Distributed Data Aggregation in Wireless Sensor Network
with Peer Verification

Sumanta Chatterjee, Alwyn R. Pais and Sumit Saurabh
Dept. of Computer Science and Engineering, National Institute of Technology, Karnataka, Surathkal, India

Keywords: Wireless Sensor Network, Secure Aggregation, Homomorphic Permutation, Commitment Protocol.

Abstract: Data aggregation in wireless sensor network is implemented to reduce the communication overhead and to reduce bandwidth utilization. Data confidentiality requires the sensor node to transmit the data in a secure manner so that the adversary is unable to read the data or transmit false data even if it compromises some of the sensor nodes or aggregation node. In this paper a distributed aggregation protocol using homomorphic trapdoor permutation is proposed. This protocol distributes the responsibility of key generation , aggregation and verification to different nodes to reduce the overall power consumption of the sensor network. The peer verification scheme is also proposed as a part of the protocol. Peer verification ensures the authentication of the data and sender node in the network, by at least k peer nodes. Security of the proposed protocol is analyzed against passive and active adversary model.

1 INTRODUCTION

The term *"Internet of Things"* was coined by International Telecommunication Unit (ITU) and European Research Cluster on the Internet of Things (IERC). A stack of four layers were proposed for the Internet of Things architecture; Sensing Layer, Aggregation Layer, Network Layer and Application Layer.

One of the important functionality of sensing layer is aggregation of data. Data aggregation often involves compaction of data from multiple sensors at intermediate nodes and transmission of the aggregated data to the base station to eliminate redundancy. It is essential to develop energy efficient data aggregation algorithms. Aggregation while providing end-to-end security is a challenging task. Data aggregation requires some security considerations. It must ensure that the information content is never revealed to the unauthorized person or malicious adversary and that the content is non-malleable ,unauthorized alteration of the content is easily identifiable and there is no replay attack due to stale data (Ozdemir and Xiao, 2009). During aggregation, network should be able to identify external attacker and balance energy consumption to increase availability of the network. It must enable the aggregator to check that the message is authenticated and also the sender of the message is authorized and authenticated. Since wireless sensor networks use a shared wireless medium,to detect maliciously injected or spoofed packet, sensor nodes require authentication mechanisms. Our contribution in this paper is as follows

- We propose a distributed aggregation protocol to distribute the responsibility of key generation , aggregation and verification among different nodes to reduce the overall power consumption of the sensor network.
- The peer verification scheme using commitment protocol is proposed to ensure the authentication of the data and sender node in the network by at least k nodes.
- To our knowledge, our paper is first to propose the use of fully homomorphic trapdoor permutation to ensure that active adversary is not able to read the aggregated data from compromised aggregator.

The rest of the paper is organized as follows. In section 2 we review some of the proposed methods for secure data aggregation in Wireless Sensor Networks (WSN). In section 2 also some preliminary concepts are defined that are used in the proposed model. In section 3 a new model for distributed data aggregation is discussed. Section 4 contains the conclusion and future scope of study.

Chatterjee, S., Pais, A. and Saurabh, S.
Distributed Data Aggregation in Wireless Sensor Network - with Peer Verification.
DOI: 10.5220/0005991604400445
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT*, pages 440-445
ISBN: 978-989-758-196-0

