CERIAS Tech Report 2002-71

A Key Transport Protocol Based on Secret Sharing Applications to Information Security

by A Eskicioglu, E Delp
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

A Key Transport Protocol Based on Secret Sharing Applications to Information Security

Ahmet M. Eskicioglu and Edward J. Delp, *Fellow, IEEE*

Abstract— **Digital multimedia content is delivered to homes via the Internet, satellite, terrestrial and cable networks. Scrambling is a common approach used by conditional access systems to prevent unauthorized access to audio/visual data. The descrambling keys are securely distributed to the receivers in the same transmission channel. Their protection is an important part of the key management problem. Although public-key cryptography provides a viable solution, alternative methods are sought for economy and efficiency.**

Message authentication is an important objective of information security in modern electronic distribution networks. 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.

This paper presents a key transport protocol based on secret sharing. Conditional access and message authentication are two important application areas for which the advantages of the proposed protocol are discussed. The protocol eliminates the need for a cipher, yet effectively combines the advantages of symmetric and public-key ciphers. It can be used to build a new key management scheme that allows the service providers to generate different keys for different sets of receivers, and to renew these keys in a convenient way.

Keywords— **cipher, conditional access, content protection, data integrity, digital signature, encryption, hashing, key transport, message authentication, multimedia, public-key cryptography, prepositioned secret sharing.**

I. Introduction

WE will focus on two application areas, i.e., conditional access and message authentication [1], [2], and discuss the advantages of the proposed key transport protocol for delivering keys to the home entertainment devices.

A. Conditional access

With the widespread availability of digital distribution technologies, consumers have access to a variety of services from satellite or terrestrial broadcasters, cable operators, and the Internet. The service providers deliver different types of multimedia content ranging from free access programs to services such as PayTV, Pay-Per-View and Video-on-Demand.

A conditional access (CA) system [3], [4] is a system that allows access to services based on payment or other requirements such as identification or authorization. The user enters into an agreement with the service provider to obtain the access rights. A typical architecture of a CA system and its major components are shown in Figure 1.

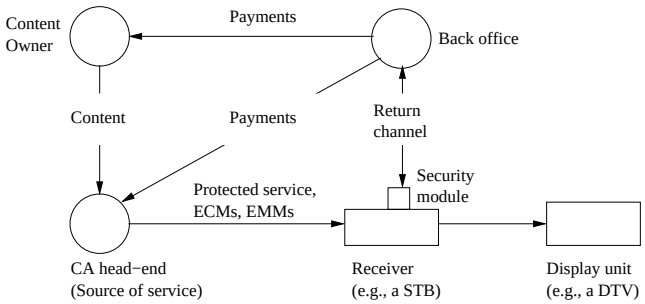

Fig. 1. CA system architecture

Presently, an important source of content is the movie studios represented by the Motion Picture Association of America (MPAA). ABC, CBS, NBC, DirecTV and Time-Warner are among the leading service providers in the US. CA systems are developed by companies commonly called the CA providers. NDS, Canal+ and Nagravision are examples of CA vendors with businesses in both the US and Europe.

The service and the entitlement messages indicating the access conditions are protected at the CA head-end before they are delivered to the customer. There are two types of entitlement messages [5] associated with each program in a service: The Entitlement Control Messages (ECMs) carry the descrambling keys (usually called the "control word"s in the terminology for CA systems) and a brief description of the program (program number, date, time, cost, etc.) while the Entitlement Management Messages (EMMs) specify the service-related authorization levels. The EMMs can be distributed on the same channel with the service or sent on a separate channel such as a telephone line. The ECMs are usually multiplexed with the associated program.

A simplified head-end architecture is given in Figure 2. Multiple streams input to the multiplexer are time multiplexed before the audio/video (A/V) data is scrambled. The signal is finally modulated for transmission through the network.

Encryption-based technologies are widely used for protecting distributed content. If the customer is authorized to watch a particular protected program, the A/V stream is descrambled, and sent to the display unit for viewing. In today's CA systems, a removable security module (e.g, a smartcard) is commonly used for securely handling the ECMs and EMMs, and handling authorization checks and

A. M. Eskicioglu is a private consultant in New York.

E. J. Delp is with the Video and Image Processing Lab, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. This research was partially funded (to EJD) by the Indiana 21st Century Research and Technology Fund.

