16

Establishing a key hierarchy for Conditional Access without encryption

J. Schwenk
Deutsche Telekom AG, Technologiezentrum
Am Kavalleriesand 3, D-64295 Darmstadt, Germany,
schwenk@fz.telekom.de

Abstract

Conditional Access systems use special key management schemes which ensure that encrypted broadcast services can only be accessed by those who are entitled to receive them. In many cases, a key hierarchy is used for this purpose. In this article, solutions are presented to improve existing Conditional Access systems by using deeper key hierarchies, and by establishing these hierarchies without the use of encryption techniques.

Keywords

Conditional Access, cryptographic key management, encryption, Pay-TV, polynomials, threshold schemes, tree structure

1 INTRODUCTION

Conditional Access systems

A Conditional Access System is „the complete system for ensuring that broadcast services are only accessible to those who are entitled to receive them" (EBU 1995). It consists of two main parts: The encryption of the signal to be protected (e.g. video, audio, data), and the key management which is necessary to guarantee that only the entitled devices will be able to decipher this signal. Since encryption techniques may also be used in the key management, the encryption of the signal itself is often called „scrambling"; this terminology will be adopted in this paper.

We will not go into the details of Conditional Access systems as they are in use now. The reader may refer to (EBU 1995), (McCormack 1994) and (Schwenk 1994) for more detailed information.

We can state two fundamental conditions for the key management:

P. Horster (ed.), *Communications and Multimedia Security II*

- It must be possible to address individual customers in order to give them a new entitlement or to delete (some of) his entitlements. To do this in a secure way, an individual secret information of cryptographic nature has to be associated with each customer.
- The signal itself can only be encrypted with a single algorithm and a single key.

In order to solve the key managment problem in Conditional Access systems, we have to find a way to effectively distribute a single key (which we will call *SK* for „signal key") to a large and changing group of individual customers, each of which is equipped with an individual personal key information PK_i.

In some of today's Conditional Access systems, this key *SK* is not directly used to decramble the signal, but to decipher short cryptograms called ECM („Entitlement Control Messages") associated with the signal. These cryptograms then contain the key („Control Word") to descramble the signal. Since this fact does not influence our argumentation, it will be ignored in the following discussions.

Key hierarchies

One solution to distribute *SK* is to use a key hierarchy which contains the signal key *SK* as a root and the individual keys PK_i as leaves. This solution will be described and refined in the next section. A solution for establishing such a key hierarchy without using encryption will then be given in section 3.

Pirate Devices

We assume that the descrambling of the signal will be done by a device owned by the customer called „decoder" or „terminal", and that all (cryptographic) computations related to the key management take place inside a security module, e.g. a smart card. A pirate device (pirate security module) is a device which has been produced by an entity who is not authorized to do so.

Some definitions from Graph Theory

In the following chapters, we will use some terminology from graph theory: A directed graph $G = (V,E)$ consists of a vertex set $V = \{v_1,...,v_n\}$ and a set of directed edges $E = \{(v_i,v_j) | v_i,v_j \in V\}$. The vertex v_i is the starting point of the edge (v_i,v_j), and v_j is its endpoint. The vertex v_i is also called predecessor of v_j. A path from vertex v to w is a sequence of edges where v is the starting point of the first edge, w the endpoint of the last edge, and the endpoint of each edge is the starting point of the next edge. A circle is a closed path, and a (directed) tree is a directed graph which contains no circles. A vertex in a tree is called a root if all paths in the graph can be extended to paths which end in this vertex. A vertex of a tree is called leaf if it is not the endpoint of any edge.

2 KEY HIERARCHIES

Key hierarchies form the basis of modern Pay-TV systems, although there are other solutions which are discussed (Fiat and Naor, 1994). Figure 1 below depicts a state-of-the-art key hierarchy which is e.g. described in (DE 33 25 858 A1). Other descriptions of key hierarchies can be found in (Schwenk 1994), (DIN EN 50 094).

