ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering and Technology, Madurai, Tamil Nadu, India

A Concealed Data Aggregation Using Privacy Homomorphism in Wireless Sensor Networks

B. Steffi Diana, S. Vijayakumar
TIFAC-CORE in Pervasive Computing Technologies, Velammal Engineering College, chennai, India.
TIFAC-CORE in Pervasive Computing Technologies, Velammal Engineering College, chennai, India.

Abstract**—** Data aggregation is a process in which data is collected from the sensor nodes and sent to the base station via cluster heads. Data aggregation schemes are vulnerable to various security attacks. Such Attacks take place compromising at the sensor node and cluster head levels. An adversary node eavesdrops the sensed data, modifies and injects forged data. Security plays a key role in the data aggregation process and the base station needs to confirm data integrity and authenticity. Data aggregation schemes based on privacy homomorphism (PH) provide better security compared with traditional aggregation. In PH based encryption the cluster heads can directly aggregate the cipher texts without decryption. The proposed System is based on privacy homomorphism encryption and uses elliptic curve Cryptography (algorithm) to encrypt the sensed data. And this system deals with transmitting difference data rather than the raw sensed data from sensor node. Compared to the existing aggregation schemes, the differential data transfer achieves more energy efficiency and increase in the packet delivery ratio.

Keywords**—** Data Aggregation, Privacy homomorphism, Data integrity and authenticity.

I. INTRODUCTION

Wireless sensor networks consist of small nodes that monitor physical and environmental conditions such as temperature, sound, vibration, process the data, and communicate through wireless links [7]. A wireless sensor network consists of a base station or gateway that communicates with a number of wireless sensor nodes by use of a radio link. Once the data is collected by some intermediate node, it is then transmitted to the gateway. Once the data reaches the base-station then it is presented to the system by the gateway connection and processed. As nodes forward the data, they remove node's redundant sensor information, and combine data at intermediate nodes reduces the overall number of messages, thus reducing communication and increasing the lifetime of the network.

Data aggregation mechanisms are proposed to reduce the power consumption of wireless sensor networks. Data aggregation schemes summarize data packets of several sensor nodes so that amount of data transmission is reduced [13]. In data aggregation scheme, a group of sensor nodes collect information from the target region. When the base station request for the data, instead of each node sensing the sensor data the data aggregator, collects the information from the sensor nodes, computes the sum or average, and sends the aggregated data to the base station over a multihop path. Thus data aggregation reduces the number of data transmissions, provides better bandwidth and energy utilization in the network.

As the majority of wireless sensor network applications require a certain level of security, security protocols require sensor nodes to encrypt and authenticate any sensed data prior to its transmission and the data has to be decrypted by the base station. Moreover, data aggregation alters the sensor data and therefore it is a challenging task to provide data encryption and authentication along with the data aggregation.

The objective of the paper is to provide solution for the security attacks taking place compromising sensor node and aggregator level. Data aggregation schemes are vulnerable to various security attacks. Secure communication requires sensor nodes to encrypt any sensed data prior to the transmission and to have end to end security with the data decrypted only at the base station. So the data aggregation and security protocol must be designed together so that data aggregation can be performed without compromising security.

II. RELATED WORKS

Numerous secure data aggregation schemes have been proposed and designed for different security requirements. A number of data aggregation schemes have been proposed based on the privacy homomorphism. In privacy homomorphism based encryption scheme [8] allows direct computation on encrypted data and affords end to end concealment of data.

In secure hierarchical data aggregation algorithm [6] achieves end to end security using an efficient public key cryptosystem. It employs homomorphism encryption and aggregate digital signatures for the end to end confidentiality and data integrity. It does not require a separate integrity verification phase and uses aggregate digital signature to relieve the extra communication cost for the network. In Security-enhanced Energy-efficient Data Aggregation [9] supports hop to hop data authentication. It uses homomorphism public key encryption into Message authentication code to offer an end to end concealment of the data and provides hop to hop verification. This method exploits homomorphism encryption to gain higher security and less computational complexity. On other hand a secured data aggregation protocol [1] attempt to detect false data injection during the data aggregation and thus to find the adversary node. Each sensor node reports the deviation to the base station and authenticates an attestation message with the attached MAC. The base station detects the attack if it receives sufficient number of attestation messages pointing at the attacking node (decision by majority). In concealed data aggregation scheme for multiple applications [7] designed for application specific environment. The base station extracts application-specific data from aggregated cipher text. It reduces the impact of compromising attacks in single application environments and degrades the damage from unauthorized aggregations. The cipher texts from different applications can be encapsulated into one cipher text and the base station can extract application specific plaintexts through the corresponding secret keys. Then PEPPDA [5] technique guarantees the privacy, authenticity and freshness of sensed data. It also promises the accuracy and confidentiality of the aggregated data without introducing a significant overhead on the battery limited sensors.

