[KRSB97] *T. Kunkelmann, R. Reinema, R. Steinmetz, T. Blecher;* **Evaluation of Different Video Encryption Methods for a Secure Multimedia Conferencing Gateway**; 4th COST-237 Workshop, Lisboa, Portugal, December 1997

# Evaluation of Different Video Encryption Methods for a Secure Multimedia Conferencing Gateway

Thomas Kunkelmann [1]  Rolf Reinema [2]  Ralf Steinmetz [1 2]  Thomas Blecher [1]

[1] Darmstadt University of Technology, 64283 Darmstadt, Germany
[2] German National Research Center for Information Technology, 64293 Darmstadt, Germany

**Abstract**

*In multimedia conferencing systems the need for confidentiality and privacy gains more and more in importance. In this paper we give an overview of the security requirements of multimedia conferencing systems and of applicable security functions. For real-time video transmissions there is a special need for selective encryption of the transmitted data. Existing methods are investigated and their strengths and weaknesses will be shown.*
*To combine different security functionalities we present the implementation of a scalable security gateway, adaptive to the requirements of specific applications and the properties of special forms of multimedia data.*

**Keywords:** Multimedia communication, Security, Encryption, Partial encryption

## 1 Introduction

Communication and cooperation in heterogeneous distributed environments are playing a rapidly increasing role in the business processes of today's enterprises. Nowadays several enterprises with distributed locations shift their personal communication and meetings more and more to so-called virtual meetings via computer links. In these cases confidential information has often to be passed securely over open networks like the Internet. Different advances were made to apply security techniques to these forms of communication [BHS94, HJRW96].

Multimedia conferencing supports synchronous communication and cooperation between two or more distributed participants. Characteristic for these systems is the combination of live media like real-time audio and video between different participants and the possibility of sharing documents and applications.

Another kind of distributed multimedia applications with a high demand for security mechanisms are video databases and *video-on-demand* (VoD) services. The security policy for these applications is not focused on optimal protection of highly confidential data, rather on protecting data against illegal access. Therefore the encryption methods needed here tend to be fast, with respect of the high data bandwidth of video streams, and to be cheap to implement in order to supply an emerging market of private users (Pay-TV, VoD). The expense to break into an encryption scheme needs not to be high, but it should be more expensive than the legal access to the video service.

In all these distributed multimedia applications, the cryptographic functionalities must cover different aspects of security, like confidentiality, integrity and authenticity. Therefore different modules of encryption mechanisms must be available to the application. Scalability for encryption methods can be achieved by partial encryption of multimedia data. An elegant way to combine several encryption modules is a *scalable security gateway*, providing different security functionalities, adaptive to the requirements of specific applications and the properties of special forms of multimedia data.

The rest of this paper is organized as follows: Section 2 points out the different qualities of security mechanisms needed for multimedia conferencing systems. In Section 3 we present different approaches of partial encryption methods for video streams, since these data need a special consideration due to their huge amount of data and the heavy CPU load they produce. Section 4 presents some methods for the partial encryption of video data streams, as suggested by others and ourselves. Those methods will be evaluated in Section 5. Based on the results found we have implemented a scalable secure multimedia conferencing gateway. Its implementation will be shown in Section 6. Section 7 concludes this paper, giving an outlook on our future work in this area.

## 2 Security Considerations for Multimedia Conferencing Systems

Nowadays multimedia conferencing systems are widely spread. Multimedia conferencing systems are supporting the synchronous (tele-)cooperation between two or more distributed individuals or groups. In multimedia conferencing, shared workspaces or shared editors are combined with live media like audio and video, which support real-time communication between several distributed sites.

Multimedia conferencing systems can be divided into two main groups:

- centralized or server-based systems, which operate mainly on ISDN and are mostly based on ITU standards (e.g., ITU-T H.320 compliant systems like *Intel ProShare*, *PictureTel*)
- decentralized or serverless systems, which operate mainly on the Internet and are mostly based on IETF standards (e.g., the MBone-Tools [Eri94], like *vic*, *vat*, *wb*)

At the moment, security plays only a secondary role in the development of multimedia conferencing systems. But the need for confidentiality and privacy gains more and more in importance. This applies especially for multimedia conferencing systems that operate on networks driven by third parties or open networks like the Internet.

### 2.1 Security Requirements

Like any other distributed multimedia system, multimedia conferencing has the following security requirements:

- *access control* to prevent unauthorized access to a conference
- *authentication* to confirm the identities of the communicating partners
- *data confidentiality* to protect data against bugging and to provide traffic flow confidentiality
- *data integrity* to protect data against loss and manipulation
- *non-repudiation* to provide proof of origin and delivery of data

The basic building blocks meeting those requirements are *encryption*, *authentication*, *certification* and *integrity preservation*. Relevant methods to fulfill these security requirements are as follows:

- Secret Key Encryption (most commonly used methods: *DES* and *IDEA*, both operating on blocks of 64 bits length)
- Public Key Encryption (e.g. *RSA* and *Diffie-Hellman*)
- Hybrid Encryption (combination of the above two methods)

- Consistency Checking
- Digital Signatures
- Copyright Information (e.g. by digital watermarks)

A general survey of encryption methods is given in [Sch96]. The main focus of this paper considers encryption methods, since their application to multimedia data streams will cause time-critical problems when encrypting the whole data stream. Besides integrity checks, the other security functionalities do not result in any time-critical operations.