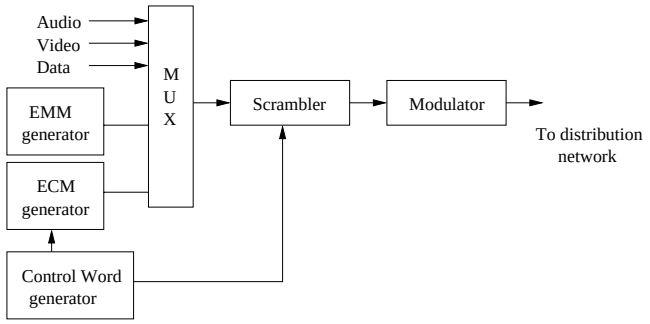

Fig. 2. Major components of the CA head-end architecture

purchases. In the US, the National Renewable Security Standard (NRSS) [6] defines a renewable and replaceable security element for use in consumer electronics devices such as digital set-top boxes and digital TVs. In Europe, the Digital Video Broadcasting (DVB) [3] project has specified the common interface (CI) between a host device and a security module.

Separating the security functionality from the navigational devices (i.e., devices that are capable of switching between the channels) has an important consequence. It will allow the consumer electronics (CE) industry to manufacture devices independent of the private CA systems. Commercial availability of CE products at retails stores is believed to be an essential factor for a fair market competition. Figure 3 depicts the architecture of a generic receiver with an NRSS-compliant security module. Note that EMM and ECM processing and content descrambling all take place in the NRSS module.

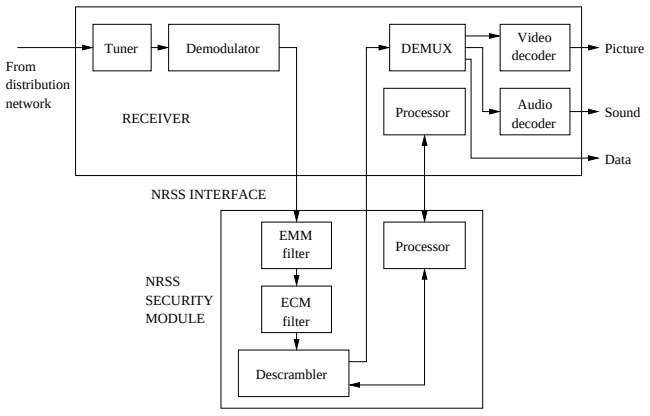

Fig. 3. NRSS security module and its host

A major component of every CA system is a back office that keeps track of all the transactions made. It is the responsibility of the security module to temporarily store the transaction records. At specified times, these records are sent to the back office for processing. As this transmission involves sensitive financial and personal data, a secure channel has to be established between the security module and the back office. In the case of Internet Service Providers (ISPs), the source of content and the back office may be co-located.

To complete the cycle, a portion of the payments received from the customers for the purchased services is sent to the content owners and service providers.

If the receiving and display units are two different devices in a home network, the interface between them should also be protected. The current DirecTV or cable systems include separate receivers popularly called set-top boxes (STBs). In the newly developed Advanced Television System Committee (ATSC) [7] system or in Internet-based CA applications, the receiver and the display unit are in the same box.

The services are usually scrambled using symmetric ciphers such as the Data Encryption Standard (DES). For security reasons, the scrambling key is changed frequently, the period of change being on the order of a few seconds. Although the protection of the ECMs is often privately defined by the CA providers, public-key cryptography [8], [9] is a viable tool for transporting the keys from the service source to the receivers. The descrambling keys are encrypted with a public key at the source , and recovered by the corresponding private key stored in the receiver.

In spite of the fact that public key cryptography is an elegant way to protect ECMs, it has major disadvantages. Public key schemes are considerably slower than symmetric key schemes, and have longer keys. Their security is based on the difficulty of solving number-theoretic computational problems. RSA (the most widely used algorithm), for example, assumes that the integer factorization problem is intractable.

B. Message authentication

Authentication is one of the four most important objectives of information security [8], [9], [10]. The others are confidentiality, data integrity and non-repudiation. In communication networks, so me or all of these objectives may need to be met.

- *Confidentiality*: Information is made accessible only to authorized parties. Encryption techniques provide confidentiality by transforming data into unintelligible format. This is a reversible process, and the entity in possession of the right key can recover the data.
- *Data integrity*: Parties have assurance that information has not been altered in an unauthorized way. Hashing functions, which produce compact representations of data, are commonly used for checking data integrity.
- *Non-repudiation*: When a dispute arises a result of a party in denial of an action, e.g., involvement an electronic transaction, it can be resolved with the participation of a trusted third party acting as a judge.

