

A Prepositioned Secret Sharing Scheme for Message Authentication in Broadcast Networks

AHMET M. ESKICIOGLU

Thomson Multimedia

101 West 103rd Street, INH 725

Indianapolis, IN 46290, USA

Abstract In modern electronic distribution networks, message authentication is an important objective of information security. This objective is met by providing the receiver of a message an assurance of the sender's identity. As physical protection such as sealed envelopes is not possible for messages expressed as binary sequences, digital tools have been developed using cryptography. A major limitation of all cryptographic methods for message authentication lies in their use of algorithms with fixed symmetric or public keys. We describe a new key transport scheme, based on secret sharing, which not only allows each new message to be authenticated with a new key, but also generates different authentication keys for different groups of receivers in broadcast networks.

Key words: ciphers, data integrity, digital signature, encryption, hashing, key transport, message authentication, public-key cryptography, prepositioned secret sharing.

1. INTRODUCTION

Authentication is one of the four most important objectives of information security.^{1,2,3} The others are confidentiality (protecting information from unauthorized disclosure), data integrity (providing assurance that information has not been altered in an unauthorized way) and non-repudiation (preventing a party from denying a previous action). In communication networks, some or all of these objectives may need to be met.

Authentication methods can be studied in two groups: Entity authentication and message authentication. Figure 1 shows a communication channel where two parties, A and B, communicate using a message protocol. Party A is the sender of a message M, and party B is the receiver. Depending on the type of communication or network, B would require one or more of the following on receipt of the message²:

- (1) Authentication of the message,
- (2) Integrity of the data included in the message,
- (3) Authentication of sender A.

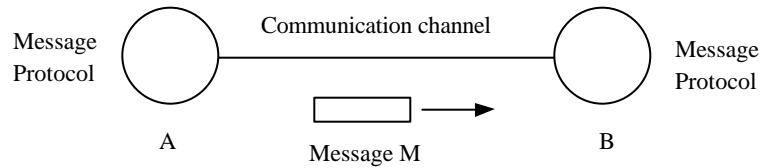


Figure 1. Two-party communications

Message authentication provides assurance of the identity of A, the originator of the message M. This type of authentication also includes an evidence of data integrity because if M is modified during transmission, A cannot be the originator. Entity authentication assures B of both the identity of A and his active participation. Although message authentication gives no guarantees of timeliness or uniqueness, it is useful in communications where one party is not active during the execution of the message protocol. To avoid replay attacks (i.e., an intruder masquerades as A, and sends a previously used message), time-variant data (sequence numbers, time stamps, etc.) can be added to the message.

2. METHODS FOR MESSAGE AUTHENTICATION

As a given message can be of arbitrary length, the process called *hashing* is an essential part of most data integrity and message authentication methods. A hash function takes a message of arbitrary finite length and produces an output of fixed length. In cryptographic applications, the hash value is considered to be a shorter representation of the actual message. Depending on the type of input parameters, hash functions are classified into two groups²: *unkeyed hash functions* (the message is the only input) and *keyed hash functions* (the message and a secret key are two inputs).

A particular class of unkeyed hash functions contains *Manipulation Detection Codes* (MDCs) in three categories: hash functions based on block ciphers, hash functions based on modular arithmetic and customized hash functions.

The keyed hash functions that are used for message authentication are grouped under *Message Authentication Codes* (MACs). MACs can be customized, constructed using block ciphers or derived from MDCs.

We will now classify the message authentication methods with a particular interest in how they exploit symmetric or public key ciphers: MACs, message encryption and digital signatures. In the rest of the paper, the following cryptographic notation will be used for denoting encryption and hashing algorithms:

$h(M)$: Hashing of message M with an MDC; $h_k(M)$: Hashing of message M with a MAC with key K ; $M_1||M_2$: Concatenation of message M_1 with message M_2 ; $E_k(M)$: Encryption of message M with key K ; $S_{K_{private}}(M)$: Signing of message M with private key $K_{private}$.

Method 1. Using a MAC

The process of producing a MAC is depicted in Figure 2. The message is input to a MAC algorithm which computes the MAC using a key K shared by both parties. A then appends the MAC to the message, and sends the pair {message || MAC} to B.

