UM MECANISMO SEGURO PARA DISTRIBUIÇÃO DE MATERIAL MULTIMÍDIA

Cíntia Borges Margi
LARC – Laboratório de Arquitetura e Redes de Computadores
EPUSP - Escola Politécnica da Universidade de São Paulo
CEP: 05508-900, São Paulo, SP, Brasil
(011) 818-5280, cbmargi@larc.usp.br

Graça Bressan
LARC – Laboratório de Arquitetura e Redes de Computadores
EPUSP - Escola Politécnica da Universidade de São Paulo
CEP: 05508-900, São Paulo, SP, Brasil
(011) 818-5280, gbressan@larc.usp.br

Wilson V. Ruggiero
LARC – Laboratório de Arquitetura e Redes de Computadores
EPUSP - Escola Politécnica da Universidade de São Paulo
CEP: 05508-900, São Paulo, SP, Brasil
(011) 818-5280, wilson@larc.usp.br

RESUMO

Este trabalho tem como objetivo levantar os principais aspectos envolvidos na distribuição segura de material multimídia e propor um mecanismo de distribuição e reprodução de vídeos MPEG que atenda a estes requisitos. Diversos métodos de criptografia para vídeos MPEG são analisados e comparados. A partir dos principais requisitos, um mecanismo seguro para distribuição de material multimídia é proposto ,inclusive com a especificação de um mecanismo visualizador.

ABSTRACT

This work intends to list the main aspects in secure multimedia distribution and proposes a mechanism for distributing and playing MPEG video that meets these requirements. Several MPEG video encryption methods are analyzed and compared. Based on the main requirements, a secure multimedia distribution mechanism is proposed, including the specification of a viewer.

1 INTRODUÇÃO

Com o aumento da distribuição de material multimídia através da Internet, principalmente com conteúdos valiosos, como é o caso do ensino a distância, a discussão dos aspectos de segurança envolvidos torna-se um assunto muito importante. Esta questão fica mais interessante quando se leva em conta a distribuição de materiais multimídia com acesso controlado em um ambiente de decodificação em tempo real.

Utilizar material multimídia na Web significa integrar e disponibilizar vídeos, áudio, textos, imagens e/ou animações. Cada uma destas mídias possui características diferentes tanto na sua codificação, como no modo de distribuição. Assim, faz-se necessário analisar estes aspectos para cada uma destas mídias isoladamente e os seus conseqüentes aspectos de integração.

Discutir os aspectos de segurança envolvidos na distribuição de material multimídia significa considerar algumas questões principais como: controle de acesso, integridade e sigilo. Estas questões estão interligadas, já que o sigilo torna-se relevante quando o acesso ao material é controlado, ou seja, somente usuários autorizados podem utilizá-lo.

Os textos, animações, desenhos e simulações são transmitidos através de SSL (*Secure Sockets Layer*), utilizando criptografia e certificados digitais. O uso de certificados digitais garante a autenticidade do servidor, e o uso de criptografia garante a confidencialidade e a integridade das informações. Portanto, esta questão possui uma solução bastante satisfatória.

Este trabalho propõe-se a analisar os aspectos de segurança envolvidos, levantar os requisitos e especificar um mecanismo seguro para distribuição e reprodução de material multimídia, mais especificamente de vídeo mpeg,.

2 SEGURANÇA NA DISTRIBUIÇÃO DE MATERIAL MULTIMÍDIA

Os aspectos de segurança a serem enfocados neste trabalho envolvem sigilo, controle de acesso e integridade. O sigilo e a integridade são obtidos através de criptografia. A troca de certificados digitais permite garantir a autenticidade do servidor e do cliente envolvidos na transação. O controle de acesso pode ser implementado através de validação de senhas, ou através de identificação por certificados digitais.

2.1 Serviços de Autenticidade, Integridade e Confidencialidade

Os serviços de segurança caracterizam os diferentes aspectos de um sistema de computadores, tais como:

Autenticidade: Requer que a origem ou o originador de uma mensagem seja corretamente identificado. A verificação da autenticidade é necessária após todo processo de identificação, seja de um usuário para um sistema, de um sistema para o usuário ou de um sistema para outro sistema.

Integridade: Consiste em proteger a informação contra modificação sem a permissão explicita do proprietário daquela informação. A modificação inclui ações como escrita, alteração de conteúdo, alteração de *status*, remoção, criação e o atraso de informações transmitidas. Deve-se considerar a proteção da informação nas suas mais variadas formas: armazenada em discos, fitas de *backup*, etc...