2 LITERATURE REVIEW

Data aggregation is widely studied in wireless sensor network (WSN). As sensors are very energy constrained devices , data aggregation is effective in conserving battery power, bandwidth utilization of overall lifetime of the sensor network (Krishnamachari et al., 2002). Compromised sensor node can disclose data and allow the adversary to inject false data into the sensor network. Several protocols are proposed for secured data aggregation in WSN. Proposed solutions are either one aggregator model where all data is aggregated by a single node or multiple aggregator model where several aggregators exchange data among themselves. One of the very first solution for secure data aggregation in WSN was proposed by Hu and Evans (Hu and Evans, 2003). In this protocol one way function is used to produce a key chain for each node and the key is used only one time. Each node sends the data with a message authentication code (MAC) to its parent. Data aggregation and authentication (DAA) is proposed by (Ozdemir and Çam, 2010) to enable false data detection in data aggregation with confidentiality. The data aggregator forwarding node is surrounded by T monitoring nodes. Each monitor node pair with another T neighbour nodes of data aggregator. Each of them compute a subMAC bit of T-bit MAC of the data and send it to the forward aggregator to verify the encrypted data. This protocol introduces additional over head of key exchange and monitor node establishment. A secure hop-by-hop data aggregation protocol (SDAP) is proposed in (Yang et al., 2008). SDAP uses probabilistic approach to dynamically partitions the topology tree of the sensor network into multiple logical groups (subtrees) of similar sizes. In (Castelluccia et al., 2009) pseudo random function and one way hashing is used to perform homomorphic encryption on the data using a secret key for each sensor node. Data can be easily aggregated in base station, but this model does not authenticate the data sender and does not support integrity of the data or prevent false data detection. A variation of DAA is proposed by (Li et al., 2015) that uses fully homomorphic encryption to encrypt the data. Privacy homomorphism over elliptic curve cryptography along with a message authentication code is proposed by (Ozdemir and Xiao, 2011) to preserves data confidentiality and integrity in hierarchical data aggregation. The idea of an end to end concealed data aggregation using privacy homomorphism is explored in (Westhoff et al., 2006).Verification of the sensor node using witness aggregation is explained in (Du et al., 2003). The witness nodes of each data aggregator compute MACs of the aggregated data and also perform data aggregation. The base station verify the correctness of the data by aggregating and verifying the MACs computed by the witness nodes. Secure data aggregation in clustered network and a technique of slicing the data and mixing is discussed in (He et al., 2007). one draw back of the model is that the intermediate sensor nodes is capable of changing the data while sending it to the the aggregator. Hence confidentiality constraints are not satisfied in this model.

Now we establish some preliminary concepts required in our proposed model.

2.1 Preliminary Concepts

Definition 1. ***Trapdoor Permutation*** *(Schmidt-Samoa, 2006): Trapdoor permutation family is a collection of finite function $\mathcal{F} = \{f_i | f_i : S_f \rightarrow D_f\}$ such that for every function f*

- *There exists a polynomial* **KEYGEN**(1^λ) *function that for a given security parameter λ generates a pair $f(.), f^{-1}(.)$ and trapdoor information $|t_f| < p(\lambda)$ such that the mapping $(x, f(x)) \rightarrow y$ and $(y, f^{-1}(y)) \rightarrow x$ can be computed in polynomial time given the trapdoor information is available.*
- *There is a probabilistic polynomial time algorithm that given $f(.)$ can output a random element S_f uniformly distributed in Range(S_f).*
- *The function f is hard to invert without trapdoor information t_f. That is for every polynomial time $\mathcal{A}$ there is a negligible function $\kappa(\lambda)$ such that*

$$Pr_{(f,f^{-1}) \leftarrow_R \mathbf{KEYGEN}, x \leftarrow_R S_f}[\mathcal{A}(1^\lambda, f, f(x)) = x] < \kappa(\lambda) \quad (1)$$

- *The domain Dom$\{D_f\}$ and range Range$\{S_f\}$ being $\{0,1\}^n$*

Hayasi, Okamoto, Tanaka proposed an algorithm to obtain RSA trapdoor permutation in common domain (Hayashi et al., 2004).

3 PROPOSED MODEL

In this section we discuss a novel approach for data aggregation in wireless sensor network in the presence of active and passive adversary. The protocol focuses on distributing the functionality of the key distribution, aggregation and verification.

3.1 Sensor Network and Adversary Model

A sensor network consists of three types of nodes. Sensor nodes collect the raw data , an aggregation

node collects the data from sensor nodes to aggregate them and base station (BS) finally receive the aggregated data from the aggregation node. A sensor network of N nodes can be represented using a graph $G =< V, E >$ where each sensor node is a vertex $SN_i; 1 \leq i \leq N \in V$ and each radio link between a pair of sensor nodes (i, j) is an edge $e_{ij} \in E; 1 \leq i, j \leq N; i \neq j$. Sensor nodes are distributed into clusters. Each cluster is modeled using a subgraph containing n nodes with each node connected to at least $k+1$ peers. A cluster head and a verifier node is selected based on residual energy. Inclusion of a sensor node into a cluster is governed by the minimization of the parameter $\frac{1}{n}\sum|e_{ih}|, \{1 \leq i \leq n; i \neq h\}$. $SN_h$is the elected cluster head.