Authentication methods can be studied in two groups: Entity authentication and message authentication. To understand the conceptual difference between the two, let us consider the following scenario:

Figure 4 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 net-

work, B would require one or more of the following on receipt of the message [9]:

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

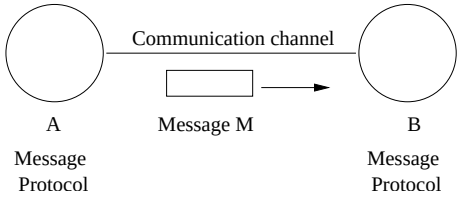

Fig. 4. Two-party communications

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, on the other hand, assures B of not only the identity of A but also his active participation. Sometimes, two parties need to authenticate each other for messages to flow in either direction. Challenge-response protocols for mutual authentication are based on symmetric or public key schemes and zero-knowledge protocols.

Although message authentication provides no guarantees of timeliness or uniqueness, it is very useful in communications where one party is not active during the execution of the message protocol. To avoid replay attacks (i.e., a intruder masquerades as A, and sends a previously used message), time-variant data (sequence numbers, time stamps, etc.) can be added to the message.

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 groups [9]:

1. *Unkeyed hash functions*: the message is the only input.
2. *Keyed hash functions*: the message and a secret key are two inputs.

Each group of functions is identified with certain properties. A particular class of unkeyed hash functions contains Manipulation Detection Codes (MDCs). They differ in the way the input message is compressed:

- Hash functions based on *block ciphers* (make use of an existing block cipher).
- Hash functions based on *modular arithmetic* (make use of an existing capability of performing modular arithmetic).
- *Customized* hash function (no assumption of block ciphers or modular arithmetic).

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.

With this background, we can now classify the message authentication methods with a particular interest in how they exploit symmetric or public key ciphers:

1. MACs
2. Message encryption
3. Digital signatures

In the rest of the paper, the following cryptographic notation will be used for denoting encryption and hashing algorithms:

$E_K(M)$:	Encryption of message M with key K
$h(M)$:	Hashing of message M with an MDC
$h_K(M)$:	Hashing of message M with a MAC with key K
$M_1 \parallel M_2$:	Concatenation of message M_1 with message M_2
$S_{Kprivate}(M)$:	Signing of message M with private key Kprivate

B.1 Method 1. Using a MAC

The process of producing a MAC is depicted in Figure 5. 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.

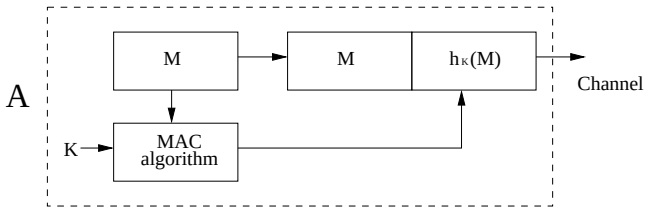

Fig. 5. Authentication with a MAC

B.2 Method 2. Encrypting the message

a) *Symmetric key encryption*: As shown is Figure 6, 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.

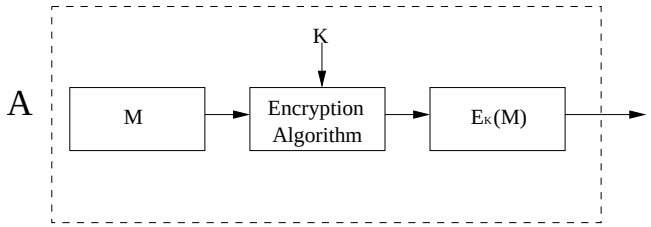

Fig. 6. Authentication with message encryption

In practice, encryption can be used together with MDCs or MACs. Some suggested basic schemes are as follows [9], [10]:

- $E_K[M \parallel h(M)]$
- $E_{K2}[M \parallel h_{K1}(M)]$
- $E_{K2}(M) \parallel h_{K1}(M)$
- $E_{K2}(M) \parallel h_{K1}[E_{K2}(M)]$
- $E_K[M \parallel h(M \parallel S)]$, where S is a shared secret.

