

SDL Specification of a Security Architecture for WorldFIP

Miguel León Chávez¹ and Francisco Rodríguez Henríquez²

¹*Benemérita Universidad Autónoma de Puebla*

Facultad de Ciencias de la Computación

14 Sur y Av. San Claudio, CP 72570, Puebla, México

Tel. (52) 222 229 55 00 ext. 7213 Fax (52) 222 229 56 72

E-mail: mleon@cs.buap.mx

²*CINVESTAV-IPN*

Sección de Computación

Av. Instituto Politécnico Nacional No. 2508, Col. San Pedro Zacatenco

México, D.F. 07300

Tel: (52) 52 55 5061 3800 ext. 6570 Fax: (52) 555061-3757

E-mail: francisco@cs.cinvestav.mx

Abstract

This paper discusses the security in the fieldbuses, and presents a security architecture for the centralized fieldbuses, such as WorldFIP. The discussion takes into account, on one hand, the security services defined by the OSI Security Architecture, and on the other hand, the security mechanisms defined by WorldFIP. The discussion shows that there are two critical points for attacking this network: the Bus Arbitrator and the bus. This paper presents then the specification, using the Specification and Description Language (SDL), of a security architecture for WorldFIP based on the RC4 algorithm, which is a lightweight cryptographic stream cipher.

1. Introduction

Fieldbuses are special purpose Local Area Networks (LAN) used to connect all kinds of devices into a factory, such as sensors, actuators, transmitters, programmable controllers, (C)NC machines, processors, and so on [1]. These networks are usually seen as a three OSI layer architecture, which includes the physical, the data link and the application layers. Some of the services of the missing layers are still present in some fieldbuses but are merged in the application layer, e.g. support for interoperability, end-to-end control, fragmentation and reassembling of messages, routing among several Fieldbuses, etc.

Typically, the fieldbuses are used by the distributed manufacturing applications in order to monitor and control the processes taking place in the factory. Examples of such applications are: factory automation, automotive industry, textile machinery, electronics manufacturing, food and beverage, chemical processing, and so on.

Up to now, the security in the fieldbuses, such as the IEC 61158, has been only considered for access protection on some objects. This is not for protection against intentional misuse of the communication facilities of a field device but in order to protect a system of accidental erroneous use of the objects.

However, there exist at least two possible security attacks that centralized fieldbuses can suffer: Non-authorized users gaining access to the communication channel and non-authorized human operators accessing the master node.

Clearly, there exists always the possibility for non-authorized users to gain access to the communication channel. If that happens, then the intruders can launch a passive attack by eavesdropping all or part of the information exchanged among the network's entities. Even worse than that, active attacks are also possible as hackers can maliciously modified the data traveling through the communication channel at will.

On the other hand, the master node stores all the network configuration as well as other important system global information, such as presence variables, i.e. variables containing summarized information on the node's global operating state. Hence, mechanisms of

user identification should be put in place in order to avoid leakage of valuable data to non-authorized human operators.

As distributed manufacturing applications become more and more diverse, complex and integrated into other kind of applications, possibly attacks to the security of the network increase in the same rate. Hence, it is of the utmost importance to incorporate security mechanisms on fieldbus communication protocols so that such kind of security attacks can be avoided and/or prevented.

In particular, the Security Architecture of the OSI Reference Model [2] considers five main classes of security services: authentication, access control, confidentiality, integrity and non-repudiation. These services are defined as follows: The *authentication* service verifies the supposed identity of a user or a system. The *access control* service protects the system resources against non-authorized users. The *confidentiality* service protects the data against non-authorized revelations. Confidentiality is considered an essential role of the cryptography systems. The *integrity* service protects the data against non-authorized modifications, insertions or deletions. The *non-repudiation* service provides certain protection against the sender of a message that refuses to be it, or against the receiver of a message that denies to have received it.

