RZ 3400 (# 93440) 02/11/02
Computer Science 13 pages

Research Report

DAB CA-PK: Conditional Access for Digital Audio Broadcast

Dirk Husemann and Michael Nidd

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{hud,mni}@zurich.ibm.com

DAB CA-PK: Conditional Access for Digital Audio Broadcast

White paper

IBM Research
Zurich Research Lab
Pervasive Computing

Dr Dirk Husemann, Dr Michael Nidd
{hud,mni}@zurich.ibm.com

January 14, 2002

Abstract

A fundamental requirement for commercial DAB data services is the availability of an easy to deploy conditional access (CA) mechanism for providing controlled access to services. Over the course of 2001 the WorldDAB Forum[10] has defined a set of requirements for conditional access for DAB data and audio services[11]. To quote from the recent *Call for Contributions for a DAB CA mechanism* "[t]he main points are minimal hardware requirements, support for DAB-only as well as DAB-mobile receiver platforms, [and] ease of deployment"[12]. Responding to this WorldDAB *Call for Contribution* we propose a DAB CA mechanism that implements these requirements.

1 Context

Commercial use of digital radio systems such as the Eureka-147 DAB system requires (among other things) a mechanism for datacasting content and services in a protected manner allowing access to the broadcast content only for consumers who have paid a subscription fee for the distributed content. Such systems are called conditional access systems and have been implemented, for example, for Internet applications and cable TV networks: the content is encrypted ("scrambled") and sent over the network to the receiver; the receiver obtains a key to the content and is able to access it.

There are different means of obtaining a content key. The consumer can acquire the key by establishing a connection to the content provider over the Internet and then download the key to her device (where the process involves some kind of payment process); the content provider can send the desired key to the consumer's receiver via the Internet or via the cable TV network; also, the content provider can configure the consumer's receiver to enable or disable reception of specific content. In the Internet case key distribution is straight forward as we have two-way one-to-one connections (using, for example, TCP) and a tree based network topology shielding the rest of the network from the bandwidth demands necessary for distributing the key to an individual receiver. Similarly, a number of cable TV systems already support a backchannel so that a bidirectional communication channel can be set up between cable system operator and the customer's set-top-box. In addition, both cable TV networks and satellite-based distribution systems have enough bandwidth at their disposal to support dedicated control channels to enable and disable specific receivers for specific broadcast content (and the receivers are more or less always reachable and connected).

The situation alters considerably with wireless broadcast networks such as digital radio networks: Here we have a *one-way broadcast channel* only. With radio broadcast networks the *bandwidth is rather limited*—at least limited enough that *entertaining a control channel* for several millions of receivers *is economically not an option* if not technically an impossibility. Furthermore, *radio receivers are not always on*—especially as portable and wearable receivers (typical for the medium radio) become available on the market. Another striking difference to the conditional access system used for cable and satellite TV is that *instead of one company controlling the content distribution*, we have *multiple independent content providers* often contributing to a shared multiplex (in the case of DAB). Finally, whereas satellite and cable TV receivers are stationary and can almost always be reached, for digital radio receivers we have to be able to support mobile receivers that might be out of reach of a digital radio network.

2 WorldDAB CA Requirements

Over the course of 2001 the DAB CA workgroup of WordDAB's DAB/Mobile Task Force has reached consensus on the following set of requirements for a DAB CA solution:[1]

Support DAB-only: Any DAB CA system shall support DAB-only receivers in addition to DAB-mobile receivers; that is, it has to support both an offline mode of operation and an online mode of operation.

Common CA base system for DAB-only and DAB-mobile receivers: DAB receiver manufacturers have a strong demand for a common CA base system; that is, the basic CA algorithms/mechanisms shall be the same for DAB-only and DAB-mobile receivers; the means of acquiring (and consequently the ease of use) content keys for CA-protected content might (and probably will) be different, with the DAB-mobile device offering a nicer user experience.

Minimal Hardware dependencies: Hardware dependencies should be as minimal as possible; if possible a software-only solution should be available; additional peripheral hardware should be avoided.

Support for streamed content: The CA system shall support streamed content (audio and data).

Support for datacasting: Granularity of the CA mechanism has to extend to individual objects/files; the content/service provider shall be able to protect different objects transported in the same service with different content keys.