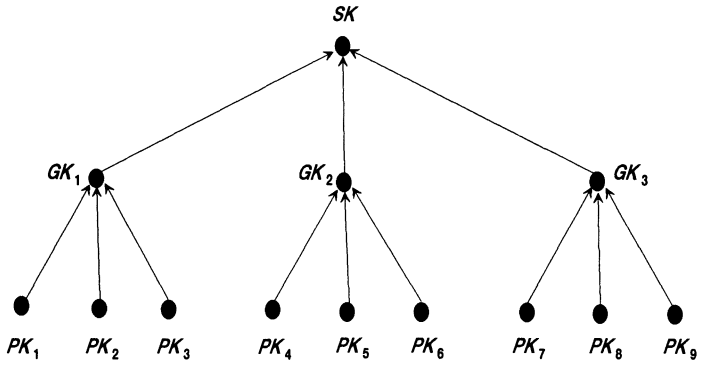

Figure 1 A key Hierarchy with three levels. This key hierarchy forms a directed graph with root SK and leaves $PK_1,...,PK_9$.

To establish (or to modify) the key hierarchy of Figure 1 using encryption techniques one has to proceed as follows:

- The personal key PK_i of user i has already been placed inside his security module during the personalization process.
- These keys are now used to encrypt and then distribute the group keys GK_i. E.g. to establish GK_1, this key is encrypted using PK_1, PK_2 and PK_3, and then the three different cryptograms are broadcasted. Now exactly the users 1, 2 and 3 can compute and store GK_1 by decyphering one of these messages. The same procedure is used to establish GK_2 and GK_3.
- Now that they are available, the keys GK_i are used to establish SK by broadcasting $GK_1(SK)$, $GK_2(SK)$ and $GK_3(SK)$ (here $A(B)$ means that data B has been encrypted using the key A). The security modules of the customers 1, 2 and 3 (who form group 1) can now get SK by decrypting $GK_1(SK)$, and the same applies for the other two groups.

Most of the key hierarchies used today are flat, i.e. they use only a small number of levels, e.g. three levels like in Figure 1. This has some advantages for practical applications, the most important one being a more flexible definition of entitlement. However, they have some disadvantages as far as security is concerned. In the following, we will describe two of these disadvantages and how to solve them by using a deeper key hierarchy.

Positive Addressing

One can think of two basic methods for deleting an entitlement of a customer. The first method would be to send a message to his security device containing a command to stop working. This method is called „negative addressing" because the message contains only negative information for the customer. Negative addressing is not very secure, since customers may find a way to filter these messages, or pirate security devices may simply ignore such commands.

A better way to switch off a customers security device is to send messages containing some necessary piece of information (e.g. new keys) to all security modules except the one to be switched off, and then use the new information in the Conditional Access system. E.g. in

Figure 1, to switch off user 5, a new key $GK_2^{(new)}$ must be given to users 4 and 6 by broadcasting $PK_4(GK_2^{(new)})$ and $PK_6(GK_2^{(new)})$, and a new SK has to be distributed using keys GK_1, $GK_2^{(new)}$ and GK_3. This method is called „positive addressing".

In a flat key hierarchy it may take some time to do this. We can give a general formula for the number of messages to be sent: If we address n customers using an m-ary tree (i.e. a tree where each vertex has exactly m predecessors) of t levels, we have

$$n \le m^{t-1},$$

where for given m and n the parameter t is chosen such that m^{t-1} is the smallest number greater or equal to n. The same applies for m if n and t are given.

To switch off one customer who corresponds to a leaf in this tree, we have to send at most

$$m(t-1)-1$$

messages in order to replace all the keys which lie on the path from this leaf to the root of the tree. For given n, this number equals $m\lceil \log_m n \rceil - 1$. Since the function

$$x \log_x n = \frac{x}{\ln x} \ln n$$

has a minimum for $x = e$, the best choice for m would be a small integer, e.g. $m = 2$ or $m = 3$.

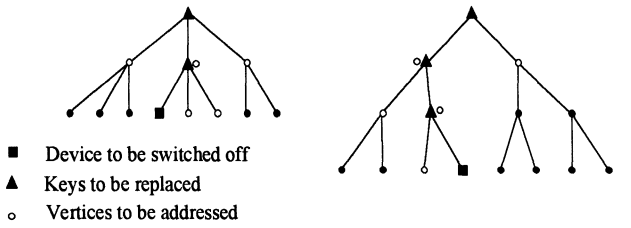

Figure 2 Positive addressing in a hierarchy with 3 and 4 levels, resp.