All the security services defined by the ISO can be achieved, in a centralized fieldbus, by using public key cryptography. That can be accomplished by assigning to each slave node in the network a unique private key and a master node's public key. During communication, slave and master nodes may mutually authenticate each other with these keys using well known protocols (such as Diffie-Hellman). To provide confidentiality, nodes may encrypt their contents using a random session key and a symmetric crypto-algorithm especially tailored for constrained environments. Integrity and non-repudiation can be obtained by signing/verifying all the messages transmitted between a particular slave node and the master node.

However, strong public key cryptography is in general an expensive fancy solution [3]. Moreover, some of the security services defined by the ISO are probably not very likely to be useful on the context of centralized fieldbuses. Particularly, non-repudiation seems to be not very useful for this kind of networks since the master node "gives permission to speak" to each slave node.

If for a given application public key cryptography solutions are too expensive, we can still design limited

security schemes for fieldbuses at a cheaper price. For instance, we can use a security scheme based on a one-way hash function optimized for heavily constrained environments [4], as those typically found in fieldbuses. Data confidentiality can then be achieved by using some lightweight cryptographic stream cipher such as RC4 or A5 GSM [5], or even a reduced version of traditional symmetric algorithms such as DES or AES.

This paper discusses the security in the fieldbuses, according to the classes of security service defined by ISO. The discussion is focused on WorldFIP because it is a typical example of the centralized fieldbuses. The paper proposes then a security architecture for this fieldbus based on RC4, which is a lightweight cryptographic stream cipher.

The remaining of this paper is organized as follows: section 2 presents WorldFIP and its security mechanisms; section 3 proposes a security architecture for WorldFIP; section 4 presents the specification and validation of the WorldFIP security architecture using the Specification and Description Language (SDL). Finally some future work directions and conclusions are given in section 5.

2. WorldFIP

This section presents WorldFIP and its security mechanisms.

2.1. WorldFIP general architecture

World Factory Instrumentation Protocol (WorldFIP) [6] is a centralized fieldbus, since it consists of one master node, named Bus Arbitrator (BA), and a set of slave nodes, named producer and consumer nodes. The BA is responsible of the maintenance of both, a configuration table and different queues of identifiers that define at any time the communication requirements of the system. On the other hand, slave nodes produce/consume data generated during the system operation. Slave nodes can be a producer and/or a consumer of one or more variables.

At network configuration time, an authorized human operator assigns a unique identifier to each one of the system's variables, and invokes the application layer services to send a set of *service request primitives* to the BA, at the data link layer level. These service request primitives describe the execution of both the basic cycle and the macro cycles (the sequential execution of one or more basic cycles). A basic cycle is composed of at least one window, called the periodic window, and at up to four windows, as follows: a

periodic window, an aperiodic variable window, an aperiodic message window and possibly a synchronization window to adjust the constant duration of the basic cycle, as is shown in Fig. 1.

At network operation time, during the periodic window, the BA reads the table of periodic variables and injects each variable identifier onto the network; each variable shown has only one producer. Consumers needing to utilize the variable, alerted by the identifier, store and use the value broadcasted by the producer. Thus the BA “gives permission to speak” to each information producer.

Whenever an aperiodic variable or an aperiodic message is produced, the producer utilizes the response of a received periodic variable identifier to request its transmission. The BA stores the identifier, which is carried by the request, into the appropriate queue. After completing the current periodic window, the requested aperiodic variables and the requested aperiodic messages are then handled in the same way that the periodic variables: within the appropriated windows, where the relevant producers reply current values.

2.2. Security in WorldFIP

WorldFIP provides some security mechanisms implemented in its protocol or in its components. The mechanisms considered by the protocol are [6]: medium redundancy, errors in the physical layer, data link layer status machines, frame check sequence (FCS), bus arbitrator redundancy and validity of variables.