Maximum flexibility for content/service provider: The CA system shall maximize the number of business models available to the content/service provider; at maximum a content/service provider shall be able to control the relevant parts of the CA chain (such as controlling encryption, key generation, customer relationship); delegation of part of the CA chain shall be possible; ease of service setup is necessary.

Crackability, hackability: The CA system should offer a reasonable amount of protection; the assumption is that a content/service provider will use a DAB CA system for "low value" content (i.e., content that is valuable for at most a week), but will use a proprietary digital rights management system for high value content (e.g., for distributing the latest version of the PC game Tomb Raider); any solution must be able to run on an open platform (e.g., Windows PC or an embedded Linux box).

Multiple subscriptions: The CA mechanism shall support switching between channels using different subscriptions/content keys; content keys/subscriptions can come from different content key/subscription providers (clearing houses).

Interoperability: The customer shall be able to acquire subscriptions/content keys from different providers within the same geographical region; the customer should be able to acquire subscriptions/content keys in different geographical regions ("DAB roaming").

Our DAB CA-PK solution fulfills all the above listed requirements; in particular, DAB CA-PK

[1] The following list of requirements is quoted from[12].

- relies on open cryptography algorithms (AES CBC/ECB, RSA);
- supports CA on the file or object level;
- supports CA for streaming data;
- is fully disclosed and open for peer review;
- is multi-operator CA capable; that is, each content provider can operate his own CA system (clearing house) if he desires to do so;
- is designed to work with limited bandwidth situations as typical for DAB channels;
- supports receivers that are mostly disconnected (i.e., no backchannel);
- does not require a smart card reader (but can make use of one if available, thereby providing a better user experience);
- can take advantage of DAB/Mobile receivers (integrated DAB/GSM or DAB/GPRS receivers) to provide a seamless DAB user experience;
- can be implemented in various degrees of complexity either as a business-to-business solution or as full-blown business-to-consumer solution.

Most prominently, through its design DAB CA-PK integrates the consumer into the protection chain instead of treating her as a threat.

In the next section we shall explain the DAB CA-PK mechanism in more detail.

3 DAB CA-PK Algorithm

Figure 1 on the following page illustrates the basic processes making up the DAB CA-PK system: the content provider, the clearing house, and the consumer's receiver.

The *content provider* encrypts the content to be transmitted via broadcast using either a stream or a block cipher and a randomly generated *content key*. In addition, it generates a randomly chosen *blocking nonce* for each content key. The sets of content keys and associated blocking nonces are then made available to a *clearing house.*

The *clearing house* handles the consumer orders for content keys for specific content. Each *receiver* contains a public–private key pair. When a consumer wants to acquire the content keys for a particular content she has to register with the clearing house the public key of her receiver together with payment details (e.g., her credit card number). The clearing house creates a new customer record containing the public key of the consumer's receiver and the credit card details of the consumer herself.

Once the consumer is registered with the clearing house she can generate *electronic orders* signed with the private key of the receiver and transmit them to the clearing house. After checking that the consumer can indeed pay for the ordered rights the clearing house encrypts the content keys using the public key of the receiver. To prevent the consumer from decrypting the content keys right away and then distribute these decrypted keys to a group of peers, the clearing house encrypts the encrypted content key a second time with the blocking nonce originally generated by the content provider—using either a stream or a block cipher. We call this double encrypted content key an *encapsulated content key*. In Figure 2 on page 5 we have illustrated the resulting double encrypted content key: it resembles an onion; the outer layer protecting the inner layer, the inner layer protecting the content key.

The consumer's receiver stores the acquired encapsulated content keys in its memory.

At broadcast time the content provider transmits the encrypted content and the blocking nonce via the broadcast network to the individual receivers.

