On the use of one-way chain based authentication protocols in secure control systems

Bogdan Groza, Toma-Leonida Dragomir
Politehnica University of Timisoara, Romania,
Faculty of Automatics and Computers
{bogdan.groza, toma.dragomir}@aut.upt.ro

Abstract

The use of one-way chains in authentication protocols is a technique of great importance which has many applications. Employing cryptographic techniques in the area of industrial control systems has gained significant interest in the last few years. This paper proposes the use of a one-way chain based authentication protocol in a robust control system. Some enhancements of a generic one-way chain based authentication protocol are required by the scenario of secure robust and they are intended for achieving lower authentication delays and computational costs while preserving the control robust in the presence of potential attackers. We also underline that the techniques described in this paper are not restricted to the subject of authentication in robust control systems since the presented protocols may be useful for other applications as well.

1. Introduction

Cryptographic techniques are playing an important role in assuring information security since cryptography offers the only security guarantee when we are working with information. Among security objectives, authentication is essential since other objectives may not be relevant as long as there is no guarantee over the source and integrity of information.

The use of one-way chains in authentication was initially proposed by Lamport in order to achieve entity authentication in a one-time password scheme [14]. However the use of one-way chains in performing entity authentication is not a very successful technique and this is mostly because of its shortcomings compared to more advanced authentication protocols such as challenge response protocols. One implementation of such a scheme is in the S-Key system by Haller [11], other one-time password schemes with distant relation to the proposal of Lamport can be found in [4], [16].

Beside this, the use of one-way chains has gained a significant impact in the last few years in the area of message authentication. This is mostly due to the success of the scheme proposed by Perrig et. al. which uses elements of one-way chains as keys for Message Authentication Codes (MAC) and time synchronization in order to assure the authenticity of broadcast information [17], [18], [19], [20]. In general, authentication schemes based on one-way chains have advantages that are close to those of schemes based on public key encryption but offer reduced computational costs since they can be built over some of the simplest one-way functions such as hash functions.

In the past, industrial control systems were isolated from public networks and their security has relied on the obscurity of custom made protocols or secure perimeters. Things have changed in the last few years, control systems now need to interact over public networks such as the Internet in this way being exposed to threats and security guarantees are needed. Therefore, the use of cryptography is becoming essential in this area. However, implementing cryptography for this purpose encounters one major issue: cryptography requires computational power which may be unavailable. In this context we believe that the use of one-way chain based authentication is of great importance for the applications in this field, since such protocols have low computational power requirements. This paper is concerned with some theoretical aspects for applying one such protocol on a control system, implementation details are subject of future work. In large, implementing such a protocol should not raise problems since the protocol is constructed on simple one-way functions which are not computational intensive - it is notable that such kind of protocols were implemented even in constrained environments such as sensor networks [17].

The protocol that we propose in this paper is an adaptation of a one-way chain based authentication protocol [9] in order to be used in a robust control scenario. Several significant improvements are

required by the scenario of secure robust control and they are intended for achieving lower authentication delays and computational costs while preserving the control robust in the presence of potential attackers. Several properties are added to the protocol while the security objectives from [9] are preserved. Each of the proposed protocols has the following properties: i) it is based only on one-way functions which do not require significant computational resources and therefore it can be used in a large variety of environments ii) it provides data authenticity which implies that information was not altered and originates from a particular entity iii) it does not depend on shared secrets; every entity stores only its own secrets (this means that the proposed protocol is not based on a secret key model, however compromising any secret from any side leads to security loss on both sides so the proposed protocol is not a public-key model either) iv) it does not require time synchronization between entities involved in a communication, e.g. there is no need for any entity to keep a secure clock on its side which is loosely synchronized with others entities clocks v) it provides a secure time-line of messages which implies that messages arrive in the order they were sent and their order cannot be switched by an intruder.