Confidencialidade: Consiste em proteger a informação contra leitura ou cópia por alguém que não tenha sido explicitamente autorizado pelo proprietário daquela informação. A informação deve ser protegida qualquer que seja a mídia que a contenha: impressa, digital, etc... Este tipo de

segurança inclui não apenas a proteção da informação como um todo, mas também de partes da informação que podem ser utilizadas para inferir sobre o todo.

3 O PADRÃO MPEG

O padrão MPEG-I, criado em 1991, foi desenvolvido para armazenar sinais digitais de áudio e vídeo colorido com qualidade VCR (Vídeo Cassete Record), e ser transmitido a uma taxa de 1,5 Mbps [LeGa91]. Como o volume de dados de vídeo é muito grande (1 segundo de vídeo na resolução de 640 x 480 resulta em 27 MB), torna-se necessário utilizar técnicas de compressão. O padrão MPEG trata separadamente vídeo e áudio, especificando como estes sinais são associados e sincronizados, possuindo assim três níveis: a camada de sistema, a camada de vídeo e a camada de áudio.

A compressão de vídeo consiste em eliminar as informações redundantes (correlatas). Estas correlações podem aparecer de duas formas: correlação espacial e correlação temporal. A correlação espacial é observada em uma mesma imagem, ou seja, são as informações redundantes que aparecem em uma imagem, como por exemplo a cor de fundo de uma imagem. Para eliminar a correlação espacial utiliza-se a Transformada Discreta de Cosseno (DCT), seguida da quantização dos coeficientes obtidos. Já a correlação temporal é observada em dois quadros consecutivos; por exemplo a primeira cena mostra uma sala com móveis e uma pessoa, enquanto na segunda cena aparece a mesma sala, porém a pessoa mudou de lugar. Para eliminar a correlação temporal, utiliza-se o processo chamado de Compensação de Movimento, que é o emprego da técnica DPCM, codificando apenas as diferenças encontradas entre os quadros.

As cadeias de vídeo podem ter três tipos de quadros:

- quadro I (*intra-frame*): é um quadro codificado somente com informações da imagem, não dependendo de qualquer quadro passado ou futuro;
- quadro P (*forward predicted frame*): este quadro é codificado relativamente ao quadro de referência precedente mais próximo (quadro I ou quadro P);
- quadro B (*bi-directional predicted frame*): sua codificação é feita relativa ao quadro de referência precedente mais próximo (quadros I ou P), ou ao quadro de referência sucessivo mais próximo, ou a ambos.

Uma seqüência típica de quadros MPEG é apresentada na Figura 1, onde a dependência entre os quadros I, P e B pode ser observada [Mitc96]. Note que se um quadro I não é decodificado corretamente, todos os quadros seguintes apresentarão erros, até a decodificação do próximo quadro I.

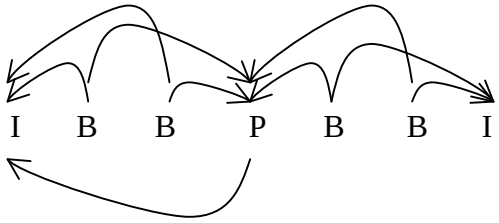

Figura 1: Interdependência de Quadros para uma Seqüência MPEG

A camada de vídeo MPEG é dividida em seis camadas: Camada de Seqüência de Vídeo, Camada de Grupos de Imagens (GOP), Camada de Imagem, Camada de *slice*, Camada de Macroblocos e Camada de Blocos, conforme observa-se na Figura 2.