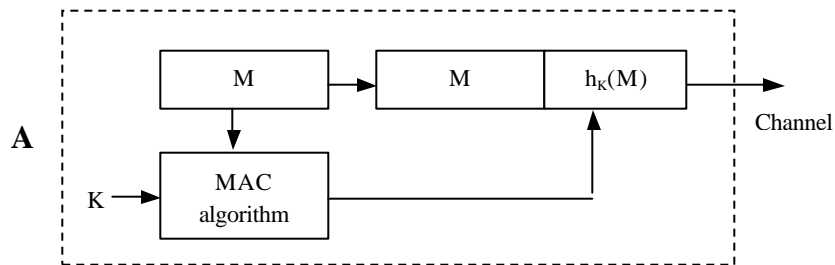


Figure 2. Authentication with a MAC

Method 2. Encrypting the message

- a) *Symmetric key encryption*: As shown in Figure 3, encrypting the entire message with a symmetric key cipher would provide both confidentiality and authentication. B is assured that the message was generated by A since A is the only other party that has a copy of the shared key. This approach is valid under the assumption that B is able to determine if the ciphertext decrypts into intelligible plaintext.
- b) *Public key encryption*: B has a public/private key pair. Using B's public key to encrypt the message provides only confidentiality but not authentication. Since all public keys are available for all, any intruder with easy access to B's public key can masquerade as A.

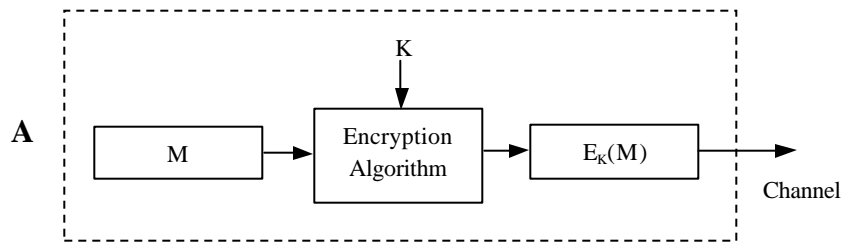


Figure 3. Authentication with message encryption

In practice, encryption can be used together with MDCs or MACs. Some suggested basic schemes are as follows^{2,3}:

$E_k[M \parallel h(M)]$; $E_{k_2}[M \parallel h_{k_1}(M)]$; $E_{k_2}(M) \parallel h_{k_1}(M)$; $E_{k_2}(M) \parallel h_{k_1}[E_{k_2}(M)]$ and $E_k[M \parallel h(M \parallel S)]$, where S is a shared secret.

Method 3. Signing the message

In Figure 4, A uses its private key to sign the message. Depending on the size of M , an appropriate signature algorithm (with message recovery or with appendix) can be used. B has assurance that the message was generated by A because A is the only party that owns the private key. It is assumed that B has the ability to distinguish between legitimate and garbled plaintexts.

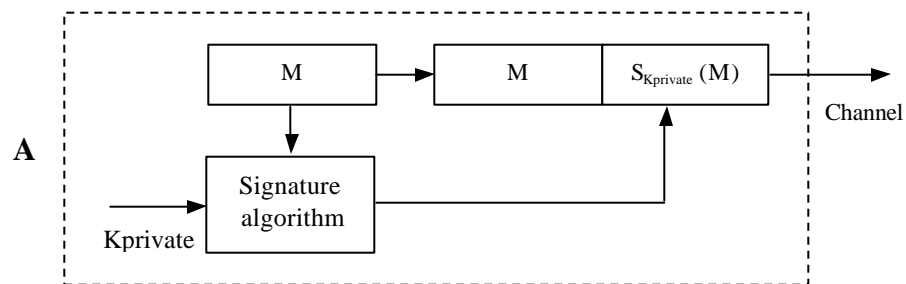


Figure 4. Authentication with a digital signature

Note that some of the above methods generate an authenticator that is appended to the message and some not. In Method 2, the encrypted message itself is the authenticator. In Method 3, if the message is short enough, a signature scheme with message recovery can be used.