An aggregation function is defined as $y(t) = f(m_1(t), m_2(t), \ldots, m_n(t))$ where at time instance t, $y(t)$ is the data aggregated and $m_i(t)$ is the data collected by $i-th$ input node. In our protocol $f(t) = \prod_{i=1}^{N} m_i(t)$. We also assume that the noise $\delta(t)$ in this cluster follow a Gaussian distribution with mean at zero and for each sensor node $\delta_i \ll m_i$. Hence, $\prod_{i=1}^{n}(m_i + \delta_i) = \prod_{i=1}^{n} m_i(1 + \frac{\delta_i}{m_i}) \approx \prod_{i=1}^{n} m_i$ as $(1 + \frac{\delta}{m}) \approx 1$

Initially it is assumed that a probabilistic polynomial time adversary is capable of eavesdropping the conversation between different nodes and an access to encryption oracle. In subsequent discussions we discuss more powerful adversary who has access to aggregation and verification oracle.

This paper used homomorphic trapdoor permutation to achieve data aggregation without revealing the data to aggregator node while providing authentication and end to end security.

3.2 Distributed Aggregation Model

One of the drawbacks of single aggregator model in energy constrained devices is that it puts additional load in the aggregator to perform the aggregation, key distribution and verification. Also, Once an aggregator is identified, an active adversary can compromise the aggregator to give permission to false sensor nodes to inject data into the network without verification. Hence we propose distributed model in algorithm 3.1 to reduce the aggregation overhead of the aggregator as well as to eliminate the effect if single cluster head is compromised by the adversary.

In this algorithm, key distribution, aggregation and verification activities are delegated to different nodes. It is further assumed that the key distribution and verifier node has the topology details of the cluster. Initially controller node and key-generator node establish a secret key using Deffie-Hellman Key exchange and key-generator creates a key-generation parameter. It distributes keys to all sensors and releases a commitment on the key-generation parameter. Sensors encrypt the data and distribute fragments among peer for authentication. After aggregation, controller release a commitment on the shared key to be used in verification.

3.2.1 Proof of Security

Theorem 1. *Distributed data aggregation model is secure against Indistinguishable Chosen Plain-text Attack.*

Proof. To prove that proposed model is secure against chosen plain-text attack we propose the following security game.

- A probabilistic polynomial time adversary $\mathcal{A}$ with access to encryption algorithm, chose a pair of message vector m_0 and m_1 of equal length on which he require the encrypted aggregation algorithm to be challenged.
- The challenger flips an unbiased coin and chose with uniform distribution $b \in \{0, 1\}$ and encrypt each message in vector m_b and sends $\{\alpha, \beta, \gamma\}$.
- Adversary chooses a random parameter $r_a \in \mathbb{Z}_n$ such that r_a^{-1} exists in $\mathbb{Z}_n$ and selects $b' \in 0, 1$ uniformly at random.
- If $[\beta H(m_{b'})^{-1}]^{r_a^{-1}} ? = p_i$, p_i is the public key information of the challenger; the adversary returns b'.
- Adversary wins the game if he is correctly able to produce $b = b'$ in t such iterations.
  Given a distinguisher

$$\mathcal{D}(b, b') = \begin{cases} 1 & if\ b' = b \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For a negligible function $\varepsilon(\kappa)$ over security parameter k advantage of adversary can be given as

$$Adv(\mathcal{A}) = Pr[\mathcal{D}(b, b') = 1] = Pr[\beta H(m_{b'})^{-1}]^{r_a^{-1}} = p_i]$$

$$= \frac{1}{2} + Pr[\mathcal{D}(\mathcal{A}(g^{ID_i}, \alpha = g^{ID_i r_a}), g^{r_a}) = 1] \leq \frac{1}{2} + \varepsilon(\kappa)$$

By DDH argument , given α and ID_i of the challenger, Adversary can successfully find a random parameter r_a such that $\alpha = g^{ID_i r_a}$ is negligible. Thus choosing the correct b', r_a such as $[\beta H(m_{b'})^{-1}]^{r_a^{-1}} = p_i$ is also negligible.