Besides these objectives we improve on our initial schemes in order to achieve the following objectives that are particular to our scenario: vi) Shorter authentication delays - by the nature of these authentication schemes delays are required in order for the authentication to be secure (for the proposed scenario with the Delayed Message Authentication protocol from [9] a delay of 4 rounds is needed, the new proposal reduces this delay to 2 rounds) vii) reducing the required computational power by eliminating any unnecessary cryptographic operation. A simplified variant of the protocol is proposed in which some cryptographic operations are removed while keeping the control robust viii) obtaining supplementary information over the authenticity of the commands and responses (this objective is in contradiction with the previous ones since we need more computational power for this; therefore we propose a distinct scheme for this purpose) ix) keeping the control robust in the presence of potential attackers (this is what we call secure robust control and is achieved by assuring the authenticity of the exchanged information).

In section 2 related work on protocols constructed on one-way chains to assure information authenticity is discussed. Section 3 outlines the concept of control system and the robust control scenario that we address and introduces our proposal; also some security issues are discussed. Section 4 holds the conclusions of our paper. For the completeness of our paper in appendix A some details are given on methods and optimizations in the construction of one-way chains; although the notion of hash chain is more widely used in this paper we will use the notion of one-way chain since in fact any one-way function can be used for this purpose.

2. Related work on authentication protocols based on one-way chains

Several authentication protocols which use elements of one-way chains as keys for MAC codes were proposed in the last few years [2], [17], [18], [19], [20]. The common idea on which all these protocols are based is to sign a message with a key that is committed in the same communication round and disclosed only in a forthcoming communication round. In principle, such an authentication protocol is secure as long as there is a guarantee that at the moment when the packet was received the key used to compute the message authentication code was not yet disclosed. In brief, in order to assure this security condition a time synchronization is used in [18], [19], [20] while in [2], [9] an authentic confirmation is used, this can be viewed as a challenge response mechanism. The difference between the proposals from [2] and [9] is at the nature of the communication which is a multiparty communication in [2] and a one to one communication in [9] and also at the initialization of the chain which in [9] is achieved also by the use of one-way chains while in [2] a generic one-time signature is required.

The Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol and several variants of this protocol were proposed by Perrig et. al. [17], [18], [19], [20]. All the proposed protocols rely on loose time synchronization, which means that the receivers must have an upper bound on the time from the side of the sender. The security condition which must be meet to make this authentication secure is the following: a packet P_i arrive safely if the receiver can unambiguously decide based on its synchronized time that the sender did not yet send the key disclosure packet $P_j, j > i$. The basic scheme from [20] does not make use of one-way chains. The principle on which the scheme is based, which makes it similar with the Guy Fawkes Protocol [1], is to issue a commitment for a random key which is used to compute MAC on some packet P_i. This key will be disclosed only later when packet P_{i+1} is sent and only then packet P_i can be authenticated, packet P_{i+1} also contains the commitment of a new random key and so on. It is easy to note that authenticity can not be achieved any more after some packet is lost. In order to improve on this

the use of one way chains is introduced in the packet loss tolerant scheme [20]. By replacing the random keys used to authenticate each packet with elements from a one way chain, even if several packets are lost the forthcoming packets can be authenticated since the lost keys can be generated from a newly received packet. Several variants of these protocols are proposed in order to: achieve fast transfer rates, deal with dynamic packet rates, immediate authentication, concurrent TESLA instances etc.

The Chained Stream Authentication (CSA) removes the requirement of time synchronization [2]. Instead of requiring time synchronization a confirmation is required (i.e. a challenge-response mechanism) for each received packet and this confirmation value is also a value from a one-way chain - a new key is released only when the arrival of the previous key is confirmed. Three variants of this protocol are proposed in [2]. The Interactive Chained Stream Authentication (I-CSA) is a chained stream authentication with one sender and one receiver, each of the two entities commit initially a new hash chain and then in each session an element from the hash chain is disclosed while the next element from the hash chain is used to compute the message authentication code on the present packet. For a multiparty communication the N-party I-CSA is proposed, this variant consists in the application of I-CSA between one sender and multiple receivers. This means that the sender waits from a confirmation from each side. However this variant is likely to be inefficient since it is impossible to send a new authentication key until a response is received from every entity and therefore the protocol succumbs when one entity fails to respond. The Timed Chained Stream Authentication T-CSA is similar to the TESLA Protocol and is based also on a time delayed disclosure of the session keys.