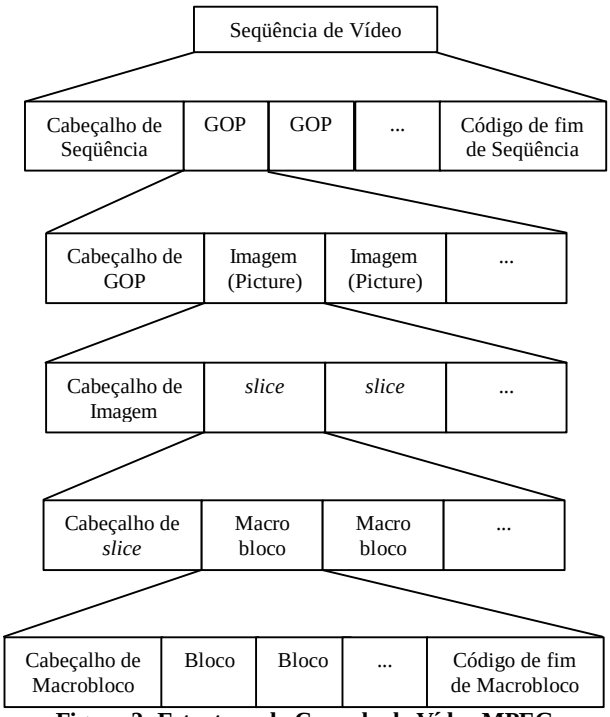

Figura 2: Estrutura da Camada de Vídeo MPEG

Cada uma destas camadas é identificada pelo seu cabeçalho, cujos valores podem ser observados na Tabela 1 [Mitc96].

Tabela 1: Códigos de início de vídeos MPEG

Nome do Código de Início	Valor em Hexadecimal
extension_start code	000001B5
group_start_code	000001B8
picture_start_code	00000100
Reservado	000001B0
Reservado	000001B1
Reservado	000001B6
sequence_end_code	000001B7
sequence_error_code	000001B4
sequence_header_code	000001B3
slice_start_code 1	00000101
...	...
slice_start_code 175	000001AF
user_data_start_code	000001B2

4 CRIPTOGRAFIA PARA MPEG

Existem diversos mecanismos de criptografia para MPEG propostos, sendo cada um deles com enfoques diferentes. Dentre estes mecanismo podem ser citados: Criptografia Simples, Criptografia Seletiva, Algoritmo de Permutação Zig-zag e VEA (*Video Encryption Algorithm*).

4.1 Criptografia Pura ou Simples

No mecanismo de criptografia pura (*Naive Encryption*), os vídeos MPEG são tratados como dados, ou seja, não são consideradas as características da codificação MPEG. O arquivo MPEG é criptografado utilizando um algoritmo de criptografia convencional, como o DES ou IDEA, e, então, o arquivo MPEG é enviado. Após receber o arquivo mpeg, este é decriptografado e o arquivo obtido pode ser assistido. Observe que o arquivo mpeg estará desprotegido (decriptografado) no disco do usuário.

Este mecanismo proporciona um nível de segurança alto, já que a dificuldade em quebrar o algoritmo é aquela apresentada ao tentar quebrar o DES ou o IDEA. Não altera o tamanho do arquivo após a criptografia, mas é muito lento [Qiao98], além de ocorrer aumento no atraso durante a decodificação da cadeia de vídeo [Span96], tornando inviável utilizá-lo em aplicações de tempo real.

4.2 Criptografia Seletiva

A criptografia seletiva procura utilizar as características das cadeias de vídeo MPEG para diminuir a quantidade de informações criptografadas. Os quadros I são aqueles que carregam mais informações, enquanto os quadros P e B representam variações da imagem em quadros I adjacentes. Assim, se os quadros I forem criptografados será difícil compreender o conteúdo de uma cadeia de vídeo. Alguns mecanismos também permitem criptografar os quadros I e P, ou todos os quadros (I, P e B).

Porém, quando criptografa-se somente os quadros I e executa-se o arquivo em player mpeg convencional que suporte erros, ainda é possível perceber o conteúdo do vídeo [Agi96]. Uma solução proposta [Agi96] é aumentar a freqüência dos quadros I, mas isto diminui a compressão, aumentando o tamanho do arquivo.

Dentre os diversos trabalhos que implementam este tipo de mecanismo podem ser citados: SE_MPEG, Aegis e SECMPEG.

SE_MPEG [Li96]: O mecanismo MPEG seguro proposto implementa proteção através de um esquema de criptografia baseado nas seqüências de codificação MPEG. O esquema de criptografia é baseado no PGP, ou seja, é feita criptografia simétrica (algoritmo IDEA) dos quadros e a chave é distribuída através de um esquema de criptografia assimétrica (algoritmo RSA). A criptografia pode ser feita somente no quadros I, nos quadros I e P, ou em todos os quadros I, P e B. Um aumento na proteção implica em maior overhead na criptografia, pois aumentam os quadros a serem criptografados. A encriptação do vídeo MPEG é feita criptografando cada um dos quadros, e mantendo as seqüências de início e fim de código do arquivo MPEG. Quando escolhe-se entre criptografar somente quadros I, quadros I e P, ou os quadros I, P e B, obtém-se uma hierarquia de proteção.

