Boletim Técnico da Escola Politécnica da USP
Departamento de Engenharia de Computação e
Sistemas Digitais

ISSN 1413-215X
BT/PCS/0107

Um Mecanismo para Distribuição Segura de Vídeo MPEG

Cíntia Borges Margi
Graça Bressan
Wilson Vicente Ruggiero

São Paulo - 2001

O presente trabalho é parte da dissertação de mestrado apresentada por Cíntia Borges Margi, sob a orientação da Profa. Dra. Graça Bressan e co-orientação do Prof. Wilson Vicente Ruggiero.: "Um Mecanismo para Distribuição Segura de Vídeo MPEG", defendida em 20/12/00, na EPUSP.

A íntegra da dissertação encontra-se à disposição com o autor e na Biblioteca de Engenharia Elétrica da Escola Politécnica da USP.

FICHA CATALOGRÁFICA

Margi, Cíntia Borges
Um mecanismo para distribuição segura de vídeo MPEG / C.B. Margi, G. Bressan, W.V. Ruggiero. – São Paulo : EPUSP, 2001.
p. – (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia de Computação e Sistemas Digitais, BT/PCS/0107)

1. Vídeo – Distribuição e segurança I. Bressan, Graça II. Ruggiero, Wilson Vicente III. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais IV. Título V. Série
ISSN 1413-215X CDD 621.38833

UM MECANISMO PARA DISTRIBUIÇÃO SEGURA DE VÍDEO MPEG

Cíntia Borges Margi, Graça Bressan, Wilson Vicente Ruggiero
LARC - Escola Politécnica
Universidade de São Paulo
05508-900 São Paulo – SP
{cbmargi, gbressan, wilson}@larc.usp.br

RESUMO

Este trabalho faz um levantamento dos principais aspectos envolvidos na distribuição segura de material multimídia, principalmente vídeo MPEG, e propõe 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 para distribuição segura de vídeo MPEG é proposto, inclusive com a especificação do servidor e do visualizador. O artigo também apresenta os resultados da implementação do mecanismo proposto.

ABSTRACT

This work intends to list the main aspects in secure multimedia distribution, mainly MPEG video, 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 distribution mechanism of video MPEG is proposed, including the server and the player specification. This paper also presents the implementation results from the proposed mechanism.

1 INTRODUÇÃO

Com a evolução das aplicações disponíveis na Internet, o uso de multimídia torna-se cada vez mais comum. Dentro deste contexto, o vídeo está sendo cada vez mais utilizado, principalmente com conteúdos valiosos, como é o caso do ensino a distância, às vezes com fins comerciais. Assim, a discussão dos aspectos de segurança envolvidos torna-se um assunto muito importante.

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.

Os principais aspectos de segurança a serem discutidos na distribuição de material multimídia são controle de acesso, integridade e sigilo. Mas, estas questões estão interligadas, já que o sigilo torna-se relevante quando o acesso ao material é controlado.

Os textos, animações, desenhos e simulações podem ser transmitidos com segurança através de SSL (*Secure Sockets Layer*), utilizando criptografia e certificados digitais. O uso de certificados digitais (Feghhi *et al.*, 1999) garante a autenticidade do servidor, e o uso de criptografia garante a confidencialidade e a integridade das informações. Portanto, esta questão possui uma solução satisfatória. Em se tratando de vídeo, outros aspectos devem ser considerados.

Este trabalho tem como objetivo levantar os principais aspectos envolvidos na distribuição segura de material multimídia, mais especificamente de vídeo MPEG, propor um mecanismo de distribuição e reprodução de vídeos MPEG que atenda a estes requisitos, e demonstrar a sua viabilidade.

Para identificar os requisitos de segurança necessários ao mecanismo proposto, este trabalho faz uma análise dos diversos métodos de criptografia para vídeos MPEG propostos em outros centros de pesquisa. A partir dos principais requisitos, um mecanismo para distribuição segura de vídeo MPEG é proposto. Este mecanismo consiste de um visualizador, o S/Viewer, de um servidor de vídeo, o S/Server, e de um protocolo para a comunicação entre ambos.

Após a especificação do mecanismo proposto, a sua viabilidade pode ser verificada através da implementação do protótipo.