The Delayed Message Authentication/Direct Chain Authentication (DeMA/DiCA) protocol uses a one-way chain on each side to exchange authentic information and two one-way chains on each side for re-initialization of the chains [9]. The DeMA component of the protocol is similar to the CSA protocol. The DiCA component of the protocol is used for the re-initialization of the one-way chains; in essence this protocol is a chained one-time signature. With respect to all previous work on one-way chain authentication protocols the DeMA/DiCA protocol is the only protocol that is based exclusively on one-way chains (for example the CSA protocol also requires a generic one-time signature for the re-initialization of the chain).

3. The use of one-way chain authentication in secure control systems

3.1. The addressed scenario

The importance of security based on cryptographic techniques for industrial control systems is widely acknowledged, a survey on the subject can be found in [6], but there are a lot of papers that address the same problem. Control systems usually have many characteristics or requirements that are different from traditional processing systems; in the last few years there is a constant effort to standardize these specifications [7].

In brief, a control system consists in a device called controller which commands (regulates) the behavior of another system which is called controlled process (the system being controlled is usually called plant). The regulation is done via a control rule which is a function that takes responses from the process as inputs and outputs a command that is sent to the process. In figure 1 the structure of a generic remote control system is depicted.

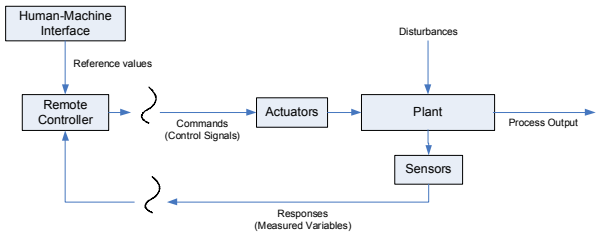

Figure 1. Generic Control System

Therefore for the rest of this paper we will be concerned in assuring security in the communication between two participants: a controller and a controlled process. The information exchanged between them will be denoted as command when it goes from the controller to the controlled process and response when it goes from the remote process to the controller.

3.2. The problems that we address and the relevance of the proposed solution

A controller is defined as robust if it operates effectively over all possible operating conditions. In large, the robust control problem is to find a control law which maintains system response and error signals within prescribed tolerances despite the effects of uncertainty on the systems [3]. Forms of uncertainties include: disturbance effects, measurement noise, modeling errors due to nonlinearities or time-varying parameters [3]. Obviously, when controllers will

operate over public networks, the operating conditions of a controller may assume as well the existence of intruders and attackers which can take actions such as: interception and modification of any number of messages. Dealing with intruders on the transmission lines is the concern of cryptology and not of control theory. Therefore we will not alter the notion of robust controller but we extend it by that of secure robust controller. We call a robust controller secure if it remains robust despite the interferences of malicious attackers. Obviously, the security issues occur in the transmission of the information between the controller and the controlled process (both on the forward line and on the feed-back line). We are not concerned with delays or with the possible loss of the communication since there is no cryptographic countermeasure against such situations, finally dealing with this is again the problem of robust control (usually, a local controller is present to undertake the control operation in the case when the communication is lost with the remote controller, this is called a fault tolerant system).

