CONSTRUCTION TECHNIQUES FOR ONE-WAY CHAINS AND THEIR USE IN AUTHENTICATION

Bogdan Groza

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

Abstract: *Authentication is one of the most important security objectives. The use of one-way chains in authentication was proved to be a very successful technique which encountered applications even in constrained environments such as ah-hoc sensor networks. This paper surveys some of the most efficient construction techniques for one-way chains and some of the most prominent authentication protocols that can be built on them. In brief, the construction techniques for one-way chains are categorized in two classes: one-way chains constructed from symmetric primitives and one-way chains constructed from asymmetric primitives. One-way chains were initially proposed for entity authentication; later their use will prove to be more successful in protocols for assuring information authenticity. We also categorize these protocols in two classes: protocols involving time synchronization and protocols involving an authentic confirmation.*

Keywords: *cryptography, one-way chain, one-way function, authentication, protocol.*

1. INTRODUCTION

Authentication is probably the most important security objective since other objectives may not matter much when there is no guarantee over the authenticity of information or of the parties involved in a communication. Also, authentication protocols are the most encountered security protocols with the widest area of applications since they are commonly used in operating systems, banking systems, mobile telephony etc.

The objective of authentication refers to both entities and information; therefore it can be subdivided in two distinct objectives: **entity authentication** which refers to a guarantee over the identity of an entity and **message authentication** which refers to a guarantee over the source of a given message (this also includes the guarantee that information was not altered during transmission, i.e. integrity). Very distinct cryptographic techniques are involved for assuring these two objectives. Generally speaking entity authentication techniques can be

categorized in two classes: **password based authentications** (or weak authentications) which rely on the disclosure of a secret called password and **challenge-response authentications** (or strong authentications) in which the knowledge of the secret is proved without revealing it (also if asymmetric encryption is used in challenge-response protocols then a shared secret is not required). Message authentication can be assured either by **Message Authentication Codes** (MAC) which are constructed on a one-way function with a secret key or by **Digital Signatures** which are public key primitives and additionally assure the non-repudiation of information but have higher computational requirements.

One-way chains are recurrent arrays generated by the successive compositions of a one-way function, i.e. a function that is easy to calculate but infeasible to invert. Such a chain can be defined as $\sigma_i = f(\sigma_{i-1})$, $\sigma_0 = x_0$, $i = \overline{1,\eta}$, here f is a one-way function, x_0 is the seed used to generate the chain and η is an integer that denotes the length of the chain. Of course $\sigma_i = f^i(x)$ and $f^i(x)$ denotes the composition of f with herself i times, i.e. $f^i(x) = f\underbrace{(...f(f(x))...)}_{i-times}$. A one-way chain has the property that it is easy to verify the connection between any two consecutive elements by checking that $\sigma_i = f(\sigma_{i-1})$ but it is infeasible to compute σ_{i-1} from σ_i since function f is one-way.

The use of one-way chains was initially proposed for assuring entity authentication by Lamport [27]. This was an important milestone; however the technique was not such a great success due to some limitations in front of more advanced challenge-response authentication protocols. Probably the greatest success in the use of one-way chains was to come twenty years later with their use in protocols for assuring the authenticity of broadcast information [3], [36], [37], [38], [39]. In brief, authentication protocols constructed on one-way chains have many characteristics that are close to public key cryptosystems (such as the fact that they do not rely on shared secrets) but offer reduced computational costs.

The importance of this paper is in providing a brief overview on some recent techniques for the construction of one-way chains and their use in authentication. In section 2 some construction techniques for one-way chains are presented. Section 3 is concerned with the use of one-way chains in authentication protocols and also outlines some applications for these protocols. Section 4 holds the conclusion of this paper.

2. CONSTRUCTION OF ONE-WAY CHAINS

In order to construct a one-way chain a one-way function is required; this gives a lot of flexibility since all cryptographic primitives are one-way functions. A partial taxonomy of the construction techniques for one-way chains is depicted in Fig. 1. In the following subsections we give a brief description of all these construction techniques.