O artigo está organizado da seguinte forma: a seção 2 trata dos aspectos de segurança envolvidos na distribuição de material multimídia, a seção 3 apresenta o padrão de compressão MPEG, enquanto a seção 4 apresenta os mecanismos existentes de criptografia para MPEG. A seção 5 apresenta os requisitos e a especificação do mecanismo proposto, enquanto a seção 6 apresenta os resultados da implementação, e a seção 7, as conclusões do trabalho.

2 SEGURANÇA NA DISTRIBUIÇÃO DE MATERIAL MULTIMÍDIA

Os diferentes aspectos de segurança que caracterizam um sistema de computadores (Stallings, 1998) são:

Autenticidade: A verificação da autenticidade é necessária ao processo de identificação, seja de um usuário para um sistema, de um sistema para o usuário ou de um sistema para outro sistema. A troca de certificados digitais e o uso de assinatura digital permite garantir a autenticidade do servidor e do cliente envolvidos na transação.

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. A integridade pode ser verificada através do *hash* da mensagem, que produz um resumo de tamanho fixo.

Confidencialidade: Consiste em proteger a informação contra leitura ou cópia por alguém que não tenha sido explicitamente autorizado pelo proprietário daquela informação. 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. O sigilo, ou confidencialidade, é obtido através de criptografia.

Controle de Acesso: Consiste na capacidade de se permitir ou negar acesso aos serviços e recursos oferecidos pelo sistema. Acessos desconhecidos ou feitos por pessoas não autorizadas podem significar a necessidade de uma verificação de todos os recursos envolvidos em busca de possíveis estragos que possam ter sido causados ao sistema, mesmo que aparentemente nada tenha ocorrido. O controle de acesso pode ser implementado através de validação de senhas, ou através de identificação por certificados digitais.

Os serviços de segurança devem considerar a proteção da informação nas suas mais variadas formas: armazenada em discos, fitas de *backup* ou impressas.

Os aspectos de segurança a serem enfocados neste trabalho envolvem sigilo, controle de acesso e integridade.

Assim, as técnicas de segurança necessárias são assinatura digital, criptografia e certificados digitais.

2.1 Algoritmos de Criptografia

Criptografia é a arte e ciência que estuda como manter informações seguras. Consiste na técnica de transformar um texto claro em um texto cifrado, ou ininteligível, para que este possa ser transmitido ou armazenado. O uso de criptografia implementa, além da confidencialidade, a autenticidade e a verificação da integridade (Stallings, 1998).

Existem dois tipos principais de criptografia: a simétrica (ou convencional) e a assimétrica.

A criptografia simétrica utiliza uma única chave no processo de criptografia, e esta chave deve ser mantida em segredo. Dentre os algoritmos de criptografia simétrica podem ser citados: DES, 3DES, IDEA, RC4.

A criptografia assimétrica utiliza duas chaves, matematicamente relacionadas, sendo uma delas para encriptar e outra para decriptar. Uma das chaves é mantida em segredo, e a outra é divulgada. O principal algoritmo é o RSA.

2.2 Funções de Hash

As funções de hash são mecanismos utilizados para verificar a integridade de uma mensagem. Estas funções são unidirecionais, possuem saída de tamanho fixo (não importa o tamanho da entrada) e devem ser resistentes a colisão (mais de uma entrada podem gerar a mesma saída, mas dada uma saída deve ser impossível encontrar duas entradas que sejam capazes de gerá-la).

Dois algoritmos para cálculo de hash serão abordados: o MD-5 e o SHA-1 (Stallings, 1998).

2.3 Certificados Digitais

Os certificados digitais (Feghhi *et al*, 1999) garantem a autenticidade de uma chave pública. Atualmente, o padrão de certificados utilizado é o ITU-T X.509. As informações que constam deste padrão são (Tremblett, 1999):

- **Versão** do formato do certificado.
- **Número serial**: é único e controlado pela Autoridade de Certificação (CA).
- **Identificação do algoritmo** utilizado para assinar o certificado.
- **Emissor** com informações sobre a CA.
- **Período de validade** inicial e final.
- **Sujeito** com informações do usuário.
- **Informação sobre a chave pública.**
- **Assinatura** da CA cobrindo todo o certificado.