Therefore, we address the problem of preserving the robust control secure in the presence of potential attackers that may alter the information sent between the controller and the controlled process. We remark that a robust controller is secure under two distinct circumstances: a) an attacker can not alter the authenticity and time-line of both the commands and the responses b) the controlled process can unambiguously decide if the command is authentic and was computed on the current state of the process and the time-line of the commands can not be altered. We try to explain both these circumstances. For both cases a requirement over the authenticity and time line of the command is obviously needed. However this may not be enough since an attacker that can alter the authenticity or time line of the responses from the process can finally trigger any command from the controller - therefore a security guarantee over the responses is requested in the first case. For the second case, even if the authenticity and time line of the response is not known as long as the controlled process can decide if the command is authentic and that it was computed on the current state of the process then the controller remains robust since a fraudulent command will not be accepted. We underline that both these circumstances are sufficient, if an attacker alters authenticity, commands and responses that are not authentic are not accepted and this results only in communication delays that are not going to affect the robust control from the security point of view.

The proposed solution consists in assuring the authenticity of the information exchanged between the controller and the controlled process by using a one-way chain based authentication protocol. Although there are several papers that address the problem of authentication by the use of one-way chains, all these proposals are not suited for assuring authenticity in the scenario of a robust control system. This is mostly because of the fact that all these proposals address the problem of broadcast or multi-cast scenarios where information is broadcasted to multiple entities without requiring their confirmation upon the received message (an exception is one variant of the CSA protocol [2]). A robust control scenario is directly linked to the concept of feed-back (in fact control theory can be referred to as the theory of feed-back) and because of this a one-way communication cannot be used. Also it is likely that the robust control scenario will require a one to one communication without involving multiple receivers which simplifies the problem and makes possible some modifications of the protocols in order to reduce the computational and communication overhead. In the next section we will introduce several variants of a one-way chain based authentication protocol which can be used for this purpose. The use of a one-way chain based authentication protocol has at least the following advantages: i) the security of this kind of protocols is well established ii) one-way chain based protocols are efficient since they can be constructed on simple one-way functions that are easy to compute iii) they provide the best alternative when expensive public key encryption is not affordable and shared secrets are not available.

3.3. A straightforward approach

As a direct approach, we consider the use of the DeMA/DiCA protocol in the addressed scenario. The protocol assumes the existence of two participants A and B. Between A and B a variable number of communication sessions is taking place and each such session consists in exactly two rounds. The keys from each round of the DeMA protocol are defined for A and B as follows:

$$\sigma_A(k) = f^{\eta-k}(x_A), 0 \le k \le \eta \qquad (1)$$

$$\sigma_B(k) = f^{\eta-k}(x_B), 0 \le k \le \eta \qquad (2)$$

Here k is the session number, η is the length of the one-way chains (also the number of sessions that can be performed with these chains), and x_A, x_B are randomly chosen values that are kept secret on each side. In session 0 the entities inform each other, in a secure manner to guarantee the authenticity of this information, of the values of $\sigma_A(0)$ and $\sigma_B(0)$ respectively which are the tips of the one-way chains (this can be done either in an offline initialization stage or by using some authenticated key exchange

protocol). Each session of the DeMA protocol is as follows:

Session $k, 1 \le k \le \eta$

$A \to B:\ M_{A,k}, MAC_{\sigma_A(k+1)}\left(M_{A,k}\right), \sigma_A(k)$

$B \to A:\ M_{B,k}, MAC_{\sigma_B(k+1)}\left(M_{B,k}\right), \sigma_B(k)$

After the exhaustion of the η keys the DiCA protocol can be used to re-initialize the one-way chains, details on the DeMA/DiCA protocol are in [9] and for this paper they are not relevant.

In order to enforce the security of the protocol we will use as key for the MAC sent in each round a key that is derived from the forthcomming session key with a generic key derivation process KD. This will assure that the session key and the key of the MAC code are cryptographically independent. Therefore the two rounds of each session became:

Session $k, 1 \le k \le \eta$

$A \to B:\ M_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(M_{A,k}\right), \sigma_A(k)$

$B \to A:\ M_{B,k}, MAC_{KD(\sigma_B(k+1))}\left(M_{B,k}\right), \sigma_B(k)$

In what follows the controller will be denoted as A while the controlled process will be denoted as B, the command issued in session k is $c_{A,k}$ while the response is $r_{B,k}$ - due to the authentication delay the authenticity of these values can be proved only after the next session. The keys from each session of the protocol are computed with relations (1), (2).