III. PRELIMINARIES

A. Network Model

A wireless sensor network (WSN) is controlled by a base station (BS). The base station has bandwidth efficiency, good computing capability, adequate memory, and stable power to support the cryptographic and routing requirements of the whole wireless sensor network. Besides the base station, sensors (SNs) which are small and low cost limited on computation, storage, and communication capability. They are deployed to sense and gather responsible results for the base station.

Generally, all sensors in a wireless sensor network may be divided into several clusters. Each cluster consists of number of cluster members. Among them one is selected as cluster head. A cluster-based wireless sensor network has several advantages such as efficient energy management, better scalability of medium access control and routing. The cluster head (CH) is responsible for collecting and aggregating sensing data from sensors within the same cluster. A cluster head then sends the aggregation results to the base station. In a homogeneous wireless sensor network, cluster heads act as normal sensors.

B. Attack Model

1) *Without compromising any Sensor node or Cluster head:* An adversary can only eavesdrop on packets in the air, so the attacker can inject the forged messages with this public information [2].
2) *Compromising Sensor node:* After compromising a sensor node, an adversary can acquire secrets such as encryption/ decryption keys. Then, an adversary can acquire sensing data and packets passed through the captured sensor node or impersonate this compromised sensor to forge malicious data.
3) *Compromising Cluster head:* After compromising a cluster head, an adversary can obtain the secrets and the adversary can decrypt the cipher text of sensing data sent by its cluster members and generate forged aggregation results.

IV. FRAMEWORK

The proposed protocol executes for two sessions, reference data transfer session and subsequent data transfer session. The reference data transfer session transfers the reference data from sensor nodes to Base station and performs verification of it in the Base station [4].

In the subsequent data transfer session, the differential data is calculated between the reference data and sensing data. It is transferred from sensor nodes to Base station so that base station can recover the individual sensed data and can check the integrity all sensing data.

The proposed approach provides more energy efficiency by transmitting and aggregating the differential data instead of raw data. The original data is recovered from the differential data is by converting the differential data to raw data using reference data. The reduced amount of data transfer from the nodes conserves energy of the sensor nodes as well as cluster head because the cluster head wants to process small amounts of data. The framework of protocol is shown in fig. 1. It consists of five phases such as Setup phase, Initial phase, Aggregate phase, Recovery phase, and Verify phase.

The first phase happens only once and it is common to both the sessions. The further four phases execute first for the reference data transfer session until a successful reference data for all nodes reaches at base station and then it repeats for subsequent data transfer sessions until a false aggregated result reach at the base station. When base station found that the aggregation result reached at the base station is false, it concludes that the data is changed during transmission or reference data kept in the node changed by because of node compromise attack.

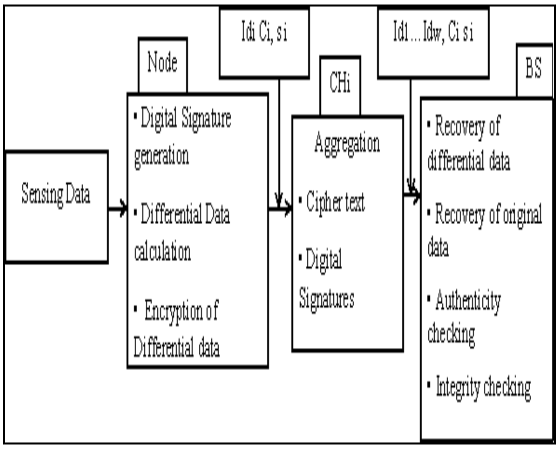

Node
Idi Ci, si
CHi
Id1… Idw, Ci si
BS
Sensing Data
• Digital Signature generation
• Differential Data calculation
• Encryption of Differential data
Aggregation
• Cipher text
• Digital Signatures
• Recovery of differential data
• Recovery of original data
• Authenticity checking
• Integrity checking

Fig. 1 Framework of the protocol

To demonstrate the proposed protocol, we consider cluster 1 (in fig. 2), here SNw is chosen as CH of cluster 1 and the remaining sensor nodes (SN1…SNw-1) is considered as cluster members. Both cluster head and cluster members perform sensing and the CH's performs data aggregation.