□

Theorem 2. *Passive adversary can not add any false data to the encrypted message.*

Algorithm 3.1: Distributed Key Generation-Aggregation-Verification.

Require: $m_1, m_2, \ldots, m_n$ as sensor input and $H(.)$ is a homomorphic trapdoor permutation over public key k, $ID_1, ID_2, \ldots ID_n$ are the unique ids given to each sensor node,multiplicative group $G =< g >$ is in $\mathbb{Z}_n^*$.

Ensure: $M = \prod_{i=1}^{n} m_i$.

Initialization: A controller(aggregator) node, a key generation node and a verifier node is selected.

1. STEP 1: Controller node and Key Generation node establish a shared secret e using secure key exchange mechanism.
2. STEP 2: Key Generation node chose a random $r_a \in \mathbb{Z}_n^*$ and generate the key generation parameter $x = er_a$.
3. STEP 3: Key generation node computes $p_i = g^{ID_i x}$ where $1 \leq i \leq n$ and share the key in one to one message.
4. STEP 4: Release the commitment $\phi = g^{r_a}$ as the session key.

Encryption:

5. STEP 1: j th Sensor Node choose a random $r_j \in \mathbb{Z}_n^*$ and computes $\alpha = \phi^{r_j ID_j}, \beta = p_j^{r_j} H(m_j)$ for k.
6. STEP 2:$i-th$ Sensor node split $H(m_i)$ in k sub messages. $H(m_i) = \sum_{j=1 \wedge j \neq i}^{k} m_i^{(j)}$ for sensor node $1 \leq j \leq n$.
7. STEP 3: Each sensor node distribute k fragments of the message to k neighbors.
8. STEP 4: After receiving k partial messages from peer sensor nodes j compute $m' = \sum_{t=1 \wedge t \neq j}^{k} m_t^{(j)} - H(m_j)$.
9. STEP 5: j-th Sensor node compute for a $z \in G, \gamma = z^{m'} p_j$ and send $\{\alpha, \beta\}$ to the aggregator node and γ to the verifier node.

Aggregation:

10. STEP 1: Aggregator computes $\frac{\prod_{i=1}^{n} \beta_i}{(\prod_{i=1}^{n} \alpha)^e} = H(\prod_{i=1}^{n} m_{(i)})$.
11. STEP 2: Aggregator releases commitment of the shared secret key e as $\delta = \phi^e$ to the verifier node.

Verification:

Verifier verifies the authentication of the message and the authentication of the sender by computing.

$$\prod_{i=1}^{n} \gamma ? = (\delta^{\Sigma_{j=1}^{n} ID_j}) \tag{3}$$

12. If the verification is successful aggregator transmit the data to the base station with its ID.
13. The base station invert the trapdoor permutation using its secret key to get $M = \prod_{i=1}^{n} m_i$.

Proof. We assume a polynomial time adversary $\mathcal{A}$ receives t tuples of $\{\alpha_i, \beta_i, \gamma_i\}$ $1 \leq i \leq t$ from a prover. Adversary computes $\{\alpha_T, \beta_T, \gamma_T\}$ for a message of his choice m_T that satisfies the following aggregation criteria.

$$\frac{\prod_{i=1}^{n} \beta_i * \beta_T}{(\prod_{i=1}^{n} \alpha * \alpha_T)^e} = H(\prod_{i=1}^{t} m_i * m_T) = \prod_{i=1}^{t} H(m_i) * H(m_T).$$

Now prover plays the following security game with the honest aggregator with an adversary $\mathcal{A}$ as a subroutine.

