range, they can be somehow guessed by hackers. Therefore, to provide higher security, in practice, the secret keys have to be selected from a set of very large prime numbers spread over a wide range. Then, hackers cannot break in and get the secret keys. Also, the range and the values of secret keys have to be changed from session to session based on the security requirement.

VI. DEMONSTRATIVE EXAMPLES FOR ACRS

The following simple examples illustrate the working of ACRS with and without data integrity.

Example 2: Non-tampered data

In this simple example, N = 3, Array **B** = [107, 127, 113].
Let **A**(1) = [100, 100, 100]. Here, all the elements of A(1) are equal. Then the aggregate x as given by (7) would also be 100. Now x reveals the data instead of hiding it. To overcome this, we append g(1) = 102, g(2) = 130 and h(2) = 139 to **A**(1) to get **C**(1) as,

C(1) = [**A**(1), g(1), g(2), h(2)] = [100, 100, 100, 102, 130, 139]

We also append h(1) = 103, bb_1 = 149 and bb_2 = 151 to **B** to get **D**(1) as

D(1) = [**B**, h(1),bb_1,bb_2] = [107, 127, 113, 103, 149, 151]
Here, it can be noted that h(1) is a prime number greater than g(1), h(2) is also a prime number greater than g(2) and bb1, bb2 are prime numbers which are greater than g(2) and h(2). These values satisfy the constraints of ACRS. Now the augmented aggregate w(1) is calculated at CH using (12) as,

w(1) = 894326823584

Then, CH sends this w(1) to the BS. On receiving w(1), the BS calculates $g(1)_{cal}$ using the Equation (16) as,

$g(1)_{\mathbf{cal}}$ = mod{w(1), h(1)}= mod{894326823584, 103}= 102

Here, $g(1)_{cal}$ = g(1). Therefore, the signature is verified correctly and the data integrity is assured. Then, the BS decrypts w(1) to get the data A(1) using the Equation (15) as,

A(1) = mod{w(1), **B**}= mod{894326823584, [107, 127, 113]}
= [100,100,100]

This reconstructed data sequence is same as the transmitted data.

Example 3: Tampered data

Here, the values of N, **B**, **A(1)**, **C(1)** and **D(1)** are same as in Example 2. The aggregate w(1), 894326823584, is also same and the CH sends this w(1) to the BS. During transmission, this aggregate w(1) is tampered as,

w(1)_tampered = w(1) + Error/Noise = 894326823584 + 10
= 894326823594

Now, BS receives this w(1)_tampered. Then the BS calculates $g(1)_{cal}$ using the Eqn. (16) as,

$g(1)_{\mathbf{cal}}$ = mod{w(1)_tampered, h(1)}= mod{894326823594, 103}= 9

Since g(1) = 102 and $g(1)_{cal}$ = 9, $g(1)_{cal} \neq g(1)$. Therefore, the signature verification fails. This means the data integrity is lost. The BS rejects the tampered aggregate and requests for retransmission. In this case, the BS does not decrypts w(1))_tampered.

VII. ACRS ALGORITHMS

ACRS aggregation process at CH is given in Algorithm VII A.

ALGORITHM VII A : ACRS_AGGREGATION at CH

Input :A(i)'s, g(i)'s, B, h(i)'s, bb1, bb2
Output : w(i)

1. Get C(i) as C(i) =[A(i), g(i), g(i+1), h(i+1)] //Augmented Data array
2. Get D(i) as D(i) = [B, h(i), bb1, bb2] //Augmented Key array
3. Compute w(i) as w(i) = CRT{**C**(i), **D**(i)} //Aggregate
4. Forward w(i) to BS
5. Over

ACRS de-aggregation at BS is given in Algorithm VII B.

ALGORITHM VII B : ACRS_DE-AGGREGATION at BS

Input : w(i)'s, g(i)'s, h(i)'s, B, bb_1 and bb_2.
Output : A(i) //Sensor Data array

1. Calculate $g(i)_{cal}$ as $g(i)_{cal}$= mod{w(i), h(i)}
2. Compare $g(i)_{cal}$ with g(i)
If $g(i)_{cal} \neq g(i)$ //Signature verification fails
Discard w(i)
Request for re-transmission of w(i)
Go to step 5
endif
3. Compute [g(i+1), h(i+1)] = mod{(w(i),bb_1, bb_2)}
Store [g(i+1), h(i+1)] //for signature verification for the next packet
4. Compute A(i) as A(i) = mod{w(i), B} //De-aggregation over
5. Over

VIII. COMPARISON WITH BONEH, GENTRY, LYNN, SHASHAM AND MERKLE HASHTREE METHOD

The data aggregation using ACRS is compared with two popular secured data aggregation methods, Boneh, Gentry, Lynn and Shasham (BGLS) [8] and Merkle Hash Tree (MHT) [20].