In the setup phase, the Base station uses this phase to generate and install all necessary secrets for each sensor node and for them also

Base Station generates public key, private key for every sensor nodes ($SN_1 \ldots . SN_w$).

- Base station randomly selects private key PR_{SNi} from Z_p.
- Base station generates public key $PU_{SNi} \in$ G2 by PU_{SNi}

$$PU_{SNi} = PR_{SNi} \times g2, \qquad (1)$$

Where g2 is the generator of G2.

Base Station generates public key and private key for them.

- Base station constructs an elliptic curve E over a finite field F_p satisfying that p is a prime number and the $E(F_p)$ has a large prime factor.
- Base station randomly selects private key PRBS from F_p.
- Base station generates public key $\eta = (Y, \varepsilon, p, G, n)$

$$Y = PR_{BS} \times G \qquad (2)$$

Where G is a generator point on the elliptic curve.

Base Station loads public key of Base Station (PUBS), private key of sensor node (PRSNi), hash function to every sensor node.

Base Station keeps public key of every sensor node as corresponds to the node ID, public key of Base Station, hash function in them.

In the initial phase, the sensor node executes this phase for transferring cipher text. The encrypted text is appended to the node id with the digital signature as a pair to Cluster head.

In aggregate phase, the Cluster head (CHi) uses this phase to aggregate the cipher texts, signatures coming from its cluster members and to generate aggregated cipher text and aggregated signature without performing any decryptions at the cluster head (CH_i).This is achieved because of additive Privacy Homomorphism property of the encryption algorithm. So this phase saves the energy spend for the decryptions and encryption at the aggregator node and mitigate the aggregator node compromise. Thus, the system provides an end to end concealment to the data.

In the recovery phase, the base station recovers each individual sensing data from the aggregated cipher text reached at the Base station. Thereby Base station overcomes the limitations of it on the aggregation functions and can perform any operations on the recovered data after verification.

In the verify phase, the Base station checks the integrity and authenticity of all sensing data in this phase. So that Base station can validate the correctness and the source of the sensing data. The scenario of proposed system is shown in fig. 2.

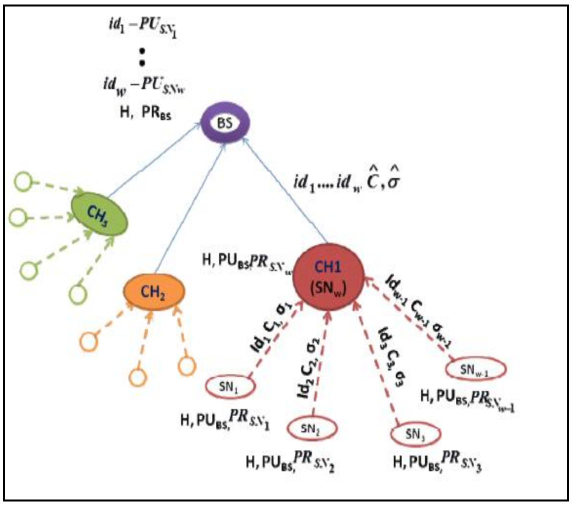

$id_1 - PU_{SN_1}$
⋮
$id_w - PU_{SNw}$
H, PR_{BS}
BS
$id_1 \ldots . id_w \; \hat{C}, \hat{\sigma}$
CH_3
CH_2
H, PU_{BS}, PR_{SN_w}
CH1 (SN_w)
$Id_1 C_1, \sigma_1$
$Id_2 C_2, \sigma_2$
$Id_3 C_3, \sigma_3$
$Id_{w-1} C_{w-1} \sigma_{w-1}$
SN_1
SN_2
SN_3
SN_{w-1}
H, PU_{BS}, PR_{SN_1}
H, PU_{BS}, PR_{SN_2}
H, PU_{BS}, PR_{SN_3}
H, PU_{BS}, $PR_{SN_{w-1}}$

Fig. 2 Scenario of proposed system

A. Reference and Subsequent Data Transfer Session

The protocol executes the reference data transfer session and then the subsequent data transfer session. The reference session transfers the sensing data value to base station and this sensing data is verified in the Base station. The Base station keep the reference data as corresponds to the node ID for recovery of original sensed data in differential data transmission session.

Then the following in the subsequent data transfer session difference data is calculated between raw sensor measurements and reference data and sent to the base station. Working procedure is explained below.

1) The Initial phase:

a. For reference session sensor nodes (SN1….SNw) take the raw measurement value as reference data and keep in them.

$$Rf_i = d_i \qquad (3)$$

And in the subsequent session sensor nodes (SN1….SNw) compute the differential data by

$$dd_i = |d_i - Rf_i|$$

Where d_i is the original sensed data of node i,
Rf_i is the reference data of node i and
dd_i is the differential sensed data of node i.