The advantages of using a deeper hierarchy cannot be seen in the toy example of Figure 2: to switch off the security module of customer 4, in both cases 5 vertices have to be addressed, i.e. 5 messages have to be sent.

In a more realistic scenario with one million customers, roughly 2047 messages have to be sent in a flat hierarchy with $t = 3$ and $m = 1024$, but only 40 messages are needed in a deep hierarchy with $t = 21$ and $m = 2$.

So positive addressing is faster (and cheaper) in a deep hierarchy.

The Mailbox Attack

The following attack could be used by pirates to distribute entitlements to pirate users: The fixed parts of the algorithm used in the security device of the Pay-TV provider are distributed as public domain software, e.g. on the internet. The variable part (e.g. the keys) are put on a mailbox which is updated regularily. The keys can then be read and given as an input to the

security device software. This attack can make ordinary anti-piracy strategies impossible if the variable information is anonymous, because in this case the person who updates the mailbox cannot be traced amongst the customers.

In a flat key hierarchy, the exchange of SK takes some time, so the lifetime of SK is quite long. This fact can be used in the Mailbox Attack because in this case the anonymous key SK can be put in the mailbox.

In contrast to this, SK has a very short lifetime in a deep hierarchy. Each time a new level is introduced into the key hierarchy, the lifetime of SK is divided by a constant factor. When a certain number of levels is reached, this lifetime will be so short that an online connection to the mailbox would be needed to get SK in time. This forces the pirate to put a key from a lower level of the hierarchy on the mailbox, which is not any longer totally anonymous.

3 ESTABLISHING A KEY HIERARCHY WITHOUT USING ENCRYPTION

In many countries there are laws to control the use of cryptographic techniques, especially encryption. In some cases these laws do not allow to use strong encryption algorithms, but only weaker ones. This may have some consequences on the security of the key management of a Conditional Access system, since a natural approach for implementing a key hierarchy is to use encryption techniques as described in section 2.

The question now arises whether it is possible to establish a key hierarchy without encryption, and the surprising answer is „yes". Two possibilities how to do this are presented in this section.

The solutions presented below work because the key management of a Conditional Access system is controlled by a central authority. This authority is the only entity who modifies the key hierarchy.

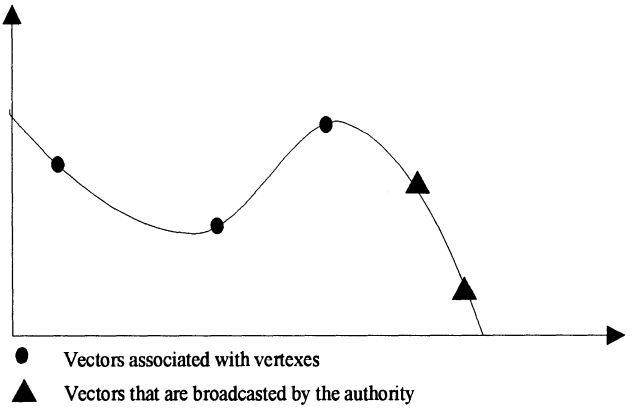

Figure 3 A broadcasting scheme based on polynomials.

Key establishment with polynomials

The basic idea is the following: A finite field GF(q) is used for all computations. For each of the m predecessors of a vertex, the central authority computes a point in GF(q)^2 from the key associated with it, and constructs a polynomial $p(x)$ of degree $m-1$ through these points.

It then broadcasts $m-1$ different points which lie on the graph of this polynomial. Now each security module which knows the parameters of one of these successors can compute by itself one additional point and therefore knows m points. It is then able to reconstruct $p(x)$. The key to be associated with the vertex is then the value of this polynomial at a certain predefined point: $k_0 = p(x_0)$.