The BGLS scheme uses bilinear maps and Gap Diffie-Hellman signatures. In [10], the ciphertext has 3 components, one for encrypting the message, next one to represent the public key and the last one to hold the aggregate signature. The security of the aggregate signature in BGLS scheme requires that the messages should be distinct from one another. In ACRS method, a single large number g(i) acts as the signature of the aggregator (CH), in association with large prime number h(i) (secret key of the aggregator). In ACRS, there is no public key as in BGLS scheme. In ACRS, the messages need not be distinct for strong security as required in BGLS method.

Przydatek, Song and Perring [20] use the Merkle Hash Tree (MHT) method for secured information aggregation. Here, the Secured Hash values of the data are aggregated by combining the individual hash values into a compound aggregate. In MHT, hash functions are cumulatively built based on the binary tree structure. The BS has the final compound hash in which all the individual hash values are embedded. Thus, in MHT, the BS can verify the authenticity of every individual data values.

The computational cost of BGLS, MHT and ACRS are shown in Table where N is the number of messages (data values) aggregated.

TABLE I
COMPUTATIONAL COST OF BGLS, MHT AND ACRS

Schemes	Signature Generation	Signature Verification
BGLS	*N* MTG + *N* PM + *N* FM	*N* MTG + (*N*+1) BP + *N* FM
MHT	2 x N HC	$Log_2(N)$ HC
ACRS	1 CRT(Nxm) + *N* FM	1 MOD(Nxm, m)

The BGLS scheme represented in Table 1, MTG denotes Map To Group operation. Symbol PM denotes Elliptic Curve Point Multiplication, FM denotes Field Multiplication (multiplication in finite field Z_p) and BP stands for Bilinear Pairing calculation. Refer [8] for details. In Table 1, with reference to MHT, symbol HC stands for Hash Calculation. In general the time complexity of HC is O(m) where m is the message (data) size in bits. For details, refer [20].

For comparing the computational overhead, the execution times taken for signature generation/verification by ACRS, BGLS and MHT methods are calculated. The execution time in seconds versus the number of sensor nodes, N is shown in Figure 4. In the experiment, the number of sensors, N assigned to the CH is varied from 8 to 26 in steps of 2.

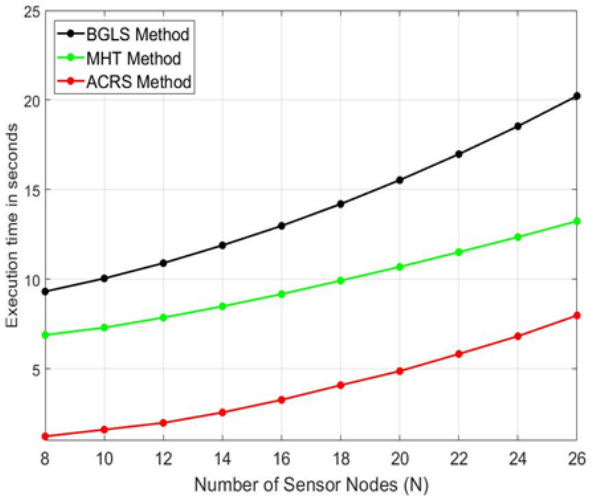

Fig. 4. Execution time versus Number of Sensor Nodes, N

From the plots, it can be seen that the ACRS method takes relatively less time for aggregation, signature generation and verification compared to BGLS and MHT methods. As N increases, the relative time gap increases between ACRS and BGLS schemes.

IX. OMNeT++ SIMULATION

The ACRS, BGLS and MHT methods are simulated using OMNeT++ network simulator. The area of the wireless network is taken as 1100m×600m (outdoor). Sensor nodes, CH and Relay Nodes are XbeeZigbee S2Cs. Communication radius max = 1,200 meters. Transmission power = 3.1 mW (5dBm). Data Rate = 250 Kbps. That is, 8/250=32 uS per byte. Energy consumed to transmit a byte = 32us×3.1mW is approximately equal to 0.1 uJ per byte. Therefore, the energy consumed depends on the length of data in bytes for ACRS, BGLS and MHT. In the simulation experiment, the number of sensor nodes is varied from 10 to 45 in steps of 5. The corresponding actual energy consumed to transmit the aggregated data is shown in Figure 5. In ACRS scheme, the CH aggregates the data and transmits the same whereas Relay Nodes RN_1 and RN_2 transmit the aggregate data. [Refer to Fig 1].