Para criptografia somente de quadros I, a degradação da performance (fps) devido à criptografia varia de 10 a 15% [Li96]. Já para os quadros I, P e B, varia de 13 a 22% [Li96]. Apesar de parecerem índices altos de degradação, segundo [Li96], estes resultados são compatíveis com aplicações para Internet.

Aegis [Span95] [Span96]: No esquema de criptografia proposto somente os quadros I de todos os grupos de quadros em uma cadeia de vídeo MPEG são encriptados, utilizando o algoritmo DES. O Aegis também encripta o cabeçalho de seqüência de vídeo. O cabeçalho de seqüência de vídeo contém os parâmetros de inicialização da decodificação, como altura e largura do quadro, taxa de quadros, taxa de bits e tamanho do *buffer*. Encriptar o cabeçalho dissimula a identidade de uma cadeia MPEG, fazendo com que esta torne-se irreconhecível. O código de seqüência final também é encriptado no Aegis, dificultando ainda mais o reconhecimento de uma cadeia MPEG. Devido a estas diferenças, um *player* MPEG convencional não é capaz de decodificar corretamente uma seqüência de vídeo codificada pelo Aegis, sendo que as imagens de fundo aparecem borradas e sem nitidez [Span96].

Segundo uma simulação feita pelos autores para o mecanismo, o desempenho do Aegis é bastante próximo ao de um sistema sem criptografia, já que este é capaz de manter constante o atraso devido à criptografia. Um sistema com criptografia completa não mantém o atraso constante, pois demora para processar as informações de entrada, acumulando atrasos. Apesar de os atrasos obtidos com Aegis serem muito próximos daqueles obtidos com um *player* convencional, o nível de segurança é aceitável, mas não é adequado para aplicações sensíveis [Agi96], já que com players com suporte a erros ainda é possível identificar a imagem criptografada.

SECMPEG [Meye95]: Este projeto propõe uma variação do padrão MPEG para a transmissão segura de vídeo, que incorpora criptografia seletiva e informações adicionais no cabeçalho. SECMPEG pode utilizar os algoritmos DES e RSA para a criptografia, e faz um cálculo de CRC para verificar a integridade do conteúdo. Implementa quatro níveis de segurança:

- 1º nível: encripta todos os cabeçalhos;
- 2º nível: encripta todos os cabeçalhos mais os coeficientes DC e os termos AC dos blocos I;

- 3º nível: encripta os quadros I e os blocos I dos quadros P e B;
- 4º nível: encripta todos os campos.

4.3 Algoritmo de Permutação Zig-Zag

O mecanismo proposto associa a criptografia a compressão da imagem e do vídeo (JPEG e MPEG) [Tang96]. Este mecanismo de criptografia utiliza uma lista randômica de permutação para fazer o mapeamento dos blocos 8 x 8 no vetor 1 x 64, ao invés de fazê-lo em zig-zag (que é utilizado pelo padrão MPEG).
A partir de quatro experimentos com a ordem dos coeficientes DC e AC no vetor 1 x 64 concluiu-se que: a posição do coeficiente DC é importante; a imagem ainda é compreensível se o coeficiente DC for zero e os coeficientes AC forem permutados em zig-zag; o último coeficiente AC pode ser mudado para zero através da matriz de quantização sem prejuízo à qualidade da imagem.
Outro mecanismo estudado é a divisão do coeficiente DC ($d_0d_1d_2d_3d_4d_5d_6d_7$) em duas partes com quatro bits, sendo uma delas colocada no lugar do coeficiente DC ($d_0d_1d_2d_3$) e outra no lugar do último coeficiente AC ($d_4d_5d_6d_7$). Assim, a codificação / criptografia utiliza este mecanismo, e em seguida aplica a lista de permutação ao vetor 1 x 64, ao invés da permutação em zig-zag.
Este algoritmo aumenta consideravelmente o tamanho das cadeias de vídeo, já que, quando altera-se a ordem do vetor 1x64, perde-se capacidade de compressão (esta é maximizada quando aplica-se a lista de permutação em zig-zag, o que aumenta o número de símbolos repetidos de Huffman).
Este mecanismo de criptografia é vulnerável ao tipo de ataque de texto limpo conhecido. Por este motivo são realizadas algumas modificações: aplica-se uma função de criptografia ao coeficiente DC, e são geradas duas listas de permutação, que são aplicadas segundo um sorteio (*flip coin*).
A Tabela 2 mostra o desempenho do algoritmo para dois vídeos: flower.mpg e tennis.mpg.

