SECURITY BASED ON CRYPTOGRAPHIC TECHNIQUES FOR REMOTE CONTROL SYSTEMS

Bogdan Groza, Dorina Petrica, Toma-Leonida Dragomir

"Politehnica" University of Timisoara Department of Automation and Applied Informatics
Bd. Vasile Parvan nr. 2, 302223 Timisoara, Romania
E-mail: bogdan.groza@aut.upt.ro, dorina.petrica@aut.upt.ro, toma.dragomir@aut.upt.ro

Abstract: Information exchange has a tremendous impact on modern society. As information became more vital so does the need for stronger security. Security based on cryptographic techniques is commonly used in many applications from many fields. Using security based on cryptographic techniques in remote control systems is certainly a subject of great interest. This paper will try to bring some points of view on the security objectives present in remote control systems and on the cryptographic primitives used to ensure them. Two solutions will be presented to ensure an authentic and confidential channel between the controller and the remote controlled systems.

Keywords: authenticity, confidentiality, secure channel, distributed control system, supervisory control and data acquisition.

1. INTRODUCTION

As long as the need for working with information increases so does the need for security. In this context cryptographic techniques are playing a special role. These techniques are commonly used in banking systems, healthcare institutions, mobile telephony, home-office applications etc.

Automation systems were in the past isolated from public networks. Things are beginning to change and remote control systems now need to communicate over public networks such as the Internet. Therefore the interest for using cryptographic techniques in industrial control systems such as DCS (Distributed Control Systems) or SCADA (Supervisory Control and Data Acquisition) has drastically increased in the last years (Dzung et al., 2005; Wright et al., 2004; Falco et al., 2004)..

Cryptographic techniques are not easy to implement in such environments because encryption requires computational power that is sometime unavailable and also introduces latencies that can became unacceptable.

However, some recent results showed that cryptographic techniques can be successfully used even in constrained environments with low computational power and were communication abilities are drastically limited. Good examples of such environments are sensor networks (Perrig et al., 2001; Liu and Ning, 2002; Du et al., 2005).

Using cryptographic techniques in remote control systems is the subject of this paper. We will investigate some security issues that are present in remote control systems and some cryptographic measurements that can be used. Two simple and efficient solutions based on cryptography will be proposed in this paper in order to ensure the security of a communication channel between a controller and some remote controlled systems. These solutions are based mostly on symmetric primitives which are fast and do not require too much computational power or storage space.

Section 2 defines the security objectives and section 3 introduces the cryptographic primitives. In section 4 we describe the environment and in section 5 symmetric primitives are use to provide and authentic and confidential channel. Section 6 introduces the concept of one-way chain and some aspects regarding the computation of one-way chains. Section 7 shows how one-way chains can improve the security of the authentic and confidential channel. Section 8 holds the conclusion of the paper.

2. SECURITY OBJECTIVES

Security objectives can vary a lot from application to application. We will distinguish between two classes of security objectives: general security objectives and particular security objectives.

General security objectives are required by almost all applications from many different fields, these objectives are the following:
1) *Confidentiality* assures that information can be accessed only by the authorized parties. This is the oldest objective of cryptography and there is a large variety of encryption algorithms to achieve it.
2) *Integrity* assures that information was not altered during transmission. This means that if any intruder modifies the information transmitted the receiver can detect this.
3) *Authentication* can be split in two classes: entity authentication and data authentication. Entity authentication or identification refers to the fact that entities which take part to the communication can prove their identity. Data authentication refers to the fact that the entity which receives information can check that this information was sent by the entity that claims to send it, in fact data authenticity can also guarantee integrity.
4) *Non-repudiation* prevents an entity from denying its previous actions. This means that if dispute arises and some entity pretends that he does not sent particular information then the receiver of the information can prove to any neutral entity that the information was sent by the entity that now denies.

