Fieldbus Protocol For Secured Wireless Sensor Network Communication in Process Automation

Dr.S.Udayakumar* and S.Ananthi**
*National Institute of Technical Teacher's Training and Research (NITTT&R), Chennai
**University of Madras
*Udayakumar_s@yahoo.com
**ananthipradeep84@gmail.com

Abstract: In Modern process Industries, Field bus model is used to transmit control information via the various bus levels to control the process units. Field-bus model makes use of a protocol structure which is similar to the OSI model and which has been customized to support process control systems in a lucid manner. There are the problems of process plant maintenance associated with sudden unexpected sabotage and consequent failure and damage to plant and personnel. When there are three levels of bus based communication – the Factory level, the Cell level and the plant or field level, - the security of the data relating to plant set points, control system parameters and the thresholds etc has to be maintained at all times absolutely. The DES algorithm has been developed and implemented and tested in the field bus data.

Keywords: FieldBus, DES Algorithm, Network Security, Process Automation, DSP, IEEE RS 485

I. NEED FOR SECURITY

Every day, the process automation is changing with a new concept. Industrial field devices are being upgraded from the earlier method of 4-20mA analog communication standard to the new digital method known as the Field Bus. This new technology promises a lot, in terms of reduction of installation costs and also providing the two way communication between the control room and the actual plant in the case of process industries. Any sudden unexpected sabotage consequent failure and damage to plant and personnel. The data can be purposely altered, for e.g. sensor data which will lead to a control signal from the Master which can lead to plant malfunction and subsequent shutdown. Even though the protocol was designed having in mind

the IEEE RS 485 as physical layer, efforts was spent to render the SFPB packets to be carried over others physical layers.

1.1. Simple Field Bus Protocol:

APPLICATION: SFBP partially implements the application layer by the two packet types "Control packet" and "Data packet". In addition it does not defines the Presentation nor the Session layers that are implicitly left to the Application. The Type of packet bits of the Packet Information section are related to the Application layer level, and in addition the NEXT bit of the Packet Information section may play a role also at the Session layer level. Byte ordering is left to the application, it simply defines the BIG ENDIAN order, where the most significant byte is the last one.

TRANSPORT: The ACK packets allow connected packets and mechanism for retransmission on fault of connected packets. The NEXT bit gives a rough mechanism to deal with large messages. However a full implementation of transmission of large messages is left at the application level.

NETWORK: ACK packets and System packets works at Network layer level, in fact they can be used also for routing purposes.

DATA LINK: This layer is covered by packets format and the CSMA/CD similar mechanism and the timing rules.

PHYSICAL: SFBP does not specify the physical layer, even though it was designed on top of a IEEE RS 485, it can run over any other physical layer such as IEEE RS 232, optic, radio and so forth. The comparison of conventional and field bus in related to security is given in figure 1.