B.3 Method 3. Signing the message

In Figure 7, 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 as surance that the message was generated by A because A is the only party that owns the private key. Again, it is assumed that B has the ability to distinguish between legitimate and garbled plaint exts.

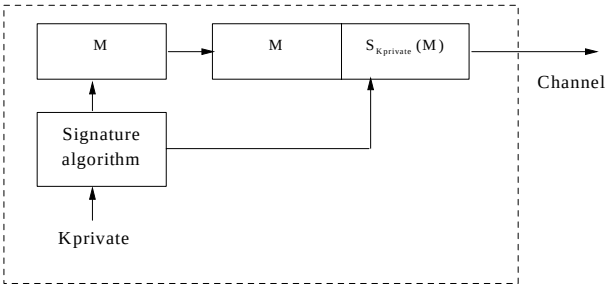

Fig. 7. Authentication with a digital signature

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. An analysis of the three approaches shows that key management is an important aspect of message authentication. Table I summarizes the need for keys in different authentication methods.

B.4 Disadvantages

Let us consider the disadvantages of using a fixed key for MAC creation, message encryption and message signing:

a) Potential cryptographic weakness

TABLE I

KEY TYPES FOR MESSAGE AUTHENTICATION METHODS

Method Key type	Symmetric Key	Private Key	Public Key
MAC creation	x		
MAC verification	x		
Encryption	x		
Decryption	x		
Signature creation		x	
Signature verification			x

The following attacks are possible for the three authentication methods [8], [9], [10]:

Method 1: The symmetric key, shared by the sender and the receiver, needs to be used for all messages during its lifetime. This makes the method vulnerable to attacks for key recovery and MAC forgery. There are two possible attacks: Attacks on the key space and attacks 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 $min(2^t, 2^n)$.

Method 2: Suppose encryption is used alone for message authentication. This method is also 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.

Method 3: 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. 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.

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 different 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.

Code authentication [11], [12], [13] is an important issue in digital distribution networks. In the future, sophisticated home entertainment devices handling audio/video data will receive software for various applications from several sources (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. The service providers would like to have assurance that their application is received and used only by authorized devices. The device manufacturers would, in turn, be concerned about unauthorized services using their devices.

II. A KEY TRANSPORT PROTOCOL BASED ON SECRET SHARING

We describe a system[1], based on secret sharing [8], [9], [14], [15], that eliminates the need for public key cryptography (or any other cipher), and facilitates the secure transmission of keys from the service providers to the receivers.

A. Threshold schemes

A (t, n) threshold scheme ($t \leq n$) 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. A perfect threshold scheme is a threshold scheme in which a knowledge of $(t-1)$ or fewer shares gives no information about the secret. 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.

After the introduction of the idea by two independent publications [14], [16] in 1979, several threshold schemes have been developed based on a common theoretical background. In Shamir's (t, n) threshold scheme, the secret S is the coefficient a_0 of a random $(t-1)$-degree polynomial

$$f(x) = (a_{t-1}x^{t-1} + \cdots + a_1 x + a_0) \pmod{p} \qquad (1)$$

over the finite Galois Field $GF(p)$, where p is a prime number larger than both S and n. Each of the n shares (x_i, y_i) is a point on the curve defined by the polynomial $f(x)$.

[1]Thomson multimedia, Inc. patent pending

As a polynomial of degree $(t-1)$ can be uniquely determined by t points, the secret can be computed from t shares. Threshold schemes have proved useful in many applications of cryptography including electronic cash, group signatures, key recovery and voting. In particular, some authors [17], [18], [19] 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.

B. A Prepositioned Secret Sharing Scheme for Key Transport

Shamir's (2, 2) threshold scheme will be used in presenting the key transport protocol. In practice, it may be important to choose the keys randomly and independent of the polynomial construction. The key generation and distribution process can then be automated by using the following steps:

1. Choose S.
2. Construct the polynomial $f(x)$ that passes through $(0, S)$ and (x_0, y_0).
3. Compute $f(x)$ at $x_1, x_1 \neq x_0$.
4. Distribute (x_1, y_1) with the data generated using S.

CA Systems: In Figure 8, the receiver (or the removable security module) is manufactured with the point (x_0, y_0) on the first degree polynomial to be constructed. The CA system at the source chooses the secret S, scrambles the A/V data, and transmits the scrambled A/V data 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 descrambles the A/V data. ES and DS denote scrambling and descrambling with the symmetric key S, respectively.