From the plots of Figure 5, it can be seen that the ACRS method consumes the lowest energy compared to the other schemes for transmitting the aggregated data. Hence, ACRS is much suited for data aggregation in WSN, where the energy consumption is a major constrain.

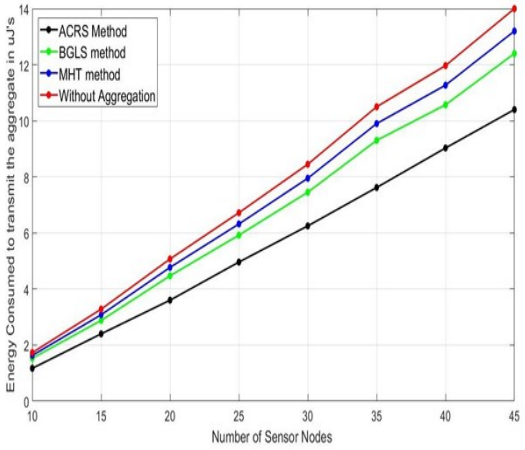

Fig. 5. Energy consumed by ACRS, BGLS, MHT and the Scheme without aggregation

CONCLUSION

A novel technique for secured data aggregation has been described for Wireless Sensor Network (WSN). This method employs Augmented Chinese Remainder System (ACRS) that uses Chinese Remainder Theorem. This ACRS approach is designed to provide data aggregation at the Cluster Head of the WSN. In addition, ACRS method also contributes data security and authentication simultaneously. Simulation results show that the ACRS aggregation is faster and consumes less energy compared to BGSL and Merkle Hash Tree aggregation methods.

REFERENCES

[1] S. Ozdemir and Y. Xiao, "Secure data aggregation in Wireless sensor networks: a comprehensive overview," *Computer Networks*, Vol. 53, No.12, pp. 2022–2037, Aug.2009. https://doi.org/10.1016/j.comnet.2009.02.023

[2] K. Akkaya, M. Demirbas and R.S. Aygun, "The Impact of Data Aggregation on the Performance of Wireless Sensor Networks", *Wiley Wireless Communication and Mobile Computing (WCMC) Journal*, Vol. 8, pp. 171–193, 2008. https://doi.org/10.1002/wcm.454

[3] M. Elhoseny, H. Elminir, A. Riad and X. Yuan, "A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption," *Journal King Saud university, Computer and Information Science*, Vol. 26, No. 3, pp. 262–275, 2015. https://doi.org/10.1016/j.jksuci.2015.11.001

[4] J. Boga, M. S. Huque1 and S. B. Saheb, "An Approach to Secure Data Aggregation in Curve Cryptography) Scheme," *International Journals of Advanced Research in Computer Science and Software Engineering*, Vol. 7, No. 7, pp. 263–267, 2017. http://dx.doi.org/10.23956/ijarcsse/V7I7/0162

[5] M. B. Omar Rafik and F. Mohammed, "Fast and secure implementation of ECC-based concealed data aggregation in WSN," *Global Information Infrastructure Symposium*. GIIS, Toronto, pp. 1–7, 2013. https://doi.org/10.1109/GIIS.2013.6684371

[6] Q. Zhou, G. Yang and L. He, "A secure enhanced data aggregation based on ECC in wireless sensor network," *Sensor Journal*, Vol. 14, No. 4, pp. 6701–6721, 2014. https://doi.org/10.3390/s140406701

[7] Ara M. Al-Rodhaan, Y. Tian and A. Al-Dhelaan, "A Secure Privacy-Preserving Data ggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems," *IEEE Access*, Vol. 5, pp. 12601–12617, 2017. https://doi.org/10.1109/ACCESS.2017.2716439

[8] D. Boneh, C. Gentry, B. Lynn and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," *Proceedings of EUROCRYPT 2003*, Vol. 2656 of LNCS. Springer, Boston, MA, USA, pp. 416–432, 2003.

[9] J. L. Tsai, "A New Efficient Certificate less Short Signature Scheme Using Bilinear Pairings," *IEEE Systems Journal*, Vol.11, No. 4, pp. 2395– 2402, Dec.2017. http://dx.doi.org/10.1109/JSYST.2015.2490163

[10] B.Waters, "Efficient Identity-Based Encryption without Random Oracles," *Springer Berlin Heidelberg Berlin*, Heidelberg, pp. 114–127, 2005.