Taking into account additional security considerations, a concrete implementation of the scheme may be the following: The authority has chosen a one-way function $f(.)$ with two arguments and a finite field GF(q), and has associated with each vertex j in the key management tree a point (x_j, y_j). Each security module knows all the information contained on the path from the leaf associated with the unique personal key PK_i to the root SK. Now let $j_1, \ldots, j_m$ be the predecessors of vertex j. To assign a new key to vertex j, the authority chooses a random number r and computes for each $k \in \{j_1,\ldots,j_m\}$ the point

$$(a_k, b_k) := (f(r, x_k) \bmod q, f(r,y_k) \bmod q).$$

This randomization via a one-way-function is needed to protect the secret key information (x_k, y_k). By using Lagrange interpolation we get the polynomial

$$\begin{aligned} g(x) &= x^m + c_{m-1}x^{m-1} + \ldots + c_0 \\ &= \sum_{k \in \{1,\ldots,m\}} \frac{(x-a_1)\cdots(x-a_{k-1})(x-a_{k+1})\cdots(x-a_m)}{(a_k-a_1)\cdots(a_k-a_{k-1})(a_k-a_{k+1})\cdots(a_k-a_m)} b_k \bmod q, \end{aligned}$$

where w.l.o.g. $\{1,\ldots,m\} = \{j_1,\ldots,j_m\}$. The authority now selects $k-1$ random points on the graph of this polynomial, and broadcasts them together with the random number r. Each security module which nows the secret key information (x_k,y_k) associated with one of the predecessoers of vertex j is now able to compute (a_k,b_k) using the publicly known one-way function $f(.)$ and the broadcasted random number r. It can then reconstruct $g(x)$ and evaluate $k_0 = g(x_0)$ to get the new key information. From this key information k_0 a new secret point $(x_j, y_j) = (x(k_0), y(k_0))$ is derived.

Key establishment with threshold schemes

A natural generalization of this approach is the use of threshold schemes. A (m,n)-threshold scheme is a scheme to distribute the knowledge about a secret s amongst n shadows $s_1, \ldots, s_n$ in such a way that the following conditions are fulfilled:

- From any m shadows from $\{s_1,\ldots,s_n\}$ the secret s can be reconstructed.
- Knowing at most $m-1$ shadows does not reveal anything about the secret.

The polynomial of degree m-1 over GF(q) from the previous subsection can be considered as a (m,n)-threshold scheme, where $m \le n \le q$.

The authority could use a (m,n)-threshold scheme in the following way. To each vertex a shadow information is associated. To replace the secret key information at vertex j, the authority uses the m shadows associated to its successors and some random information to construct a $(m, 2m-1)$-threshold scheme. Then it broadcasts the random information and the $m-1$ shadows which do not belong to the successors of j. Each successor may then compute an additional shadow, and is therefore able to reconstruct the secret s, which will be used as the new secret information associated with vertex j.

Advantages

The key management schemes presented in this section have the advantage that no encryption is needed to implement them. In the concrete implementation of the polynomial-based scheme, no hidden channel is available to the authority, because choosing the random number such that $p(x_0)$ contains some information implies inverting the one-way-function $f(.)$. Only the key *SK* has to be used for decryption of the signal itself, but if a deep key hierarchy is used, this key can be changed very frequently, so a week key is acceptable in this place.

4 REFERENCES

EBU (1995) Technical Review No. 266 (Winter 1995/96).

Fiat, A. and Naor, M. (1994) *Broadcast Encryption.* Advances in Cryptology - CRYPTO 93 Proceedings. Springer-Verlag, Berlin, 480-491.

McCormac, J. (1994) *European Scrambling Systems (4th edition).* Waterford University Press.

Schwenk, J. (1994) *Security of Pay-TV Systems.* In: Deutsche Telekom 1994 Research Review, pp. 9-21.

DIN EN 50 094, *Zugriffskontrollsysteme für die MAC/Packet-Familie: Eurocrypt.* Beuth Verlag, Burggrafenstr. 6, 1000 Berlin 30.

DE 33 25 858 A1, *Mehrebenen-Verschlüsselungssystem zum Senden verschlüsselter Informationen* (cf. US 401258).

5 BIOGRAPHY

The author studied mathematics at the University of Gießen, Germany. In 1989 he received his masters degree for a work on interactive Zero-Knowledge proofs, and in 1993 his Ph. D. for a work on extended triple systems, a work which is related to the theory of elliptic curves over finite fields. Since November 1993 he is working at the research institute of Deutsche Telekom AG, mainly in the area of Conditional Access systems. Other research interests include the design and analysis of cryptographic protocols, and digital signatures.