Key management is an important aspect of message authentication. Let us consider the disadvantages of using a fixed key for MAC creation, message encryption and message signing.^{1,2,3}

a) Potential cryptographic weakness

MACs: There are two attacks: Attack on the key space and attack on the MAC value. If the hacker can determine the MAC key, he is able to create a MAC value for any message. For a key size of t bits and a fixed input, the probability of finding the correct n -bit MAC is about 2^{-t} . The objective of MAC forgery is to create a MAC for a given message or to find a message for a given MAC without knowing the key. For an n -bit MAC algorithm, the probability of meeting this objective is about 2^{-n} . In summary, the effort needed for a brute-force attack on a MAC algorithm would be the min ($2^t, 2^n$).

Encryption: If encryption is used alone for message authentication, it is vulnerable to brute-force attacks. In the recent years, several powerful attacks have been developed against modern ciphers. For a 56-bit DES algorithm, an exhaustive search requires 2^{55} DES operations. More efficient attacks like linear or differential cryptanalysis allow key recovery with less processor time.

Digital signatures: From a theoretical viewpoint, no popular public-key signature algorithm is proven to be secure. Their security is based on the difficulty of computing discrete logarithms or factoring large numbers. With a fixed public/private key pair, attacks are possible using the public key or signatures on messages.

b) Public-key infrastructures

In some applications, the authenticity of the sender's public key is a major problem requiring complex public-key infrastructures. A public-key certificate is a data record that includes a public key and some other information such as the owner identity, the issuer identity and the validity period. It is digitally signed by a trusted third party called a Certificate Authority (CA) who creates, distributes, maintains and revokes public-key certificates.

c) Lack of capability to authenticate messages with different keys

Another disadvantage associated with a fixed key is that it is used by the entire population of the receivers. In some applications, there may be a need

to send a message to a specific group of receivers. In general, we would like to have a scheme that makes it possible to use a new key for each new message and to generate different keys for different groups of receivers.

The key sharing scheme* that we will describe circumvents the problems associated with fixed keys. It has been proposed for protecting audio/video content in conditional access systems⁴.

3. A KEY TRANSPORT SCHEME FOR MESSAGE AUTHENTICATION

3.1 Threshold Schemes

A (t, n) threshold scheme^{1,2,5-7} ($t \leq n$) is a method by which n secret shares S_i , ($1 \leq i \leq n$), are computed from a secret S in such a way that at least t shares are required to reconstruct S . For example, with a $(2,5)$ threshold scheme, a bank manager can divide the combination of the bank safe among his five tellers in such a way that any two tellers can use their secret pieces to construct the combination and open the safe. In a *perfect* threshold scheme, a knowledge of $(t-1)$ or fewer shares does not change the probability distribution of the possible values of the secret.

After the introduction of the idea by two independent publications^{5,6} in 1979, several threshold schemes have been developed based on a common theoretical background. In Shamir's (t, n) threshold scheme, a random $(t-1)$ -degree polynomial, i.e., $f(x) = (a_{t-1}x^{t-1} + \dots + a_1x + a_0)$, is used over the finite Galois Field $GF(p)$:

1. Choose a prime p larger than n and the secret S .
2. Define S to be the constant term a_0 .
3. Construct $f(x)$ by selecting $(t-1)$ random coefficients a_1, \dots, a_{t-1} .
4. Compute the shares by evaluate $f(x)$ at n distinct points, and distribute them to n users.

The secret S can be computed by constructing the polynomial from any t of the n shares.

* Thomson multimedia, Inc. patent pending.

Electronic cash, group signatures, key recovery and voting are some of the cryptographic applications for which threshold schemes have proved useful. In particular, some authors^{7,8,9} discuss the application of threshold schemes to key distribution in broadcast networks. Their basic idea is to construct a (t,n) threshold scheme, and to assign a distinct share to each receiver in the network. If $(t-1)$ shares are broadcast, the secret can be constructed by any receiver using the $(t-1)$ shares and its distinct share. A limitation of this approach is the generation of a key common to all intended recipients. From a security viewpoint, the hacker needs to know only a single share to break the system. We will use Shamir's threshold scheme in a new way that allows the broadcaster to create different keys for different sets of receivers, and to renew these keys conveniently.