It can be noted that FCS provides the integrity service, according to the definition given by the ISO security architecture, i.e. FCS is calculated when the frame is transmitted and when it is received. If the code received matches the code calculated there is a very high probability that the frame is correct.

However, the other mechanisms included in the WorldFIP protocol do not provide the following security services: authentication, access control and confidentiality.

Previous section has shown that the authorized human operator configures the BA, which stores the network configuration into a table and several queues.

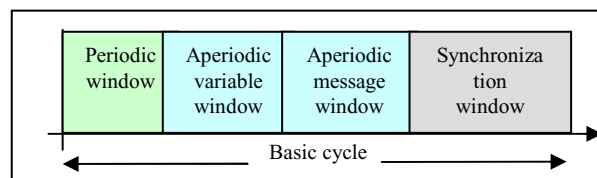


Figure 1. WorldFIP basic cycle

Therefore, WorldFIP must provide the authentication service in order to put in place some mechanisms of user identification in the BA, such as logging and password, to avoid access from non-authorized human operators. Access-rights can also be added to the authentication mechanism so that authorized human operators can have access rights to configure, read and modify the network configuration. Also, in order to avoid passive and active attacks from intruders that have managed to gain access to the communication channel, WorldFIP must instrument a data-confidentiality service via feasible encryption/decryption schemes.

3. Security architecture for WorldFIP

This section presents the security goals that we wish to obtain from the security architecture to be proposed here.

3.1. Security goals

In our model we shall assume that all nodes belonging to the network possess some kind of processing capabilities, and that both of them, BA and producer/consumer nodes trust each other. Deceiving from any party should be difficult or impossible in practice. Ideally, fault induction and power disruption should not compromise protocol’s security, nor should it open avenues for hijack attempts. BA and producer/consumer nodes should be resistant to session hijacking, replay or man-in-the-middle attacks.

As it was mentioned in the introduction section, all the customary security services for communication protocols can be achieved by using public key cryptography schemes. However, public key cryptography requires a processing power that is typically well beyond the reach of fieldbus slave node processors. Fortunately we can still design limited security schemes for a fieldbus at a cheaper price by using alternative cryptographic options.

Data confidentiality can be achieved by using some lightweight cryptographic stream cipher such as RC4 or A5 GSM [5], or even a reduced version of traditional symmetric algorithms such as DES or AES, which can be obtained by reducing the size of the encryption key or by limiting the standard number of rounds used during the encryption/decryption processes (16 in the case of DES and 10 for AES).

Due to the fact that password-based security mechanisms are needed only for data protection at the BA node, where typically processing power is not a

concern, we can use any of the traditional schemes based on symmetric ciphers.

Summarizing, this paper proposes to achieve the above stated security features by incorporating the following mechanisms into the WorldFIP architecture:

- A lightweight stream cipher in order to guarantee data confidentiality by encrypting all the relevant data to be transferred by the network's entities.
- A password-based security mechanism to prevent non-authorized users to gain control in the BA node.

These security mechanisms are discussed in detail in the next section.

3.2. WorldFIP Security Architecture

In this paper we propose to use RC4 as the stream cipher needed to implement data encryption/decryption. RC4 is a symmetric-key stream cipher that was developed in 1987 by Ronald Rivest and kept as a trade secret by RSA Data Security, Inc. In September 1994 the algorithm was anonymously posted on the Internet, and since then it has been studied and used widely in academic circles and is now available for public analysis. RC4 is commonly used as the default cipher for SSL/TLS connections and currently being standardized by the IETF under the name "Arcfour" [7].

RC4 uses a 2048-bit key-length needed to initialize a 256-byte state table, called the S-box. If a given application wishes to use a shorter key, then that shorter key is repeated as many times as needed to fill the 2048-bit key. Typically, RC4 is used in a mode where 16-byte (128-bit) keys are repeated sixteen times.