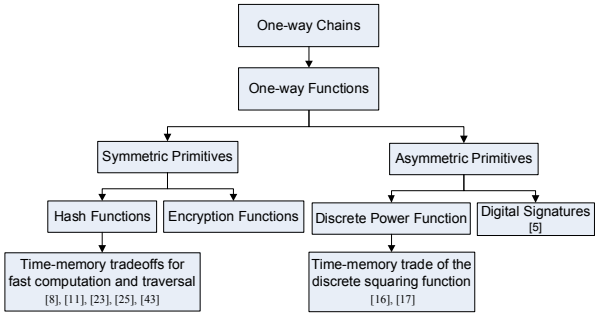

Fig. 1. Taxonomy of the construction techniques for one-way chains.

2.1 The use of symmetric primitives

Constructing one-way chains from symmetric primitives offers the advantage of reduced computational costs. The drawback in using these functions is that the one-way chain has a fixed length, if the length of the chain is chosen too large then computing the chain requires more computational power if it the length is too short then the chain can be exhausted too quickly.

Hash functions are the most widely used functions for this purpose because of their computational simplicity (the most commonly used hash functions are from the SHA (Secure Hash Algorithm) family [12]). A one-way chain can be constructed on a hash function by taking $f(x) = H(x)$, here $H(x)$ denotes a hash

function. Of course the chain can now be also defined as $\sigma_i = H(\sigma_{i-1}), \sigma_0 = x_0, i = \overline{1,\eta}$ and obviously $\sigma_i = H^i(x_0)$.

However symmetric encryption functions may be used as well for this purpose (the standard symmetric encryption algorithm for today is AES (Advanced Encryption Standard) [13]). By using the symmetric encryption function $E_k(m)$, here k is the secret key and m is the message, we can define the one-way function $f(x) = E_x(0)$, here 0 denotes a message of integer value 0. Obviously by successive compositions this function can be further used to generate a one-way chain in which $\sigma_i = E_{\sigma_{i-1}}(0), \sigma_0 = x_0, i = \overline{1,\eta}$. Since symmetric encryption functions require more computational power than hash functions there is no advantage in using them for this purpose.

In order to optimize the computation and traversal of hash chains improved methods based on time-memory trade-offs can be used. The essence of these methods is that several values from the chain can be stored in order to avoid the complete re-computation of the chain, several proposals can be found in [8], [11], [23], [25], [43].

When the elements of the one-way chain are exhausted a digital signature can be use for the re-initialization of the chain. It is obvious that the use of a one-time signature may be more convenient since these digital signatures are also based on simple one-way functions. One such technique is proposed in [14] and an improvement on it in [47]. Finally this signature can be also generated from one-way chains as in the case of the DiMA protocol [15].

2.2 The use of asymmetric primitives

The discrete power function, which is a function from asymmetric cryptography and forms the basis of the RSA and other cryptosystems [40], can be used for the construction of one-way chains. This function is more computational intensive and also increases the communication overhead since the size of the keys is also larger - therefore the use in practice of the discrete power function is limited. Beside these, this function has the advantage that the length of the chain does not influence the computational cost.

While working with the discrete power function:

$$f(x) = x^{\varepsilon} \bmod n \tag{1}$$

the exponents can be reduced modulo the order of the group. This is due to the theorem of Euler which states that:

$$x^{\phi(n)} \equiv 1 \bmod n \tag{2}$$

This leads to the fact that there is no need for multiple compositions in order to compute $f^{\eta}(x)$ since:

$$f^{\eta}(x) = x^{\varepsilon^{\eta} \bmod \phi(n)} \bmod n \tag{3}$$

Therefore computing $f^{\eta}(x)$ requires two modular exponentiations: one for the computation of the exponent $e = \varepsilon^{\eta} \bmod \phi(n)$ and the other for the computation of $f^{\eta}(x) = x^{e} \bmod n$; due to the repeated square and multiply algorithm this involves only logarithmical complexity.

Three different cases may depicted according to the value of the exponent ε [16], [17]. The general case, which holds for any value of ε, in which exponents can be reduced modulo the order of the group, i.e. $\phi(n)$. The particular case in which the exponent ε and the order of the group $\phi(n)$ are relatively primes, i.e. $\gcd(\varepsilon, \phi(n)) = 1$, in this case the function has an inverse $f^{-1}(x) = x^{\delta} \bmod n, \varepsilon \cdot \delta \equiv 1 \bmod \phi(n)$ and each element of the chain can be computed from the previous one by inverting the function. The case of $\varepsilon = 2$ which is the most advantageous case from the computational point of view since exponents may be computed in a time-memory trade at the reduced cost of almost one modular multiplication for each element of the chain [16], [17].