- If it is negative, $dd_i = d_i - Rf_i$ and append a negative sign to its node ID.

- If it is zero, the corresponding sensor node (SNi) does not go for further steps.

b. Encoding It generally transforms data into another format using a scheme that is publicly available. Here we consider the binary encoding scheme. Here sensor nodes ($SN_1 \ldots . SN_w$) encode the sensed data set as the reference data into binary format and append β number of zeros to it.

$$\text{Encoded} \quad \text{data}(m_i) = d_i || 0^{\beta}, \quad \text{Where } \beta = l.(i-1) \quad (5)$$

l is the number of bits required to represent all sensing data.

d_i is the binary value corresponding to the sensed value.

c. Mapping This is based on map() function. It maps values (plain text) into points on the elliptic curve (mapped message). Here sensor nodes map the encoded message into the elliptic curve.

$$\text{Mapped message} \quad (M_i) = map(m_i) = m_i . G \quad (6)$$

d. Encryption Sensor nodes encrypt the mapped message corresponds to the differential data using the Base station public key.

- Randomly select k_i from 0 to n-1, where n Є PU_{BS}.
- Sensor nodes generate cipher text of sensed data.

$$\text{cipher} \quad \text{text}(C_i) = (r_i, s_i) = (k_i \times G, M_i + ! \quad (7)$$

e. Sensor nodes compute the digital signature for their sensed data using its private key.

- Compute hi=H(d_i), where hi Є G1
- Compute signature

$$\sigma_i = PR_{SN_i} \times h_i \qquad (8)$$

Every sensor node appends its cipher text and signature to its ID's and sends to its Cluster head.

2) *The Aggregate phase:*

In the aggregate phase the cluster head aggregates cipher text and send to Base station.

Aggregated Cipher text $(\hat{C}_1) = (\hat{R}_1, \hat{S}_1) = \sum_{i=1}^{w} \mathrm{I}$

Aggregated signature $(\hat{\sigma}) = \sum_{i=}^{w}$ (10)

Cluster head appends the aggregated cipher text and signature to it node ID's and passes it to Base Station.

3) *The Recovery phase :*

In the recovery phase base station recovers $d_1 \ldots . . d_i$ from the aggregated cipher text.

a. Decryption Base station generates M' by decrypting the cipher text with PR_{BS}.

$$\hat{C}_1 = (\hat{R}_1, \hat{S}_1) \quad \text{and} \quad M' = -PR_{BS} \times \hat{R}_1 + \hat{S}_1 \qquad (11)$$

b. Remapping Remapping using rmap() function to remap elliptic curve points into plain text.

Remapped data $m' = rmap(M^{+}) =$

c. Decoding Base station recovers the sensing data.

$$d_i = m'[l.(i-1),(l.i)-1]$$

4) *The Verify phase*

a. The IDi of each recovered data at the Base Station is checked with the ID's stored in Base Station.

b. If it matches, Base Station confirms that the data is from a valid node (ID).

c. Integrity of each recovered data is checked by Base Station.

$$e_n(\hat{\sigma}, g_2) = \prod_{i=1}^{w} e_n(h_i, PU_{SN_i}) \qquad (14)$$

$$h_i = H(d_i) \qquad (15)$$

- If the equation holds, the recovered data d_i is accepted and the base station keeps the reference data to the corresponding node ID. Acknowledgement (ACK) is sent to the sensor nodes through Cluster head. Then the nodes perform subsequent data transfer session in next iteration calculating the difference data. If the equation holds, the recovered data in subsequent data transfer d_i is accepted. Otherwise, d_i is rejected. Base station concludes that the data integrity is lost because of the modification occurred during the transmission or to the reference data kept in the node due to node compromise attack.
- Otherwise, Base Station sends negative acknowledgement (NACK) to sensor nodes through Cluster Head. The sensor node performs the above phases again until the reference data and subsequent data successfully reaches the Base Station. If sensor nodes do not receive any ACK or NACK, it repeats this phase again.