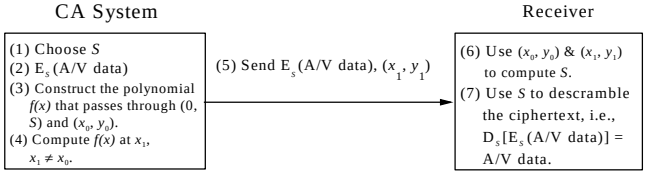

Fig. 8. Transport of descrambling keys with prepositioned secret sharing

Message Authentication Systems: In Figure 9, 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 the 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 computes the authenticator. If the

authenticator from the sender is not the same, the message is rejected.

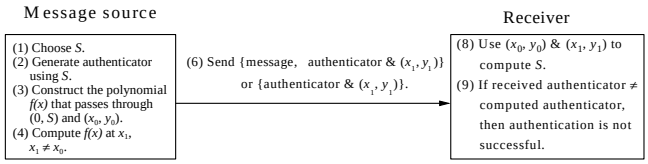

Fig. 9. Transport of authentication keys with prepositioned secret sharing

Note that since the shared secret is used as a symmetric key, the proposed scheme is applicable to methods 1 and 2, and not method 3 which requires an asymmetric key pair.

This key transport protocol an example of a prepositioned shared secret scheme [15], [20] where it is possible to reconstruct different keys by communicating different activating shares for the same prepositioned information. The use of such a scheme has been discussed within the context of critical military applications. An interesting scenario is when a private piece of information must be communicated from a commander to a group of subordinate officers to launch a missile. Two desired requirements of the system would be:

1. the officers should not be able to cooperate to find the launch code without their commander's participation.
2. the commander should be able to send a different piece of private information to activate a different launch code.

C. Generalization of the scheme

In a generalization of the proposal, the value of t is a system parameter. Choosing a higher value for t, and storing $(t-1)$ shares in the receiver would increase the system's resistance to ciphertext-only attacks, but lead to more computations for polynomial construction.

Multiple shares can also be used to build a convenient key management scheme in a CA or message authentication system. The system operators may define three levels of keys: individual, group and regional. Receivers can be assigned different authorization levels by storing different numbers of shares. The simple scenario below will explain how the required key hierarchy can be established with secret sharing.

Consider a system in which a population of receivers is used for keeping authorizations. Three different receivers are specified:

- Level 1: All the receivers in the broadcast region are assigned one common share (the polynomial is of first degree).
- Level 2: All the receivers in a given group are assigned an additional common share (the polynomial is of second degree).
- Level 3: Each receiver is assigned a unique additional share (the polynomial is of third degree).

Figure 10 shows the generation of the keys at the specified three levels. For demonstration purposes, modular arithmetic was not used in obtaining the graph.

TABLE II
POINTS FOR POLYNOMIAL CONSTRUCTION

Point Degree of polynomial	First	Second	Third
The activating share = (5,10)	x	x	x
The common share for Level 1 = (17,15)	x	x	x
The common share for Level 2 = (12,6)		x	x
The unique share for Level 3 = (3,12)			x

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

Message Authentication Systems: If the code is broadcast for all the receivers in the region, the receivers will construct a first degree polynomial using the common share, and obta in 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 receivers in the region.

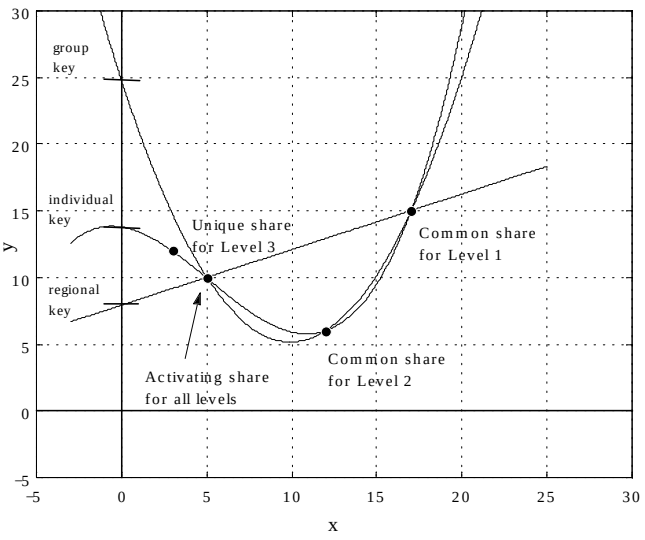

Fig. 10. Key generation for three authorization levels

Example: Three polynomials will be constructed using the marked points on the graph in Figure 10. The first degree will pass through the common share for Level 1and 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 $p = 23$. The coordinates of the points needed for the three polynomials are given in Table II.