- Honest Key Generator with shared secret key e and secret key r_a , generates public key $p = g^{IDx}$ for the prover with identifier as ID and sends p to prover and broadcast ϕ.
- Prover chose a message vector $m = \{m_0, m_1, \ldots, m_t\}$ and send t-tuples $\{\alpha_i, \beta_i, \gamma_i\}$ such that $1 \leq i \leq t$ to $\mathcal{A}$
- Adversary chose a message m_T and send back $\{\alpha_T, \beta_T, \gamma_T, m_T\}$.
- Prover invoke the **Verification** function with message vector $M = m \cup m_T$. and $t+1$ tuple $\{\alpha_i, \beta_i, \gamma_i\}$ such that $1 \leq i \leq t$ and $\{\alpha_T, \beta_T, \gamma_T\}$.
- If verification is correct then Prover asked adversary for random parameter $r_a \in \mathbf{Z}_p$ and $H(m_T)^{-1}$.
- Prover then computes $[\beta_T H(m_T)^{-1}(g^{ID^{-1}})] = g^x$

Thus if adversary is able to add a new data of his choice to the encrypted message, a polynomial time prover can use the adversary to break discrete log problem. Thus by the hardness assumption of discrete log problem, adversary can not append data of his choice to the encrypted message. □

3.2.2 Security Analysis for Active Adversary

In this security analysis we enhance the capability of the adversary to be able to compromise the controller node or the key distributor node or the verifier node. If the adversary $\mathcal{A}$ compromises the aggregator, he can run the key exchange conversation with the key generator to establish a shared secret key. To generate a public key p_t for an unauthorized sensor node t adversary need to compute $p_t = g^{xt}$ where $g^x = g^{er_a}; r_a \in Z_n$ is the secret of the key generator node and $\phi = g^{r_a}$ is the session key. Assume the polynomial time Adversary (A) receives $e, \phi \in \mathbb{G}$ as input and generate a valid key generation parameter g^x. Given a distinguisher $\mathcal{D}$,

$$Pr[\mathcal{D}(g^x, g^{er_a}) = 1] = Pr[\mathcal{D}(\mathcal{A}(\mathbb{G}, g, \phi, e), g^{er_a}) = 1]$$
$$= Pr[\mathcal{D}(\mathcal{A}(\mathbb{G}, g, g^e, g^{r_a}), g^{er_a})] \leq \varepsilon(\kappa)$$

by Decision Diffie-Hellman assumption where $\varepsilon(\kappa)$ is negligible function defined over security parameter κ.

Thus the probability that a polynomial time adversary can generate an unauthorized key is negligible.

But the adversary can aggregate the data on behalf on the compromised aggregator and also can inject false data in the process as follows.

- Adversary computes $\frac{\prod\beta}{(\prod\alpha)^e} = H(\prod_{i=1}^{n} m_i)$
- Adversary computes for a random $m_T; H(\prod_{i=1}^{n} m_i * m_T) = H(\prod_{i=1}^{n} m_i) * H(m_T)$

If the adversary compromises the key generator node, then he can run the key exchange protocol with aggregator to establish a shared key. Since it will gain access to the secret r_a of the key generator node, it will be able to generate public key for unauthorized node in the sensor network. But the advantage of the adversary is nullified as the unauthorized node will not able to send the data to the base station as the verifier node possess all the IDs of the node authenticated to send data for aggregation.

3.3 Secure Aggregation against Active Attacker

The model of data aggregation described so far allows an active adversary to add a data to the aggregated result, Verification process provide authentication of the sensors nodes and authentication of the message being sent by the sensor but fail to detect alteration of the data by compromised aggregator. And also this verification method fails to identify if the sensor node compute $H(m_i)$ but distribute the component of $H(m\prime)$ among peers thus violating the non repudiation criteria. Therefore we propose modification of the algorithm 3.1 in 3.2 to accommodate the needs to protect against active adversary and malicious sensor node. In this modification, instead of sending the sensor data into pieces, sensor node compute the compliment of the sensor data and split it into k segment. After aggregation, aggregator releases the commitment on the shared secret and the compliment of the aggregated data.

3.3.1 Security Analysis for Active Adversary

Now we check the probability of an adversary $\mathcal{A}$ to inject the false aggregated data into the network if he is able to compromise aggregator node.