Finally it is easy to remark that any digital signature, which is a public key primitive, can be used to construct a one-way chain of unbounded length by computing a chain of signatures in which each new signature is computed on the previous one. In [5] the notion of signature chain is used to denote such a chain.

Since digital signatures are the most computational intensive cryptographic primitives it is likely that their use for such purpose is inefficient. Also in [4] the notion of infinite length hash chain is introduced, however this notion is a little misleading since the length of the chain became unbounded due to the use of a digital signature – therefore the chain is no longer a hash chain.

2.3 Some security issues

Although hash functions are used in almost all cryptographic systems, they are less known than other cryptographic primitives such as for example block ciphers [10, page 84]. The most commonly used hash functions are from the SHA family [12]. A recent result [45] shows that it is possible to find collisions on SHA1 function therefore for long term security in some applications SHA-256 or a more powerful hash function is recommended. However, finding random collisions does not affect the security of hash chains since it requires only second pre-image resistance. Also for short term security, as in the case of broadcast transmissions where keys are useful only for short periods of time, the use of hash functions with low levels of security is appropriate.

Table 1: Computational performance of some cryptographic primitives in Java (time is taken as the average time of 1.000.000 runs, small variations on timings are expected).

CPU	Hash with MD5 (128 bit)	Hash with SHA1 (160bit)	Modular Multiplication (1024 bit module)	Modular Exponentiation (1024 bit module and exponent)
Intel Centrino 1.6 Ghz	$1.6\times10^{-6}s$	$2.8\times10^{-6}s$	$74\times10^{-6}s$	$50\times10^{-3}s$
AMD Athlon 64 2800+ 1.8 Ghz	$1.2\times10^{-6}s$	$2.4\times10^{-6}s$	$60\times10^{-6}s$	$43\times10^{-3}s$

For the use of the discrete power function the size of the modulus affects the level of security, since the only known way to invert this one-way function is by factorizing the modulus. A 1024 bit module can be considered secure for today standards; a good point of view on the security of the integer factorization problem can be driven from the challenges offered by RSA [42]. For a more accurate look on the computational efficiency of different cryptographic primitives in table 1 the computational time measured in Java [26] is given. It is obvious that hash functions are the most computational efficient while the discrete squaring function (i.e. discrete power function in the case of $\varepsilon = 2$) may be useful in the case of the time-memory trade when the chain is generated at the cost of almost one modular multiplication. The general case of the discrete power function is very expensive being in thousands of times more expensive than a hash function.

3. ONE-WAY CHAINS IN AUTHENTICATION

In the previous section several construction techniques for one-way chains were introduced, in this section we describe several authentication protocols that can be built with one-way chains.

3.1 One-way chains in assuring entity authentication - Lamport's scheme

Although they offer the weakest level of security fixed passwords are still the most commonly used authentication technique. The greatest disadvantage of conventional time-invariant passwords is that they can be stolen either from the system where they are stored or by intercepting user's communication over insecure channels. In this context, one-time passwords are a first step towards strong authentication. These are passwords which are valid only once for an authentication and the advantage they offer is that a previously disclosed password can not be used to impersonate the user.

Lamport has proposed a functional one-time password scheme which is based on the use of a one-way chain [27]; this is probably the first proposal for using one-way chains in authentication. The proposed scheme has the following advantages: secrets are stored only on the user's side, which prevents an intruder to learn the secret by gaining access on the system side, and each password is valid only once, therefore intercepting user's communication with the system will not lead to an impersonation [27].