Particular security objectives may appear in some specific applications. The following two objectives are certainly needed in the context of remote control systems and this is mostly because such systems are working in real-time:
5) *Availability* ensures that a particular service is available to its users when requested (Stajano and Ross, 1999). In the context of remote control systems we may translate this in the fact that the remote controlled systems can be assured that the controller is functional.
6) *Data freshness* ensures that received information is fresh (Perrig et al.,2001). This may be interpreted in two ways: first it will be the fact that information can expire after a period of time and secondly it can be the fact that the order in which information packets is received, e.g. command send by the controller, is not altered.

3. THE ENVIROMENT

We will consider the following environment in which a controller sends commands to a number of n remote controlled systems. We will denote the controller as C and the remote controlled systems as $R_i, i=\overline{1,n}$. The nature of the remote controlled systems is not important they can be PLC (Programmable Logic Controller), Display Stations etc. - any-kind of terminal that is able to receive, interpret and confirm commands.

From the perspective of any remote controlled system R_i we want to assure the following security parameters according to the objectives described in section 2:
1) The command received remains confidential between the controller and the remote controlled system to which is intended.
2) The command is authentic – this means that the controller has generated the command and it is intended to the particular remote controlled system, this also ensures integrity of the command.
3) The command has a secure timeline – this means an intruder can not change the order in which commands are received. We will also remark that in such environments other real time conflicts may appear, however this paper deals only with conflicts that can be generated by a possible attacker who would try to mislead the control system by changing the order in which commands are sent.

As from the perspective of the controller C he must be assured that the command was received by R_i. This objective is easy to assure, compared to the previous objectives, because R_i can simply send to C an authentic confirmation message.

Such an environment is suggested in Figure 1, the communication channel from the controller to the remote controlled systems is an authentic and confidential channel while the communication channel from the remote controlled systems to the controller, denoted by a dotted line, is only an authentic channel since it is only required to transmit an authentic confirmation for the received command.

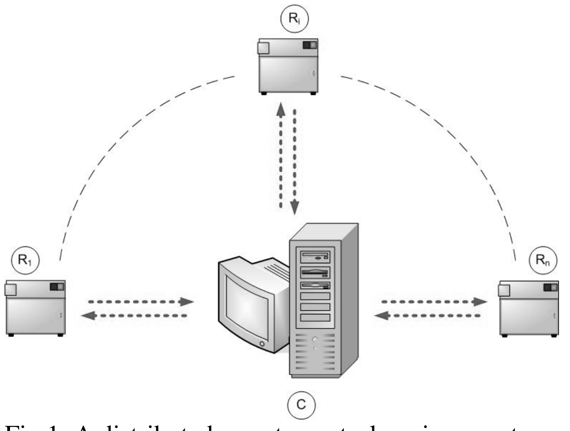

Fig 1. A distributed remote control environment

4. NECESSARY CRYPTOGRAPHIC PRIMITIVES

There is a large variety of cryptographic techniques which may be used to ensure the previous objectives.

Unfortunately the most sophisticated cryptographic techniques are based on asymmetric encryption, or public-key encryption which use different keys for encryption and decryption, and they require more computational power and more storage space than conventional symmetric techniques. By contrary, symmetric techniques, which use the same key for encryption and decryption, require low computational power and low storage space but their use depends on a shared secret which is usually called key.

Asymmetric techniques offer more flexible security but they became impracticable in constrained environments when computational power, storage space or communication abilities are limited. In constrained environments, such as a sensor networks, in order to assure security goals techniques based on symmetric techniques were proposed (Perrig et al., 2001; Liu and Ning, 2002; Du et al., 2005).

Additionally, to have low computational cost and low space requirements we will base the model in the following section only on symmetric primitives. In the context of the security objectives described previously we will consider the following symmetric primitives with the respective notations:

- $E_k(x)$ - symmetric encryption of message x under key k. There is large variety of symmetric encryption algorithms. The preferd method which is the standard for today is AES (Advanced Encryption Standard) (FIPS 197, 2001).
- $H(x)$ - hash function applied over message x. A hash function is a one way function that can be applied on message of arbitrary length and outputs a value of fixed length from which the message can not be recovered. These functions are commonly used to assure data integrity. The most commonly used hash functions are from the SHA (secure Hash Algorithm) family (FIPS 180-2, 2002).
- $MAC_k(x)$ - message authentication code applied on message x with key k, this is a keyed symmetric primitive. Such primitives are used to ensure the authenticity of a message. There is large variety of construction for a MAC (Message Authentication Code). The easiest-one would be to hash a message concatenated with a key, however in order to increase security more complex construction with imbricate hash-functions should be used (Bellare et al., 1996).