For the clarity of the exposition we now write the protocol steps for four consecutive sessions between the controller and the controlled process:

Session k Standard variant

$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}\right), \sigma_A(k)$

$B \to A: r_{B,k}, MAC_{KD(\sigma_B(k+1))}\left(r_{B,k}\right), \sigma_B(k)$

Session $k+1$

$A \to B:\ c_{A,k+1}, MAC_{KD(\sigma_A(k+2))}\left(c_{A,k+1}\right), \sigma_A(k+1)$

$B \to A: r_{B,k+1}, MAC_{KD(\sigma_B(k+2))}\left(r_{B,k+1}\right), \sigma_B(k+1)$

Session $k+2$

$A \to B:\ c_{A,k+2}, MAC_{KD(\sigma_A(k+3))}\left(c_{A,k+2}\right), \sigma_A(k+2)$

$B \to A: r_{B,k+2}, MAC_{KD(\sigma_B(k+3))}\left(r_{B,k+2}\right), \sigma_B(k+2)$

Session $k+3$

$A \to B:\ c_{A,k+3}, MAC_{KD(\sigma_A(k+4))}\left(c_{A,k+3}\right), \sigma_A(k+3)$

$B \to A: r_{B,k+3}, MAC_{KD(\sigma_B(k+4))}\left(r_{B,k+3}\right), \sigma_B(k+3)$

On a first view it will be reasonable to decide that the command which corresponds to the response $r_{B,k}$ can be emitted by the controller only after the authenticity of this value is proved. But if the controller waits until round 2 of session $k+1$, to test the authenticity of the response $r_{B,k}$, then the command can be send only in round 1 of session $k+2$ and the authenticity of this command can be tested by the controlled process only after round 1 of session $k+3$; this will cause a delay of four rounds. We underline here that authentication delays can be also solved from the design of the controller, dealing with authentication delays in the design of a controller is potential subject for our future work; finally for slow processes delays may not be relevant.

3.4. An improved solution

Due to the nature of the scenario in which the protocol is to be used, a robust control scenario, some significant improvements can be done on the initial approach. There are two distinct objectives which we will separately achieve by these improvements: reducing the authentication delay and reducing the computational costs. We underline that the order in which the packets are sent, i.e. the time line, is secure due to the use of the session keys which are elements of a one-way chain and therefore a possible intruder cannot alter it.

First, we want to reduce the authentication delay. Fortunately the delay of four rounds can be significantly reduced by binding the response with the corresponding command. We remark that upon receiving the response from the controlled process in session k the controller can already compute the command that will be sent to the controlled process, of course since the response value is not yet authentic the command can be wrong but if this happens the controlled process will observe this when checking the authenticity of the command which is bound to the response and will simply omit the command if authenticity fails. To bind the command with the response is easy and can be achieved by computing a MAC on both these values, therefore instead of $MAC_{KD(\sigma_A(k+2))}\left(c_{A,k+1}\right)$ the controller will compute $MAC_{KD(\sigma_A(k+2))}\left(c_{A,k+1}, r_{B,k}\right)$ - this binds the authenticity of the command with the authenticity of the response. Therefore it is a good option to choose to send the command immediately in session $k+1$ and upon

receiving the authentication key in session $k+2$ the controlled process can decide if the command from session $k+1$ corresponds to the response from session k and is authentic. This will change the rounds of the authentication protocol in the following:

Session $k, 1 \le k \le \eta$ Direct Command Variant (DCV)

$$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}, r_{B,k-1}\right), \sigma_A(k)$$

$$B \to A: r_{B,k}, MAC_{KD(\sigma_B(k+1))}\left(r_{B,k}\right), \sigma_B(k)$$

By binding the command with the response in the Direct Command Variant (DCV) of the protocol the authentication delay can be reduced to two rounds; i.e. the authentication of the command issued for the response from session $k-1$ is achieved after round 1 of session $k+1$.