The scheme is constructed on a very simple principle. The user can compute on its side the one-way chain $\{x_A, f(x_A), f^1(x_A), f^2(x_A), ..., f^\eta(x_A)\}$, here f is a commonly known one-way

function, x_A is a secret value known only to the user and η is the number of authentications that can be performed with this chain. In the initialization session, $i = 0$, the user transfers the value of $f^{\eta}(x_A)$ to the system in a manner that guarantees the authenticity of this value. Then, when the user needs to authenticate for the first time to the system he will present $f^{\eta-1}(x_A)$. Generally for the i^{th} authentication the user will prove his identity by sending $f^{\eta-i}(x_A)$ to the system which can easily verify that $f(f^{\eta-i}(x_A)) = f^{\eta-i+1}(x_A)$. This scheme may also be viewed as a challenge-response protocol where the challenge is defined by the position of the password in the password sequence [29, page 396].

The use of Lamport's scheme in entity authentication has not encountered a large number of applications and this is probably due to its disadvantages compared to more advanced challenge-response authentication protocols. The greatest disadvantage of the scheme is the pre-play attack (also suggested in Note 10.7 from [29]): an attacker can intercept (or impersonate the system in order to extract) a yet unused password for subsequent impersonation. One implementation of Lamport scheme is in the S-KEY system by Haller [18], [19], [20]. This system is vulnerable to the "host impersonation" attack where a false host can obtain passwords from the user for subsequent impersonation; several comments on the S-Key are in [32].

A proposal of authentication scheme which is resistant to such an attack and which require only public information to be stored at the verifying host is in [33]. Another one-time password scheme is proposed in [44] which is supposed to have some advantages compared to Lamport's scheme, but this proposal along with several variations of it is examined in [7] and all of them are proved to be insecure. Also in [7] the Robust and Simple Authentication Protocol (ROSI) is presented and this protocol is secure against various attacks that are examined. Other simple variations of one-time password schemes include: shared lists of one-time passwords, in which the user and the system use a pre-shared list of passwords, and sequentially updated one-time passwords, in which each time the user authenticates to the system it also transmits a new password [29, pp. 396]. However, all these one-time password schemes have only distant relations with the scheme proposed by Lamport and they are not constructed on a one-way chain.

A disadvantage in using hash functions for the construction of the one-way chain is the fact that the chain will have a fixed length and when exhausted it will require re-initialization which can raise security issues. In order to remove this disadvantage the use of the discrete power function may be considered for the construction of the one-way chains, obviously by using this function the chain will never exhaust. The use of chains constructed over the discrete power function in Lamport's scheme is discussed in [17].

3.2 One-way chains in assuring information authenticity

In this section we investigate several protocols that can be built on one-way chains for assuring the authenticity of information in one-to-one or one-to-many communications. As this section outlines one-way chains are an excellent solution in providing keys for MAC's that can be later used in verifying the authenticity of information.

The common idea on which they are all based is to compute a MAC on message with a key which is committed in the current communication session and disclosed only in a forthcoming session. In principle, such an authentication scheme is secure as long as there is a guarantee that at the moment when the MAC was received the key of the MAC was not yet disclosed. In brief, this can be guaranteed by the use of two distinct mechanisms:

a) **Time synchronization.** A time synchronization can be used between the sender and the receivers and as long as keys are disclosed at exact time intervals the receivers can decide based on their time synchronization if the key of the MAC was or not disclosed at the time when the MAC is received. In fact here time can be seen as a challenge, therefore it may be more appropriate to call this technique as time-driven challenge-response; however, for the simplicity of the terminology, we will avoid this. This technique was proposed by Perrig et. al. and used in the schemes from [36], [37], [38], [39].

b) **Authentic Confirmation.** Instead of time synchronization the sender can wait for an authentic confirmation of the arrival of the MAC from each receiver before releasing the key of the MAC. This closely resembles a challenge-response mechanism in which the challenge is implicitly defined by the current position of the key in the chain (a similar idea is stated for Lamport's scheme in [29, 396]). This technique is used in [3], [15].

3.2.1 The TESLA Protocol

The Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol was proposed by Perrig et. al. [36], [37], [38], [39]. The excellent idea which stands behind this scheme is the use of loose time synchronization between senders and receivers. This implies that the receivers must have an upper bound on the time from the sender's side. Therefore, the security condition which must be meet to make the 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 $P_j, j > i$ [38].

Several variants of the TESLA protocol are proposed in [37], [38].

