A Cryptographic Protocol to Obtain Secure Communications in Extended Ethernet Environment.

X. Simón, J.L. Melús, F. Recacha.
Dept. Applied Mathematics and Telematics.
Polytechnic University of Catalonia.
C/ Jorge Girona Salgado s/n
tel: 34-3-401.68.28 fax.: 34-3-401.68.01 Email: teljmm@mat.upc.es
Barcelona - April 1992

Abstract:
Nowadays, the use of local area networks to exchange information between different users is very common in all the world. Usually, this information is very critic (economical and financial management, academic, etc.) and it is necessary to dispose of a security mechanism in the network in these cases. Here we present a real implementation that we are developing in the U.P.C. Campus (Polytechnic University of Catalonia). The implemented security is based on cryptographic techniques and on an extended local area network. This network is formed by a backbone that links every one of the departments (subnetworks) that make up this institution. The basic idea behind this implementation consists in to utilize the existing resources in the network, more specifically the bridges, and to introduce the wished security in them.

The used algorithms are the DES [1] algorithm for the ciphered/deciphered information and the transference of session keys. The other algorithm is the RSA [2] algorithm, and it is used for the transmission of master keys and the setting up of the exchanged information between the different "secure bridges" (Cryptonets) and the Key's Administrator. In this work, we describe the cryptographic protocol utilized to cipher/decipher the Ethernet frames between the secure bridges and the established procedure for the renewal of session and master keys. The used method is clear for the user and it does not reduce the previous network's resources (traffic, delays, etc.).

1. Introduction

In this paper, we present the implemented security developed in our local area network. The extended network is formed by a principal backbone and several segments assigned to each one of the University Departments, joined to it by means of other devices (routers, bridges, etc.). In this network, we dispose of "secure areas" and others that can be designated as "insecure areas" (see fig. 1-1).

In a "secure area" we think information is protected. We consider potential users in this zone will not attempt against the security of the information. We will associate this area with a Department subnetwork.

b) Analogously, an "insecure area" is defined when the security can be committed and any intruder may have access at the information that circulate in the network.

Figure 1-1: Secure and unsecure zones in network.

In the last case it is necessary to implement the appropriate security mechanisms (cryptographic algorithms) that allows to offer the level of secrecy required and dispose of the methods of authenticity for the integrity of the transmitted messages and users. The security will be laid on the bridges as we can see in fig. 1-2. This option is based on some points:

1) The maximum number of cipher devices we need in the network when we implement the security in the bridges is considerably reduced. Also the modification introduced in the network lacks significance and the additional cost is very small.

2) We can obtain a more simple management of the keys than in the other case and the natural
complexity associated with this process always can be limited.

Figure 1-2. Bridges are cryptographic devices, between secure departments and unsecure backbone.

This network is installed in the U.P.C. (Polytechnic University of Catalonia). In the ordinary mode, the protocol utilized is the LAT [3]. This protocol works at level LLC (Logic Link Control). In this work, for the normal handling of the frames (filtering and ciphering) we have chosen the MAC (Medium Access Control) level. The IEEE standard 802.3 [4] and the facto standard Ethernet [5] define this level for our environment. In fig. 1-3, we can observe schematically the process of filtered and ciphered of Ethernet's packets that circulate on the backbone. As we said before, this procedure is applied at MAC level [6].

2. The Secure Bridge (Cryptonet)

The Cryptonet is an especial bridge with some different particularities. They do the same basic functions as normal bridges, that means, the filtered function of the traffic that comes into the bridge according to the contents of some tables (the addresses of different nodes in the network). These addresses are dynamically renewed (see [7][8]). In this paper we will dedicate to describe the security that the Cryptonet must carry out. Any bridge installed in the network will be connected by one port to the backbone and the other port to the associated section of a secure area. The bridge must have the capacity of discerning if one frame that arrives from a secure area must be ciphered or not, depending on the terminal source and the treated information. The same reasoning is applied when the frame arrives by the backbone and only some of it must be deciphered.

The bridge disposes of some tables to consult [7][8] if the nodes with capacity to generate ciphered traffic are recorded. Here, we will not concern how they are created or their maintenance. These tables are utilized and also we can determine the associated bridge of each of the nodes (the associated bridge joins the respective node of the department subnetwork with the backbone).

In summary, only the frames that are guided between two secure areas are ciphered and consequently the information is passed by the backbone, which is the "insecure area" of our system. In this case the nodes will always be registered in the respective tables. If the information is transmitted between two nodes of the same department subnetwork, the originated frames are not forwarded to the backbone and the filtered and ciphered processes will not be carried out (see fig. 2-1).