Once the S-box is initialized with the key, the RC4 algorithm enters in a loop that updates the S-box and generates a byte of pseudo-random keystream. That pseudo-random keystream is XOR-ed with the plaintext message to produce the ciphertext. Thus, the RC4 algorithm can be broken into two phases: initialization and operation. In the initialization phase the 256-byte state table S is populated using the key K as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted. The C-like pseudo-code for this phase of RC4 is as follows:

Initialization Phase:

1. $j = 0;$
2. for $i = 0$ to 255:
3. $S[i] = i;$
4. for $i = 0$ to 255:
5. $j = (j + S[i] + K[i]) \bmod 256;$
6. swap ($S[i], S[j]$);

Notice that what the RC4 initialization phase basically does is to swap the location of the numbers 0 to 255 (each of which occurs only once) in the state table.

Let us assume that the initialization phase has been completed and that the input message M consists of N bytes stored in the array $M[0..N-1]$. Then, the following algorithm implements the operation phase of RC4 by encrypting/decrypting the input message M :

Operation Phase:

1. $i = j = 0;$
2. for ($k = 0$ to $N-1$) {
3. $i = (i + 1) \bmod 256;$
4. $j = (j + S[i]) \bmod 256;$
5. swap $S[i]$ and $S[j]$;
6. $D = S[(S[i] + S[j]) \bmod 256]$
7. output $C = M[k] \text{ XOR } D$
- }

The above algorithm outputs a ciphertext C which is the input message M , XOR-ed byte by byte with the values of the random stream D . As long as the same initial S-box S is used, both of them encryption and decryption processes are completely symmetric. Therefore, if the operation phase algorithm is used to obtain an encrypted ciphertext C from an input plaintext M , then M can be recovered from C by re-executing the same algorithm once again, provided that the same initial state table S is used.

Regarding the authentication service, a password-based security system can be implemented by using any symmetric cipher (such as DES or AES) with the password as the key to encrypt a fixed plaintext (usually an all zeroes text) [8, 9]. The ciphertext so produced is then stored in a public file. When a given human operator wants to have access to the BA, he/she needs to input his/her login and personal password. The security system then proceeds to encrypt the fixed all-zeroes plaintext using the password that was just input by the user. Finally the ciphertext so produced is compared with the one stored in the public file under the login name of the user allowing user's identification.

In order to use a password as a key to encrypt, each one of its characters are converted to their 7-bit ASCII equivalent. The password is padded with zeroes as needed in order to obtain a bit-length equal to the key-length of the symmetric cipher being used. Typical choices for ciphers for password security systems are DES and more recently AES.

4. Specification of the WorldFIP security architecture

This section presents the specification of the WorldFIP Security Architecture using the Specification and Description Language (SDL) [10], because it is a language that is intelligible to human beings, but still formal enough to support analysis and comparison of behaviors.

For validation purposes, the term validation model is utilized by SDL. A validation model is a description of the system, which is suitable for validation, i.e. to apply validation techniques such as testing the formal model, exhaustive validation (reachability analysis), non-exhaustive validation (analysis of a random subset of the reachable states), simulation and informal validation techniques (checklist) [11].

A validation model is always executable. In order to construct the validation model of the WorldFIP Security Architecture, shown in Fig. 2, the toolset ObjectGEODE was utilized.

In SDL, behavior is always performed in the context of a *system*, beginning with a top-level description of the system to be validated. Fig. 2 shows the WorldFIP system, which is composed of the following *blocks*: the application and data link layers of the Bus Arbitrator (BAApplicaLayer and BADataLinkLayer), the data link layer of the producer/consumer nodes (PCDataLinkLayer), and the physical layer. These blocks are interconnected through *channels*, named ca, cb, c1, c2 and c3. The channel ca connects the BA with the environment, by this means the human operator configures the network.