Tabela 2: Desempenho do Algoritmo de Permutação Zig-Zag

Tempo para codificação / Vídeo	Algoritmo Original	Algoritmo de Permutação Zig-Zag
flower.mpg	37.985 seg	37.969 seg
tennis.mpg	14.213 seg	14.403 seg

4.4 VEA (Video Encryption Algorithm)

O algoritmo *Video Encryption Algorithm* (VEA) utiliza o comportamento estatístico do vídeo comprimido [Qiao97]. A análise estatística feita com as cadeias de vídeo MPEG trata as cadeias de vídeo como *bytes*. A primeira observação feita é que a freqüência de ocorrência dos valores destes *bytes* (0 a 255) é praticamente a mesma para qualquer valor do *byte*. Analisando esta distribuição para meio *byte*, em qualquer posição da cadeia, não ocorre nenhuma alteração na distribuição de freqüência. Ainda observa-se que diferentes cadeias MPEG possuem o mesmo comportamento!
Outro estudo realizado é relacionado a freqüência de ocorrência de diagramas (pares de números adjacentes). Esta análise divide o quadro I em porções, e então verifica-se o número de ocorrências do par de maior freqüência na porção. Se um destes pares se repetir, então um diagrama se repetiu. Observou-se que não há nenhum padrão de *byte* repetido com porções de 1/16 de um quadro I. Esta informação é relevante para o desenvolvimento do algoritmo VEA.
O algoritmo VEA assume que uma porção do quadro I terá a seguinte forma: $a_1a_2...a_{2n-1}a_{2n}$. Separa-se os *bytes* pares dos *bytes* ímpares, obtendo duas novas cadeias (lista par e lista ímpar). Aplica-se a função Ou-exclusivo entre as listas par e ímpar, obtendo-se $c_1c_2...c_n$. Escolhe uma função de criptografia E, e aplica-se a lista par. O texto criptografado é $c_1c_2...c_n$ E ($a_2a_4...a_{2n}$).
Os autores fazem uma comparação entre o Aegis e o VEA, sendo que o VEA proporciona um ganho de 47% em relação a criptografia com o IDEA no tempo total de criptografia [Qiao97].

4.5 Permutação Pura

Os resultados estatísticos que permitiram o desenvolvimento do VEA, também validam o uso da Permutação Simples. A permutação simples embaralha os *bytes* das cadeias por permutação. A cardinalidade da chave de permutação depende do nível de segurança desejado, podendo variar de 64 números até 1/8 de um quadro I [Qiao98].

4.6 Comparação Entre Os Mecanismos Descritos

Alguns dos mecanismos de criptografia descritos podem alterar o tamanho da cadeia de vídeo MPEG, como o de Permutação em Zig-Zag. O nível de segurança de cada um dos mecanismos é diferente, além do tempo necessário para a criptografia [Qiao98] (ou velocidade de criptografia).

Tabela 3: Comparação dos Algoritmos de Criptografia MPEG

Algoritmo	Nível de Segurança	Velocidade	Tamanho das Cadeias
Criptografia Pura	Alto	Lento	Sem alterações
Criptografia Seletiva	Moderado	Rápido	Aumenta
Permutação Zig-zag	Muito baixo	Muito rápido	Aumenta muito
VEA	Alto	Rápido	Sem alterações
Permutação Pura	Baixo	Super rápido	Sem alterações

Assim, pode-se comparar estes algoritmos segundo três parâmetros: velocidade de criptografia, nível de segurança e tamanho das cadeias de vídeo. A Tabela 3 mostra os resultados desta comparação.

5 UM MECANISMO SEGURO PARA A DISTRIBUIÇÃO DE MATERIAL MULTIMÍDIA

Uma vez analisados os mecanismos de criptografia existentes, é possível levantar quais as características importantes para um mecanismo seguro de distribuição de vídeo MPEG. A implementação deste mecanismo resultará em: esquema de codificação / criptografia MPEG, método de acesso ao vídeo e o *player* MPEG seguro.

5.1 Requisitos

Este mecanismo de distribuição segura de vídeo MPEG deve considerar os seguintes aspectos de segurança:

- Controle de acesso: somente usuários identificados e autorizados têm acesso à reprodução do material;
- Somente material autenticado será reproduzido, identificando a fonte geradora;
- Existem diversos níveis de segurança possíveis, para que haja um compromisso adequado com a velocidade de decodificação.

Estes diversos níveis de segurança podem ser obtidos através de dois modos: utilizando diferentes mecanismos de codificação / criptografia MPEG, ou através do uso de diferentes chaves de criptografia do arquivo MPEG para diferentes clientes.