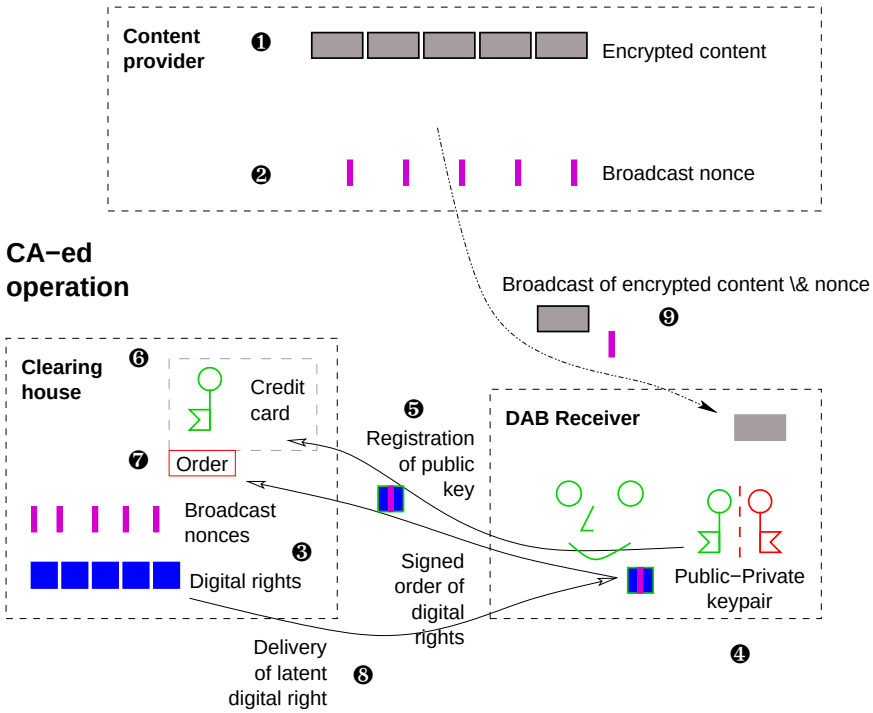

Figure 1: Overview of the DAB CA-PK system. Content is encrypted by the *content provider* using a stream or block cipher (❶). The content keys (including, if necessary for the operation of the cipher, the initialization vector) and a randomly chosen *blocking nonce* (❷) are transfered to a *clearing house* (❸). The consumer registers the public key of her *receiver* (❹) with the clearing house (❺ and ❻) and transmits an order for content keys for the desired content (❼). The clearing house encrypts the requested content keys with the public key of the consumer's receiver; before transmitting the encrypted content key to the consumer, though, the clearing house additionally encrypts the encrypted content key with the blocking nonce using a suitable stream or block cipher (❽). The consumer cannot decrypt the double encrypted content encryption key right away because she does not know the blocking nonce. The content provider eventually broadcasts the content along with the blocking nonce to the consumer's receiver (❾); using the received blocking nonce the receiver can then decrypt the double encrypted content key and, finally, the encrypted content.

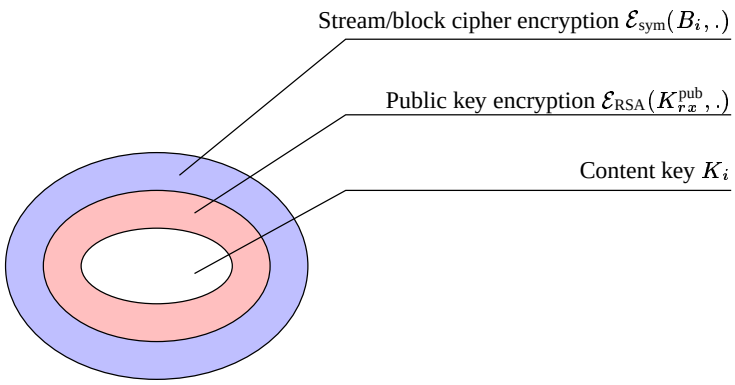

Figure 2: Encapsulated content key as delivered to the consumer's receiver. It resembles an onion where the outer layer (generated through the stream or block cipher encryption with the blocking nonce B_i) protects the inner layers. The outer but one layer (generated by public key encrypting with the receiver's public key K_{rx}^{pub}) in turn protects the content key K_i.

Once the receiver has received the blocking nonce, it can then remove the outer encryption shell from the encapsulated content key; use its private key to strip away the inner shell to arrive at the content key; and, finally, use the content key to decrypt the received content. Figure 3 on the following page illustrates the cryptographic operations and information flow in a more abstract manner.

Different methods of distributing the encapsulated content keys to the receivers are possible. The most obvious is for receiver with a backchannel to contact the clearing house via the Internet and directly carry out the above described transactions. We shall describe this and others methods in section 4 on page 8 and section 5 on page 10.

In the following sections we describe each step of our invention in further detail.