B. Analytical Approach

Elliptic curve is constructed over field F_{29} with p=29, G=(5, 7), n=31

$$Y = PR_{BS} \times G = 85$$

$$E = y^2 - x^3 - x + 16 \bmod 29$$

- Select the base station private key (PI.
- Select the base station public key (PU_{BS}) :

Raw measurement sensed by sensor node is taken as d_i. Sensing data as d_1=8, d_2=9, d_3=15, d_4=18, d_5=25.First during Reference data transfer session, the reference data, Rf_i=d_i. Consider reference data corresponds to the nodes $SN_1 \ldots . SN_5$ areRf_1=10, Rf_2=8, Rf_3=11, Rf_4=15, Rf_5=20

Differential data is calculated using the equation (4) as shown below.

$$dd_1 = d_1 - Rf_1 = |-2| = 2$$

$$dd_2 = d_2 - Rf_2 = 9 - 8 = 1$$

$$dd_3 = d_3 - Rf_3 =$$

$$dd_4 = d_4 - Rf_4 = 18 - 15 = 3$$

$$dd_5 = d_5 - Rf_5 = 25 - 20 = 5$$

Consider l=3, because 3 bit is sufficient to represent all difference data.

Encoding of data is calculated using the equation (5) as shown below.

$$\beta_1 = l.(i-1) = 0, m_1 = dd_1 0^{\beta_1} = 010$$

$$\beta_2 = l.(i-1) = 3.(2-1) = 3, m_2 = dd_2 0^{\beta_2} = 001000$$

$$\beta_3 = l.(i-1) = 3.(3-1) = 6, m_3 = dd_3 0^{\beta_3} = 100000000$$

$$\beta_4 = l.(i-1) = 3.(4-1) = 9, m_4 = dd_4 0^{\beta_4} = 011000000000$$

$$\beta_5 = l.(i-1) = 3.(5-1) = 12, m_5 = dd_5 0^{\beta_5}$$
$$= 101000000000000$$

Mapping of data is calculated using the equation (6) as shown below.

M1 = map(m1) = m1.G=010 X 5 =050

M2 = m2.G=005000

M3 = m3.G=500000000

M4 = m4.G=055000000000

M5 = m5.G=505000000000000

Encryption of data is calculated using the equation (7) as shown below. Consider k_1=3, k_2=4, k_3=5, k_4=2, k_5=6

$$C_1 = (r_1, s_1) = (k_1.G, M_1 + k_1 \times Y)$$

= (3 5,010)=(15,265)

$$C_2 = (r_2, s_2) = (k_2.G, M_2 + K_2 \times Y)$$

= (4 5, 5000+4 85)=(20,5340)

$$C_3 = (r_3, s_3) = (k_3.G, M_3 + k_3 \times Y)$$

=

$$C_4 = (r_4, s_4) = (k_4.G, M_4 + k_4 \times Y)$$

= (1

$$C_5 = (r_5, s_5) = (k_5.G, M_5 + k_5 \times Y)$$

= (30.5

Cipher text is aggregated using the equation (9) as shown below.

$$(\hat{C}_1) = (\hat{R}_1, \hat{S}_1) = \sum_{i=1}^{5} r_i, \sum_{i=1}^{5} S_i$$
$$= (100,5055500006710)$$

Cipher text is decrypted using the equation (11) as shown below.

$$M' = -PR_{BS} \times \hat{R}_1 + \hat{S}_1 = -1700 + 50505550000671 0$$

= 505055500005010

Data is remapped using the equation (12) as shown below

$$m' = \frac{M'}{G} = 101011100001010$$

Decoding of the data is done using the equation (13) as shown below.

$$d_1 = m'[i.(i-1), (i.l) - 1] = [0,2] = 010 = 2$$

$$d_2 = m'[3.(2-1), (3.2) - 1] = [3,5] = 001 = 1$$

$$d_3 = m'[3.(3-1), (3.3) - 1] = [6,8] = 100 = 4$$

$$d_4 = m'[3.(4-1), (3.4) - 1] = [9,11] = 011 = 3$$

$$d_5 = m'[3.(5-1), (3.5) - 1] = [12,14] = 101 = 5$$

Finally decoded data is verified at base station using the equation (14) and (15).

V. SIMULATION WORKS

We used a network simulator called Cooja in Contiki operating system platform. Cooja simulates networks of Contiki nodes such as emulated nodes, cooja nodes and java nodes. Contiki [3] is an open source, portable, light weight, multi-tasking operating system for memory constraint networked embedded systems and wireless sensor networks. Contiki is designed for embedded systems and microcontrollers with small amounts of memory for wireless sensor nodes.

We selected Micaz mote for wireless sensor nodes and programmed using MIB520B programming board and sample program is uploaded into micaz mote.