(a) First degree polynomial:

The coefficients of the first degree polynomial

$$f(x) \quad = \quad a_1 x + a_0 \pmod{23} \tag{2}$$

are obtained by solving

$$\begin{array}{rcrcrcl} a_1 17 & + & a_0 & = & 15 & \pmod{23}, \\ a_1 5 & + & a_0 & = & 10 & \pmod{23}. \end{array} \tag{3}$$

The solution gives $(a_1, a_0) = (10, 6)$. Hence, $S = 6 \pmod{23}$.

(b) Second degree polynomial:

The coefficients of the second degree polynomial

$$f(x) \quad = \quad a_2 x^2 + a_1 x + a_0 \pmod{23} \tag{4}$$

are obtained by solving

$$\begin{array}{rcrcrcrl} a_2 17^2 & + & a_1 17 & + & a_0 & = & 15 & \pmod{23}, \\ a_2 12^2 & + & a_1 12 & + & a_0 & = & 6 & \pmod{23}, \\ a_2 5^2 & + & a_1 5 & + & a_0 & = & 10 & \pmod{23}. \end{array} \tag{5}$$

The solution gives $(a_2, a_1, a_0) = (10, 20, 5)$. Hence, $S = 5 \pmod{23}$.

(c) Third degree polynomial:

The coefficients of the third degree polynomial

$$f(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0 \pmod{23} \tag{6}$$

are obtained by solving

$$\begin{array}{rcrcrcrl} a_3 17^3 + a_2 17^2 + a_1 17 + a_0 & = & 15 & \pmod{23}, \\ a_3 5^3 + a_2 5^2 + a_1 5 + a_0 & = & 10 & \pmod{23}, \\ a_3 12^3 + a_2 12^2 + a_1 12 + a_0 & = & 6 & \pmod{23}, \\ a_3 3^3 + a_2 3^2 + a_1 3 + a_0 & = & 12 & \pmod{23}. \end{array} \tag{7}$$

The solution gives $(a_3, a_2, a_1, a_0) = (18, 19, 0, 22)$. Hence, $S = 22 \pmod{23}$.

In general, the coefficients of the polynomial $f(x)$ of degree at most $(t-1)$ are computed from

$$f(x) \quad = \quad \sum_{i=0}^{t-1} y_i \prod_{0 \le j \le t-1, j \ne i} \frac{(x - x_j)}{(x_i - x_j)}, \tag{8}$$

where $(x_i, y_i), 0 \le i \le (t-1)$ are the coordinates of the points defining $f(x)$.

With this notation, the expression for the shared key obtained from t secrets is

$$f(0) = \sum_{i=0}^{t-1} y_i \prod_{0 \le j \le t-1, j \ne i} \frac{x_j}{(x_j - x_i)}. \tag{9}$$

D. Security Analysis

In the prepositioned shared secret scheme for missile launching, the activating share is the only data communicated to the officers to determine the corresponding missile launch code. In our scheme, the shared secret is used to obtain the data (i.e., the ciphertext in CA systems or the authenticator in message authentication systems) which is broadcast with the activating share. The system is therefore exposed to brute-force attacks for small values of t, i.e., lower degree polynomials. A potential hacker may use the available data 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.

Case $t = 2$:

The system is most vulnerable if first degree polynomials are used. If the hacker finds two keys, he can compute the prepositioned information by constructing two straight lines and finding their intersection.

Let the polynomials of degree one be denoted by $f_i(x) = a_{i1} x + a_{i0}$.

$(0, K_1)$, where K_1 is the first key found, and (x_1, y_1), the corresponding activating share AS_1, determine the first polynomial $f_1(x) = a_{11} x + a_{10}$. Similarly, $(0, K_2)$, where K_2 is the second key found, and (x_2, y_2), the corresponding activating share AS_2, determine the second polynomial $f_2(x) = a_{21} x + a_{20}$. The intersection of these two polynomials reveal the permanent key in the receiver. Note that the polynomials associated with the other keys K_i, $i > 2$, also pass through the same intersection because of the linearity property.