### 2.2 Integration of Security Functionalities in the System

Security functionalities can be built up on two different layers:

- Security in the transmission or networking layer, i.e., security is already provided by the networking protocol used (e.g., SSL, RTP [SCFV96]). An additional data manipulation by security functions is not necessary. For secure communication over ATM networks an approach with a cryptographic protocol unit is presented in [SHB95], able to cope with different encryption keys.
- Security in the data layer, i.e., before data is transmitted from a sender to a receiver it will be manipulated by the appropriate security functions in the application. The security functionality can either be applied to the application as in [BHS94] for a remote conferencing tool, or the application itself is designed to gain security for other programs.

One of the drawbacks of network layer security mechanisms is the need for secure underlying transport protocols, which are not available at the moment. IPnG and ATM will provide these functionalities in the near future. The advantage of data layer security is that the transmitted data can be subdivided into parts with sensitive and insensitive data with respect to the human perception.

In comparison to providing security on the data layer, all transmitted data are protected in the network layer. This tends to problems in transmitting huge amounts of data, which is typical for multimedia conferencing. The network layer is not capable of subdividing the data stream in parts with a higher need for protection and parts with a lower or no need for protection. Implementing security functions in the data layer has the advantage that only some parts of the data need to be protected and so the amount of time spent on protecting them can be extensively reduced.

## 3 Multimedia Data Considerations

Several multimedia data formats require a special treatment in terms of encryption. In particular, these are data formats with real-time properties, like audio and video communication. Here encryption methods cannot be applied straightforward due to the severe time constraints for data processing and the complexity of secure encryption standards. Either encryption must be realized with special hardware, which is not available on many platforms, or the data streams have to be subdivided in order to separate data portions relevant to the human perception for encryption. The latter case is known as *partial encryption* schemes. In this section we investigate this topic in order to examine practicable solutions for scalable encryption mechanisms to be used in our security gateway.

## 3.1 Data Formats for Video Transmission

For the partial encryption of multimedia data it is important to see how video data is organized in the data stream, in order to develop applicable methods for extracting the relevant data portions. So we first give a short survey over the common data formats used in toady's video conferencing systems. A more general survey can be found in [Ste94] and [StNa95].

### Motion-JPEG

The *Motion-JPEG* (M-JPEG) video format is not standardized, it consists of a sequence of single video images (frames) encoded with the *JPEG* format. JPEG (Joint Pictures Expert Group) is a format to encode still images with continuous colors or greyscale values, like natural objects. The JPEG format became an ISO International Standard [JPEG93]. The JPEG image encoding technique leads to a high compression ratio for continuous-toned images. It is based on a combination of applying the *Discrete Cosine Transformation* (DCT) to blocks of 8x8 image pixels, followed by an entropy coding (Huffman and run-length encoding) of the resulting coefficients [Ste94]. The M-JPEG video format is used mainly for video conferencing tools due to a symmetrical expense for encoding and decoding, which is important for real-time applications.

### MPEG-1 and MPEG-2

The *MPEG* format for coding and transmitting video signals along with the corresponding audio information has been standardized by the ISO [MPEG93]. For MPEG there are three different standards specified, MPEG-1, MPEG-2 and MPEG-4 (standard scheduled for November 1998). MPEG-1 is today's commonly used video compression standard due to its availability for many platforms and appropriate hardware support. It covers data rates of about 1.2 to 1.85 Mbit/s. MPEG-2 is an enhancement over the possibilities of MPEG-1 by coping high-definition TV and multiple audio channels. Nowadays the first hardware and software solutions for MPEG-2 encoders emerge.

An MPEG data stream is formed of different layers, responsible for the synchronization of audio and video, and providing pre-defined starting points for re-synchronization. MPEG utilizes the compression techniques of JPEG, along with inter-frame relationships (*prediction* and *motion compensation*). The enhancements of MPEG-2 are an extended parameter set for image resolution, pixel aspect ratio and motion compensation techniques. The most remarkable extension is the scalability of video image resolution, leading to an excellent fit into the priority concept of ATM cell transmission, thus making the compression standard interesting for video transmission over ATM links.

### H.261 and H.263

H.261 and H.263 are widespread standards adopted by the ITU [ITU96] for transmitting video data streams. The intention of H.261 is to provide video information at a data rate of px64 Kbit/s (with $p \in \{1, \ldots 30\}$), matching the ISDN specification. Therefore H.261 is toady's mostly used video compression standard for ISDN video conferencing systems (also known as ITU-T H.320 compliant systems) like *Intel ProShare* or *PictureTel*.

The codec (en*cod*ing and *dec*oding functionality) is designed for a symmetrical encoding and decoding process with a maximum end-to-end delay of 150 ms.

The H.261 standard also specifies many format parameters. The resolutions supported by H.261 are CIF (*Common Interface Format*, 352x288 pixels) and QCIF (1/4 CIF). The frame rate is defined as 29.97 fps. The encoding schemes for H.261 are similar to those used in MPEG, *intraframe* and *interframe* blocks combined with a motion vector perform the basic units of an H.261 image.