Este MPEG *player* deve considerar diversos mecanismos, dentre os quais:

- Mecanismo de identificação de usuários, por exemplo certificados digitais;
- Mecanismo de integridade do material distribuído;
- Mecanismo de distribuição de chave secreta (simétrica) para decriptografia do vídeo MPEG;
- Esquema e algoritmo de criptografia para vídeos MPEG.

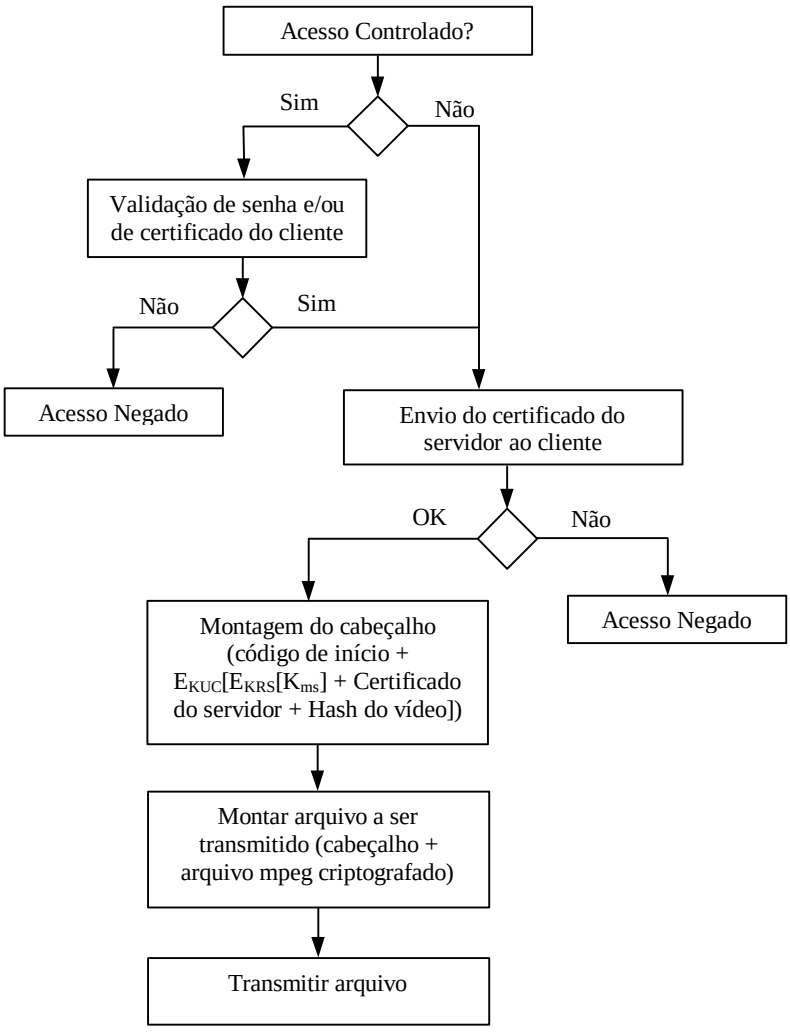

Figura 3: Fluxograma da Distribuição de vídeo segura

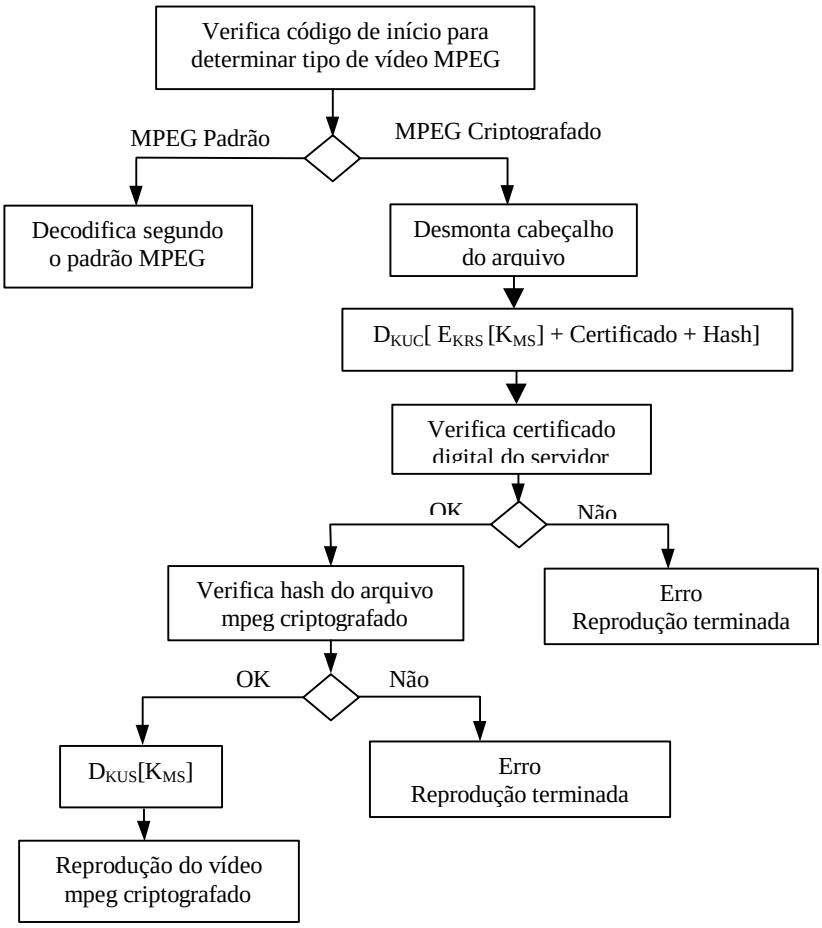

Figura 4: Fluxograma do Mecanismo de reprodução de vídeo mpeg criptografado

5.2 Especificação

Como o mecanismo proposto considera as duas etapas principais, o acesso ao vídeo e a sua reprodução, é necessário especificar cada uma delas.

5.2.1 Acesso Ao Vídeo MPEG

Podem existir dois tipos de vídeo mpeg disponíveis: aqueles de acesso restrito e aqueles de acesso público. Assim, o primeiro passo é verificar se o vídeo solicitado possui ou não acesso controlado. Caso haja necessidade, efetua-se um controle de acesso através de validação de senhas, ou através de identificação por certificados digitais.
O passo seguinte trata da autenticação do servidor e do usuário através da troca de certificados digitais e verificação dos mesmos.