[11] C. Jie, S. Lili, Z. Hong, X. Yan and L Lu, "Data aggregation with end-to- end confidentiality and integrity for large-scale wireless sensor networks," *Peer-to-Peer Networking Applications*, Vol.11, No.5, pp. 1022–1037, 2018. https://doi.org/10.1007/s12083-017-0581-5

[12] J. Jose, J. Jose, and H. Muhammed Ilyas, "Symmetric concealed data aggregation techniques in wireless sensor networks using Privacy Homomorphism: A review," *International Conference on Information Science (ICIS),* pp. 275–280, 2016. http://dx.doi.org/10.1109/INFOSCI.2016.7845340

[13] K. Parmar, C. Devesh and Jinwala, "Symmetric-Key Based Homomorphic Primitives for End-to-End Secure Data Aggregation in Wireless Sensor Networks," *Journal of Information Security* Vol. 6, pp. 38–50, 2015. http://dx.doi.org/10.4236/jis.2015.61005

[14] K. Shah and D. C. Jinwala, "A secure expansive aggregation in Wireless Sensor Networks for linear infrastructure," *IEEE Region 10 Symposium (TENSYMP)*, pp. 207–212, 2016. https://doi.org/10.1109/TENCONSpring.2016.7519406

[15] E. Choudhari, K. D. Bodhe and S. M. Mundada, "Secure data aggregation in WSN using iterative filtering algorithm," *International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. pp. 1–5, 2017. https://doi.org/10.1109/ICIMIA.2017.7975603

[16] M. Mansouri, L. Khoukhi, H. Nounou and M. Nounou, "Secure and robust clustering for uantized target tracking in wireless sensor networks," *Journal of Communications and Networks*, Vol. 15, No. 2, pp. 164–172, 2013. https://doi.org/10.1109/JCN.2013.000029

[17] G. Priyanka, Padmane and K. G. Bagde, "Secure Data Aggregation in Wireless Sensor Network using BECAN Scheme," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, No. 10, pp. 205–209, 2015.

[18] S. Nath, P. B. Gibbons, S. Seshan and Z. R. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, ACM, New York, NY, USA, pp. 250–262, 2004. https://doi.org/10.1145/1340771.1340773

[19] S. Roy, M. Conti, S. Setia and S. Jajodia, "Secure Data Aggregation in Wireless Sensor Networks: Filtering out the Attacker's Impact," *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 4, pp. 681–694, 2014. https://doi.org/10.1109/TIFS.2014.2307197

[20] B. Przydatek, D. Song, and A. Perrig, "Secure information aggregation in sensor networks," *Proceedings of ACM SenSys, ACM*, Los Angeles, CA, USA, pp. 255–265, 2003

[21] R. Dian-xu, Z. Xiao-Guang and Li Li-jun, "Safety Data Fusion Algorithm in Wireless Sensor Network," *Journal of Networks*, Vol. 8, No. 5, pp. 1121–1129, 2013. http://dx.doi.org/10.4304/jnw.8.5.1121-1129

[22] J. Grossschadl, "The Chinese Remainder Theorem and its application in a high-speed RSA crypto chip," *Computer Security Applications, ACSAC, 16th Annual Conference*, pp. 384–393, 2000. https://doi.org/10.1109/ACSAC.2000.898893

[23] Y. Mo and S. Li, "Base Extent Optimization for RNS Montgomery Algorithm," *IEEE Trustcom/BigDataSE/ICESS*, PP. 1004–1009, 2017. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.344

[24] N. Shi, Z. Hou, M. Tan, K. Shao and X. Zhu, "A threshold encryption scheme without a dealer based on Chinese remainder theorem," *IEEE 9th qnternational Conference on Communication Software and Networks (ICCSN)*, PP. 90–96, 2017. http://dx.doi.org/10.1109/ICCSN.2017.8230085

[25] W. Stalling (2010), "Cryptography and Network Security: Principles and Practices (3rd. ed.)", *Prentice-Hall*, NJ, USA, 2019.

[26] V. Kumar and N. Srivastava, "Chinese Remainder Theorem based Fully Homomorphic Encryption over Integers," *International Journal of Applied Engineering Research (ISSN0973-4562)*, Vol. 14, No. 2, (Special Issue), pp. 203-208.

[27] Y.H. Ku and X. Sun, "The Chinese remainder theorem," *Journal Franklin Institute,* Vol. 329, pp. 93–97, 1992.

[28] D. E. Knuth, *Semi-numerical Algorithms (3rd ed.), The Art of Computer Programming,*" Vol.2. Addison-Wesley, Reading, MA, 1997.

[29] X. Zheng, C.T. Huang and M. Matthews, "Chinese remainder theorem based group key management*," Proceedings of the 45th annual southeast regional conference*. ACM–SE45, New York, NY, pp. 266–271, 2007. https://doi.org/10.1145/1233341.1233389