The **basic scheme** from [38], which is similar to the Guy Fawkes Protocol [2], does not use one-way chains. The principle on which it is based is the following: a random key is committed in packet P_{i-1}, this key is used to compute a MAC on packet P_i and is disclosed later in packet P_{i+1}, packet P_i also contains the commitment for the key which is used to compute the MAC for packet P_{i+1} and so on. As long as the first packet is authentic and the security condition holds the other packets can be also checked for authenticity. For this purpose the first packet is committed with a regular digital signature scheme and the security condition is checked based on the time at which the packet arrives and the loose time synchronization between the sender and the receiver.

However this scheme succumbs after some packet is lost because the next key commitment is also lost. The **packet loss tolerant scheme** introduces the use of one-way chains [38]. By replacing the random keys used for the MAC of 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 key by successive composition of the one-way function. The **fast transfer rate scheme** proposes the disclosure of the key for packet P_i in a later packet P_{i+d} where d is computed based on the synchronization uncertainty and maximum tolerable network delay [38]. This can increase the transfer rate since in the previous schemes packet P_{i+1} can be sent only after the P_i was received. The **dynamic packet rate** scheme introduces the use of the keys on a time interval basis rather than on a packet index basis. With this scheme the same key is used on all packets sent on a particular time interval and therefore the packet rate is dynamic. The **scheme dedicated for a broad spectrum of receivers** proposes the use of multiple one-way chains with different disclosure periods. By using this scheme each receiver can pick the chain with the minimal disclosure period that suits the speed of its network access.

The **immediate authentication scheme** makes it possible to authenticate the packets at the time when they are received; this is possible by including a commitment of the current packet in the previous packet and therefore the packet can be checked for authenticity based on the authenticity of the commitment [39].

3.2.2 The CSA Protocol

The Chained Stream Authentication (CSA) removes the requirement of time synchronization by introducing the requirement of an authentic confirmation for each received packet; this can be viewed as a challenge response mechanism. To preserve the asymmetry of the scheme, i.e. not involving shared secrets, a value from a one-way chain can be used as a confirmation. Three variants of CSA are proposed in [3]. The **Interactive Chained Stream Authentication** (I-CSA) addresses the scenario of one sender and one receiver, each of the two entities commits 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 package and so on. Since CSA was intended for a broadcast communication the **N-party**

Interactive Chained Stream Authentication is proposed for communications between one sender and multiple receivers. However this scheme is likely to be inefficient since the sender must wait for a response from each entity and this raises two problems: first if one of the receivers fails to respond the scheme succumbs, and secondly the speed of the communication is given by the entity with the weakest communication resources. In order to fix the second problem delaying just the secret keys and sending information before the secrets was the proposed alternative, but this will cause either the receivers to use information that is not yet proved to be authentic either to store information until the keys are received (this can cause storage problems on the receivers side). The **Timed Chained Stream Authentication** T-CSA is proposed for the removal of the confirmation; this variant is similar to the TESLA Protocol.

As a partial conclusion a taxonomy of the authentication protocols presented so far is depicted in Fig. 2.

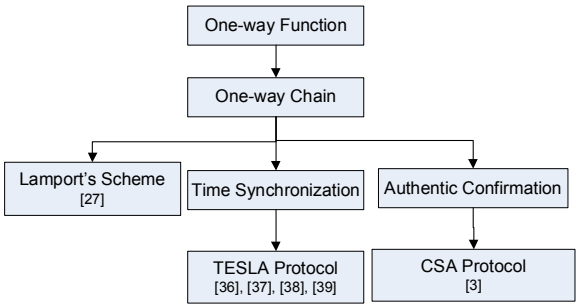

Fig. 2. Taxonomy of one-way chain based authentication protocols.

3.2.3 The DeMA/DiCA Protocol

The Delayed Message Authentication/ Direct Chain Authentication (DeMA/DICA) is also based on the use of a one-way chain and an authentic confirmation [15].

In large the differences between DeMA/DiCA and CSA protocols are the following: the nature of the communication which is a multiparty communication for CSA and a one-to-one communication for DeMA/DiCA, the initialization of the chains (i.e. the commitment of the chains) which is different for the two protocols (DeMA/DiCA uses also one-way chains while CSA a generic one-time signature) and the construction of the one-way chains, in CSA hash-chains are used while the DeMA/DiCA protocol was proposed in the general setting of the use of a one-way function (also a description of the DeMA protocol for the case of the discrete power function is given and experimental results a for the DeMA protocol in the case of the discrete power function are presented [16]).