3.1 Content Provider

The content provider carries out the following tasks:

- Content generation or preparation
- Generation of the content keys
- Generation of the blocking nonces
- Generation of appropriate signaling elements
- Encryption of the content
- Broadcast of the content and signaling elements

Content in the context of our invention can be almost anything: data files, data objects, audio files, video files, streaming data, streaming audio, streaming video, IP packets, and so forth. The only assumption that we make is that we can subdivide the data into suitably sized blocks (which is not a real restriction).

$$\begin{aligned} & C_i, K_i, B_i \\ & \mathfrak{C}_i = \mathcal{E}_{\text{sym}}(C_i, K_i) \\ \Rightarrow \quad & \mathfrak{C}_i, K_i, B_i \end{aligned} \tag{1}$$

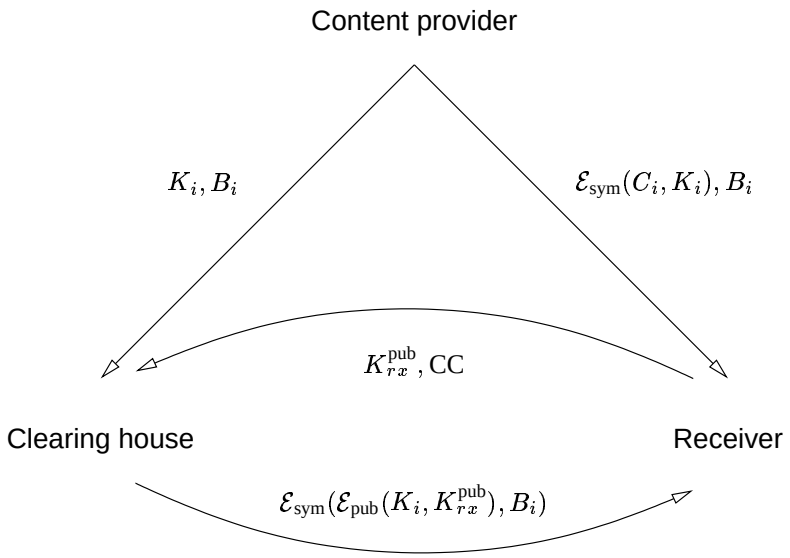

Figure 3: Cryptographic operations and information flow. The content provider sends the randomly chosen content key K_i and the randomly chosen blocking nonce B_i to the clearing house. The clearing house receives the public key K_{rx}^{pub} of the receiver and the credit card details CC of the consumer from the receiver. Using the public key K_{rx}^{pub} of the receiver it encrypts the content key K_i arriving at $\mathcal{E}_{\text{pub}}(K_i, K_{rx}^{\text{pub}})$; next it further encrypts the already encrypted content key using the blocking nonce B_i arriving at the encapsulated content key $\mathcal{E}_{\text{sym}}(\mathcal{E}_{\text{pub}}(K_i, K_{rx}^{\text{pub}}), B_i)$ which it returns to the receiver. At broadcast time the content provider transmits the encrypted content $\mathcal{E}_{\text{sym}}(C_i, K_i)$ and the associated blocking nonce B_i to the receiver. The receiver uses the blocking nonce to remove the outer encryption layer of the encapsulated key and then its private key K_{rx}^{priv} to obtain the content key; using the content key it then decrypt the received encrypted content.

For generating the content keys K_i and blocking nonces B_i we assume a generator that is producing random bit sequences sufficiently fast and of cryptographically sound quality. Both content key K_i and blocking nonce B_i are made available to the clearing house (see also Figure 3 on the page before).

For the encryption process $\mathcal{E}_{\text{sym}}(.,.)$ shown in Equation 1 on page 5 we can use stream or block ciphers—we propose to use AES in CBC mode [8, 7]. In case of block ciphers (and, hence, for AES) we might need to include extra padding content depending on the size of the content blocks and the block size used by the block cipher.

Part of the encryption process is the generation of appropriate signaling elements, entitlement checking messages (ECM) in case of DAB, for conveying the content key identifier (content key ID), the serial number of the key, and the blocking nonce B_i to the receiver. Also, if we had to apply padding we need to provide this information in the signaling element as well.

Finally, encrypted content $\mathfrak{C}_i$ and signaling elements are passed on to the lower layers of the transmission system.