H.263 enlarges H.261 by providing more resolution formats, better prediction methods for motion compensation, better error correction schemes and higher compression ratios than H.261. H.263 is not restricted to data rates of px64 Kbit/s anymore. Because of these enhancements it will probably replace H.261 in the near future.

### 3.2 Performance Aspects for Encrypted Video

As pointed out in [BGU95], modern high-performance workstations and servers are capable of playing MPEG-1 or M-JPEG video, leaving about 20 to 60 percent CPU time for other jobs when using hardware JPEG support. On most desktop workstations such a computing power is not available. Here the frame rate or the pixel resolution has to be reduced (e.g. from CIF to QCIF format) to meet the limited CPU capacity. Performance measurements on a PC (100 MHz Pentium, Linux) showed that such a system can playback about three H.261 QCIF video streams with frame rates sufficient for video conferencing (between 11 and 12 fps).

Table 1 shows the performance evaluations of several hardware platforms decrypting video streams in software, with standard library implementations of the DES algorithm. The reason for investigating DES is the fact that cryptanalysists consider it to be a safe algorithm for ciphertext-only and known-plaintext attacks [Sch96], except for the key length of 56 bits, which opens a door for brute-force attacks with massive hardware power. The IDEA algorithm [Lai92] is an alternative which provides a far better resistance against such attacks by using a key length of 128 bit, its implementations are slightly faster as DES [Sch96]. For modern encryption methods, e.g., FEAL and Blowfish, this safety cannot be guaranteed due to the short time period in which cryptanalysists could gain experiences with them. They may become a faster alternative if time will show that there exist no known attacks against them.

| *DES CPU usage* | *1.5Mbit MPEG* | *2Mbit M-JPEG* | *3x128 Kbit H.261* |
|---|---|---|---|
| Intel Pentium-100, Linux | 86.70 % | ★115.62 % | 21.67 % |
| DEC Alpha 1000/ 266 | 65.63 % | 87.50 % | 16.41 % |
| Sparc 20 (Solaris) | 76.01 % | ★101.34 % | 19.00 % |
| Sparc 4c (SunOs) | ★312.77 % | ★417.03 % | 78.19 % |

**Table 1:** CPU utilization of different hardware systems for DES software encryption. The MPEG and M-JPEG cases represent e.g. Pay-TV scenarios (16 - 25 fps), while the H.261 scenario describes an ISDN video conference with three video channels open (12 - 15 fps).

For the MPEG and M-JPEG scenarios we examined, the need for reducing the encryption effort is obvious, the slower workstations are already overloaded with the DES decryption (★ = projected values). For the H.261 scenario, an encryption CPU usage of 20 percent implies a frame reduction from e.g. 11 to 8.8, violating the lower bounds for human image perception. Therefore partial encryption is a suitable solution also for this case, as well as the usage of one of the modern, faster encryption algorithms, with the drawback of no guarantees for the security of these algorithms and maybe no availability of the decryption algorithm at the receiver's site.

## 4 Partial Video Cryption Methods

Considering the results from performance measures in secure video systems, several methods for partial encryption of video data have been proposed in the last few years, which are summarized in this section.

### 4.1 SEC-MPEG

SEC-MPEG [MeGa95] is a toolkit for partial encryption of MPEG-1 data. The development of SEC-MPEG is based on the Berkeley-MPEG player [PSR93]. The aim of this toolkit is to achieve confidentiality and integrity checks. Confidentiality is achieved by using the DES algorithm, integrity checks are carried out by a *cyclic-redundancy check* (CRC) due to performance issues, at the expense of a weak integrity certification. The toolkit supports four levels of confidentiality (*C-levels*) and three levels of integrity (*I-levels*), beginning with encrypting the header information, up to an encoding of the whole MPEG stream. In C-level 2 a subset of DCT blocks is selected, which will be partially encrypted, while C-level 3 encrypts all intracoded image information.
The drawback of SEC-MPEG is that it produces a data stream with a proprietary format, so it is not compatible with conventional MPEG players.

### 4.2 Partial encryption of intracoded frames

Some work has been done in partially encrypting only the intracoded frames (I-Frames) of an MPEG stream [MaSp95] or the intracoded blocks in intercoded frames, as in C-level 3 of SEC-MPEG. In [AgGo96] an example of this kind of encryption is given, the authors also show the limits of this technique. Video sequences with a high degree of motion still show a lot of details of the original scene. As a remedy the increase of intracoded-only frames is suggested, but this will also vastly increase the size of video data. In the case of video transmissions over a channel with limited bandwidth this can be no solution.

### 4.3 Encryption of DCT block information

A method for an encoding/ decoding process with no significant delay resulting from additional encryption is applicable to video compression techniques based on the JPEG algorithm. In [Tan96] this method is described for the MPEG standard. It is based on the zigzag ordering of the DCT coefficients before entropy coding is applied. This order is randomly permuted, the secret key of the encryption is the permutation itself. Due to a table lookup for computing the zigzag coefficients in usual video decoders this permutation generates no temporal overhead. The drawback of this method is the worse performance of the run length encoding of the DCT coefficients, which results in an expansion of the encoded video data of about 20% to 40% for the tested video sequences.