Since computational power is a major issue in robust control we can simplify further the DCV variant. The computations performed on the side of the controlled process can be reduced by removing the MAC on the response. Although indeed the controller can not test the authenticity of the responses, this will not affect the robust control since the command will be accepted by the controlled process only if it was bound to the correct response. This results in the following:

Session $k, 1 \le k \le \eta$, Simplified variant (SV)

$$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}, r_{B,k-1}\right), \sigma_A(k)$$

$$B \to A: r_{B,k}, \sigma_B(k)$$

In the Simplified variant the controller can never check if the received response is correct, which obstructs the observability of the system since the process states can not be tested for authenticity any more, but the controlled process can verify that the command was computed on the correct response and will not accept the command otherwise - the controller robustness is preserved.

It may be also important in some conditions for both the controller and the remote process to check if the command or the response were correctly received, for example, the human-machine interface usually includes measurement displays and also on the controlled process status led are sometimes available. For the case when both this requirements must be reached we propose a complete variant which allows both the controller and the controlled process to check what part of the message was incorrectly received either the command or the response. In order to achieve this, one MAC is computed separately on the command and on the response, the following are the rounds for this variant:

Session $k, 1 \le k \le \eta$, Complete variant (CV)

$$A \to B$$

$$c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}\right), MAC_{KD(\sigma_A(k+1))}\left(r_{B,k-1}\right), \sigma_A(k)$$

$$B \to A:$$

$$r_{B,k}, MAC_{KD(\sigma_B(k+1))}\left(c_{A,k}\right), MAC_{KD(\sigma_B(k+1))}\left(r_{B,k}\right), \sigma_B(k)$$

In this variant both the controller and the controlled process can individually check the authenticity of both the command and response at a delay of two rounds. The same can be achieved in the standard variant without requiring the computation of one additional MAC but at the delay of four rounds, when reduced communication delays are more important than computational power the CV variant shall be preferred.

For the completeness of our description a method for the initialization of the one-way chain is needed. The method must guarantee between the parties which are initializing their chains that the new chains were not used before and that they are indeed generated by their claimants, finally any protocol for assuring an authentic key exchange can be used. In [10] the following protocol is used for this purpose: $A \to B$: $\{ A, B, N_A, Sig_A(A, B, N_A) \}$; $B \to A$: $\{ B, A, N_B, Sig_B(A, B, N_A, N_B) \}$; $A \to B$: $\{ A, \sigma_A(0), Sig_A(A, B, N_A, N_B, \sigma_A(0)) \}$; $B \to A$: $\{ B, \sigma_B(0), Sig_B(A, B, N_A, N_B, \sigma_B(0)) \}$. Here Sig_A, Sig_B denotes a digital signature performed by the controller A and the controlled process B while N_A, N_B are two nonces to ensure the uniqueness of this communication. Finally, the Direct Chain Authentication protocol proposed in [9] can be also used for the same purpose.

3.5. Some security issues

Some of the proposals of one-way chain based authentication have encountered different security issues. For the completeness of our paper we underline why they do not represent a concern for our proposal.

A pre-play attack on the S-Key one-time password system [11] is suggested in Note 10.7 from [15], the attack consist in storing multiple elements of the one-way chain (which play the role of passwords in the S-Key system) for subsequent use in the impersonation of the user. Such an attack does not apply since in the proposed protocol only one key is revealed and a new one will be released only when a confirmation is

received from the other side. The best thing an attacker can do is to alter the messages sent between the entities but these messages will prove in the next session not to be authentic. In [18] a man-in-the-middle attack on one variant of the CSA protocol is described, this attack is removed by the CSA proposal from [2]. Such an attack is not possible on the initial proposal of the DeMA/DiCA protocol from [9] because the addressed scenario had only one sender and one receiver and the chains are committed in an off-line initialization stage. Also the man-in-the-middle attack from [18] is not feasible on the commitment of the chains from the previous section and from [10].