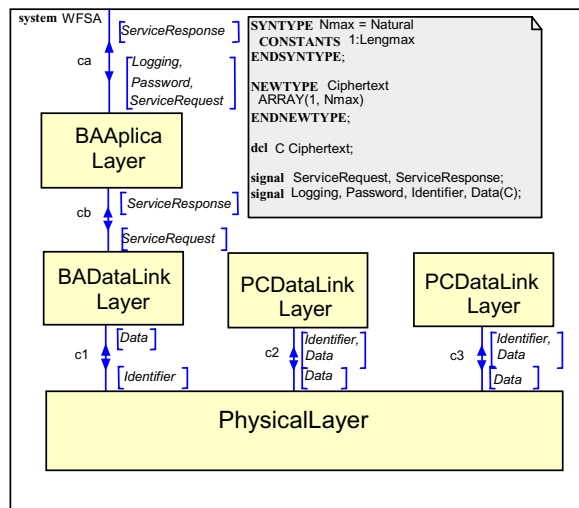


Figure 2. WorldFIP security architecture system

In addition, Fig. 2 shows the *signals* passed in each direction over the channels, as indicated by the arrows on the channels, and the declaration of the signals as well as the declaration of the data structures.

It can be noted that the operation of WorldFIP has not changed. At network operation time, the BA continues injecting each variable identifier onto the network, but now the producer broadcasts the ciphertext (Data(C)), which is the variable's value encrypted by the operation phase of the RC4 algorithm depicted in the previous section. On the other hand, the consumer nodes read and decrypt the ciphertext to obtain the variable's value. This SDL specification has assumed that the initialization phase of the RC4 algorithm has been done at network configuration time, hence producer and consumer nodes have the same initial state table S.

In a SDL specification, the system occurs only at the top level, while blocks only occur inside. The system is decomposed into blocks and channels recursively over as many levels as desired until the basic components, called *processes*, are reached. One *SDL process* is a concurrent object with its own control flow, described by an Extended Communicating Finite State Machine (FSM), which is composed of the following four main parts: input port, FSM, timers and variables. This FSM state-transition behavior is expressed in terms of *process diagrams*, which are not presented in this paper for lack of space.

Fig. 3 and 4 present the data link blocks of the BA and the producer/consumer nodes, respectively. These blocks are composed of processes, which are defined by the WorldFIP Standard. It can be noted that there exist only one new process, called RC4, which executes the initialization and the operation phases of the RC4 algorithm.