3.2 Clearing House

The clearing house

- registers public keys and payment details of customers,
- accepts and verifies content key orders from customers,
- carries out the payment process,
- encrypts the requested content keys with the receiver's public key,
- encrypts the encrypted content keys with the blocking nonce,
- delivers the double encrypted content keys to the receivers or customer,
- keeps track of which keys the customer has paid for and received.

$$\begin{array}{rc} & K_i, B_i, K_{rx}^{\text{pub}} \\ \mathfrak{K}_{i\cdot rx} = & \mathcal{E}_{\text{sym}}(\mathcal{E}_{\text{pub}}(K_i, K_{rx}^{\text{pub}}), B_i) \\ \Rightarrow & \mathfrak{K}_{i\cdot rx} \end{array} \tag{2}$$

Orders for content keys K_i are cryptographically signed using public–private key cryptography algorithms. As equation 2 shows, for the encryption of the requested content keys K_i we use the registered public key K_{rx}^{pub} of the receiver—thus, only the receiver (being in possession of the matching private key) can decrypt the content key. To double encrypt the public key encrypted content key we can use any stream or block cipher and the blocking nonce B_i associated with the content key K_i. In case of a block cipher size of the encrypted content key should be divisible by the block size of the block cipher (otherwise padding has to be carried out and signaled, see previous section).

The encapsulated $\mathfrak{K}_{i\cdot rx}$ is returned to the costumer's receiver.

3.3 Consumer's Receiver

$$\mathfrak{K}_{i\cdot rx}, K_{rx}^{\text{priv}} \tag{3}$$

$$\mathfrak{C}_i, B_i \tag{4}$$

$$\Rightarrow \quad K_i = \mathcal{D}_{\text{pub}}(\mathcal{D}_{\text{sym}}(\mathfrak{K}_{i\cdot rx}, B_i), K_{rx}^{\text{priv}}) \tag{5}$$

$$\Rightarrow \quad C_i = \mathcal{D}(\mathfrak{C}_i, K_i) \tag{6}$$

The receiver finally

- acquires the encapsulated content keys $\mathfrak{K}_{i\cdot rx}$ from the clearing house (equation 3, see also previous section),
- receives the encrypted content $\mathfrak{C}_i$ and the associated signaling elements (equation 4),
- extracts the content key ID, blocking nonce B_i and content key serial number from the signaling elements,
- selects the appropriate double encrypted content key $\mathfrak{K}_{i\cdot rx}$ using the content key ID and serial number,
- decrypts the double encrypted content key and obtains the public key encrypted content key using the blocking nonce (equation 5),
- decrypts the public key encrypted content key using the receivers private key K_{rx}^{priv} (equation 5), and, finally,
- decrypts the encrypted content $\mathfrak{C}_i$ using the content key K_i (equation 6).

The ciphers used by the content provider and the clearing house determine the ciphers used by the receiver.

4 Eureka-147 Implementation

In this section we describe the implementation of our CA solution for the Eureka-147 Digital Audio Broadcasting (DAB) technology [4]. For the sake of discussion we shall concentrate on multimedia file transfer (MOT) [2]; note, however, that the implementation that we describe in the following paragraphs is by no means restricted to file transfer applications (though that is one of its main applications) but we can implement it for DAB audio, video, and IP streaming as well.

Aside from using Eureka-147 DAB as the broadcast technology, we shall assume for the time being that we have a receiver platform with a backchannel (*interaction channel* in Eureka-147 terms) such as a mobile Internet connection (e.g., a GPRS mobile phone component) for acquiring and storing content keys. We shall address the non-backchannel receiver platform later. For our proposed implementation we use for the block cipher of section 3 the AES block cipher in CBC and ECB mode. For the content encryption we use the AES algorithm in CBC mode with a key, initialization vector, and block size of 128 bits. Key, initialization vector, and blocking nonce are random bit sequences. The concatenation of AES key and initialization vector produces the 256 bit content key of our invention. The content provider generates a sufficiently large number of content keys and associated blocking nonces for a particular content set and stores them for later use in a database.

4.1 MOT Content Scrambling

With DAB MOT files are transmitted as an MOT header and an MOT body object. Both MOT header and MOT body objects can be segmented. Each segment (consisting in turn of segment header and segment body) is carried in an MSC data group[2] The MOT header object contains attributes describing the transmitted file as well as auxiliary, application specific, attributes (such as path names, file types, etc.); by interpreting the header object a DAB receiver can reassemble the transmitted file and, if it is part of a set of files (e.g., if part of an MOT directory or MOT Broadcast Web Site (BWS) application [3]), deposit it in the right place in the directory hierarchy of this set of files.

MOT data group header and data group body objects are linked through a transport identifier (transport ID); the transport ID changes with each new MOT object. In the unscrambled mode of operation the MOT header object is carried in data groups of type 3, the MOT body object is transported in type 4 data groups.