2.4 SSL

O protocolo SSL (Secure Sockets Layer) foi desenvolvido pela Netscape Communications e submetido ao IETF como Internet *Draft* em fevereiro de 1995. Este protocolo é implementado nos sockets, protegendo dados de qualquer aplicação (por exemplo, http, ftp, telnet) que seja construída utilizando-o, ignorando detalhes de implementação das mesmas (Shostack 1995). O HTTP utilizado junto com SSL é chamado HTTPS, e faz uso da porta 443.

SSL provê encriptação de uma sessão; autenticação de um servidor e, opcionalmente, do cliente; e autenticação da mensagem. O SSL é composto por duas camadas, a primeira é a *SSL Record Protocol*, e a outra é composta pelo *Handshake*, *Alert* e pela Aplicação, que utilizam a camada anterior (Stallings, 1998).

O SSL supõe que existe uma estrutura de certificação digital para a autenticação do servidor, não provê renegociação de chaves de sessão, e utiliza um protocolo confiável de camada de transporte.

Os algoritmos de criptografia simétrica suportados pelo SSL incluem: RC2 e RC4, ambos com 128 e 40 bits, DES 3DES e IDEA.

3 O PADRÃO MPEG

Um vídeo com qualidade VCR é composto de quadros na freqüência de 30 quadros por segundo. A digitalização produz quadros em uma resolução com

640 x 480 pixels, sendo que 1 segundo de vídeo resulta em 27 MB.

Como o volume de dados de vídeo é muito grande, tornam-se necessárias técnicas de compressão. O padrão MPEG-1, 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 (LeGall, 1991). O padrão MPEG trata a compressão de vídeo e áudio, especificando como estes sinais são associados e sincronizados, sendo definido em três camadas: 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 temporal. A correlação espacial é observada em uma mesma imagem, ou seja, são as informações redundantes que aparecem em uma imagem, como a cor de fundo. 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 (*Differential Pulse Code Modulation*), codificando apenas as diferenças encontradas entre os quadros.

Em MPEG-1 o quadro é dividido em blocos 16 x 16 amostras para luminância, e blocos de 8 x 8 amostras para cada sinal de crominância. Um macrobloco é composto por um bloco de luminância (4 x (8 x 8) amostras) e dois blocos de crominância (1x (8 x 8) + 1x (8 x 8) amostras), conforme observa-se na Figura 1. O vetor de movimento indica a translação espacial de um bloco para o outro, sendo utilizado na Compensação de Movimento para eliminar a correlação temporal.

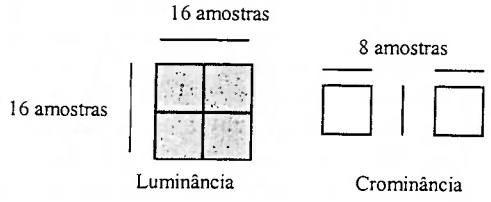

Figura 1: Constituição do Macrobloco MPEG

As cadeias de vídeo MPEG 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 2, onde a dependência entre os quadros I, P e B pode ser observada (Mitchel *et al.*,1996). Note que se um quadro I não é decodificado corretamente, todos os quadros seguintes apresentarão erros, até a decodificação do próximo quadro I.

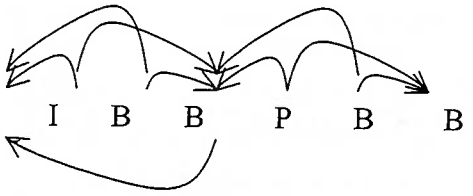

Figura 2: Interdependência de Quadros para uma Seqüência MPEG

A camada de vídeo MPEG é dividida em seis camadas (Figura 3):

- Camada de Seqüência de Vídeo
- Camada de Grupos de Imagens (GOP)
- Camada de Imagem
- Camada de *slice*
- Camada de Macroblocos
- Camada de Blocos