Figure 2-1. Only frames between registered nodes are ciphered.
3. Need for a cryptographic protocol

Two bridges participate in the cipher/decipher process, so it is necessary to create a protocol for each pair. Now we will explain the basic structure of the developed protocol:

a) The interchanged frames between each pair of Cryptonets must be ciphered with the same key, which has to be previously shared and only known by them. These keys are called session keys and constitute the lower level in our protocol. They are used to cipher the information field of Ethernet's frames.

b) It is periodically necessary to renew the pre-existing session keys to obtain a good strength in our security system. The master keys are utilized for this purpose and they will allow the interchange of the new session keys in the prefixed time.

c) Similarly, the master keys will be periodically renewed to enhance the robustness of our system. Then, the administration's keys will be defined in the higher level of key's hierarchy. At this level, the keys don't depend on each pair of bridges, but on each bridge and a common element for all the bridges of our network. This new element is called the manager or administrator of the system. In fig. 3-1 we can see the three levels of keys that form the defined protocol.

![KEY HIERARCHY](image)

Figure 3-1. The three protocol levels: administration, master and session.

The transmission speed in our Ethernet network is about 10 Mbit/sec. This speed will determine the cryptographic techniques that we can need in this case. For the implementation we have selected a hardware version of the DES algorithm (the Cry 12C102 chip, made by CRYPTECH [9]), since this hardware chip avoids the possible bottleneck effect due to high-speed requirements in this application.

Another problem is solved with the speed offered by the DES algorithm. When the session keys have to be renewed by the master keys, it is necessary to bring up this process very fast due to the utilized protocol (as we can say in the next paragraph).

Finally, the RSA algorithm is employed when the master keys are renewed. As we have said before, this process is carried out by the Cryptonet together with the Administrator. To implement this algorithm we have used the PQR-6 board from Cryptech [10].


Using the DES algorithm with a session key, the Ethernet frames will be ciphered, only the data field, at the MAC level, according to the standard 802.3, as we can see in fig. 3-2.

![Figure 3-2. Only Data Field is ciphered. CRC is recalculated.](image)

Such as we said, the process of ciphering/deciphering in a bridge is possible because in each one of them there is a table with all the nodes that use the same procedure to obtain security in our system. Every node is associated to a Cryptonet. Each Cryptonet will have registered the respective nodes associated with it and will share a set of DES keys (session and master) with each other. These keys will always be secret for the rest.

The mode of DES selected is the triple CBC (Cipher Block Chaining), with two different keys of 64 bits each one. The choice of the CBC mode is due to the following reasons:

a) The CBC mode spreads the errors until the end of the cipher process.
b) The operation of the DES in any of the different methods is reversible, that is to say, we can cipher in reverse mode and decipher in direct mode or we can make these processes in the other way.

c) If we consider the preceding items, we can add a secret part at the end of the message. This last position is called the MAC (Message Authentication Code) and it allows to detect the following situations:

c.1) A message altered by an intruder, when we don't obtain the expected MAC in reception, after the respective deciphered process.

c.2) Also it allows to detect some mistakes in the used key by the cipher or decipher Cryptonet. This situation is due to the use of different keys or the change of any bit of the key. The acknowledge of wrong frames will be avoided with this procedure.

Finally, using this method the MAC will be generated after the same only processed used to cipher the frame. This is very important for the implementation, because it won't be required additional time to obtain the MAC. This process to obtain the MAC is shown in fig. 3-3.

The key's space size is: $2^{8\times164-8} = 2^{112}$
(triple ciphered, with 8 bits of parity)

The reduced key's space size is: $2^{112} / 2^{32} = 2^{80}$

In this case, $2^{80}$ is bigger than $2^{56}$ (key's space size with simple ciphered), normally accepted as safe. When a 32 bits length MAC is chosen, there will be some frames that will generate the same ciphered MAC. The probability of one of these frames to be wrong and accepted as right is the following:

$$P(\text{error}) = \frac{\text{Total number of frames}}{2^{32}} / \text{Total number} = 1 / 2^{32}$$

If the MAC length was 8 bytes:

The reduced key's space size is: $2^{112} / 2^{64} = 2^{48}$
and the error probability is:

$$P(\text{error}) = 1 / 2^{48}$$

There is a serious compromise between the key's space and the error probability. Here we have decided not to limit the key's space unreasonably against the error probability. In fig. 3-4 MAC's generation is seen for the frame's subspace.

Figure 3-4. Each MAC is generated by a frame's subspace.