Case $t > 2$:

Higher values of t make cryptanalysis increasingly more difficult. The security is based on the difficulty of estimating the prepositioned information in the receiver. For each 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 on the polynomial. In general, each pair can be used to construct 2 linear equations in t variables. In reduced form, there is 1 linear equation in $(t-1)$ variables as the ordered pair $(0, K_i)$ determines the value of a_{i0}.

Consider the case $t = 3$. Let the polynomials of degree two be denoted by $f_i(x) = a_{i2} x^2 + a_{i1} x + a_{i0}$.

Each pair $(0, K_i), (x_i, y_i)$, $i = 1, 2, 3, \cdots$, introduces two linear equations in 3 variables or, in reduced form, one linear equation in 2 variables. As the permanent key (i.e., the set of points kept in the receiver) represents the points of intersection of *all* the second degree polynomials associated with the keys K_i, it is not possible to derive a complete set of equations to find the key.

For higher value of t and a big set of K_i, the problem of finding the permanent key practically becomes intractable.

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

1. Define the authentication key as a function of the shared secret: In Shamir's threshold scheme, the secret (hence 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. Alternative definitions are also possible using the characteristics of the polynomial. 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. The number of shares kept in the receiver is an important piece of information for key recovery.
3. "Mask" the activating share before distribution: Although the transmission of the activating share in-the-clear does not introduce any major weakness, it may be masked for additional security. An unkeyed hash function can be used for this purpose, avoiding the need for key management. The sender would use the hash value of the activating share for generating the ciphertext, 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.

Copy protection is another important issue for the content providers. Delivery systems will carry the information along with the copyrighted content that indicates if the consumer is authorized to make a copy. Other methods and key management schemes are needed to prevent unauthorized access to content across the interfaces and in storage [21]. Conditional access and copy protection are two critical issues that need to be addressed in parallel for the management of rights associated with the consumption of digital content.

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.,

$$(M \parallel activatingshare) \parallel (h_K(M \parallel activatingshare)).$$

III. CONCLUSIONS

The proposed scheme can be a convenient key transport mechanism for conditional access and code authentication systems associated with home entertainment devices. It can also be used in other architectures requiring secure communications. The major strengths of such an approach include:

- 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.
- 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.
- Unlike the popular public-key systems, the security does not rely on unproven mathematical assumptions. The degree of the polynomial can be determined based on the desired level of security.

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 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. Prepositioned secret sharing schemes may also be used in ID-based key distribution protocols.

Ramp schemes [22] or dynamic threshold schemes [23], [24], which are extensions of conventional threshold schemes, may prove useful in developing key transport protocols.

References

[1] A. M. Eskicioglu, "A key transport protocol for conditional access systems," *Proceedings of SPIE: Security and Watermarking of Multimedia Content III*, vol. 4314, pp. 139–148, January 22-25 2001.

[2] A. M. Eskicioglu, "A prepositioned secret sharing scheme for message authentication in broadcast networks," *Proceedings of the Communications and Multimedia Security Issues of the New Century*, pp. 363–373, May 21-22 2001.

[3] R. de Bruin and J. Smits, Artech House, Inc., 1999.

[4] H. Benoit, Arnold, 1997.

[5] ISO-IEC, *International Standard 13818-1*, 1996, First Edition.

[6] EIA, *EIA-679B National Renewable Security Standard*, September 1998.

[7] "Advanced television systems committee standard a/53," available at http://www.atsc.org.

[8] B. Schneier, John Wiley and Sons, Inc., 1996.

[9] J. Menezes, P. C. van Oorschot, and S. A. Vanstone, CRC Press, 1997.

[10] W. Stallings, Prentice-Hall, Inc., 1999.

[11] "www.atsc.org," .

[12] "www.havi.org," .

[13] "www.dvb.org," .

[14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, November 1979.

[15] G. J. Simmons, "How to (really) share a secret," *Advances in Cryptology – CRYPTO '88 Proceedings, Springer-Verlag*, pp. 390–448, 1990.

[16] G. R. Blakley, "Safeguarding cryptographic keys," *Proceedings of the National Computer Conference*, vol. 48, pp. 313–317, 1979.

[17] C. S. Laih and J. Y. Lee, "A new threshold scheme and its application in designing the conference key distribution cryptoystem," *Information Processing Letters*, vol. 32, no. 3, pp. 95–99, 24 August 1989.