3.2 A Prepositioned Secret Sharing Scheme for Key Transport

The receiver is manufactured with the point (x_0, y_0) on the first degree polynomial to be constructed. The source of the message chooses a secret S , generates the authenticator, and transmits, depending on the method, either the message and the authenticator or just the authenticator with (x_1, y_1) in-the-clear. On receiving (x_1, y_1) , the receiver constructs the polynomial passing through the two points, recovers the secret, and checks the authenticator. If authentication is not successful, the message is rejected.

Each new key requires the construction of a different polynomial passing through the point (x_0, y_0) . Our proposal can therefore be considered to be a *prepositioned shared secret scheme*^{10,11} which makes it possible to reconstruct different keys by communicating different activating shares for the same prepositioned information.

The proposal can be generalized by defining t is a system parameter. A higher degree polynomial would result in slightly more computations in polynomial construction, but increase the resistance of the system to attacks. In this case, the receiver may store some of the $(t-1)$ shares, and obtain the others from the sender.

Multiple shares can also be used to build a convenient key transport scheme in a communications network. *Code authentication*¹²⁻¹⁴, an important issue in digital networks, will be used as a small case study. In the future, sophisticated home entertainment devices handling audio/video data will receive software for various applications via digital distribution networks

(satellite, cable, terrestrial or Internet). Identification of the source of this code is an essential requirement for both the service providers delivering content and the manufacturer of the devices using the content. Suppose in a given broadcasting system different groups of devices are to be authorized in different ways. The simple example below will explain how secret sharing can be used to establish the required key hierarchy.

Example: A broadcast system with three different authentication levels for code authentication. Level 1: All the devices are assigned one common share, Level 2: The devices in a given group are assigned an additional common share, and Level 3: Each device is assigned a unique additional share.

If the code is broadcast for all the receivers in the region, the devices will construct a first degree polynomial using the common share, and obtain the same authentication key. If only a particular group or an individual device is authorized to have access to the application, the additional share(s) will result in a key that cannot be constructed by the other devices in the region.

The first degree polynomial will pass through the common share for Level 1 and the activating share. The second degree polynomial will pass through the common share for Level 1, the common share for Level 2 and the activating share. For the construction of the third degree polynomial, the additional point will be the unique share for Level 3.

Let $p = 31$. The coordinates of the points needed for the three polynomials are given in Table 2.

Table 2. Points for polynomial construction

Point \ Degree of polynomial	First	Second	Third
The activating share = (10, 15)	x	x	x
The common share for Level 1 = (30, 20)	x	x	x
The common share for Level 2 = (20, 10)		x	x
The unique share for Level 3 = (5, 25)			x

The solution gives:

$$f_1(x) = 8x + 28; f_2(x) = 21x^2 + 5x + 4; f_3(x) = 27x^3 + 13x^2 + 7x + 10.$$

Hence, $(S_1, S_2, S_3) = (28, 4, 10) \pmod{31}$.

Message (code, etc.) authentication is performed by consumer electronics and information technology devices such as digital set-top boxes, digital TVs and PCs. Their memory capacity does not impose a serious limitation on the storage of authentication keys.

3.3 Security analysis

In our scheme, the shared secret is used to generate a message authenticator which is broadcast with the message and the activating share. For small values of t , i.e., lower degree polynomials, the system may be exposed to brute-force attacks. A potential hacker may use the available authenticators in an attempt to find the prepositioned information i.e., the “permanent key” in the receiver. The vulnerability of the system to attacks for key recovery is reduced for increasing values of t . In the following analysis, we will assume that a new key is used for each broadcast message, and the activating share is sent in-the-clear.

- $t = 2$: The system is most vulnerable if first degree polynomials are used. If the hacker finds two authentication keys, he can compute the prepositioned information by constructing two straight lines and finding their intersection.
- $t > 2$: The security is based on the difficulty of estimating the prepositioned information in the receiver. For a polynomial of degree $(t-1)$, there are $(t-1)$ pieces of the shared secret in the receiver. The only data available to estimate these pieces is a pair of points (the activating share and the hacked authentication key) on the polynomial. In general, each pair can be used to construct 2 linear equations in t variables. We are currently investigating how this data may weaken the system.