An informal proof of security may be useful. The security of the proposed protocols is based on the fact that given $M_{A,k}, MAC_{KD(\sigma_A(k+1))}(M_{A,k}), \sigma_A(k)$ it is not possible to compute a $MAC_{KD(\sigma_A(k+1))}(M')$ for any $M' \neq M_{A,k}$ (standard assumptions are made for the MAC code: it must be a function family $f_K(x), K \in \{0,1\}^l$, where l is the security parameter, unforgeable under an adaptively chosen message attack). Because function f is one-way, from $\sigma_A(k) = f^{\eta-k}(x_A)$ one can not compute $\sigma_A(k+1) = f^{\eta-k-1}(x_A) = f^{-1}(\sigma_A(k))$ and therefore it is obvious that the key remains secret and the MAC cannot be forged. Therefore the only guarantee that is needed is that the key $\sigma_A(k+1)$ was not released at the time when the package $M_{A,k}, MAC_{KD(\sigma_A(k+1))}(M_{A,k}), \sigma_A(k)$ is received. This is guaranteed since A will release $\sigma_A(k+1)$ only when the confirmation $\sigma_B(k)$ is received from B and obviously a potential attacker cannot forge the value of $\sigma_B(k)$ since again function f is one way. Formal proofs of security for one-way chain based protocols can be found in [2], [20] and we believe that it should be straight forward to give a similar proof for the proposed protocol.

3.6. Implementation aspects

Control systems are usually implemented on microcontrollers since they are cheap, however in modern days networked control systems are used which combine control devices with computers that communicate over public networks such as the Internet (recent papers address the problem of control over such communication links [12]). According to figure 1 the implementation of a control system requires the presence of two equipments: one for the implementation of the controller and the other for the controlled process. Since the communication over a public network is assumed, on both sides the presence of standard computers is needed – control devices can be embedded or connected to these computers. Any one-way function can be used in the protocol, for example a hash function, and the MAC code is also one of the cheapest cryptographic primitives – therefore there should be no computational constraints in implementing this protocol. The proposed solution can be implemented on any standard computer that communicates over public networks or even on more constrained environments such as microcontrollers.

4. Conclusions

The use of one-way chains in assuring information authenticity was discussed and several protocol variants are proposed. The addressed scenario consists in a robust control system and as far as we known this is the first paper which proposes the use of one-way chain authentication for such purpose. Because these protocols require low computational resources (they can be built entirely on simple one-way functions such as hash functions) we believe that is likely that they will have good perspectives for practical use. As future work within our research group we plan to do some practical implementation of the proposed protocols in a concrete robust control scenario, this will be useful in order to test the performance of these protocols in practice.

Acknowledgements: This work was partially supported by national research grant MEDC-CNCSIS TD-122/2007.

5. References

[1] R. Anderson, F. Bergadano, B. Crispo, J.H. Lee, C. Manifavas, R. Needham, "A New Family of Authentication Protocols", ACM OSR, 1998.

[2] F. Bergadano, D. Cavagnino, B. Crispo, "Individual Authentication in Multiparty Communications". Computer & Security, Elsevier Science, vol. 21 n. 8, 2002, pp.719-735.

[3] Burns, R., *Advanced Control Engineering*, ISBN: 0-7506-5100-8, 464 pages, Elsevier, 2001.

[4] H.-Y. Chien, J.-K. Jan, "Robust and Simple Authentication Protocol", Oxford Journal, The Computer Journal, Vol. 46, No. 2, 2003.

[5] D. Coppersmith, M. Jakobsson, "Almost Optimal Hash Sequence Traversal", 2003.