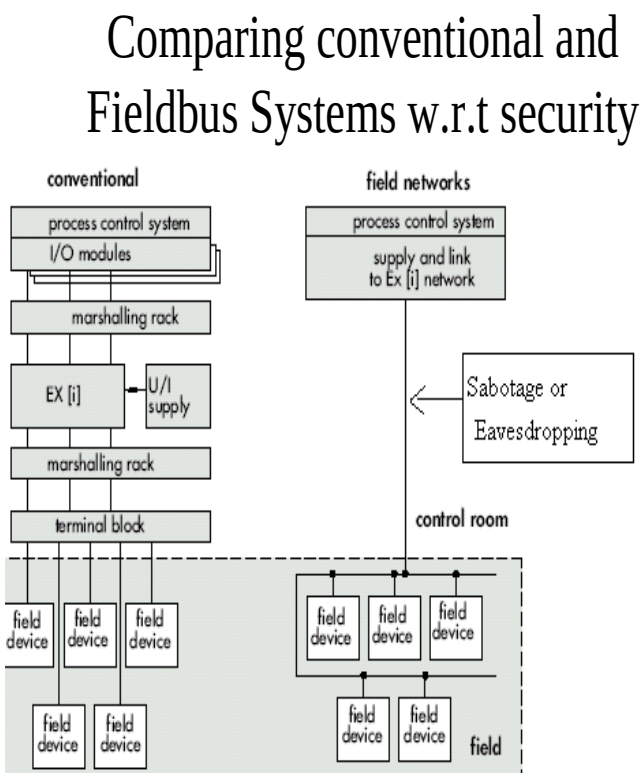


Fig.1. Typical Fieldbus Systems

II. SECURITY IN PROCESS CONTROL

The field bus along with its associated networks are problems of sudden unexpected sabotage and consequent failure and damage to plant and personnel. Secure data communication relies on suitable encryption of the data flowing in across the network and a decryption at the receiver end. In large process plant, Using popular encryption algorithms, such as the Data Encryption Standard (DES) and a protocol for secure key exchange has been developed for Symmetric Encryption). The possibility for intervention at various levels is shown in fig.2.

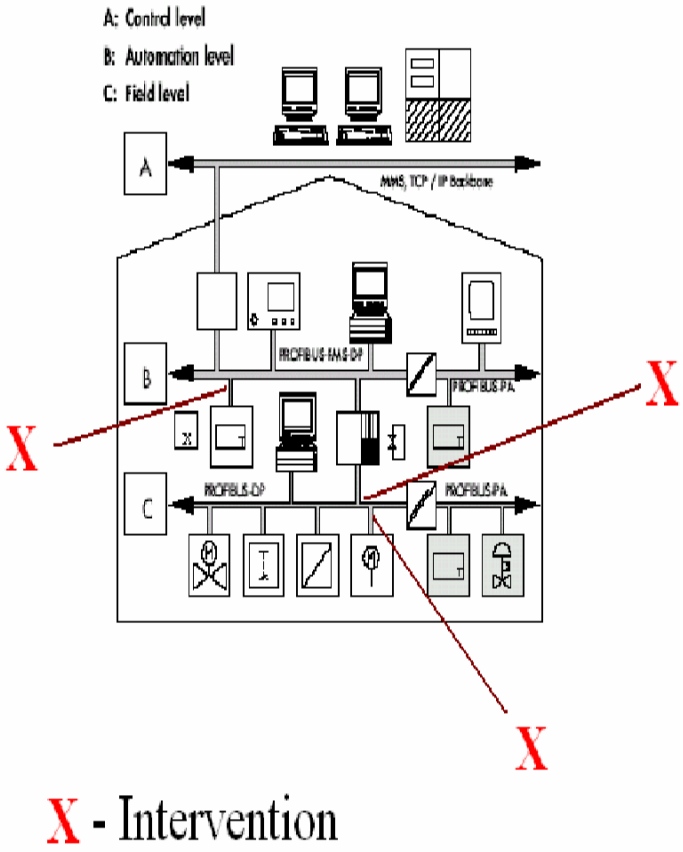


Fig.2 Shows the possibility of intervention at various levels of the Fieldbus

Higher level devices are connected to the supervisory levels using commercial networks based on TCP/IP. (Fig.2). In those levels, security protocols are easy to implement, and are also commercially available At the Field level, implementation of security features is difficult because field devices are not sufficiently powerful. It is difficult to co-ordinate among the several sets of field devices. Difficulties exist in incorporating the contemporary cryptographic algorithms over Fieldbus networks. Some have been given below.

- Field us standards are not well laid out
- The end devices must be capable of supporting the volume of computation required to decrypt the data and apply it to the control of process.
- Encryption must be two-way. When the process variable data are being sent upwards over the bus, the same should also be encrypted. This imposes an additional requirement for the field devices to run encryption algorithm.

The simple Fieldbus packet details are given below:

Simple Field Bus Protocol packet

SFBP packets are made up of 11 bytes, or optionally more, streamed in frames.