[2] Main Service Channel data group, see [4, section 5.3.3].

When we apply conditional access protection to an MOT file, the MOT header object stays unencrypted, only the MOT body object is encrypted (*scrambled* in DAB terminology) before being segmented to fit the required data group size.

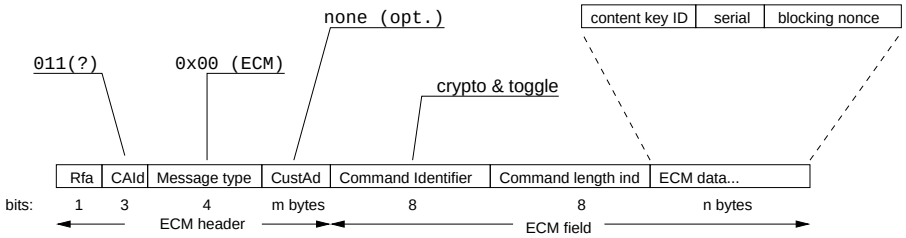

Figure 4: CA-PK ECM structure. CA-PK at the moment does not make use of the customer address (`CustAd`) field in the ECM header. The ECM data part carries the content key ID, the serial number of the content key, and the blocking nonce. `CAId` is provisionally set to binary `011` as CA-PK has not yet been registered with WorldDAB as an access control system.

To allow the receiver to decrypt an encrypted MOT body we need to include signaling elements in the broadcast data stream to specify

- the *conditional access system* used,
- the *cryptographic algorithms* involved, and
- the *content key identifier.*

These signaling elements are specified in the DAB standard as Entitlement Checking Messages (ECM) [4, section 9.3]. An ECM consists of an ECM header and an ECM data field. The ECM header carries the CA identifier (specifying the conditional access system used). Optionally it can contain an address field—the CA-PK system does not make use of that addressing field. The ECM field carries the type of crypto algorithm used and the length of the ECM data part. CA-PK transports the content key ID, the serial number of the content key, and the associated blocking nonce in the ECM data part (see Figure 4).

Encrypted MOT body segments are transmitted as data groups of type 5, and the associated ECM objects as type 1 data groups.

4.2 MOT Content Unscrambling

On the receiver side we have to collect all arriving data groups carrying MOT header segments, body segments, and ECMs with the same transport ID. The key object really is the ECM: it signals which CA system is used and its data group session header contains the transport ID. Obviously we can ignore any ECMs and their associated type 3 and type 5 data groups that have not been produced by CA-PK.

From the CA-PK ECMs the receiver system can extract the transport ID of the associated MOT header and body data groups as well as the content key ID and the blocking nonce. If the receiver system is in possession of the matching (double encrypted) content key, we then

- remove the outer shell and inner shell of the double encrypted content key by AES-ECB decrypting with the blocking nonce followed by an RSA decryption using the private key of the receiver;
- then we decrypt the received MOT body data groups and pass them on to the rest of the normal DAB receiver processing chain (together with the transport ID so that they can be matched up with the MOT header data groups).

4.3 Signaling Missing Content Keys

Curiously, the current specification of DAB does not provide a mechanism for signaling the CA operator [4, 13]; thus, if the DAB receiver does not have the required content keys to decrypt received content, there currently is no information transmitted to indicate where to acquire missing content keys from. We propose

- to use ECM objects with the customer address set to 0 to convey the address of the controlling CA clearing house;
- to define a format for initiating content key acquisition requests.

The address format depends on the communication medium used to initiate the content key acquisition; in the case of a Web based transaction process, the address would be a URL. Other address formats are possible; for example, a GSM SMS address, a phone number, and so forth.

For the current implementation we assume a Web based clearing house service: the address would then be the URL of the clearing house and the transaction would be initiated by sending an HTTP GET request ([5]) to the specified URL, using the content key ID and content key serial number from the ECM as request parameters. A DAB receiver with a backchannel can then directly initiate the acquisition process with the clearing house server. In case of a DAB receiver without a backchannel, the receiver would store the ECM details for later use; once the receiver is connected to a downloading device (e.g., a PC with Internet connection) it could then proceed as described.

5 Business-to-Business Application

In the preceding sections we have described the normal mode of operation for DAB CA, the business-to-consumer (B2C) case: A content provider broadcasts his CA protected content to a large number of (mobile) receivers.