Theorem 3. *Adversary can not add any false data to the encrypted message.*

Proof. We assume a polynomial time adversary $\mathcal{A}$ receives a t tuple of $\{\alpha_i, \beta_i, \gamma_i\}$ $1 \leq i \leq t$ from a prover. $\mathcal{A}$ can request the random oracle to get aggregation

Algorithm 3.2: Secure Distributed Aggregation.

Require: $m_1, m_2, \ldots, m_n$ as sensor input and $H(.)$ is a homomorphic trapdoor permutation over public key k, $ID_1, ID_2, \ldots ID_n$ are the unique identifiers given to each sensor nodes, $G =< g >$ is in $\mathbb{Z}_n^*$

Ensure: $M = \prod_{i=1}^{n} m_i$

Initialization: A controller node, a key generation node and a verifier node is selected.

1. STEP 1: Controller node and Key Generation node establish a shared secret e using secure key exchange mechanism.
2. STEP 2: Key Generation node choses a random $r_a \in Z_n^*$ and generate the key generation parameter $x = er_a$
3. STEP 3: For each sensor node, key generation node computes $p_i = g^{ID_i x}$ where $1 \leq i \leq n$ and share the key in one to one message.
4. STEP 4: Release the commitment $\phi = g^{r_a}$ as the session key.

Encryption:

5. STEP 1: j th Sensor Node choose a random $r \in \mathbb{Z}_n^*$ and computes $\alpha = \phi^{r_j ID_j}, \beta = p_j^{r_j} H(m_j)$.
6. STEP 2: $i-th$ Sensor node compute complement of $H(m_i)$ as $[H(m_i)]^c$ and split in k sub messages. $[H(m_i)]^c = \sum_{j=1}^{k} m_i^{(j)}$ for sensor node $1 \leq j \leq k$
7. STEP 3: Each sensor node distribute k fragments of the message to k neighbours.
8. STEP 4: After receiving k partial messages from peers, sensor node j computes $m' = \sum_{t=1, t \neq j}^{k} m_t^{(j)}$
9. STEP 5: j th Sensor node computes for a $z \in G, \gamma = z^{m'} p_j$ and send $\{\alpha, \beta\}$ to the aggregator node and γ to the verifier node.

Aggregation:

10. STEP 1: Aggregator computes $\frac{\prod_{i=1}^{n} \beta_i}{(\prod_{i=1}^{n} \alpha)^e} = H(\prod_{i=1}^{n} m_i) = M$.
11. STEP 2: Aggregator compute commitment of the shared secret key e as $\delta = \phi^e$ and aggregated message as $C = z^{[M]^c}$ and release $(\delta, C)$to the verifier node.

Verification:

Verifier verifies the authentication of the message and the authentication of the sender by computing.

$$C? = \frac{\prod_{i=1}^{n} \gamma}{\delta^{\sum_{j=1}^{n} ID_j}} \tag{4}$$

12. If the verification is successful aggregator append its ID and send C data to the base station.
13. The base station verifies $C? = z^{[M]^c}$ and invert the trapdoor permutation using its secret key to get $M = \prod_{i=1}^{n} m_i$.

$M = H(\prod_{i=0}^{t} m_i) = \frac{\prod_{i=1}^{n} \beta_i}{(\prod_{i=1}^{n} \alpha)^e}$ and release of the complement of the aggregation as $C = \frac{\prod_{i=1}^{t} \gamma}{\delta^{\sum_{j=1}^{t} ID_j}}$ for the set of data. Adversary computes $\{\alpha_T, \beta_T, \gamma_T\}$ for a message of his choice m_T such that it satisfies the aggregation and verification criteria.

$$\frac{\prod_{i=1}^{n} \beta_i * \beta_T}{(\prod_{i=1}^{n} \alpha * \alpha_T)^e} = H(\prod_{i=1}^{t} m_i * m_T)$$

$$z^{[M*H(m_T))]^c} = \frac{\prod_{i=1}^{t} \gamma * \gamma_T}{\delta^{\sum_{j=1}^{t} ID_j + ID_T}} = C\frac{\gamma_T}{\delta^{ID_T}} \quad (5)$$

Now prover plays the following security game with the honest aggregator with an adversary $\mathcal{A}$ as a subroutine.