A. Porting Contiki code to Micaz mote

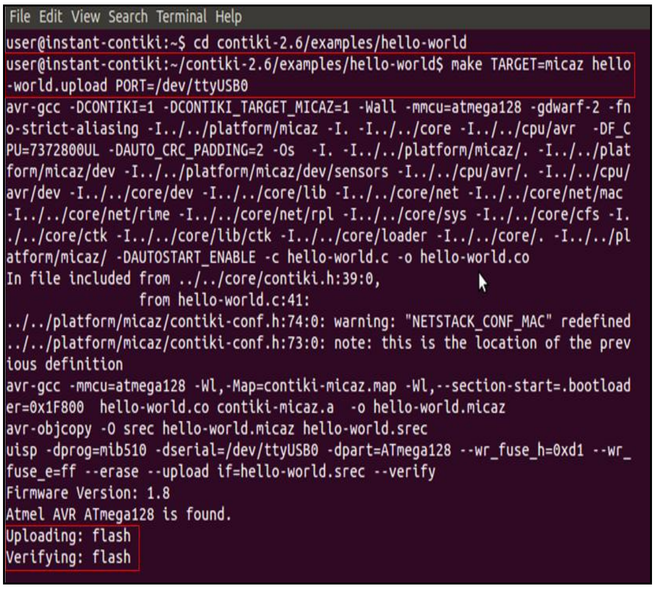
The serial connection to the ports USB0 and USB1 must be enabled for uploading the code. The following commands are given to enable the ports.

```
sudo chmod 777 /dev/ttyUSB0
sudo chmod 777 /dev/ttyUSB1
```

Then the UISP is downloaded and installed. The following command is given to upload the sample code to Micaz mote which is shown in Fig. 3

Make TARGET = micaz hello-world.uplod PORT =/dev/ttyUSB0

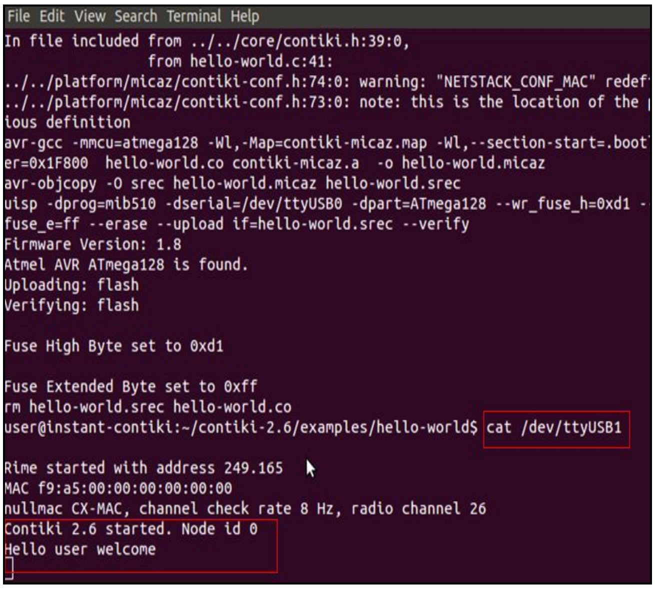
And the code being uploaded in Micaz is shown in Fig. 3



```
user@instant-contiki:~$ cd contiki-2.6/examples/hello-world
user@instant-contiki:~/contiki-2.6/examples/hello-world$ make TARGET=micaz hello
-world.upload PORT=/dev/ttyUSB0
avr-gcc -DCONTIKI=1 -DCONTIKI_TARGET_MICAZ=1 -Wall -mmcu=atmega128 -gdwarf-2 -fn
o-strict-aliasing -I../../platform/micaz -I. -I../../core -I../../cpu/avr  -DF_C
PU=7372800UL -DAUTO_CRC_PADDING=2 -Os  -I. -I../../platform/micaz/. -I../../plat
form/micaz/dev -I../../platform/micaz/dev/sensors -I../../cpu/avr/. -I../../cpu/
avr/dev -I../../core/dev -I../../core/lib -I../../core/net -I../../core/net/mac
-I../../core/net/rime -I../../core/net/rpl -I../../core/sys -I../../core/cfs -I.
./../core/ctk -I../../core/lib/ctk -I../../core/loader -I../../core/. -I../../pl
atform/micaz/ -DAUTOSTART_ENABLE -c hello-world.c -o hello-world.co
In file included from ../../core/contiki.h:39:0,
                 from hello-world.c:41:
../../platform/micaz/contiki-conf.h:74:0: warning: "NETSTACK_CONF_MAC" redefined
../../platform/micaz/contiki-conf.h:73:0: note: this is the location of the prev
ious definition
avr-gcc -mmcu=atmega128 -Wl,-Map=contiki-micaz.map -Wl,--section-start=.bootload
er=0x1F800  hello-world.co contiki-micaz.a  -o hello-world.micaz
avr-objcopy -O srec hello-world.micaz hello-world.srec
uisp -dprog=mib510 -dserial=/dev/ttyUSB0 -dpart=ATmega128 --wr_fuse_h=0xd1 --wr_
fuse_e=ff --erase --upload if=hello-world.srec --verify
Firmware Version: 1.8
Atmel AVR ATmega128 is found.
Uploading: flash
Verifying: flash
```