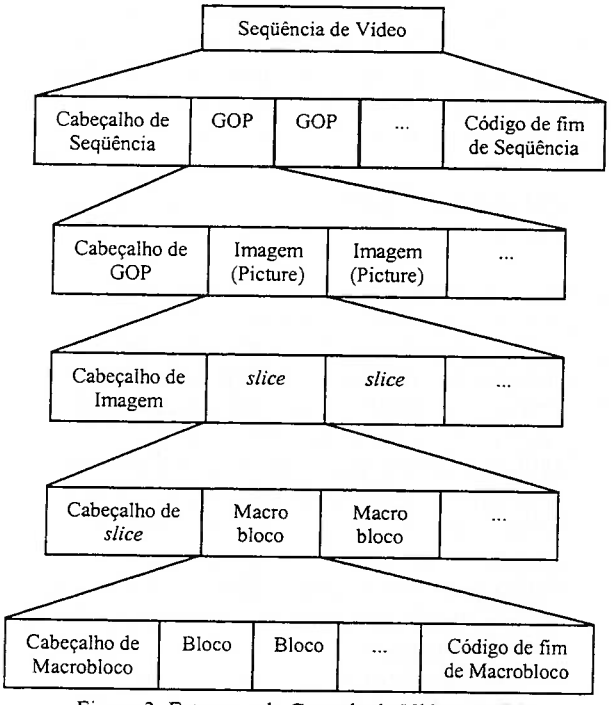

Figura 3: Estrutura da Camada de Vídeo MPEG

Cada uma destas camadas é identificada pelo seu cabeçalho, cujos valores podem ser observados na Tabela 1 (Mitchel *et al,* 1996).

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

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*.

O processo de compressão MPEG segue os seguintes passos:

1. processo de identificação dos quadros;
2. preparação dos blocos de dados;
3. codificação: transformada discreta de coseno (DTC), quantização, supressão de seqüências repetidas (aplicada em zig-zag) e codificação de Huffman.

4 CRIPTOGRAFIA PARA MPEG

Existem diversos mecanismos de criptografia para MPEG, porém cada um deles possui enfoque diferente. Dentre estes mecanismo podem ser citados: Criptografia Simples, Criptografia Seletiva, Algoritmo de Permutação Zig-zag, VEA (*Video Encryption Algorithm*) e Permutação Pura.

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. Outra característica deste mecanismo é não alterar o tamanho do arquivo após a criptografia. A desvantagem deste algoritmo é ser muito lento (Qiao *et al.,* 1998), e causar aumento no atraso durante a decodificação da cadeia de vídeo (Spannos *et al.,* 1996), 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 (Agi *et al.*, 1996). Uma solução proposta (Agi *et al,*. 1996) 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(Li *et al,*. 1996), Aegis (Spannos *et al.*, 1995) (Spannos *et al.,* 1996) e SECMPEG (Meyer *et al,*. 1995).

No SE_MPEG com criptografia somente de quadros I, a degradação da performance (fps) devido à criptografia varia de 10 a 15% (Li *et al.*, 1996). Já para os quadros I, P e B, varia de 13 a 22% (Li *et al,*. 1996). Apesar de parecerem índices altos de degradação, segundo (Li *et al.*, 1996), estes resultados são compatíveis com aplicações para Internet.

O Aegis também encripta o cabeçalho de seqüência de vídeo, dissimulando a identidade de uma cadeia MPEG. O código de seqüência final também é encriptado, dificultando ainda mais o reconhecimento de uma cadeia MPEG.

Os atrasos obtidos com Aegis são muito próximos daqueles obtidos com um *player* convencional e o nível de segurança é aceitável, mas não é adequado para aplicações sensíveis (Agi *et al.,* 1996), já que com *player* com suporte a erros ainda é possível identificar a imagem criptografada.

O SECMPEG propõe uma variação do padrão MPEG para a transmissão segura de vídeo, que incorpora criptografia seletiva e informações adicionais no cabeçalho.

4.3 Algoritmo de Permutação Zig-Zag

O mecanismo proposto associa a criptografia a compressão da imagem e do vídeo (JPEG e MPEG) (Tang, 1996). 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, reduz-se a capacidade de compressão (esta é maximizada quando aplica-se à lista de permutação em zig-zag, o que aumenta o número de símbolos repetidos de Huffman).

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 (sem criptografia)	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 (Qiao *et al.*, 1997). 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 digramas (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 digrama 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-se uma função de criptografia E, e aplica-se a lista par. O texto criptografado é $c_1c_2...c_n$ E ($a_2a_4...a_{2n}$).