### 4.4 Reducing the amount for strong encryption

Statistical analysis of MPEG streams show that it is still sufficient to reduce the effort for encryption to one half of the video stream, and use these data as a *one-time pad* for the other half of the stream, in order to obtain a strong cryptographic protection for the whole MPEG data [QiNa97]. The proposed algorithm parses the MPEG stream down to the slice layer, but does not touch the macroblock information. So it is suited to operate in a separate encryption module, independent of the encoding or decoding process. The method needs about 53% of the effort for encrypting the whole data stream, its drawbacks are the usage of multiple encryption keys and the overwriting of some MPEG header fields, which makes the solution infeasible for most existing applications.

### 4.5 Scalable method for JPEG-based video

In [KuRe97] we present a scalable partial encryption method, which allows a security level of nearly every granularity. It can be applied to all video compression methods based on the JPEG standard, in particular the formats mentioned above. Our method takes advantage of decreasing importance for the image composition of the DCT coefficients, so it is sufficient to encrypt only the first few of them. The algorithm starts with encrypting a data block at the beginning of a DCT block and guarantees the protection of at least the first *n* DCT coefficients of a block, encrypting consecutive data portions in the video stream of the encryption method's block size. The parameter *n* of encrypted coefficients provides scalability for the security level. Figure 1 gives an example (with *n*=3), which parts of an MPEG streams will be encrypted.

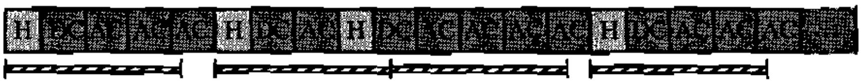

**Figure 1:** Encrypted parts of a video stream with our partial encryption method

### 4.6 Combination of Bitstream- and VLC Encryption

We performed some experiments on the partial encryption of "confidential" video material to find practicable solutions with low encryption effort, but a high level of confidence for the protected video stream. The experiments were made with a modified version of the Berkeley MPEG encoder and player [PSR93]. We have investigated three approaches for encryption:

- Encryption of the *DCT coefficients* before they are encoded with the Huffman tables of MPEG or H.263.
- Partial encryption of the *Variable Length Codes* (VLC), which occur after applying the Huffman encoding.
- Selective encryption of the MPEG or H.263 data stream, depending on the index of the coefficients represented by the VLC.

The first approach is only useful for security analysis and the computation of optimal parameters for the other approaches. Due to a pseudo-random bit distribution after encryption, the DCT coefficients cannot be compressed effectively, resulting in video sequences which occupy the same disk space as the raw digitized material.

Our experiments have shown that a combination of VLC encryption and bit stream encryption is most useful for partial encryption schemes, when keeping a high level of confidentiality is the primary goal.

### 4.7 DVB - Conditional Access

*Conditional Access* (CA) is a method for video encryption used in the *Digital Video Broadcasting* (DVB) project [DVB96]. It provides a *Common Scrambling Interface*, which is supported by every DVB program vendor. It is a combination of a block cipher and a stream cipher and is fed with two control words, which are transmitted in the video control stream. The individual access to specific programs of a video transmission (pay-per-view) is regulated by *Entitlement Management Messages* (EMMs), which are only valid in combination with a unique receiver ID number.

The system supports the encryption of a whole MPEG-2 transport stream, or the encryption of specific packeted elementary streams, e.g. only a video or an audio stream. The encryption unit can switch between two keys, so a key exchange is possible during a video transmission.

## 5 Evaluation of Results

We first present some aspects on the safety of partial encryption methods for video data. Based on these considerations, we compare the different methods with respect to safety, time consumption and communication overhead.

### 5.1 Possible Reconstruction of Protected Data

With methods used in cryptanalysis, e.g., statistical and entropy evaluations, it may always be possible to detect those portions of a data stream which have been encrypted. However, this will be a difficult job for partially encrypted (MPEG or similar encoded) video streams due to the nearly redundancy-free Huffman encoding. An eavesdropper who succeeded in analyzing a partially encrypted video stream might probably reconstruct a video frame as in the examples of Figure 2. Here the non-reconstructible protected information is set to zero, otherwise the random encrypted information would still obscure the reasonable information.

These examples motivate to protect truly confidential video information with an adequate method, e.g. the scalable approach presented in [KuRe97]. In other scenarios, where encryption is merely used to aggravate the access for the public, e.g., video-on-demand systems, the expense for reconstructing parts of a video is out of all proportion to the fee for joining the movie broadcast legally. In these scenarios a simple encryption method might be considered as sufficient.

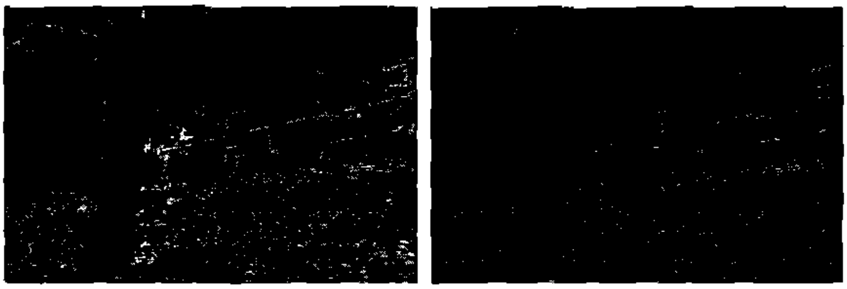

**Figure 2:** Maximal possible reconstruction for intracoded block encryption (left) and with the method of [KuRe97] (right), both frames with about 46% encrypted data (video *flowers*, 1/2 original size).