Fig. 3 uploading a sample program to micaz mote

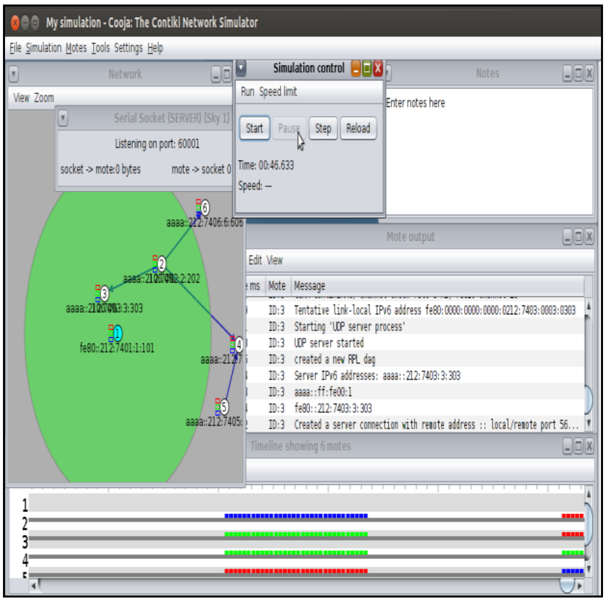
The output of the sample program uploaded in Micaz mote is viewed in the terminal window shown in Fig. 4



```
In file included from ../../core/contiki.h:39:0,
                 from hello-world.c:41:
../../platform/micaz/contiki-conf.h:74:0: warning: "NETSTACK_CONF_MAC" redef
../../platform/micaz/contiki-conf.h:73:0: note: this is the location of the
ious definition
avr-gcc -mmcu=atmega128 -Wl,-Map=contiki-micaz.map -Wl,--section-start=.boot
er=0x1F800  hello-world.co contiki-micaz.a  -o hello-world.micaz
avr-objcopy -O srec hello-world.micaz hello-world.srec
uisp -dprog=mib510 -dserial=/dev/ttyUSB0 -dpart=ATmega128 --wr_fuse_h=0xd1 -
fuse_e=ff --erase --upload if=hello-world.srec --verify
Firmware Version: 1.8
Atmel AVR ATmega128 is found.
Uploading: flash
Verifying: flash

Fuse High Byte set to 0xd1

Fuse Extended Byte set to 0xff
rm hello-world.srec hello-world.co
user@instant-contiki:~/contiki-2.6/examples/hello-world$ cat /dev/ttyUSB1

Rime started with address 249.165
MAC f9:a5:00:00:00:00:00:00
nullmac CX-MAC, channel check rate 8 Hz, radio channel 26
Contiki 2.6 started. Node id 0
Hello user welcome
```

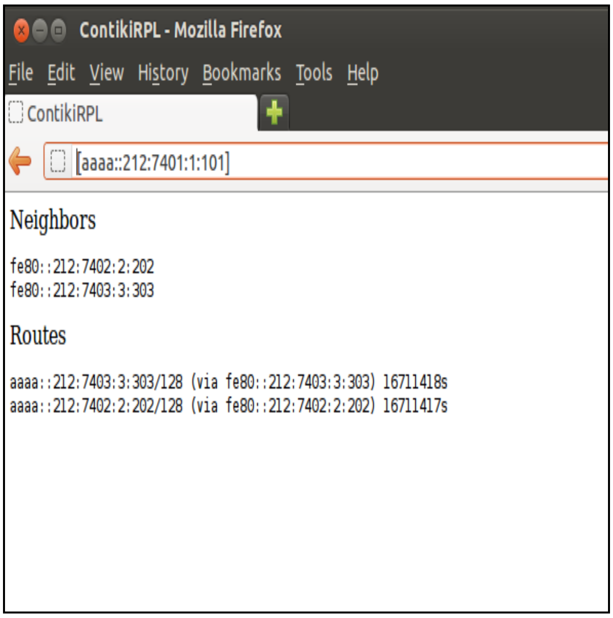
Fig. 3 Output in terminal window

A simple network scenario in cooja simulator is shown in Fig. 5. The routes discovered were displayed in the web interface which is shown in Fig. 6



```
ms  Mote  Message
    ID:3  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7403:0003:0303
    ID:3  Starting 'UDP server process'
    ID:3  UDP server started
    ID:3  created a new RPL dag
    ID:3  Server IPv6 addresses: aaaa::212:7403:3:303
    ID:3  aaaa::ff:fe00:1
    ID:3  fe80::212:7403:3:303
    ID:3  Created a server connection with remote address :: local/remote port 56...
```