Frames Space  
(Dimension=TOT)  
MAC's Space  
(Dimension= 32)

There are some differences between the clear and the ciphered frame. In the last one it will be necessary to add 8 bytes. These bytes are the Initial Vector (V_I) of the CBC mode (DES operation). The bridge must know the V_I to do the deciphered operation. Usually, this V_I is transmitted in clear because it is not important in a cryptoanalytic attack, but it will be changed for every ciphered frame. Moreover, in some cases we need to add some bytes (pad bytes) to obtain the required multiplicity for the operation of CBC mode (multiplicity of 64 bits, 8 bytes). The content of these bytes must be unknown and only it is necessary to codify the length of the additional pad with
three bits. Finally the Cryptonet removes the suggested number of bytes in the deciphered operation, which have been previously assigned in the preestablished three bits codification. In the fig.3-5 the format of ciphered frames can be seen.

![Diagram of ciphered frames](image)

Figure 3-5. New Data Field: Initial-Vector, Flags/Pad_len byte, Counter byte, Data bytes, Pad bytes and MAC.

If we observe this figure, one byte as byte counter and some flag bits are also included. The flag bits share the same byte that the prefixed pad-len bits. When the length of the clear frame is longer than 1484 bytes, it will be in the best case only increased in 14 bytes for the ciphered frames, as we said before:

- Initial Vector (8 bytes)
- Flags and Pad_len (1 byte)
- Counter (1 byte)
- Pad (0 to 7 bytes)
- MAC (4 bytes)

The number 1484 is the biggest multiple of 8 that allows to add 14 bytes without exceeding the maximum size permitted in an Ethernet frame (1500 bytes). Therefore all the clear frames whose length is between 1484 and 1500 will have to be split in their data field in two new frames (subframe). Each subframe has the same header and a new data field (see fig. 3-6).

With this procedure, when a Cryptonet receives a subframe needs to know if it is the first or the second part of the previous divided frame to reset it properly. Moreover in this process it is necessary to verify if the two subframes proceed from the same old frame and check that:

- The two subframes come from the same Cryptonet.
- The two subframes have been sent consecutively. In this case, the sequence numbers are consecutive and these numbers are obtained by simple inspection of the byte counter.

The Cryptonet keeps the arrived subframe and waits for the other one. If it does not arrive in a prefixed time, the Cryptonet will refuse the previous subframe.

The pursuit of this task is carried out by two flags. These bits are placed in the same byte of the pad_len:

\[ F_3, F_4 \]

1 byte (flags and pad_len) \( (X: \text{not used}) \)

a) \( F_3 \): when this bit is '1' it indicates that this frame is a subframe.
b) \( F_4 \): when the bit \( F_4 \) is '0' and the previous bit \( F_3 \) is '1', this subframe is the first one. If the bit \( F_4 \) is '1' this is the last subframe.

3.2. Master Protocol. Renewal of session keys

The new proposed session keys will be renewed with the master keys. The new session keys will be ciphered in ECB DES mode with the master keys. These keys are private for each pair of Cryptonet, similarly to the session keys.

It is always very difficult to decide the right time for the process of renewal. We have chosen the number of frames that have been ciphered with the same session key to start this process. Each Cryptonet will have a different counter number for each ciphered channel. If we need to avoid a cryptoanalytic attack it will be necessary to change these session keys adequately.

Another problem to be solved is to avoid that two Cryptonets change the session keys simultaneously. The probability of this event is very low because the circulating traffic between the two bridges are not the same.
If we consider that "N" is the limit of the maximum number of frames ciphered with the same session key and "n" is the frame's number that a bridge has ciphered with the same key, when the number "n" is the same that "N", the total number of frames ciphered with the same key is as maximum 2N. Therefore, if we avoid the coincidence in time of renewal, we obtain the following aspects:

- A more simple protocol.
- To avoid the concurrence of the cipher/decipher processes, usually running in parallel in the bridge.

![Diagram of frame structure](image)

Figure 3-6. Obtained sub-frames by partitioning the initial frame. Second one has a 32 byte Data Field length and the first, L - 32.

### 3.3. Modification of the frames in the renewal process

When a Cryptonet proposes to change the actual session key, it must notify it to the other bridge, because it is impossible to be pointed out by simple inspection, although the frame contains the new key. The way to carry out this process is to activate the corresponding *Change Request* flag (CH_REQ), F1, setting it to '1'. The bridge obtains the new key randomly and it is ciphered with the master key in ECB mode and added at the end of the clear data field, before being all ciphered with the session key. There is something more: the MAC is added after the new session key. In fig. 3-7 we can see the modified frame in the renewal process.