- Honest key generator with secret key $x = er_a$, generates public key $p = g^{ID_T x}$ for the prover with identifiable information as ID_T. And sends p to prover.
- Prover chose a message vector $m = \{m_0, m_1, \ldots, m_t\}$ and send t-tuples $\{\alpha_i, \beta_i, \gamma_i\}$ such that $1 \leq i \leq t$ to adversary.
- Adversary chose a message m_T and send back $\{\alpha_T, \beta_T, \gamma_T, m_T\}$.
- Prover invoke the **Verification** function with message vector $M = m \cup m_T$. and $t+1$ tuple $\{\alpha_i, \beta_i, \gamma_i\}$ such that $1 \leq i \leq t$ and $\{\alpha_T, \beta_T, \gamma_T\}$ and $C' = z^{[M*H(m_T))]^c} = C\frac{\gamma_T}{\delta^{ID_T}}$
- If verification is correct then Prover asked adversary for random parameter $r_a \in \mathbf{Z}_p$ and $H(m_T)^{-1}$ and $m'_T = \sum_{i=1}^{k} m_i^{(T)}$.
- Prover then computes

$$e = \frac{1}{r_a ID_T} \log_g \left(\frac{C'\delta^{ID_T}}{C z^{m'_T}}\right) \quad (6)$$

Thus if adversary $\mathcal{A}$ is able to add a new data of his choice to the encrypted message, a polynomial time prover can use the adversary to break a discrete log problem. Thus by the hardness assumption of discrete log problem, adversary can not append data of his choice to the encrypted message. □

4 CONCLUSION

We proposed an algorithm to reduce over all energy consumption of the sensor nodes by distributing the task of aggregation, verification and key generation. Peer verification is essential to eliminate the possibility of aggregator or key generator being compromised. Future scope of study is to prove the efficiency of the algorithm in the presence of cheating verifier and aggregator.

REFERENCES

Castelluccia, C., Chan, A. C., Mykletun, E., and Tsudik, G. (2009). Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(3):20.

Du, W., Deng, J., Han, Y. S., and Varshney, P. K. (2003). A witness-based approach for data fusion assurance in wireless sensor networks. In *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, volume 3, pages 1435–1439. IEEE.

Hayashi, R., Okamoto, T., and Tanaka, K. (2004). An rsa family of trap-door permutations with a common domain and its applications. *Public Key Cryptography–PKC 2004*, pages 291–304.

He, W., Liu, X., Nguyen, H., Nahrstedt, K., and Abdelzaher, T. (2007). Pda: Privacy-preserving data aggregation in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2045–2053. IEEE.

Hu, L. and Evans, D. (2003). Secure aggregation for wireless networks. In *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, pages 384–391. IEEE.

Krishnamachari, B., Estrin, D., and Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 575–578. IEEE.

Li, X., Chen, D., Li, C., and Wang, L. (2015). Secure data aggregation with fully homomorphic encryption in large-scale wireless sensor networks. *Sensors*, 15(7):15952–15973.

Ozdemir, S. and Çam, H. (2010). Integration of false data detection with data aggregation and confidential transmission in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 18(3):736–749.

Ozdemir, S. and Xiao, Y. (2009). Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, 53(12):2022–2037.

Ozdemir, S. and Xiao, Y. (2011). Integrity protecting hierarchical concealed data aggregation for wireless sensor networks. *Computer Networks*, 55(8):1735–1746.

Schmidt-Samoa, K. (2006). A new rabin-type trapdoor permutation equivalent to factoring. *Electronic Notes in Theoretical Computer Science*, 157(3):79–94.

Westhoff, D., Girao, J., and Acharya, M. (2006). Concealed data aggregation for reverse multicast traffic in sensor networks: Encryption, key distribution, and routing adaptation. *Mobile Computing, IEEE Transactions on*, 5(10):1417–1431.

Yang, Y., Wang, X., Zhu, S., and Cao, G. (2008). Sdap: A secure hop-by-hop data aggregation protocol for sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 11(4):18.