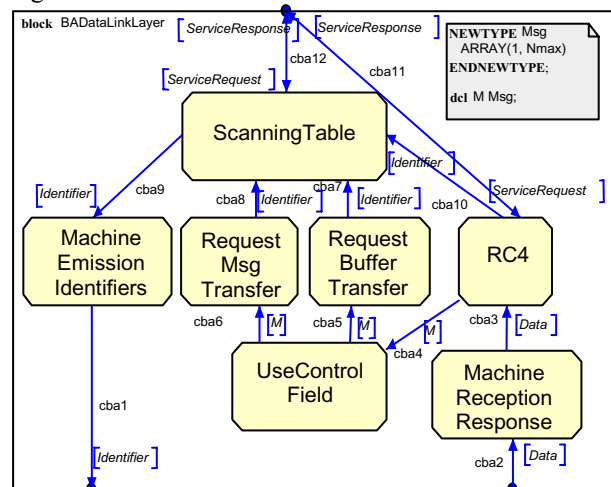


Figure 3. Bus Arbitrator data link layer block

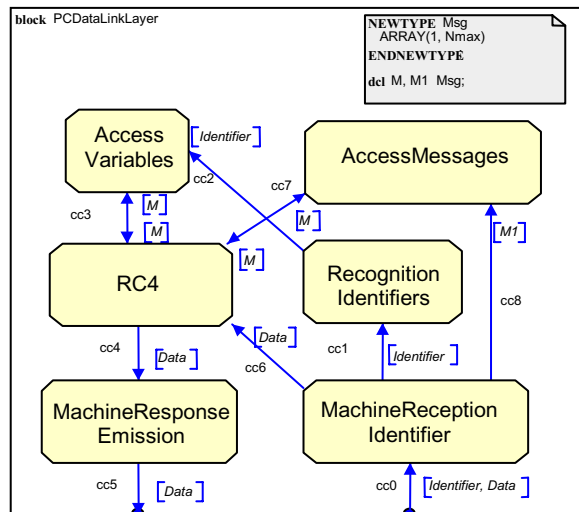


Figure 4 Producer/consumer Data Link Layer block

5. Conclusion

This paper has discussed the security in centralized fieldbuses, such as WorldFIP. The discussion has shown that this fieldbus provides only limited security, namely the integrity security service defined by the OSI Security Architecture. WorldFIP implements that service, until certain extend, by using a frame check sequence (FCS). However, centralized networks are vulnerable to at least two possible security attacks: Non-authorized users gaining access to the communication channel and non-authorized human operators accessing the master node.

Both types of security attacks can be avoided using public key cryptography schemes. However, public key cryptography requires a processing power that is typically well beyond the reach of many fieldbus node processors. To overcome this difficulty, this paper has proposed a Security Architecture that provides the authentication and confidentiality services at a cheaper price by using alternative cryptographic options. We have proposed to use reduced versions of AES or DES symmetric crypto-algorithms at the master side, and RC4, which is a lightweight cryptographic stream cipher at the slave nodes side.

The proposed Security Architecture has been applied to WorldFIP and specified using SDL. For simulation purposes a validation model has been constructed using the toolset ObjectGeode. From the original WorldFIP Standard, it can be noted that only two processes are required to be added into the WorldFIP architecture: the RC4 process and the password process (not shown in the application layer block of the Bus Arbitrator). The RC4 process is used

by the producer/consumer nodes for encrypting/decrypting the variable's value. Therefore, there exists an extra processing into these nodes that must be quantified and measured in order to verify that the timing constraints on the messages are met; and this is our future research work, as well as to use the A5 GSM into the Security Architecture proposal.

6. References

- [1] J.P. Thomesse, "A Review of the FieldBuses", *Annual Reviews in Control*, vol. 22, 1998, pp. 35-45.
- [2] ISO 7498-2, International Organization for Standardization. Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture, 1989.
- [3] C. Paar, "Cryptography in Heavily Constraint Environments", In *Proceedings of the first Ad-hoc Workshop*, University of Bochum, Germany, December 2-3, 2002.
- [4] S.E. Sarma, S. A. Weis, and D. W. Engels, "Low Cost RFID and the Electronic Product Code", In *Cryptographic Hardware and Embedded Systems-CHES 2002, Lecture Notes in Computer Sciences*, Springer-Verlag, Heiselberg, Germany, 2002.
- [5] A. Biryukov, A. Shamir, and D. Wagner, "Real Time Cryptoanalysis of A5/1 on a PC", In *Fast Software Encryption Workshop*, New York City, April 10-12, 2000.
- [6] EN 50170-3, WorldFIP, General Purpose Field Communication System, CENELEC EN 50170-3, 1995.
- [7] K. Kaukonen, and R. Thayer, "A stream cipher Encryption algorithm Arcfour", *IETF*, Internet draft-kaukonen-cipher-arcfour-03.txt, July, 1999.
- [8] Menezes, Oorschot, and Vanstone, *Handbook of Applied Cryptography*, CRC Press, New York, fifth ed., 2001.
- [9] Trappe, W., and L. Washington, *Introduction to Cryptography with Coding Theory*, Prentice-Hall, 2002.
- [10] ITU-T Z.100, ITU Recommendation Z.100, *the Specification and Description Language (SDL)*, 2000.
- [11] D. Hogrefe, "Validation of SDL systems", *Computer Networks and ISDN Systems*, vol. 28, no. 12, pp. 1659-1668, 1996.