The DeMA/DiCA consists in two components: the DeMA component of the protocol which is used for exchanging messages and is similar to the CSA protocol and the DiCA component of the protocol which is used for the re-initialization of the one-way chains (in essence DiCA 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 TESLA protocol requires time synchronization while the CSA requires a one-time signature for the re-initialization of the chain).

DeMA/DiCA uses two one-way chains on each entity's side, a longer chain of $\eta+\delta$ and a shorter chain of δ elements. The first η elements from the longer chains are used to exchange messages with the DeMA protocol while the last δ elements from the longer chain with all the elements from the shorter chain are used to re-initialize the one-way chains. The structure of the one-way chains and their use in the DeMA/DiCA protocol is depicted in Fig. 3. Details on this protocol can be found in [15].

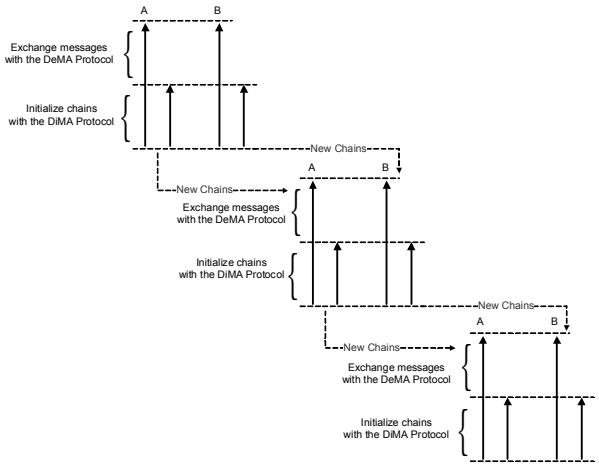

Fig. 3. The use of the one-way chains in the DeMA/DiCA protocol

3.2.4 Applications of one-way chained based authentication protocols

The TESLA protocol is probably the most promising authentication protocol for practical use. It is of course suitable for broadcast communication and offers great security advantages at reduced computational cost, also several variants and improvements on it were proposed [37], [38], [39].

Probably the most important application of this protocol was in ad-hoc sensor networks [36], this is an environment with reduced computational power and communication abilities. An improvement over the scheme from [36] is presented in [28]. In brief, the proposal from [28] improves the distribution of the keys which is done also in a broadcast manner and makes the solution more practical for networks with large number of nodes.

An application of the CSA protocol is presented in [3]. In this application the timed version of the CSA protocol is used to broadcast audio information. So far we are not aware of any other practical application that uses the CSA authentication protocol.

The DeMA/DiCA protocol [15] was proposed in the context of a one-to-one communication. This was because of our intention to use this protocol in a secure robust control scenario which is expected to be a one-to-one communication between a controller and a controlled process. The use of a confirmation mechanism instead of time synchronization was needed because control systems are based on the concept of feed-back and a response from the receiver of the information is usually required. In principle we can distinguish between two scenarios which are also illustrated in Fig. 4. In the first scenario, Fig. 4 a), a synchronous communication takes place between the controller and the controlled process: each command is issued on the previously received response; in this scenario one instance of the DeMA/DiCA protocol is needed. In the second scenario, Fig. 4 b), an asynchronous communication takes place: commands and responses are sent independently between the controller and the controlled process; in this scenario two instances of the DeMA/DiCA protocol are needed. The use of the DeMA/DiCA protocol in such scenarios is subject of our future work.

Besides these, it may be also relevant to remark that one-way chains are also used for assuring security in: electronic payment schemes [21], [34], [35], [41], routing protocols [6], [22], [24], [46] digital certificate revocations [1], [31], digital signatures [9], [30].