A special case is the business-to-business (B2B) application of DAB CA. In the B2B case a content provider also distributes CA protected content, but his customers are business partners. The number of receivers is limited (typically between 100–10'000) and the receivers usually are controlled (if not owned) by the content provider. For the B2B we can simplify the operation of CA-PK: Both content encryption and content decryption follow the processes described in sections 4.1 and 4.2. As the number of receivers is quite small (compared to the B2C case) and as we know the public keys of the receivers in advance (they are under control of the content provider) we can distribute the content keys via DAB itself using a DAB data channel; for example, using the DAB Broadcast Website (BWS) application [3]. Each set of content keys for a specific receiver would be transported in a content key file of its own using for the file name the cryptographic hash of the receiver's public key (its *fingerprint*). Each receiver can then select its content key files from the BWS.

We should like to note that you can reduce the size (and, hence, the required bandwidth) of the content key file BWS by creating suitable receiver groups that share a public–private key pair; for example, a business customer who has offices or shops in different locations but has just one collective contract with the content provider could constitute a receiver group.

6 Security

The main purpose of any CA system is to provide a protected channel for the transmission of content. Access to the protected channel is typically controlled through cryptographic algorithms; only those receivers in possession of a valid content key are able to access the content. One of the most interesting questions that we can ask is, therefore, how secure against attacks any given CA system really is.

Different attacks against a CA system are possible. The most obvious purpose is to gain access to the content without recompensating the content provider—possibly with the intention of redistributing the content. Other attack forms are denial-of-service attacks intended to disrupt service delivery. For the time being we shall focus on the first kind of attacks: gaining acccess to the content without authorization from the content provider.

6.1 Unauthorized Access

Looking at CA-PK we immediately can identify the following points of attack:

- the content provider (content, content keys, blocking nonces),
- the clearing house (content keys, blocking nonces),
- the encrypted content,
- the acquired content key.

Let us take a look at each of these items in turn.

6.1.1 Content Provider and Clearing House

Obviously gaining access to the content provider's content storage system would be the most effective way of obtaining the content without renumeration. It is the responsibility of the content provider to keep the unencrypted content protected (both physically and if necessary also while in transit to the CA system).

Getting hold of either the content provider's or the clearing house's copy of the of the content keys is the next best and effective way of gaining unauthorized access—followed by obtaining the set of blocking nonces for certain content. Having access to the unencrypted content keys is as good as having access to the content itself. Even if the attacker only gets hold of the blocking nonce, that would be sufficient to essentially share a legally acquired key ahead of broadcast time. Again, both content provider and clearing house have to establish safeguards and checks to keep the content keys and blocking nonces protected from unauthorized access.

6.1.2 Encrypted content

The security of the encrypted content depends on two factors:

- the encryption cipher chosen and
- the protection of the content key.

We recommend using a proven and well-examined stream or block cipher. In the Eureka-147 case (which section 4 on page 8 describes) we selected the AES block cipher for both content encryption (AES CBC mode) and the outer layer encryption of the content key (AES ECB mode). The US *National Institute of Standards and Technology* has selected AES as the follow-on to DES [8, 7]. The selection process took place in an open and public manner—making AES a rather well-reseached and analyzed, and, thus, secure cipher.

The originally specified scrambling method for DAB CA [4, section 9] essentially consists of an XOR operation of a *control word* with the content. From a crypto-analysis point of view applying an XOR operation on some content means that one has to change the control world rapidly to maintain some sense of security. For AES on the other hand no crypt-analysis attacks are known, and AES is considered secure by the cryptography community. Except to prevent a content key from spreading (see next section) we would not have to change it for long periods of time.

6.1.3 Content Keys

To decrypt (unscramble) the broadcast content the costumer needs to be in possession of the content key. As we have described in section 3 (and in more detail in sections 3.2 and 3.3) the customer only gets to see the double encrypted content key. Before the attacker can access the content key itself, he has to decrypt the outer layer (see also Figure 2 again). Again, as we are using AES for the outer layer encryption, it is currently impossible for the attacker to remove that outer layer encryption without having access to the blocking nonce used.

Once the blocking nonce has been broadcast, however, the attacker could then redistribute the extracted content key to his peers. To foil this redistribution (or at least make it more difficult) we can increase the rate of changing the content key for any given content.