If two bridges pretend to agree on a new session key, they cannot break the normal process of ciphering/deciphering frames until they are sure that they share the same key. Then they can use this key.

To avoid any delay because of the process of key renewal, the proposed key is considered as reserve's key. It will be utilized as the actual session key at the next proposed renewal. The interchanged keys are not used immediately with this operation mode. The two involved Cryptonets have the same key (reserve's key) at the time of the renewal, because this key was previously decided (at the last renewal's process).

The bridges will proceed as follows:

a) The bridge that is proposing the change activates the CH_REQ flag and will cipher all the frames with the reserve's key (note that now it is the actual key) since this moment (including this one). The new proposal key is added to this frame in the corresponding data field, previously ciphered with master key, as we said before. It is going to be included in every frame until the renewal's process has ended.

b) The bridge that receives the renewal proposal, when active CH_REQ is detected, switches to the reserve's key (this key is already known by both bridges). Now, it attempts to extract the new key sent by the other bridge.

If the receiver bridge succeeds obtaining the new key (it gets a right MAC's check), will activate the *Change Acknowledge* flag (CH_ACK), set to '1', in all the ciphered frames that it sends to the other bridge since this moment.

When the bridge that is proposing the change detects an active CH_ACK flag, will ceases to activate the CH_REQ flag and to include the new reserve's key. It follows the usual ciphering way since this moment.

Finally, the Cryptonet that is activating the CH_ACK flag makes safe it immediately when stops detecting CH_REQ flag and goes on with the usual ciphering way. The reserve's key has been renewed and there are some frames ciphered with a new key yet, the new session key.
3.4. Obtaining a session DES key

As we said before, when a new key is proposed, only 8 bytes are needed. But we have to dispose of two keys of 8 bytes to utilize the triple CBC ciphered mode at session level. We can see how to get the two keys with a key's renewal of only 8 bytes in fig. 3-8. sense

\[ K_{\text{triple}} = K_1 K_2 K_3^{-1} \]

c) K4 is the MAC. As we know, the MAC is added at the end of the data field. This secret is shared only by the two Cryptonets. With our implementation.

The MAC is established only with 8 bytes of K4 key, as we said before (see 3.3). This key is secret and it is only shared by the two Cryptonets just as we need for our purposes. Really we will take only four bytes of the all 8 bytes for become the MAC. In the next change of keys the process will follow as we have indicated here and we will similarly obtain the new MAC.

The renovation of the only 8 bytes of the total 16 bytes does not jeopardize the system because if a cryptoanalyst does not know the present key, the renewal of only 8 bytes has the same effect as if we dispose of another whole new key.

4. The management Protocol: renewal of Master Keys

As we said before, the ciphering/deciphering of frames is made by CBC mode with triple encryption of DES using session keys. The renewal of these keys is obtained by means of master key and this function is carried out by the two bridges without any additional device.

These master keys also must be renewed. To do this, we use the RSA algorithm. In the network there will be an administrator or manager as we can
see in the figure 4-1. The established protocol allows to work in a separate way each one of the secure bridges (CryptoNets) with the general administrator. It decides the new master key for each pair of CryptoNets and after this it tells them this key by means of RSA algorithm. We have chosen this method because in this way we don't need to dispose of any additional table in each one of the bridges and to keep the public's key of every one of the bridges in the network. This procedure allows us to save a considerable amount of memory if we think that the keys used are 512 bits large. The manager will dispose of a pair of keys (public and private keys) similarly as each one of the CryptoNets. There are some opportunities that these keys could be used in our system, too:

a) In a certain chaotic situations:
   a.1) When there are some continuous errors in the established session between two bridges. For instance, if we consider any mistake in the used session key. The MAC will detect this problem.
   a.2) When we suspect that any intruder is violating our system (the secret or the authenticity). In this case it is necessary to create the new session keys and master key.
   a.3) When we cannot do the renewal of session keys and the licensed number of frames with the actual key is exceeded.

b) When the manager decides to do a reset in the overall system.

5. Conclusions

In this paper we have described a real implementation of a security protocol in an extended LAN environment. The developed protocol have a hierarchical architecture. At the first level we dispose of the session key, in the second level there are the master keys and finally at the highest level there are the public and private key of the manager and the CryptoNet's keys. The relations between them, the right time to do the keys' renewal and the appropriate form to do it are also characteristics that allows to obtain a security based in two algorithms very well known, just as the DES and RSA.

This protocol now is implemented in two secure bridges (CryptoNets) in the U.P.C. extended network and the obtained results allow to conclude that the usual performance offered by this network is not being degraded by these new security devices and the information carried out through this network is protected against some active or passive intruder.

6. References