IDEA / APPROACH DETAILS

Ministry Category: Department of Space (ISRO)

Problem Statement: Geofencing of data on NavIC/IRNSS signals

Team Leader Name: M.Benhur Rodriguez

Problem Code: #ISR12

College Code: 2115

IDEA/SOLUTION/PROTOTYPE

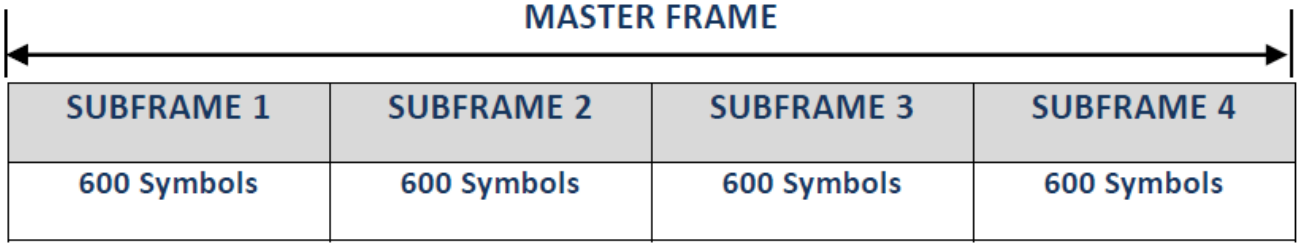
The solution's functioning levels are divided into two phases:

1) Generation of data's to be processed and Uplink.
2) Data Receiving and processing at receiver end.

INTRODUCTION:

- Since the IRNSS satellites operate on L5 band of 1176.45 MHZ and S band of 2492.028 MHZ and the data formats of stream data's from GSAT satellites follows the given below format, the challenge arises in utilizing the data formats into effectively well structured data format schemes. The given text messages should be properly divided into fragments of data's and these fragments should also be properly encrypted using proper encryption protocols and these chunks of data's should be properly queued and uplinked to satellite. The real problem is that the data's from the satellites should be properly Geo-Fenced at the receiver end such that the corresponding data's should only be received and shown to the user only He/She(user) is inside the given geofenced boundaries.

DIAGRAMS OF PRESENT FRAMES:



1	9	26	27	28	30	31	263	287
TLM	TOWC	ALERT	AUTONAV	SUBFRAME ID	SPARE	DATA	CRC	Tail
8 BITS	17BITS	1 BIT	1 BIT	2 BIT	1 BIT	232 BITS	24BITS	6BITS

1	9	26	27	28	30	31	37	257	263	287
TLM	TOWC	ALERT	AUTONAV	SUBFRAME ID	SPARE	MESSAGE ID	DATA	PRN ID	CRC	Tail
8 BITS	17BITS	1 BIT	1 BIT	2 BIT	1 BIT	6 BITS	220 BITS	6	24 BITS	6 BITS

- As defined by ISRO's guide to IRNSS, the sub frames 1&2 is being used for transferring the location and time for the purpose of trilateration at receiver end and 3&4 can be used for transfer of text information in case of disaster warning and LBS.
- When the sub frame 3&4's MESSAGE ID equals to eighteen (18) then the DATA block of 220 BITS follows an protocol to send text messages. That is

Parameter	Size (bits)
Text ID	4
Block count	8
Block ID	8
Text data (25 chars of 8 bits each)	200
PRN ID	6

Bit Index	37-40	41-48	49-56	57-256	257-262
Parameter	Text ID	Block count	Block ID	Text data (25 chars of 8 bits each)	PRN ID
Size (bits)	4	8	8	200	6

- Since the service of IRNSS is divided UN-Encrypted for general public use and Encrypted for Military & other authorized use , the message scheming of our idea make uses of SPARE BIT in Sub Frame's 3&4's body. If the SPARE BIT is set then the received data is in encrypted mode and reset for un-encrypted mode.

I-Generation of data's to be processed and uplink:

I-(a)-Mode-1 (un-encrypted for public use):

- The Text Messages to be broadcasted is collected and assigned a local MESSAGE ID's to every other message files.
- Geo Fenced latitude and longitude coordinates are prepared and made into a default text file with TEXT ID #00 (or) TEXT ID#(default id).maximum of four boundaries followed by a polygon should be used as geo fenced boundary.
- The Text Data capacity has been limited to 24 chars of 8-bit and an extra 8-bit (Bit 57- 64) (TEXT ID COUNT) is reserved for future use.