### 5.2 Experimental Results

Our experiments are based on a series of different video sequences, which reflect several scenarios where digital video can be used. Movies for video-on-demand (VoD) applications are represented by the test sequences "Flowers" and "Biker" (action movie), several sporting scenes ("Soccer" and "Skating") with different ratio of movement, and video conference scenes ("Conference", portrait of a male speaker, and "Talk", two talking persons with the fade-in of a telephone number) are used for testing.

In VoD scenarios the encryption effort need not to be high, even with a few percent of encrypted data the quality of the video material becomes intolerably poor. In Figure 3 we present an example for an encrypted video image with about 25% of the data encrypted. We consider about 10 percent encryption as a satisfactory level for VoD applications, which complies with the fact that here the software and hardware effort must be minimized to keep the costs per set-top unit cheap.

For truly confidential video sequences (e.g. the phone number in the "Talk" sequence) it is not sufficient at all to pick some few video blocks or DCT coefficients for encryption, as it is done in most partial encryption schemes. Here the combination of bit stream and VLC encryption seems a good approach, although the security of the stream cipher method in our VLC encryption is considered not to be as secure as, e.g., DES. A replacement for this algorithm with a bloek cipher would provide a security benefit. When using our scalable approach it is necessary to protect at least the first 10 to 12 DCT coefficients in order to keep a high level of confidence. This results in an encryption rate of 40% or more of the video data. To protect numbers or letters in the video image from being read by an eavesdropper, a ratio of 50% encrypted data is necessary.

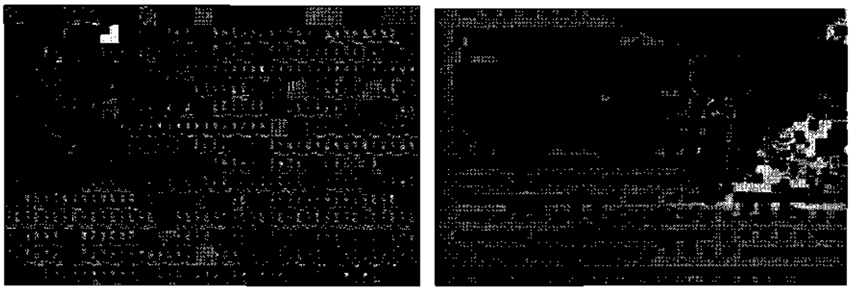

**Figure 3:** Video sequence *biker* with 25% encrypted data, playback (left) and maximal possible reconstruction (right)

### 5.3 Comparison of the Encryption Methods

In Table 2 we compare the different partial encryption methods with respect to security, scalability, time effort, protocol signaling overhead and feasibility for the usage in a security gateway for video conference applications.

| Method | Security | Scalability | Time overhead | Protocol overhead | Isolation |
|---|---|---|---|---|---|
| *SEC-MPEG* | high | 3 levels | DES encryption | about 17 to 32%(own data format) | possible |
| *Frame-type encryption* | high | I: 25-40%<br>IP:70-85%<br>IPB: 99% | DES encryption | none | possible (low overhead) |
| *Intra-block encryption* | high | no | DES encryption | none | possible |
| *DCT permutation* | breakable | no | none | none, data volume + 20% to 40% | not possible |
| *Scalable method* | high | full, from 8% to 100% | DES encryption | 3-5% | possible (10.5%) |
| *VLC/ bit stream combination* | high (medium for VLC bits) | full | DES encryption + VLC re-order | 3-5% | possible |
| *DVB - Conditional Access* | n.a. (details are secret) | no | Hardware: none, SW: low (XOR) | yes (control words, EMM, etc.) | default |

**Table 2:** Comparison of different partial encryption methods

One important aspect for the implementation of a partial encryption method in a security gateway, independent form the video coding unit, is the possibility of separating the data portions relevant for encryption in an efficient fashion (*Isolation* in Table 2). For a method like e.g. [Tan96], this implies a complete decoding and re-encoding of the whole video stream, which is not feasible. For the scalable method of [KuRe97] the gateway process must identify the start of a DCT block in the video sequence. In [PSR93] the parsing time is estimated with 17.4% of the whole play-back time, [MeGa95] achieved values of about 30%. Our experiments showed, however, that this time can easily been reduced to a value of 10.5% of the total playback time for the complete separation of encryption data. Re-implementing the parsing algorithm with a finite state machine might furthermore reduce this time effort. So this partial encryption algorithm is well suited for an operation in a separate security gateway.

An integrated solution of the decoding process and the security gateway needs only about 1.01% time overhead for the separation of encrypted data, again compared to the video playback time.

Another important factor is the signaling or control data overhead an encryption scheme generates. These data can be embedded in the video stream as it is done in SEC-MPEG with a special encryption header flag, or it can be transmitted via a separate control channel as in our security gateway approach. Using a special encryption header flag implies a length of at least 32 bit, resulting in a protocol overhead of 17% (intercoded) to 32% (intracoded) additional bandwidth. In our solution the signaling data is encoded with fewer bits, so the total overhead is about 4.1% in average, at a level of ten coefficients protected.