É importante registrar estas transações de controle de acesso e troca de certificados em um arquivo de log do sistema. Com estas informações é possível identificar quais os usuários do sistema e quem acessa os vídeos de acesso restrito, permitindo uma auditoria.

O terceiro passo é montar um cabeçalho para que o visualizador possa reproduzir o vídeo mpeg. Este cabeçalho é composto pelo código de início de seqüência de vídeo ms (mpeg seguro), pela chave de criptografia do vídeo ms (K_{ms}), pelo certificado digital do servidor e pelo hash do vídeo ms.
Para garantir a confidencialidade da chave de criptografia do vídeo, do certificado do servidor e do hash do vídeo ms, estas informações são criptografadas com a chave pública do usuário (K_{UC}). Além disto, para garantir a autenticidade da chave de criptografia do vídeo, esta é criptografada com a chave privada do servidor (K_{RS}). Assim o cabeçalho fica: código de início + $E_{KUC}[E_{KRS}[K_{ms}]$ + Certificado do servidor + Hash do vídeo].
Então, acrescenta-se o cabeçalho ao arquivo mpeg criptografado, e a transmissão do mesmo é iniciada.
A Figura 3 ilustra a etapa de acesso ao vídeo mpeg criptografado.

5.2.2 Reprodução do Vídeo MPEG

O *player* seguro deve ser capaz de reproduzir vídeos mpeg padrão e vídeos mpeg criptografados. Para identificar o tipo de vídeo a ser reproduzido é