ALGORITHM OR STEPS INVOLVED:

1) Allocate and store LAT/LONG coordinates of geo fence and store it in TEXT ID #00 (or) TEXT ID #DEF

DEF-default text id (INT)

2) Set the SPARE BIT as '0'.
3) Get the number of text files to be uploaded and let the number be 'n'.
4) For every text files upto 'n' perform step 5.
5) For every i'th Text message file ,if the length in bytes is greater that message threshold then goto step 7 else goto step 6.

Message threshold = 24 chars of 8-bit each * 255(block count max) * 8 = 48,960 (in bytes)

=48,960 bits / 1024 = 47.8125 Kb (in Kbytes)

6) Set the TEXT ID as value of 'i' and TEXT ID COUNT as 00.Divide the text message file into chunks of data's of 192 bytes each and for each block assign a unique BLOCK ID(Bit 49-56) in ascending order and set the BLOCK COUNT(Bit 41-48) value to total blocks of chunks of data.goto step 9.
7) Divide the text file into further chunks of text files and these files are provided with TEXT ID's of i+1,i+2, so on upto EOC(until the text file is split up until all text files contains only 48,960 bytes of data) and the current TEXT FILE is given TEXT ID of 'i' and TEXT ID COUNT is initialized to BCD value of total number of text files that has been generated during this step. For every text file ID's generated in this step perform step 8.
8) Divide the message file into chunks of data's of 192 bytes each and for each block assign a unique BLOCK ID(Bit 49-56) in ascending order and set the BLOCK COUNT(Bit 41-48) value to total blocks of chunks of data.
9) All the generated files are then uplinked to IRNSS constellation of satellites.
10) End Of Process.

I-(b)-Mode-2 (encrypted for Military/privileged use):

- In this mode of operation an hardware module or an software module capable of decryption and making internet request is needed.
- An SSH(Secure Shell) server is created and initialized at a local ground station and the login for this shell is secretly maintained in-between the privileged users of this mode.
- An combination of RSA Keys(Private and Public keys) to maintain a SSL/TL security are generated and the encryption key is automatically changed in a certain interval of time (say 1 day) for maintaining security and to avoid flaws.
- The Private Key is stored in remote file location in SSH servers.

ALGORITHM OR STEPS INVOLVED:

1) Allocate and store LAT/LONG coordinates of geo fence and store it in TEXT ID #00 (or) TEXT ID #DEF

DEF-default text id (INT)

2) Set the SPARE BIT as '1'.
3) Create RSA keys and store the Private Key in a SSH directory.
4) Get the number of text files to be uploaded and let the number be 'n'.
5) For every text files upto 'n' perform step 5.
6) For every i'th Text message file ,if the length in bytes is greater that message threshold then goto step 7 else goto step 6.

Message threshold = 24 chars of 8-bit each * 255(block count max) * 8 = 48,960 (in bytes)

=48,960 bits / 1024 = 47.8125 Kb (in Kbytes)

7) Set the TEXT ID as value of 'i' and TEXT ID COUNT as 00.Divide the text message file into chunks of data's of 192 bytes each and for each block assign a unique BLOCK ID(Bit 49-56) in ascending order and set the BLOCK COUNT(Bit 41-48) value to total blocks of chunks of data. goto step 9.
8) Divide the text file into further chunks of text files and these files are provided with TEXT ID's of i+1,i+2, so on upto EOC(until the text file is split up until all text files contains only 48,960 bytes of data) and the current TEXT FILE is given TEXT ID of 'i' and TEXT ID COUNT is initialized to BCD value of total number of text files that has been generated during this step. For every text file ID's generated in this step perform step 8.
9) Divide the message file into chunks of data's of 192 bytes each and for each block assign a unique BLOCK ID(Bit 49-56) in ascending order and set the BLOCK COUNT(Bit 41-48) value to total blocks of chunks of data.
10) Then all the text messages corresponding to TEXT ID's are then encrypted using the RSA's Public Key and separately store in a directory.
11) All the generated files are then uplinked to IRNSS constellation of satellites.
12) End Of Process.