[18] S. Berkovits, "How to broadcast a secret," *Advances in Cryptology – EUROCRYPT '91 Proceedings, Springer-Verlag*, pp. 535–541, 1991.

[19] C. S. Laih 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, March 30 - April 1 1993.

[20] G. J. Simmons, "How to (really) share a secret," *Advances in Cryptology – CRYPTO '88 Proceedings, Springer-Verlag*, pp. 390–448, 1990.

[21] A. M. Eskicioglu and E. J. Delp, "An overview of multimedia content protection in consumer electronics devices," *Signal Processing: Image Communication*, vol. 16, no. 5, pp. 681–699, April 2001.

[22] G. Blakley and C. Meadows, "Security of ramp schemes," *Advances in Cryptology – CRYPTO '84 Proceedings, Springer-Verlag*, pp. 242–268, 1985.

[23] C. S. Laih, L. Harn, J. Y. Lee, and T. Hwang, "Dynamic threshold scheme based on the definition of cross-product in an n-dimensional linear space," *Advances in Cryptology – CRYPTO '89 Proceedings, Springer-Verlag*, pp. 286–298, 1990.

[24] C. Blundo, A. Cresti, A. De Santis, and U. Vaccaro, "Fully dynamic secret sharing schemes," *Advances in Cryptology – CRYPTO '93 Proceedings, Springer-Verlag*, pp. 110–125, 1994.

Ahmet M. Eskicioglu Ahmet M. Eskicioglu received the B.S. degree from the Middle East Technical University (METU), Ankara, Turkey, and the M.S. and Ph.D. degrees from the University of Manchester Institute of Science and Technology (UMIST), England. He was with the Computer Engineering Department, METU from 1983 to 1992, the Department of Computer Sciences, University of North Texas from 1992 to 1995, and Thomson Multimedia Corporate Research, Indianapolis from 1996 to 2001. He has participated in the development of several national and international standards for conditional access and copy protection in the US and Europe. These include the Advanced Television Systems Committee (ATSC) conditional access system, the Electronics Industries Alliance (EIA) National Renewable Security Standard (NRSS), the Digital Video Broadcasting (DVB) Content Protection and Copy Management (CPCM) System, and the Content Scramble System (CSS) for DVD players. Dr. Eskicioglu is a consultant in New York. His interests include image compression, system simulation, data security, conditional access, and digital rights management.

Edward J. Delp (S'70-M'79-SM'86-F'97) was born in Cincinnati, Ohio. He received the B.S.E.E. (cum laude) and M.S. degrees from the University of Cincinnati, and the Ph.D. degree from Purdue University. From 1980-1984, Dr. Delp was with the Department of Electrical and Computer Engineering at The University of Michigan, Ann Arbor, Michigan. Since August 1984, he has been with the School of Electrical and Computer Engineering and the Department of Biomedical Engineering at Purdue University, West Lafayette, Indiana, where he is a Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering. His research interests include image and video compression, multimedia security, medical imaging, multimedia systems, communication and information theory. He is a Fellow of the IEEE, a Fellow of the SPIE, and a Fellow of the Society for Imaging Science and Technology (IS&T). In 2000 he was selected a Distinguished Lecturer of the IEEE Signal Processing Society. From 1997-1999 he was Chair of the Image and Multidimensional Signal Processing (IMDSP) Technical Committee of the IEEE Signal Processing Society. He was Co-Chair of the SPIE/IS&T Conference on Security and Watermarking of Multimedia Contents that was held in San Jose in January 1999, January 2000, January 2001, and January 2002. He is the Program Co-Chair of the IEEE International Conference on Image Processing that will be held in Barcelona in 2003. From 1991-1993, he was an Associate Editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. From 1992-1999 he was a member of the editorial board of the journal *Pattern Recognition*. From 1994-2000, Dr. Delp was an Associate Editor of the *Journal of Electronic Imaging*. From 1996-1998, he was an Associate Editor of the *IEEE Transactions on Image Processing*. In 1990 he received the Honeywell Award and in 1992 the D. D. Ewing Award for excellence in teaching. In 2000 he received the Raymond C. Bowman Award for fostering education in imaging science from the Society for Imaging Science and Technology (IS&T). During the summers of 1998, 1999, and 2001 he was a Visiting Professor at the Tampere International Center for Signal Processing at the Tampere University of Technology in Finland. In 2002 he received a chaired professorship and is currently the Silicon Valley Professor of Electrical and Computer Engineering at Purdue University.