SM DA SA PI DU DU DU DU DU DU CS [+]

SM	start marker	1 byte	value: fixed, 254
DA	destination address	1 byte	value: 0...127 (0x00...0x7f)
SA	sender address	1 byte	value: 0...127 (0x00...0x7f)
PI	packet information	1 byte:	
		7 6 5 4 3 2 1 0	
		L L L A N T T T	

L DU valid length, this field is 3 bits long, defines how many bytes are valid in the data unit.
A ACK bit field *
N NEXT bit field *
T Type of packet, this field is 3 bits long, defines the type of packet:
0 echo
1 control packet
2 data packet
3 time
4 priority (experimental, see priorituary packets)
5 reserved
6 system packet

III. THE CHOICE OF ALGORITHM

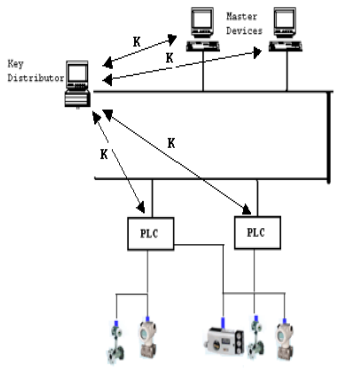
Of all the standard cryptographic algorithms available in the market today, this work uses the Data Encryption standard 9DES) as the backbone for the Fieldbus security protocol. There are several reasons behind this preference for the DES. The DES is absolutely symmetrical. The algorithm is public and the encrypted message is a function of the input text and the "key"> Description is done using the same key. The algorithm is probably the fastest of all the commercial cryptographic algorithms available today. The key length is manageable at 56-bits and is more than enough to guarantee security for Fieldbus networks carrying short data chunks. The algorithm is IEEE prescribed standard.

IV. FIELDBS PROTOCOL- DEVELOPMENT AND IMPLEMENTATION

The implementation of the protocol is based upon a set of primitives which is common to the various levels of the Fieldbus architecture. The one major problem in the implementation of symmetric encryption algorithms like the DES, is the process of key exchange. There has to be a way by which the communicating parties can securely exchange the key. This cannot be done by directly transmitting the key without encryption, which in turn requires another key(infinite recursion!). A straight forward solution could be to manually be exchange the keys .Which would be a one-time process during network setup. Communication could then proceed using the manually distributed keys and should theoretically remain secure , but for how long? Network interveners could employ guessing schemes such as the known message or the probable message attacks , or some form of key deduction by analyzing encrypted samples , and figure out the key .this process could happen within days of installations , depending upon the amount and frequency of messages transmitted over the network .This is an added advantage for fieldbus systems as many of the control messages are usually short and frequented over the network..

To get over this problem , the highest level in the fieldbus network, namely the Master supervisory level is equipped with a secure trusted system known as the Key Distributor (KD).The key distributor is installed in a physically sealed environment with access permitted only to trusted management level personnel. It does not require frequent manual intervention as the operating protocol is fully automated. In fact , ZERO manual intervention is a reasonable option.

The Protocol – Initial Configuration



- **The Master (viz. Supervisory) level has a trusted Key Distributor**
- **Every computer at the Master level shares a key with the Key distributor(KD)**
- **All the Field Level PLCs also share a key with the Key Distributor**
- **The Initial Key configuration is done in secrecy, manually by the higher authorities.**

Fig.3 Shows the presence of the key distributor at the highest level

4.1.The Initial Configuration

To kick start the protocol operation, the following initial configuration should be established .

- The key distributor should share a DES key with each of the devices in Master level and also the Field Level PLCs.
- The key distributor should be recognized as the trusted system on the network and every message sent by the KD should be encrypted (except for few allowable exceptions , such as acks, to be discussed later)

  This initial configuration is only required to turn on the security system. The manually established keys are very much transient ,and will change automatically (without manual intervention)from time to time, to ensure tighter system security.

4.2. Procedure for Key Exchange

Now that the initial configuration has been put in place, the KD can securely exchange messages with other devices in its level and also the Field level devices (usually PLCs).This secure channel is made use to facilitate key exchange between master and slave devices so that they can proceed to communicate directly. The actual key exchange occurs as follows.