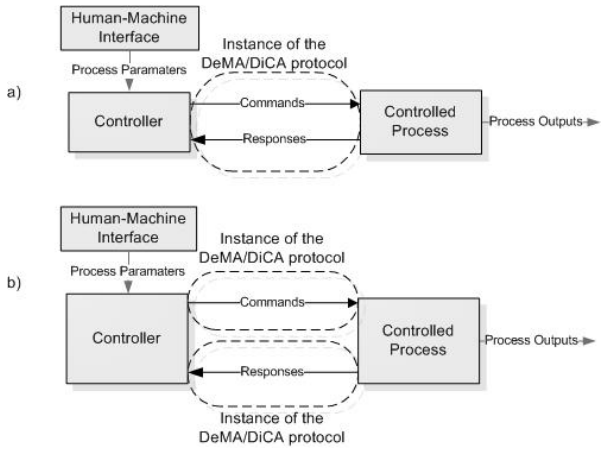

Fig. 4. Generic control systems with synchronous or asynchronous communication between the controller and the controlled process

4. CONCLUSION

Obviously authentication is a central objective in the field of information security. It is almost certain that authentication protocols based on one-way chains will have a great perspective for use in the forthcoming years. This paper has made a concise classification on the construction techniques for one-way chains and on the protocols built upon them. As future work, further investigations over the possible use of such protocols can be done. We are especially concerned with the use of such protocols in industrial control systems where computational resources are limited.

5. ACKNOLEDGEMENT

Research reported in this paper was partially supported by National University Research Council of Romania under research grants MEdC-CNCSIS-A-309/2005, MEdC-CNCSIS-TD-90/2006.

REFERENCES

[1] Aiello, W., Lodha, S., Ostrovsky, R., "Fast digital identity revocation". In Hugo Krawczyk, editor, Proceedings of Crypto'98, LNCS 1462, Springer, 1998.

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

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

[4] Bicakci, K., Baykal, N., "Infinite Length Hash Chains and Their Applications", 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WETICE, IEEE, 2002.

[5] Bicakci, K., Baykal, N., "Improving the Security and Flexibility of One-time Passwords by Signature Chains". Turkish Journal of Electrical Engineering & Computer Sciences, 11, p.223-236, 2003.

[6] Cheung, S., "An efficient message authentication scheme for link state routing", 13th Annual Computer Security Applications Conference, ACSAC, pages 90-98, 1997.

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

[8] Coppersmith, D., Jakobsson, M., "Almost Optimal Hash Sequence Traversal", Sixth International Conference on Financial Cryptography 2002, LNCS 2357, Springer, 2003.

[9] Even, S., Goldreich, O., Micali, S., "On-line/offline Digital Signatures". In Advances in Cryptology: Crypto '89, pp 263-277, Springer, 1989.

[10] Ferguson, N., Schneier, B., "Practical Cryptography", Wiley Publishing Inc., 432 pages, 2003.

[11] Fischlin, M., "Fast Verification of Hash Chains", Topics in Cryptology – CT-RSA 2004, Springer, 2004.

[12] FIPS 180-1, National Institute of Standards and Technology (NIST). "Announcing the Secure Hash Standard"., U.S. Department of Commerce, 1995.

[13] FIPS 197, "Announcing the Advanced Encryption Standard". http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[14] Goyal, V., "How To Re-initialize a Hash Chain".URL: http://eprint.iacr.org/2004/097.pdf, 2004.

[15] Groza, B., "Using one-way chains to provide message authentication without shared secrets", 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, pp. 82-87, IEEE, 2006.

[16] Groza, B., Petrica, D., Dragomir T.L., "Using the Discrete Squaring Function in the Delayed Message Authentication Protocol", accepted for Proceedings of ICISP'06, IEEE, 2006.

[17] Groza, B., Petrica, D., Dragomir T.L., "A time-memory trade solution to generate one-time passwords using quadratic residues in Zn", Studies in Informatics Control, ISSN 1220-1766, pp. 201- 212, 2005.

[18] Haller, N., "The S/KEY One-Time Password System", Proceedings of the ISOC Symposium on Network and Distributed System Security, 1994.

[19] Haller, N., Metz, C., Nesser, P., Straw, M., "The S/KEY One-Time Password System", Internet RFC 1760, 1995.

[20] Haller, N., Metz, C., Nesser, P., Straw, M., "A One-Time Password System". Internet RFC 2289, 1998.