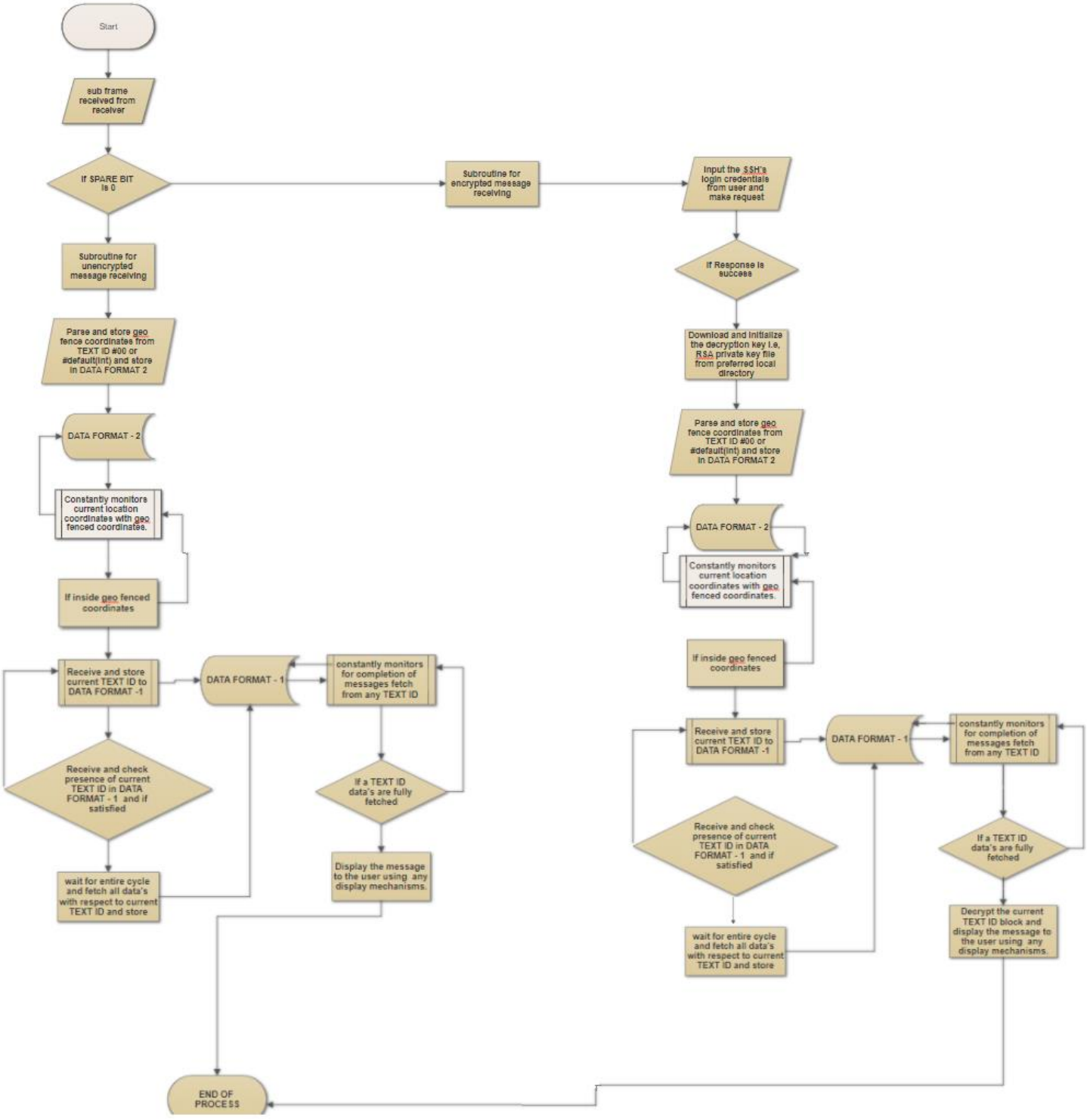
II-Data Receiving and processing at receiver end:

- A software module / hardware module capable of receiving signals at L5 and S Bands could be created and designed in such a way that it could constantly receives location and time data's from IRNSS's satellite clusters and in addition to that it could properly parse the text messages being transmitted.
- For in case of general public usage applications, the terms of the receiver end could be limited to passive data receiver and a parser and some special routines to display text messages if transmitted.
- But for in case of privileged and military purposes the receiver end should be capable of making a internet request , routines to decrypt messages , protocols to perfectly parse the incoming streamed signals should be developed. This may be a software module running on some system platform or it could be a hardware module created using microcontrollers specifically designed for this purpose alone.