[6] D. Dzung, M. Naedele, T.P. Hoff, M. Crevatin, "Security for Industrial Communication Systems", Proceedings of the IEEE, vol. 93, no. 6, 2005.

[7] J. Falco, J. Gilsinn, K. Stouffer, "IT Security for Industrial Control Systems: Requirements Specification and Performance Testing", NDIA Homeland Security Symposium & Exhibition, 2004.

[8] M. Fischlin, "Fast Verification of Hash Chains", 2004.

[9] B. Groza, "Using one-way chains to provide message authentication without shared secrets", Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPerU 2006, IEEE, 2006.

[10] B. Groza, T.-L. Dragomir, D. Petrica, "Using the discrete squaring function in the delayed message authentication protocol", International Conference on Internet Surveillance and Protection, ICISP'06, IEEE, 2006.

[11] N. Haller, C. Metz, P. Nesser, M. Straw, "A One-Time Password System", RFC 2289, Bellcore, Kaman Sciences Corporation, Nesser and Nesser Consulting, 1998.

[12] O. C. Imer, S. Yuksel, T. Basar, "Optimal control of lti systems over unreliable communication links", Automatica, (42), 2006.

[13] M. Jakobsson, "Fractal hash sequence representation and traversal", 2002.

[14] L. Lamport, "Password Authentication with Insecure Communication", Communication of the ACM, 24, 770-772, 1981.

[15] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A., *Handbook of Applied Cryptography*, CRC Press. 1996.

[16] C. J. Mitchell, "Remote user authentication using public information", 9th IMA International Conference on Cryptography and Coding, LNCS 2898, 2003, pp.360-369.

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

[18] A. Perrig, R. Canetti, D. Song, J. D. Tygar, "Efficient and Secure Source Authentication for Multicast", Proc. of Network and Distributed Systems Security Symposium,2001.

[19] A. Perrig, R. Canetti, J. D. Tygar, D. Song, "The TESLA Broadcast Authentication Protocol", In CryptoBytes, 5:2, Summer/Fall, pp. 2-13, 2002.

[20] A. Perrig, R. Canetti, J. D. Tygar, D. Song, "Efficient Authentication and Signing of Multicast Streams Over Lossy Channels", IEEE Symposium on Security and Privacy, 2000.

[21] Y. Sella, "On the Computation-Storage Trade-offs of Hash Chain Traversal", 2003.

Appendix A

Any cryptographic primitive is in essence a one-way function and any one-way function can be used to generate such chain. The reason for which hash-function are preferred in the construction of one-way chains is that they are easy to compute, however other functions can be used as well. We conclude with two construction perspectives of one-way chains, of which the most useful and widely spread is obviously the first one: a) Constructing one-way chains from symmetric primitives. Using symmetric primitives, and in particular hash functions which are the simplest ones, offers the advantage of reduced computational cost. In order to optimize the computation and traversal of hash chains improved methods based on time-memory trade-offs have been proposed [5], [8], [13], [21]. b) Constructing one way chains from asymmetric primitives. Using the discrete power function, i.e. $f(x) = x^{\varepsilon} \bmod n$, which is a primitive from public key encryption, has the advantage that the length of the chain is not fixed nor does it influence the computational cost. This is because exponents can be reduced modulo the order of the group and therefore the computational time depends only logarithmically on the order of the group (it is easy to observe that indeed $f(x) = x^{\varepsilon^{\eta} \bmod \phi(n)} \bmod n$). However this function is more computational intensive and the size of the keys is also larger and therefore the use in practice of the discrete power function is limited. The cheapest way to generate such chains is at the cost of about one-modular squaring in the case of $\varepsilon = 2$ by using a time-memory trade-off, the implementation of such a protocol is discussed in [10]. Due to the area of application for the protocols presented in this paper, a low computational power environment, the use of hash functions for the construction of one-way chains will be preferred. Since this paper contains a theoretical result for the purpose of generality we have used the notion of one-way chain instead of hash chain.