O VEA proporciona um ganho de 47% no tempo total de criptografia em relação à criptografia com o IDEA (Qiao *et al,*. 1997).

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 (Qiao *et al.*, 1998).

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 (ou velocidade de criptografia) (Qiao *et al.*, 1998).

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 (Qiao *et al.*, 1998), parecendo o VEA ser o melhor algoritmo.

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

5 Um Mecanismo Seguro para a Distribuição de Vídeo MPEG

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 resultou em:

- um *player* MPEG seguro, o **S/Viewer**,
- na definição do esquema de codificação / criptografia MPEG para o servidor de vídeo, o **S/Server**,
- no protocolo de acesso ao vídeo.

5.1 Requisitos

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

- Controle de Acesso;
- Reprodução de Material Autenticado;
- Diversos Níveis de Segurança.
- Confiabilidade do Protocolo

5.1.1 Controle de Acesso

Os vídeos armazenados no servidor são divididos em dois grupos: vídeos de acesso público e de acesso restrito. Os vídeos de acesso público podem ser reproduzidos por qualquer usuário, fazendo parte dos privilégios de todos os usuários que acessem o servidor de vídeo após a sua identificação. Já os vídeos de acesso restrito, como é o caso daqueles pertencentes aos cursos *online*, possuem associados a eles uma lista com os usuários que tem a autorização para reproduzi-los. A lista de permissão de acesso de um vídeo é criada quando da inserção do mesmo no servidor de vídeo, podendo ser alterada por seu responsável posteriormente.

A identificação do usuário é feita através de seu certificado digital, e os seus privilégios são determinados verificando as listas de acesso dos vídeos.

5.1.2 Reprodução de Material Autenticado

Para iniciar a reprodução de um vídeo MPEG seguro, o S/Viewer solicita o certificado digital do servidor de vídeo, garantindo desta forma a autenticidade do material a ser reproduzido.

Além disto, o *player* também verifica a integridade do vídeo a ser reproduzido, através da verificação do *hash* do mesmo, que está encriptado na assinatura digital.

5.1.3 Níveis de Segurança

Os vídeos disponíveis no servidor possuem diferentes requisitos de segurança. Estes níveis podem ser obtidos através de dois modos:

- utilizando diferentes esquemas de codificação / criptografia MPEG;
- através do uso de diferentes chaves de criptografia para diferentes clientes.

A Tabela 3 (Qiao et al,. 1998[b]) mostra que diferentes esquemas de criptografia oferecem níveis diferentes de segurança e de velocidade. Maior segurança, em geral, resulta em menor velocidade de criptografia.

Desta forma, a escolha do esquema de criptografia deve considerar um compromisso entre atender aos requisitos de segurança e oferecer um desempenho adequado em termos de tempo de codificação / decodificação.

Um nível adicional de segurança, para casos mais críticos, pode ser obtido através do uso de chaves diferentes de criptografia para clientes diferentes terem acesso a um mesmo vídeo.

5.1.4 Confiabilidade do Protocolo

Outro aspecto que deve ser considerado é a confiabilidade do protocolo de comunicação entre o servidor e o *player*. Este não pode ser vulnerável a ataques conhecidos, possuindo a máquina de estados bem definida, sem falhas, tratando todos os possíveis erros.

5.2 Arquitetura do Sistema

A Figura 4 ilustra a arquitetura do sistema proposto e implementado.

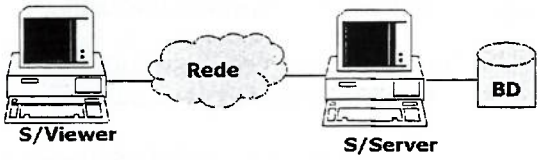

Figura 4: Arquitetura do Sistema

O servidor de vídeo, S/Server, possui uma base de dados associada, onde estão cadastrados os vídeos com as informações necessárias (nível de segurança e usuários que terão acesso aos vídeos) e os usuários com as suas permissões.

O S/Viewer permite ao usuário assistir vídeos MPEG padrão ou seguros (vms - vídeo MPEG seguro), além de poder escolher entre assistir um vídeo gravado localmente ou acessar um servidor de vídeo.