ALGORITHM/STEPS USED:

a. Start of process
b. The incoming signals are read from receiver and data's in binary are stored in a local memory.
c. The sub frame data's of 1 & 2 are passed into a special subroutine where in the location values are calculated and mapped in display module containing maps.
d. The sub frame data's of 3 & 4 are then continuously monitored for a match where MESSAGE ID equals to 18 in bit levels of 31-36. If a match is found then goto step (e) else goto step (d).
e. Now, the SPARE BIT of corresponding sub frame is checked and verified to 'o' (Zero).
f. Now, the bit levels in range 37-262 is then separated and stored in local memory and the further parsing and storing process follows the data format specified below.
g. The data from TEXT ID #00 or #(int) default which containing the geo fenced coordinates will follow a special data format of DATA FORMAT -2.
h. Then all fields from all bit levels are then parsed and stored locally using data format -1.
i. Initially with the received TEXT ID a dictionary is created using key as TEXT ID and then all the fields are then parsed and stored according to the format specified.
j. Processes are executed in order of flow chart.
k. End

FLOW CHART OF PROCESS:



LINK : https://drive.google.com/open?id=1DKZ9xwprWlQQSbQWOF5o0H2tKtUzZCEb

DATA FORMAT - 1:

Dictionary(

TEXT ID (integer) : Dictionary(

“BLOCK COUNT”: (integer),

“BLOCK ID” : list() and append to list(),

“TEXT DATA”: Dictionary(

‘ID_1’: String of data for BLOCK ID 1,

‘ID_2’: String of data for BLOCK ID 2,

‘ID_n’: String of data for BLOCK ID n

),

“is_extendible”: Boolean,

“TEXT ID COUNT”: (integer),

“COMPLETED TEXT ID”: list() and append to list(),

“is_complete”: Boolean

)

,

TEXT ID (integer) : Dictionary(

“BLOCK COUNT”: (integer),

“BLOCK ID” : list() and append to list(),

“TEXT DATA”: Dictionary(

‘ID_1’: String of data for BLOCK ID 1,

‘ID_2’: String of data for BLOCK ID 2,

‘ID_n’: String of data for BLOCK ID n

),

“is_extendible”: Boolean,

“TEXT ID COUNT”: (integer),

“COMPLETED TEXT ID”: list() and append to list(),

“is_complete”: Boolean

)

, ……..,

)

DATA FORMAT – 2:

Dictionary(

Geo fence id 1 (int): Dictionary(

“pt_1”:[latitude(double),longitude(double)],

“pt_2”:[latitude(double),longitude(double)],

“pt_3”:[latitude(double),longitude(double)],

“pt_4”:[latitude(double),longitude(double)]

),

Geo fence id ‘n’ (int): Dictionary(

“pt_1”:[latitude(double),longitude(double)],

“pt_2”:[latitude(double),longitude(double)],

“pt_3”:[latitude(double),longitude(double)],

“pt_4”:[latitude(double),longitude(double)]

),

……..,

)

TECHNOLOGY STACK

Front end – Python, QGIS (for Map services and geo fencing coordinates), Tkinter (python-module for GUI frameworks) , etc,

Back end – Python .

USE CASE

- The required text files should be collected and be separated according to public use / private use
- Then the protocols of message structure forming , encrypting , login authentication processes should be performed using the specified protocols above.
- With the help of C&C centers / Indian Master Control Center using the GSAT Uplink stations , the made up file should be uplinked properly.
- The GSAT satellites should transmit the message data’s in events of disaster warnings and other military based applications when a reference signal is set in satellite by control stations.
- The receiver should be a software/hardware depending on the type of application used, should continuously receive , parse , process and display the data’s streamed through the satellites signals.
- The receiver module should be given in terms of public use (un-encrypted) and for the private use (encrypted) for military or for other standard privileged use.
- The module at receiver end constantly checks that the current device is in or out of geo fenced boundaries and particular actions for any options were carried out.

DEPENDENCIES / STOPPER

- This project’s use in case of privileged for military activities and for other activities require an active internet connection for a period of interval to make authentication’s and for file transfer.
- The commonly shared private key for all privileged users should not be misused and in case of any misuse then encrypted data’s in the stream could be possibly read by intruders/hackers. This case could be avoided by keeping the refresh rate for key exchange in app/module to shorter span of time and the keys for encryption could be changed at Command and Control centre in same short span of time.
- The receiver module doesn’t provide any anti-spoofing modules for detection of spamming of location coordinates. This could result in receiving the data’s even without inside the geo fenced boundaries. This can be eliminated by using some hardware based signal authentication schemes by using methods of interference detection and localization,etc.