5. USING SYMMETRIC PRIMITIVES TO PROVIDE AN AUTHENTIC AND CONFIDENTIAL CHANNEL

For the remote control environment described in section 3 we will use the previously defined cryptographic primitives in order to ensure an authentic and confidential channel.

We will suppose that the controller C has shared secret keys with every remote controlled system $R_i, i = \overline{1,n}$, let the secret keys be $K_{C,R_i}, i = \overline{1,n}$. Every system will need two different keys, one for encryption and one for computing message authentication codes. Let K^E_{C,R_i} be the encryption key and K^M_{C,R_i} the key for the message authentication code. These keys, K^E_{C,R_i} and K^M_{C,R_i}, can be derived from the secret key K_{C,R_i}. The derivation process should be irreversible because otherwise the master key can be recovered by an attacker who manages to break one of the keys. If the process is irreversible then if one of the keys is broken the other is still safe. As an example of such derivation, it is possible to derive keys from the master key by computing $K^E_{C,R_i} = E_{K_{C,R_i}}(r_0)$ and respectively $K^M_{C,R_i} = E_{K_{C,R_i}}(r_1)$ where r_0, r_1 are random values, alternatively the encryption function can be replaced by a MAC (Menezes et al., 1996, p. 568).

We will also suppose that the controller keeps a counter θ_{C,R_i} for every remote controlled system which is incremented after each information exchange. All the remote controlled systems will also independently update their counter after each command correctly received.

In order to sent a confidential and authentic command to a remote controlled system the controller will do the following operations:
1) represent the command as a message M
2) encrypt the command as $E_{K^E_{C,R_i}}(M)$
3) increment the counter $\theta_{C,R_i} = \theta_{C,R_i} + 1$
4) compute the message authentication code for the message M concatenated with the incremented counter θ_{C,R_i} as $MAC_{K^M_{C,R_i}}\left(E_{K^E_{C,R_i}}(M) \| \theta_{C,R_i}\right)$ (the symbol $\|$ denotes concatenation)

The message sent from the controller to the remote controlled systems will be the following:

$$C \to R_i:$$
$$\left\{i, \theta_{C,R_i}, E_{K^E_{C,R_i}}(M), MAC_{K^M_{C,R_i}}\left(E_{K^E_{C,R_i}}(M) \| \theta_{C,R_i}\right)\right\}$$

The remote controlled system has to do the following operations:
1) verify that the message is addressed to it by checking the value of i from the newly received message, if this fails the message is ignored and the remote system will wait for another transmission
2) verify that the counter is fresh (new), this means that the newly received value of the counter is bigger than the last received value of the counter, if this fails

the message is ignored and the remote system will wait for another transmission
3) verify that the message and the counter are authentic and are intended to it by checking the MAC on the encrypted message, if this fails the message is ignored and the remote system will wait for another transmission
4) update the counter with the newly received counter value θ_{C,R_i}
5) decrypt the newly received message and use the information
6) increment the value of the newly received counter $\theta_{C,R_i} = \theta_{C,R_i} + 1$ and then reply to the controller with the new counter value and its MAC:

$$C \rightarrow R_i : \left\{ i, \theta_{C,R_i}, MAC_{K^M_{C,R_i}} \left(\theta_{C,R_i} \right) \right\}$$

The controller will wait for a response and will verify the correctness of such a response by checking that the value of the counter is new and that the MAC is correctly computed. If all this succeeds the command was correctly received and the counter is again updated on the control system side. Otherwise the command was not correctly received and it must be resent.
In Figure 2 an example of one to one communication, in which a controller communicates via a secure channel with a remote controlled system, is suggested (the continuous line denotes that the communication is between C and R_i).