[21] Hauser, R., Steiner, M., Waidner, M.. "Micro-Payments based on IKP". In Worldwide Congress on Computer and Communications Security Protocol, 1996.

[22] Hauser, R., Przygienda, T., Tsudik, G., "Reducing the cost of security in link-state routing", in Symposium of Network and Distributed Systems Security, 1997.

[23] Hu, Y.-C., Jakobsson, M., Perrig, A., "Efficient Constructions for One-way Hash Chains", Proceedings of Applied Cryptography and Network Security ACNS'05, LNCS 2947, Springer, 2005.

[24] Hu, Y.-C., Johnson, D. B., Perrig, A., "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks" in The 4th IEEE Workshop on Mobile Computing Systems and Applications, IEEE, 2002.

[25] Jakobsson, M., "Fractal hash sequence representation and traversal", IEEE International Symposium on Information Theory, IEEE, 2002.

[26] Java.sun.com: The Source for Java Developers, http://java.sun.com, 2006.
[27] Lamport, L., "Password Authentication with Insecure Communication". Communication of the ACM, 24, 770-772, 1981.
[28] Liu, D., Ning., P. "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks". In Proc. of the 10th Annual Network and Distributed System Security Symposium, pages 263-276, 2003.
[29] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A., "Handbook of Applied Cryptography", CRC Press. 1996.
[30] Merkle, R. C., "A digital signature based on a conventional encryption function". In CRYPTO '87, pages 369-378, LNCS 293, Springer, 1988.
[31] Micali., S., "Efficient Certificate Revocation". Technical Report MIT/LCS/ TM542b, 1996.
[32] Mitchell, C., Chen, L., "Comments on the S/KEY User Authentication Scheme", Operating Systems Review, 1996.
[33] Mitchell, C. J., "Remote user authentication using public information", Cryptography and Coding, 9th IMA International Conference on Cryptography and Coding, LNCS 2898, Springer, pp.360-369, 2003.
[34] Nguyen, K.-Q., Mu, Y., Varadharajan, V., "Digital coins based on hash chain", Proceedings of the 20th National Information Systems Security Conference, Baltimore, USA, pp. 72-79, 1997.
[35] Pedersen, T. P., "Electronic payments of small amounts", Security Protocols International Workshop, LNCS 1189, Springer, 1997.
[36] Perrig, A., Szewczyk, R., Wen, V., Culler D., Tygar, J.D., "SPINS: Security Protocols for Sensor Network", Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM, 2001.
[37] Perrig, A., Canetti, R., Tygar, J. D., Song, D., "The TESLA Broadcast Authentication Protocol", In CryptoBytes, 5:2, Summer/Fall, pp. 2-13, 2002.
[38] Perrig, A., Canetti, R., Tygar, J. D., Song, D., "Efficient Authentication and Signing of Multicast Streams Over Lossy Channels", IEEE Symposium on Security and Privacy, 2000.
[39] Perrig, A., Canetti, R., Song, D., Tygar, D., "Efficient and Secure Source Authentication for Multicast", Proceedings of Network and Distributed System Security Symposium, 2001.
[40] Rivest, R., Shamir, A., Adleman, L., „A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, 1978.
[41] Rivest, R., Shamir, A., "Payword and Micromint: Two simple micropayment schemes". CryptoBytes, volume 2, no. 1, RSA Laboratories, 1996.
[42] RSA Laboratories - RSA Factoring Challenge,http://www.rsasecurity.com/rsalabs/challenges/ factoring/numbers, 2006.
[43] Sella, Y., "On the Computation-Storage Trade-offs of Hash Chain Traversal", 2003.
[44] Tsuji, T., Shimizu, A., "A one-time password authentication method", Master Thesis, Kochi University of Technology, 2003.
[45] Wang, X., Yin, Y.L., Yu, H., "Collision search on SHA1", http://theory. csail.mit.edu/ ~yiqun/shanote.pdf, 2005.
[46] Zhang, K., "Efficient Protocols for Signing Routing Messages", In Proceedings of Network and Distributed System Security Symposium, 1998.
[47] Zhao, Y., Li, D., "An Improved Elegant Method to Re-initialize Hash Chains", URL: http://eprint.iacr.org/2005/011.pdf, 2005.