Several modifications are possible to increase the robustness of the system⁴:

1. **Define the authentication key as a function of the shared secret:** In Shamir’s threshold scheme, the key is defined to be the y-intercept of the constructed polynomial. This definition can be generalized to allow other ways of defining the key. One approach is to evaluate the value of a predefined function at the secret. Ideally, two additional requirements may be desired: Keeping the function definition secret, and choosing a function that preserves entropy (i.e., entropy of the secret = entropy of the value of the function at the secret).

2. **Make t a time-dependent secret system parameter:** If the system allows the parameter t to be a time-variant secret, the adversaries would encounter one more dimension of difficulty for cryptanalysis.
3. **“Mask” the activating share before distribution:** An unkeyed hash function can be used for this purpose, avoiding the need for key management. Alternatively, since a common key is available for both parties, the message authentication method itself would be appropriate. In either case, the sender would use the hash value of the activating share for generating the authenticator, but transmit the share instead.
4. **Add redundant activating shares:** Inclusion of redundant multiple shares in transmission would conceal the actual activating share. A predefined process would then be needed for the receiver to select the proper value, and ignore the remaining shares.

In symmetric key based authentication methods that do not provide confidentiality, the sender can use the activating share as part of the message to ensure its integrity, i.e., $(M \parallel \text{activating share}) \parallel (h_k(M \parallel \text{activating share}))$ in Figure 2.

4. CONCLUSIONS

A key transport scheme is presented for message authentication in communications networks. Its major strengths are:

- The receiver has minimal computational requirements for symmetric key recovery. For the generation of each new key, a simple operation (i.e., construction of a polynomial) is performed. The degree of the polynomial is not a critical design factor for consumer electronics or information technology devices.
- Although the prepositioned information shared between the receiver and the message source is fixed and functions as a permanent key, each distinct activating share allows a new symmetric key to be derived and used.
- Depending on the application in use, different customer authorization levels can be conveniently defined by assigning different shares to different receivers.

It is worth mentioning an interesting analogy with the public key systems. The prepositioned information can be considered to be the “private key” of the receiver. The public information, i.e., the activating share, sent as part of the

message determines the symmetric authentication key to be constructed. On the other hand, as the authentication keys are not generated at the message source, no additional cipher is needed to protect them in distribution.

The reader is encouraged to look for other applications of the key transport protocol. One particular area of interest is information hiding¹⁵. Prepositioned secret sharing schemes may also be used in ID-based key distribution protocols⁹.

REFERENCES

1. B. Schneier, *Applied Cryptography*, John Wiley and Sons, Inc, 1996.
2. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
3. W. Stallings, *Cryptography and Network Security, Principles and Practice*, Prentice-Hall, Inc, 1999.
4. A. M. Eskicioglu, "A Key Transport Protocol Based on Secret Sharing – An Application to Conditional Access Systems," IS&T/SPIE's 13th International Symposium on Electronic Imaging 2001, San Jose, CA, 21-26 January, 2001.
5. A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, November 1979.
6. G. R. Blakley, "Safeguarding cryptographic keys," Proceedings of the National Computer Conference, American Federation of Information Processing Societies, Vol. 48, pp. 313-317, 1979.
7. C. S. Lai and J. Y. Lee, "A new threshold scheme and its application in designing the conference key distribution cryptosystem," *Information Processing Letters*, Vol. 32, No. 3, pp. 95-99, 24 August 1989.
8. S. Berkovits, "How to broadcast a secret," *Advances in Cryptology – EUROCRYPT '91 Proceedings*, Springer-Verlag, pp. 535-541, 1991.
9. C. S. Lai and S. M. Yen, "On the design of conference key distribution systems for the broadcasting networks," Proceedings of IEEE INFOCOM '93, Vol. 3, pp. 1406-1413, San Francisco, CA, March 30-April 1, 1993.
10. G. J. Simmons, "How to (really) share a secret," *Advances in Cryptology – CRYPTO '88 Proceedings*, Springer-Verlag, pp. 390-448, 1990.

11. G. J. Simmons, "Prepositioned shared secret and/or shared control schemes," *Advances in Cryptology – EUROCRYPT '89 Proceedings*, Springer-Verlag, pp. 436-467, 1990.
12. www.atsc.org
13. www.havi.org
14. www.dvb.org
15. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding – a survey," *Proc. IEEE*, Vol. 87, No. 7, July 1999.