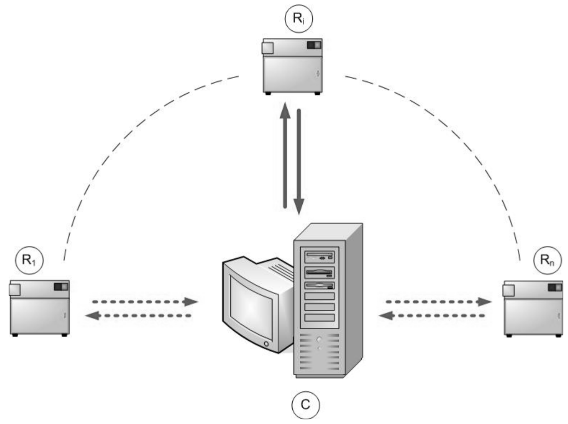

Fig. 2. Example of one to one communication

6. ENTITY AUTHENTICATION WITH ONE-WAY CHAINS

We will now consider the objective of entity authentication, or identification. We authenticate almost every day when we log on a computer, or when we talk through our mobile phone. This is probably the most important objective since the fact that information is not altered or is confidential may remain marginal in the case that an entity does not know to who is talking. The process of authentication is usually based on the existence of a secret which is used to prove someone's identity.

Password based authentication is probably the most common authentication technique. The disadvantage of password based authentication is that it provides only a weak level of security since passwords can be stolen from the system where they are stored or by intercepting user's communication over insecure channels. However the advantages of password based security are great when we consider computational time which is vital in most control systems. A better solution which does not require to much computational power is to use one-time passwords.

One-time passwords are passwords which are valid only once for an authentication. The main advantage in using them is that by disclosing an already used password the user may not be impersonated, since a one-time password may not be used twice.

In (Lamport, 1981) a functional one-time password scheme was proposed in which secrets are stored only on one entity's side and intercepting a password sent from entity to another would not lead to an impersonation since it can't be used twice. Lamport's authentication requires the entity which needs to authenticate to compute the sequence $\{x, F(x), F^1(x), F^2(x), ..., F^{N_A}(x)\}$, where x is an arbitrary value chosen by the entity and kept secret, N_A is the number of authentications to be performed, F is a known one way function. This sequence is also called a one-way chain.

Initially the entity to which identity is proven must know $F^{N_A}(x)$ and then when the other entity needs to authenticate for the first time ($i = 1$) it will present $F^{N_A - 1}(x)$ as the first one-time password. The authenticity of this password can be verified by checking that $F(F^{N_A - 1}(x)) = F^{N_A}(x)$ and if this proves to be correct than $F^{N_A}(x)$ will be replaced by $F^{N_A - 1}(x)$. At the i^{th} authentication the entity will prove it's identity by sending $F^{N_A - i}(x)$ and the other entity will simply verify the authenticity by computing $F(F^{N_A - i}(x))$ and also checking that $F(F^{N_A - i}(x)) = F^{N_A - i + 1}(x)$, where $F^{N_A - i + 1}(x)$ is the previous authentic one time password, again if this proves to be correct $F^{N_A - i + 1}(x)$ is replaced by $F^{N_A - i}(x)$.

One-way chains have many applications in authentication, for example they are used in the S-Key system to authenticate users (Haller et al., 1998) or in an electronic payment scheme to authenticate transactions (Rivest and Shamir, 1996).

Since computational speed can became vital in remote controlled systems constructing such a one-way chain may also raise some issues. There are mainly two solutions to construct a one way chain:

A) *Constructing one-way chains from symmetric primitives.* Using symmetric primitives, and more exactly hash functions, has the advantage that they are fast to compute but their use in Lamport's scheme also has a major disadvantage in the fact that the one-way chain has a fixed length – if all the values are used then it will be impossible to generate new values since the chain is irreversible. If the length of the chain is chosen too large than it requires more computational power if it is too short then it can be exhausted too quickly.