Fig. 5 Sample communications between nodes in simple network


ContikiRPL - Mozilla Firefox
File Edit View History Bookmarks Tools Help
ContikiRPL
[aaaa::212:7401:1:101]

Neighbors

```
fe80::212:7402:2:202
fe80::212:7403:3:303
```

Routes

```
aaaa::212:7403:3:303/128 (via fe80::212:7403:3:303) 16711418s
aaaa::212:7402:2:202/128 (via fe80::212:7402:2:202) 16711417s
```

Fig. 6 Network statuses in web interface

IV. CONCLUSIONS AND FUTURE WORK

Thus the proposed technique recovers sensing data from the aggregated result and verifies data integrity and authenticity. It improves the network lifetime by achieving energy and bandwidth efficiency while preserving security requirements such as confidentiality, data authentication, and integrity. It reduces the energy consumption by transferring differential data rather than raw data from nodes so that the cluster head also wants to process small amounts of bits. Thereby it gives energy relief to sensor nodes as well as the cluster head.

Our future works aims at integrating the proposed technique with the network layer security primitive ContikiSec that uses symmetric key encryption algorithms that are available in the Contiki operating system.

REFERENCES

[1] Baratchi, M., &Jamshidi, K. (2011, February). *A distributed verification scheme for encrypted data aggregation in wireless sensor networks.* In Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on (pp. 210-215). IEEE.

[2] Boonsongsrikul, A., Lhee, K. S., & Hong, M. (2010, February). *Securing data aggregation against false data injection in wireless sensor networks.* In Advanced Communication Technology (ICACT), 2010 The 12th International Conference on (Vol. 1, pp. 29-34). IEEE.

[3] Chen, C. M., Lin, Y. H., Lin, Y. C., & Sun, H. M. (2012). *RCDA: recoverable concealed data aggregation for data integrity in wireless sensor networks.* Parallel and Distributed Systems, IEEE Transactions on, 23(4), 727-734.

[4] Dunkels, A., Gronvall, B., & Voigt, T. (2004, November).*Contiki-a lightweight and flexible operating system for tiny networked sensors*.In Local Computer Networks, 2004. 29th Annual IEEE International Conference on (pp. 455-462). IEEE.

[5] Dunkels, A., &Osterlind, F. (2008). *Contiki Programming Course: Hands-On Session Notes.*

[6] Dunkels, A., & Schmidt, O. (2005). *Protothreads-lightweight stackless threads in c. SICS Research Report*.

[7] Jose, J., Kumar, M., & Jose, J. *Energy Efficient Recoverable Concealed Data Aggregation in Wireless Sensor Networks*.

[8] Jose, J., Princy, M., & Jose, J. *PEPPDA: Power Efficient Privacy Preserving Data Aggregation for Wireless Sensor Networks.*

[9] Karl, H., &Willig, A. (2007). *Protocols and architectures for wireless sensor networks.Wiley.com.*

[10] Kumar, V., &Madria, S. K. (2012, July). *Secure Hierarchical Data Aggregation in Wireless Sensor Networks: Performance Evaluation and Analysis*. In Mobile Data Management (MDM), 2012 IEEE 13th International Conference on (pp. 196-201).IEEE.

[11] Lin, Y., Chang, S., & Sun, H. (2012). *CDAMA: Concealed Data Aggregation Scheme for Multiple Applications in Wireless Sensor Networks.*

[12] Ozdemir, S. (2007, July). *Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism.* In Pervasive Services, IEEE International Conference on (pp. 165-168).IEEE.

[13] Sohraby, K., Minoli, D., &Znati, T. (2007). *Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons.*

[14] Quy, N. X., Kyung, M., & Min, D. (2008, September). *Security-enhanced energy-efficient data aggregation for cluster-based wireless sensor networks.*In Internet, 2008.ICI 2008. 4th IEEE/IFIP International Conference on (pp. 1-5). IEEE.

[15] Zhao, F., &Guibas, L. J. (2004). *Wireless sensor networks: an information processing approach. Morgan Kaufmann.*