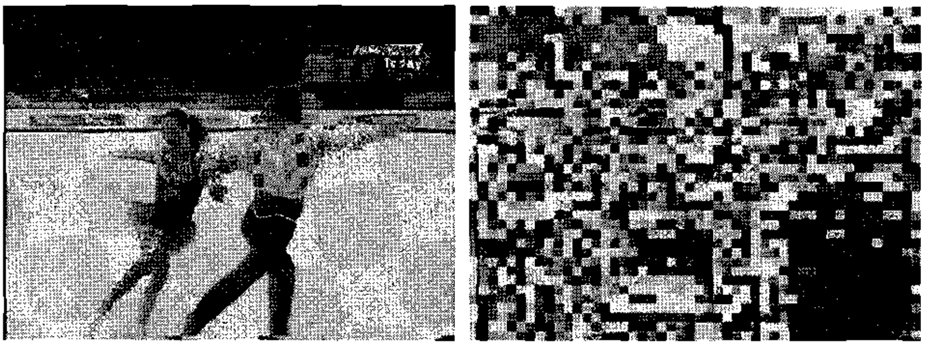

**Figure 4:** Video clip "skating" with encryption in ECB mode. The uniform areas of the original image are also visible in the encrypted frame

The impact of the encryption method used can be seen in Figure 4, where DES in standard ECB mode (Electronic Codebook, the same input pattern generates identical encryption output) was chosen. Large uniform areas in the video image result in identical block encoding and can be identified in the encrypted output data. Using CBC mode (Cipher Block Chaining, the output data is fed back to the next encryption block) avoids this security leak, but this method cannot be used with unreliable network connections where single bits or bytes (but not the whole encrypted block) can get lost, after a packet loss the rest of the encrypted data cannot be decrypted.

Figure 5 shows the difference between VLC encryption and encryption of only some DCT values. Fine-scaled objects like the text across the video image are expressed by higher DCT coefficients and thus can be reconstructed from a partially encrypted video. So we suggest also to protect the VLC coefficients to overcome this security leak.

Table 3 shows that there is always a gain in the frame rate when using partial encryption in software solutions. The performance measures were calculated on an UltraSparc station with Solaris. For common desktop PCs with no multimedia support the performance gain will be comparable. In the case of inter-coded frames, the encryption rate was 35% (VoD example), for the intra-coded (Motion-JPEG, videoconference scenario) frame example, the encryption rate was assessed with 50%. The experiments show that the performance gain is at least two frames/ second, with an effective frame rate of about 12 to 15 fps this results in a visible profit.

| *Video clip* | *Mode* | *No enc.* | *Full enc.* | *Partial enc.* | *Encryption rate* |
|---|---|---|---|---|---|
| Skating | intra | 20.82 | 15.14 | 17.92 | 50% |
| Talk | intra | 17.86 | 12.70 | 15.28 | 50% |
| Soccer | intra | 16.94 | 10.88 | 14.02 | 50% |
| Skating | inter | 25.50 | 22.52 | 24.32 | 35% |

**Table 3:** Frame rates(fps) with full and partial encryption

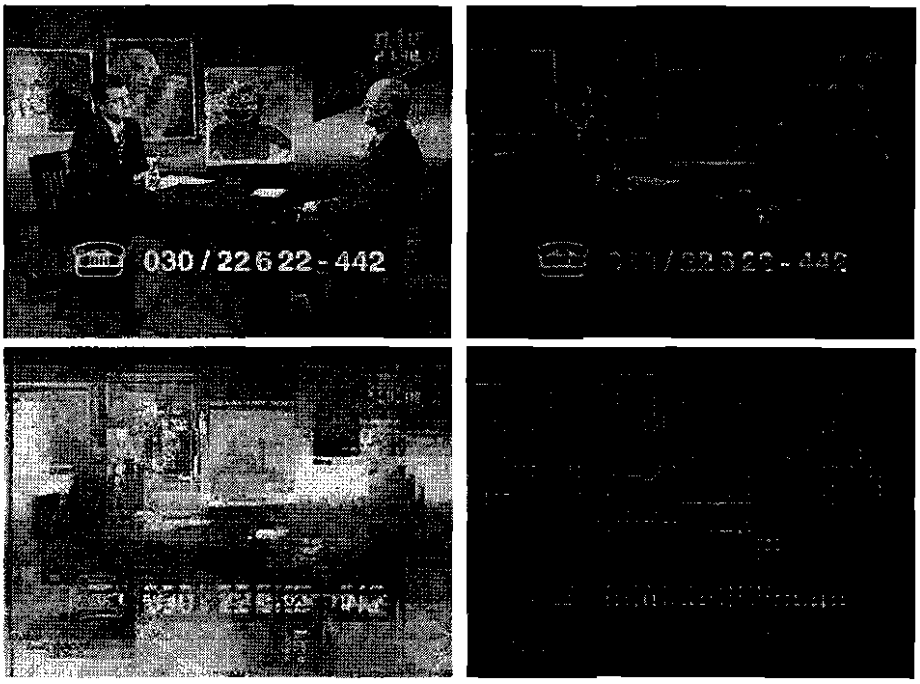

**Figure 5:** Video clip "talk" original (top left), 1 DCT coefficient protected (right), 1 VLC bit encrypted (bottom left), combination of both (right)

## 6 Implementation of a Scalable Secure Conferencing Gateway

For the secure end-to-end transmission of multimedia conferencing data streams over open internetworks between two or more distributed sites an application independent *secure conferencing gateway* has been implemented. It provides the following advantages:

- security can be achieved even if the used conferencing applications do not support it
- conferencing systems can be implemented independently of the used encryption methods

Currently the secure conferencing gateway is only restricted to the secure transmission of video data streams using the above described methods. It provides various ports for the different applicable video compression modes: M-JPEG, MPEG-1/ MPEG-2 and H.261/ H.263.