B) *Constructing one way chains from asymmetric primitives.* Using primitives from public key encryption, and more exactly functions over groups of integers, has the advantage that the length of the chain can take almost "infinite" values and the chain is never exhausted. However even if these primitives offer more flexible security their computational cost is also higher. The notion of infinite length hash chain was introduced in (Bicakci and Baykal, 2002). The use of functions from public key encryption was discussed in (Groza and Petrica, 2005a), and an optimized solution which significantly reduces computational requirements is in (Groza et al., 2005b).

In the next section we will discuss how one-way chains can be used to improve the security of the communication channel.

7. USING ONE-WAY CHAINS TO IMPROVE THE SECURE CHANNEL

We will now reconsider the construction proposed in section 5. Notice that in this construction the message has a significant meaning only for the remote system to which is intended and can decrypt it. But it may be useful for the other systems to know that at least the controller is on-line - in this way assuring temporary availability of the controller. In order to guarantee to all remote systems that the controller is on-line the direct solution will be to send an authentic message to all remote systems – this means that the controller has to authenticate simultaneously to all systems. In order to implement this it will require computing a MAC for every remote system and one message for every remote system. The same objective can be achieved more elegantly with only one message to all remote systems if one-way chains are used to authenticate the controller to the remote systems.

Instead of the counters θ_{C,R_i} which the controller shares with each remote controlled system, a one-way chain based counter θ_{OWC} will be used with all remote systems. Assume that this sequence is generated on the controller side $\{x, F(x), F^1(x), F^2(x),..., F^j(x),..., F^\eta(x)\}$ for a sufficiently large value η and a random value x. In the initialization stage the controller will share with each remote system the value of $\theta_{OWC} = F^\eta(x)$ then each value from the sequence will be used as a new value for the one-way chain based counter in each communication session. In this way the counter became an authentic counter.

In order to send a confidential and authentic command in the j^{th} session to the remote controlled systems the controller will do the following operations:

1) represent the command as a message M
2) encrypt the command as $E_{K^E_{C,R_i}}(M)$
3) compute the new value for the one-way chain based counter $F^j(x)$
4) compute the message authentication code for the message M concatenated with the new value of the one-way chain based counter $F^j(x)$

The following new structure for the message will result:

$$C \to R_i: \\ \left\{ i, F^j(x), E_{K^E_{C,R_i}}(M), MAC_{K^M_{C,R_i}}\left(E_{K^E_{C,R_i}}(M) \,\|\, F^j(x) \right) \right\}$$

This message can be broadcast to all the remote systems, but only the system to which the MAC on the encrypted command corresponds can decrypt the message. All other systems can verify that the counter is correct and therefore can be assured that the controller is able to function.

Each remote system has to do as follows:

1) verify that $F\left(F^j(x)\right) = \theta_{OWC}$ and if this is true then the controller is on-line and also set $\theta_{OWC} = F^j(x)$, if this fails the message is ignored and the remote system will wait for another transmission
2) verify that the message is addressed to it by checking the value of i from the newly received message, if this fails the message is ignored and the remote system will wait for another transmission
3) verify that the message is authentic and is intended to it by verifying the MAC on the encrypted message, if this fails the message is ignored and the remote system will wait for another transmission
4) decrypt the message and use the information
5) the remote system that has successfully recover the message can reply to the controller with:

$$R_i \to C: \left\{ i, \theta_{OWC}, MAC_{K^M_{C,R_i}}(\theta_{OWC}) \right\}$$

The controller will wait for a response and if the response is received it will verify the authenticity of the response by checking the MAC of the received message. Otherwise, in the case that a response is not received or the response is not authentic, he will resend the same command.

However by sending the same command with the same value for the counter θ_{OWC} the availability of the controller is not guarantee to the other remote systems since this value was already sent, so it is recommended that new values for the counter to be used. Finally, if all the systems can have synchronized clocks it will be preferable to associate each value from the one-way chain with a particular time interval and disclosed the value at the respective time, this solution is used in (Perrig et al., 2001).

The following problem may occur: if one package is lost by any of the remote systems then it will fail to verify that $F\left(F^{j}(x)\right)=\theta_{OWC}$. In order to remove this shortcoming the remote system will have to verify that $\underbrace{F\left(...F\left(F^{j}(x)\right)...\right)}_{k}=\theta_{OWC}$ for some value of is k (this is exactly the number of packages that are lost).