6.2 Digital Right Management System

CA-PK in its current incarnation implements a conditional access system only, following the WorldDAB CA requirements ([11, 12]); that is, CA-PK provides a protected "pay-to-access" channel from the content provider to the receiver. A Digital Rights Management system (DRM system) goes far beyond such a protected channel; it in addition provides the content provider control over

- how often certain content is (dis)played, and
- how content might be used (e.g., can the consumer make a copy? Can the consumer transfer the content to a different device?)

We can extend CA-PK to include DRM features. The prerequisites for a CA-PK based DRM system would include some kind of secure token, such as a smart card, as well as specially adapted content players to enforce the DRM conditions.

6.3 General Comments

We should like to point out that all CA and DRM systems can only provide a certain level of security. One of the primary objectives for a content provider is the distribution of content (the other objective is to generate revenue by doing the content distribution); that is, any content provider wants the content to be "consumed" at some point in time. Taking the current state of the anthropological evolution this objective means, that at some point in the distribution chain we have to have the content in an unscrambled form. For most any content it is fairly easy to recapture it in electronic form (with little if any loss of quality) and redistribute it unencrypted and freely accessible (see, for example, [6, 9]). The fact that not only "hacker" software exist for this purpose but also commercial software packages means that redistribution is not a process available only to the "hackers" but to the average consumer.

7 Intellectual Property Declaration

IBM has applied for a patent on the CA technology described in sections 3, 3.1, 3.2, 3.3, 4, 4.1, 4.2, and 5. IBM is generally willing to grant nonexclusive licenses under its patents, upon reasonable and non-discriminatory terms and conditions to those who respect IBM's intellectual property rights (see also IBM's Patent Licensing Web site [1]).

References

[1] IBM Corporation. IBM Intellectual Property & Licensing: Patent Licensing Practices. IBM Patent Licensing Web site http://www.ibm.com/ibm/licensing/license_info/index.html, November 1999. Cited 2002-01-10.

[2] European Telecommunications Standards Institute (ETSI). EN 301 234: Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol. ETSI Web site http://www.etsi.org/, February 1999. Cited 2002-01-08.

[3] European Telecommunications Standards Institute (ETSI). TS 101 498-1: Digital Audio Broadcasting (DAB); Broadcast website; Part 1: User application specification. ETSI Web site http://www.etsi.org/, August 2000. Cited 2002-01-08.

[4] European Telecommunications Standards Institute (ETSI). EN 300 401: Radio Broadcasting Systems: Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers. ETSI Web site http://www.etsi.org/, May 2001. Cited 2002-01-08.

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1. W3C Web site http://www.w3.org/Protocols/rfc2616/rfc2616.html, June 1999. Cited on 2002-01-11.

[6] NTONYX. Ntonyx: Virtual audio cable 2.0. NTONYX Web site http://www.ntonyx.com/vac.html, November 2001. Cited 2001-12-04.

[7] National Institute of Standards and Technology. ADVANCED ENCRYPTION STANDARD (AES). NIST Web site http://csrc.nist.gov/encryption/aes/, November 2001. Cited 2001-12-20.

[8] National Institute of Standards and Technology. AES Homepage. NIST Web site http://csrc.nist.gov/encryption/aes/, December 2001. Cited 2001-12-20.

[9] Slashdot. Rent Music Over the Net. Slashdot.org Web site http://slashdot.org/article.pl?sid=01/12/04/1335214&mode=thread, December 2001. Cited on 2001-12-04.

[10] WorldDAB Forum. The WorldDAB Forum Web Site. Web site http://www.worlddab.org/. Cited on 2001-12-18.

[11] WorldDAB Forum: DAB/CA Workgroup. Beating a path through the DAB CA Jungle: Terms and Requirements - September 2001. WorldDAB Web site http://www.worlddab.org/worlddabmember/docs/madocs/tcc-231.pdf, September 2001. Cited on 2001-12-18. Requires WorldDAB password.

[12] WorldDAB Forum: DAB/CA Workgroup. Call for Contribution (CfC) for a DAB Conditional Access Mechanism. Emailed to WorldDAB members on 2001-12-14, December 2001. Available on request from WorldDAB project office (dorta@worlddab.org).

[13] WorldDAB Forum: DAB/CA Workgroup. Minutes of the 2001-07-31 WorldDAB DAB/Mobile Conditional Access Workgroup Meeting. WorldDAB Web site http://www.worlddab.org/worlddabmember/docs/madocs/tcc-215.pdf, August 2001. Cited on 2002-01-11. Requires WorldDAB password.