The gateway consists of two parts, an encryptor and a decryptor using our scalable encryption method. Between two or more such gateways there are two channels established, a data channel that carries the encrypted video data and a control channel. The control channel is used for authentication, exchange of session keys, the security level and synchronization during the session (e.g., changing the session key or the security level during a transmission). This architecture is characterized by its high degree of modularity and scalability to reflect different application scenarios and different levels of security needed in various conferencing situations.

Unlike in [BHS94] where all session keys are transmitted to the participants in advance by secure e-mail, the session keys were chosen by random during the conference. The initial session key is chosen by the first site entering a conference. When the next site enters the conference, it recognizes that there is already another participant and requests the current session key from it. The session key is then encrypted using the public key of the requester. The initial security level is set to a default value, which is suitable for typical conferencing situations.

The session key and the security level can be changed by any site at any time, which requires synchronization through the control channel. First, the change of the session key or security level will be announced. After the acknowledgment of all sites, changes take place at the pre-scheduled time.

The implementation of the secure conferencing gateway is not just restricted to point-to-point communications between two distributed participants. It can also be used for point-to-multipoint transmissions between two or more distributed sites. To achieve this goal we make use of IP-Multicast [Eri94].

In some application scenarios like seminars and discussions people often enter and leave a conference at their convenience. This is reflected in the fact that there is no master or server gateway that would supervise the change of session keys, security levels and ensure the consistency between all involved gateways. The change of the master role would generate more communication overhead than the distributed negotiation for changing the session keys and security levels using IP-Multicast.

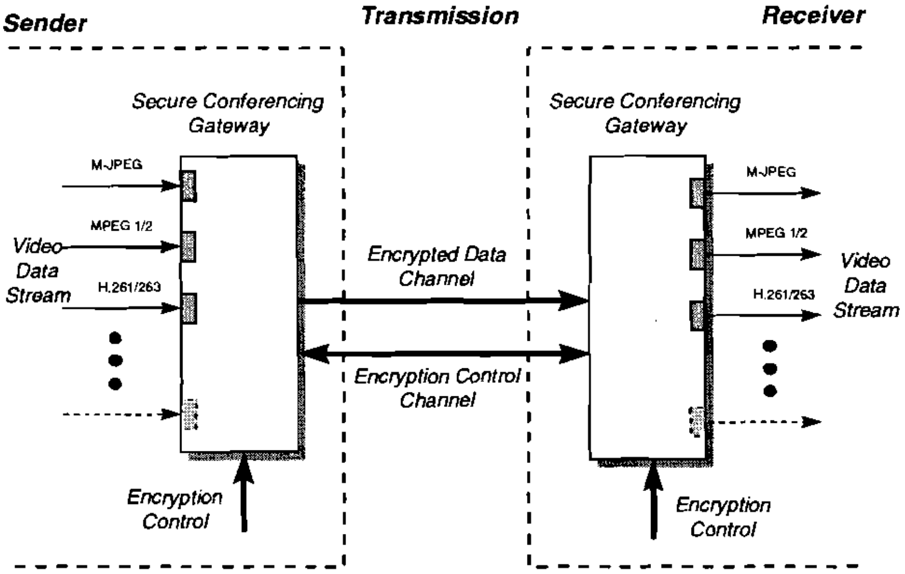

**Figure 6:** Secure Conferencing Gateway

The implementation of the partial encryption algorithm is based on the Berkeley MPEG encoder/ decoder [PSR93]. To provide encryption, decryption and certification of public keys SecuDE is being used [GMD97]. SecuDE is a toolkit (for UNIX and MS-DOS/MS-Windows platforms) which offers a library of various security functions. It provides basic cryptographic functions (like RSA or DES), digital signatures, X.509 key certification, operation of certification authorities, secure access to public X.500 directories for the storage and retrieval of certificates, cross-certificates and revocation lists. SecuDE provides a so-called Personal Security Environment (PSE) which contains the user's private and public key pair.

## 7 Summary and Future Work

In this paper we pointed out the different security requirements needed for multimedia conferencing systems. A special treatment has to be applied for live media, especially for real-time video data due to the large amount of data to be protected. Partial encryption is a solution to solve this problem.

MPEG-1/MPEG-2 and H.261/H.263 are widespread compression standards used in most of today's video conferencing applications. They are well suited for partial encryption because on the one hand they make use of DCT, which has a high potential for dividing data in more relevant less relevant parts (entropy of the coefficients). On the other hand, large amounts of video data are encoded by reference to preceding or following blocks (intracoded blocks), from this it follows that only the referenced blocks have to be protected. Motion-JPEG, which also uses DCT, is not well suited for all partial encryption schemes. In contrast to MPEG-1/MPEG-2 and H.261/H.263 it does not make use of referenced frames or blocks and so the redundancy of Motion-JPEG data streams is much higher.