An example of one to many communication, in which a controller communicates via a secure channel with one remote controlled system and broadcasts authentic messages to the other remote controlled systems to ensure its availability, is suggested in Figure 3.

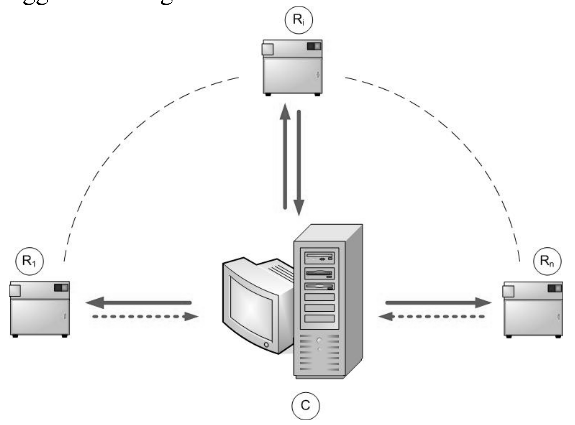

Fig. 3. An example of one to many communication

8. CONCLUSION

Defining security objectives on remote control systems is certainly a problem of interest. In this paper two solutions were proposed by which a controller can send confidential and authentic commands to a number of remote systems and also ensure them of his availability. We expect that these solutions are secure and are suitable to be used in practice.

REFERENCES

Bellare, M., Canetti, R., Krawczyk, H., (1996) *Keying Hash Functions for Message Authentication*, Advances in Cryptology – CRYPTO 96, LNCS vol. 1109, Springer-Verlag.

Bicakci, K., Baykal, N., (2002) *Infinite Length Hash Chains and Their Application*, IEEE 11th International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)

Du, W., Wang, R., Ning, P. (2005) *An Efficient Scheme for Authenticationg Public Keys in Sensor Networks.* Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005).

Dzung, D., Naedele, M., Hoff, T.P., Crevatin, M., (2005) *Security for Industrial Communication Systems*, Proceedings of the IEEE, vol. 93, no. 6, June 2005.

Falco, J., Gilsinn, J., Stouffer, K. (2004) *IT Security for Industrial Control Systems: Requirements Specification and Performance Testing*, NDIA Homeland Security Symposium & Exhibition 2004

FIPS 197, (2001) *Announcing the Advanced Encryption Standard.* http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS 180-2, (2002) *Secure Hash Standard*, http://csrc.nist.gov/publications/fips/fips180/fips-180-2.pdf

Groza, B., Petrica, D., (2005) *One-time passwords for uncertain number of authentications*, in Proceedings of CSCS15.

Groza, B., Petrica, D., Dragomir T.L., (2005) *A time-memory trade solution to generate one-time passwords using quadratic residues in Zn,* to be published in Studies in Informatics and Control.

Haller, N., Metz, C., Nesser, P., Straw, M., (1998). *A One-Time Password System.* RFC 2289, Bellcore, Kaman Sciences Corporation, Nesser and Nesser Consulting.

Lamport, L., (1981). *Password Authentication with Insecure Communication.* Communication of the ACM, 24, 770-772.

Liu, D., Ning, P., (2002) *Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks.* Proceedings of the 10th Annual Network and Distributed System Security Symposium.

Menezes, A.J., van Oorschot, P.C., Vanstone, S.A., (1996). *Handbook of Applied Cryptography*. CRC Press.

Perrig, A., Szewczyk, R., Wen, V., Culler D., Tygar, J.D., (2001) *SPINS: Security Protocols for Sensor Network,* Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM 2001.

Rivest, R., Shamir, A. (1996) *Payword and Micromint: Two simple micropayment schemes.* CryptoBytes, volume 2, no. 1, RSA Laboratories

Stajano, F., Ross, A., (1999) *The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks*, 7th International Workshop on Security Protocols, Cambridge, UK

Wright, A.K., Kinast, J.A., McCarty, J.,(2004) *Low-Latency Cryptographic Protection for SCADA Communications*, Proceedings of Applied Cryptography and Network Security 2004