necessário determinar um código de início para o vídeo mpeg criptografado, uma vez que todo vídeo mpeg padrão é identificado pelo seu código de início de seqüência.
Assim, o primeiro passo para a reprodução de um vídeo é identificar o tipo de mpeg: padrão ou criptografado. Se for um arquivo mpeg padrão, a reprodução é executada normalmente. Caso seja um vídeo mpeg criptografado, são necessários os outros passos.
Se o vídeo a ser reproduzido for um mpeg criptografado, é necessário desmontar o cabeçalho e decritografar as informações criptografadas utilizando para tanto a chave privada do cliente ($D_{KRC}[E_{KRS}[K_{ms}]$ + Certificado do servidor + Hash do vídeo]).
Então, o certificado do servidor é verificado. Em seguida, calcula-se o hash do arquivo mpeg criptografado e compara-se com o hash recebido. Caso ocorra algum erro, a reprodução do arquivo é terminada.
Esta descrição considera que o hash do vídeo é calculado para todo o arquivo mpeg, assim é necessário ter recebido o arquivo todo para iniciar a sua reprodução. Quando considera-se distribuição de vídeo por *streaming*, torna-se necessário alterar o método de cálculo do hash do arquivo mpeg todo para parte do arquivo mpeg.
O próximo passo é obter a chave K_{ms} para, então, iniciar a reprodução do vídeo.
A Figura 4 ilustra o processo de reprodução de um vídeo mpeg pelo *player* proposto.
Reproduzir um vídeo mpeg criptografado significa decriptografá-lo e decodificá-lo simultaneamente e em tempo real. Ou seja, o vídeo não fica disponível ao usuário decriptografado.
Como mecanismo de criptografia para vídeo mpeg, deve-se utilizar o mais eficiente, podendo ser tomado como base a comparação feita por [Qiao98], que implicaria na utilização do VEA. Para garantir a eficiência do algoritmo utilizado é interessante realizar alguns testes comparativos entre os mecanismos descritos, de modo a obter uma comparação numérica do desempenho dos mesmos.

6 CONSIDERAÇÕES FINAIS

A partir da análise dos mecanismos de criptografia MPEG existentes e do levantamento dos requisitos de um MPEG *player* seguro, é possível especificá-lo e, então, implementá-lo.
O *player* MPEG seguro proposto será desenvolvido a partir do software implementado na parte 5 do padrão ISO/IEC 13818-5 e 11172-5. Este programa foi desenvolvido em linguagem C, que será a linguagem adotada para o desenvolvimento do *player* seguro.
Com a implementação do MPEG *player*, o próximo passo é a realização de testes para verificar o nível de segurança do mecanismo e sua velocidade de criptografia.
Um critério que deve ser utilizado para avaliar a segurança do vídeo criptografado é executá-lo em um *player* que suporte erros, e verificar se é possível identificar objetos na cena.
A implementação do mecanismo proposto associado ao uso de SSL para a distribuição de textos, animações e imagens resolve o problema de segurança na distribuição de material multimídia para ensino à distância.

7 REFERÊNCIAS BIBLIOGRÁFICAS

[Agi96] I. Agi and L. Gong; " **An Empirical Study of Secure MPEG Video Transmission**". In Proceedings of the Internet Society Symposium on Network and Distributed System Security, San Diego, CA, Feb. 1996.

[LeGa91] D.J. LeGall; **"MPEG: A Video Compression Standard for Multimedia Applications".** In Communications of the ACM, Vol. 34, nº 4, April 1991.

[Li96] Y. Li, Z. Chen, S. Tan and R. H. Campbell; **"Security Enhanced MPEG Player"**. In Proceedings of the First International Workshop on Multimedia Software Development (MMSD '96), Berlin, Germany, March 1996.

[Meye95] J. Meyer and F. Gadegast; **"Sicherheitsmechanismen für Multimedia-Daten am Beispiel MPEG-I Video"**. Projektbericht, TU Berlim, 1995. http://www.mpeg1.de

[Mitc96] J.L. Mitchel, W.B. Pennebaker, C.E. Fogg and D.J. LeGall; **"MPEG Video Compression Standard"**. Chapman and Hall, 1996.

[Qiao97] L. Qiao and K. Nahrstedt; **"A New Algorithm for MPEG Video Encryption".** In Proceedings of the First International Conference on Imaging, Science, Systems and Technology (CISST '97), Las Vegas, Nevada, July 1997.

[Qiao98] L. Qiao and K. Nahrstedt; "**Comparison of MPEG Encryption Algorithms**". In Computer & Graphics, vol. 22, nº 4, 1998.

[Span95] G. A. Spanos and T. B. Maples; **"Performance Study of a Selective Encryption Scheme for the Security Networked, Real-Time Video"**. In Proceedings of Fourth International Conference on Computer Communications and Networks, Las Vegas, Nevada, September 1995.

[Span96] G. A. Spanos and T. B. Maples; **"Security for Real-Time MPEG Compressed Video in Distributed Multimedia Applications"** . In Fifteenth IEEE International Phoenix Conference

on Computers and Communications, Scottsdale, AZ, March 1996.

[Tang96] L. Tang; **"Methods for Encrypting and Decrypting MPEG Video Data Efficiently".** In Proceedings of The Fourth ACM International Multimedia Conference (ACM Multimedia '96), Boston, MA, November 1996.