The Master (M) starts off by sending an encrypted message to KD (encrypted by the key km, kd shared between M and KD) that includes the master ‘s identity (optional –depending upon underlying communication protocol), the identifier of the Field device (F, usually the PLC , terms used interchangeably) with which it wants to communicate a newly generated 56-bit key Km, Fd for communication with F and a Time stamp TS1. KD then proceeds to verify the key and diverts it over to the Field device F, encrypted by the key Kfd, kd (shared between KD and F). Along with the key , the identifier of M , a new Timestamp(current clock)and the original timestamp value when M initially placed the request(TS1) are also included in the message . The Field devices stores Km,fd in its key database against the identifier of M. It follows up sending a key acknowledgement (k-ack)back to KD. Note that this k-ack need not be encrypted because of the implicit restriction that only one master can enter key acquisition phase with a particular field level device , at a time.

In case multiple masters issue key proposals ,the requests are queued up by KD and forwarded to F one by one. The k-ack need not include the identifier of the master to whom it corresponds, thus saving on encryption /decryption processing times. It however , includes a digest of the key , so that spoofed acks can be counteracted. In the final phase ,KD redirects this received k-ack back to master M. Once again , there is the similar restriction that prevents a particular master from making key proposal for more than one Field device . Although these restrictions appear to be stringent , they serve to improve performance by reducing the need for transmitting encrypted acknowledgements during key exchange. The key exchange protocol is seen below.

- Between Master Device(M), Key Distributor (KD) and Field Device (FD)
  - M → KD : Ekm,kd(IDM | ID F | km,fd | TS1)
  - KD : Verify the key
  - If key is acceptable

KD → FD : Ekfd,kd(IDM | km,fd | TS2 | TS1)
FD stores the key in its DB

FD → KD : *k-ack* (Note: Only one Master Device can
KD → M : *k-ack* (Note: Any Master can acquire key to one field device at a time)
M stores the key in its DB

Legend : ID-Address Identifier, TS – Timestamp
kx : Key x, Ekx,y: Encrypted with Key shared by x and y, *k-ack* : Key Ack

The Protocol – Key Acquisition and Communication has been developed if key is not acceptable.
KD → M : *n-ack*

i). Communication between Master Device (M) and Field Device (FD) is done as under:

- FD → M : Ekm,fd(reply | TS)
- M → FD : Ekm,fd(msg | TS) (Only for ymmetric Encryption)

ii). Timestamps guard against replay attacks

Legend : *n-ack* : Negative ack

Once Key exchange is completed , both parties can directly exchange encrypted information over the network . It is advisable to include timestamps in each message before encryption so that replay attacks are not possible . I n addition to timestamps, The messages should be appended with a hash (or CRC) prior to encryption . This will ensure that no attacker is able to transmit well –furnished junk packets in an attempt to confuse the communicating parties.

4.3 Filed Level Security

At the field level , the individual field devices are equipped with the TMSC50 and C240 to run the DES cipher. In case of a multi master configuration , with more than one PLC connected to a bus , the same protocol can be operated with each node acting as the local key Distributor for itself

V. CONCLUSION

The TMS C 240 is present embedded in many of the state –of –art field devices such as motorized control valves and sensors. Hence, it is easier to implement the SecFB protocol without the need for additional hardware. It was tested with the Simple Field Bus Protocol (SFBP)[2] stack as the underlying platform. The DES algorithm was implemented on the TMS family of DSP processors This chip(240) is already in use for motorized control valves, as a field device Data scanners which send large amounts of data from the process could also use such DSP chips to support encrypted data transfer over the Field bus.

ents of key acquisit

REFERENCES

1. Date Encryption standard (DES) Implementation on the DES320C6006, Internet Whitepaper
2. Clauio Ghiotto and Paolo and Paolo Marchetto, "Simple Field Bur Protocol" © Softmedia 2003, HEXEL, Electronic Lab.
3. P.Swaminathan and R.Pradeep, Network Communication Security in Field bus Protocol for secure Process control, Proc. Of the ACHEMA'06 Conference, Germany, 2006.
4. P.Swaminathan and R.Pradeep,The Secure Field Bus (SecFB) Protocol -Network Communication Security for secure Industrial Process control, IEEE Conference, Honkong, 2006. (1-4244-0549-1/©2006 IEEE).