The newly emerging *MPEG-4* standard has to be further examined with respect to encryption. MPEG-4, which is still in the specification phase, provides specific solutions for different types of objects during the encoding process because of this it might offer totally new possibilities for partial encryption methods.

The implementation of our scalable secure conferencing gateway is based on the performed evaluation. One of the next steps in the gateway implementation is the integration of security functions for other media beyond video data streams.

With the specification of the new *Security Service* as a *Common Object Service* for CORBA (Common Object Request Broker Architecture) by OMG (Object Management Group) end of last year [OMG96], the integration of our security gateway in a CORBA security service will be investigated. Along with the integration in a middleware platform for heterogeneous environments, a new service called *transcoding* [AMZ95] can additionally be offered. Transcoding means the translation between different encoding formats, transmission protocols and bit rate adaptation. The specification in CORBA IDL will make it much easier for application programmers to use these services particularly in open distributed environments, since the service interfaces are then accessible in a standardized manner.

## Literature

[AgGo96] I. Agi, L. Gong: *An Empirical Study of Secure MPEG Video Transmissions.* ISOC Symposium on Network and Distributed System Security, San Diego, CA, 1996

[AMZ95] E. Amir, S. McCanne, H. Zhang: *An Application Level Video Gateway.* Proc. 3rd ACM Multimedia, San Francisco, CA, 1995

[BGU95] P. Bahl, P.S. Gauthier, R.A. Ulichney: *Software-only Compression, Rendering, and Playback of Digital Video.* Digital Technical Journal Vol. 7(4), 1995

[BHS94] K. Bahr, E. Hinsch, G. Schulze: *Incorporating Security Functions in Multimedia Conferencing Applications in the Context of the MICE Project.* Proc. 2nd Int'l Workshop IWACA'94, Heidelberg, Germany, 1994

[DVB96] *The Digital Video Broadcasting Project.* http://www.ebu.ch/dvb_homc.html, 1996

[Eri94] H. Eriksson: *The Multicast Backbone.* Comm. of the ACM, 37(8), pp. 54-60, 1994

[GMD97] W. Schneider (Ed.): *SecuDE Documentation.* ftp://ftp.darmstadt.gmd.de/secude/, Darmstadt, Germany, 1997

[HJRW96] E. Hinsch, A. Jaegemann, I. Roper, L. Wang: *The Secure Conferencing User Agent - A Tool to Provide Secure Conferencing with MBONE Multimedia Conferencing Applications.* Proc. IDMS '96, Berlin, Germany, 1996

[ITU96] ITU-T Recommendation H.263: *Video coding for low bit rate communication.* 1996

[JPEG93] ISO/ IEC International Standard 10918: *Digital Compression and Coding of Continuous-Tone Still Images.* 1993

[KuRe97] T. Kunkelmann, R. Reinema: *A Scalable Security Architecture for Multimedia Communication Standards.* Proc. 4th IEEE Int'l Conference on Multimedia Computing and Systems, Ottawa, Canada, 1997

[Lai92] X. Lai: *On the Design and Security of Block Ciphers.* ETH Series in Information Processing, 1, H. Gorre Verlag, Konstanz, 1992

[MaSp95] T.B. Maples, G.A. Spanos: *Performance Study of a Selective Encryption Scheme for the Security of Networked Real-time Video.* Proc. 4th Int'l Conference on Computer and Communications, Las Vegas, NV, 1995

[MeGa95] J. Meyer, F. Gadegast: *Security Mechanisms for Multimedia Data with the Example MPEG-1 Video.* http://www.cs.tu-berlin.de/~phade/secmpeg.html, 1995

[MPEG93] ISO/ IEC International Standard 11172: *Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s.* 1993

[OMG96] Object Management Group: *Security Service: v1.0.* 1996

[PSR93] K. Patel, B.C. Smith, L.A. Rowe: *Performance of a Software MPEG Video Decoder.* Proc. ACM Multimedia, Anaheim, CA, 1993

[QiNa97] L. Qiao, K. Nahrstedt: *A New Algorithm for MPEG Video Encryption.* Proc. 1st Int'l Conf. on Imaging Scienee, Systems and Technology, Las Vegas, NV, 1997

[SCFV96] H. Schulzrinne, S. Casner, R. Frederick, V. Jaeobson: *RTP: A Transport Protocol for Real-Time Applications.* RFC 1889, 1996

[Sch96] B. Schneier: *Applied Cryptography.* 2nd Edition, ISBN 0-471-11709-9, John Wiley, New York, 1996

[SHB95] D. Stevenson, N. Hillery, G. Byrd: *Secure Communications in ATM Networks.* Comm. of the ACM, 38(2), pp. 45-52, 1995

[Ste94] R. Steinmetz: *Data Compression in Multimedia computing - standards and systems.* Multimedia Systems, 1(4), pp. 187-204, Springer Verlag, Berlin 1994

[StNa95] R. Steinmetz, K. Nahrstedt: *Multimedia: Computing, Communications and Applications.* Prentice Hall, München 1995

[Tan96] L. Tang: *Methods for Encrypting and Decrypting MPEG Video Data Efficiently.* Proc. 4th ACM Int'l Multimedia Conference, Boston, MA, 1996