O cliente se comunica com o servidor através do protocolo TCP, utilizando uma porta entre as disponíveis. A implementação do servidor suporta múltiplas conexões, através da criação de novos processos.

5.3 Especificação

A especificação do mecanismo de distribuição segura de vídeo MPEG consta de três partes: do protocolo de acesso, do servidor de vídeo (**S/Server**) e do *player* (**S/Viewer**). As próximas seções tratam desta especificação.

5.3.1 Protocolo de Acesso

Para acessar um vídeo disponível no **S/Server** (servidor de vídeo), o **S/Viewer** (o *player* MPEG seguro proposto) irá estabelecer uma conexão com o

mesmo. A Figura 5 ilustra as mensagens trocadas durante o processo de acesso ao vídeo.

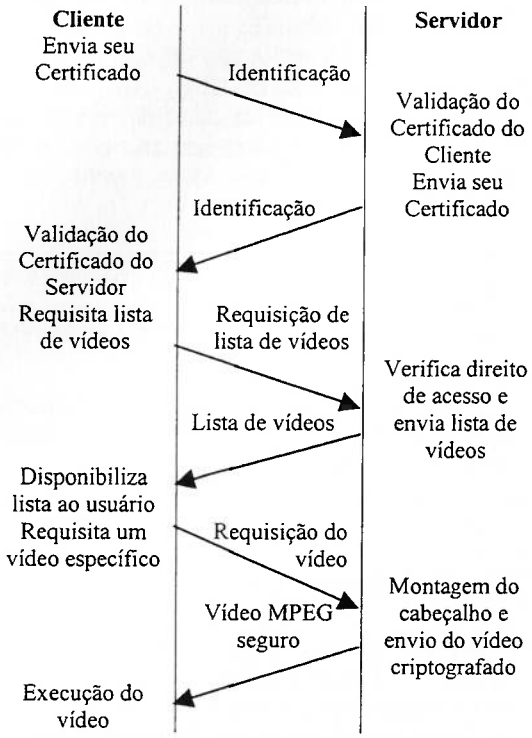

Figura 5: Diagrama de Tempo das Mensagens Trocadas entre o Cliente e o Servidor para Acesso ao Vídeo MPEG Seguro

A primeira mensagem que o cliente envia contém um campo de controle, o seu certificado digital e o horário da requisição criptografado com a sua chave privada (E_{KRC}[tempo]), conforme observa-se na Figura 6.

Controle	Certificado Digital	E_{KR}[tempo]

Figura 6: Formato da Mensagem de Identificação do Cliente

O campo de controle é utilizado para identificar o tipo da mensagem, neste caso uma mensagem de identificação. O horário da requisição deve ser indicado para que o servidor possa rejeitar requisições antigas, que se atrasaram no trajeto até o servidor, ou então para evitar ataques do tipo *replay*, onde um usuário inválido reenvia uma requisição válida para tentar obter acesso aos vídeos.

Ao receber esta mensagem, o servidor irá verificar o certificado digital do cliente, e decriptografar o horário recebido com a chave pública do cliente. Se este horário estiver num intervalo de tempo válido, o servidor envia o seu certificado digital, o horário recebido do cliente encriptado com a chave pública do mesmo, e o horário atual criptografado com a sua chave privada em uma mensagem de identificação (Figura 7).

Controle	Certificado Digital	E_{KU}[tempo recebido]	E_{KR}[tempo]

Figura 7: Formato da Mensagem de Identificação do Servidor

O cliente irá verificar o certificado digital do servidor, se o horário recebido criptografado com a sua chave pública é o mesmo que enviou, e se o horário recebido assinado pelo servidor é válido. Se os horários estiverem corretos e o certificado digital do servidor de vídeo corresponder ao escolhido pelo cliente, este terá a garantia de estar acessando o servidor desejado.

Após a verificação dos certificados, o cliente irá efetuar uma requisição dos vídeos disponíveis (Requisição de Lista de Vídeos), ou então solicitar um vídeo específico (Requisição de Vídeo). Observe que no caso de um curso a distância, o próprio curso indicará o nome do vídeo a ser exibido, fazendo assim uma requisição direta do vídeo.

No caso da requisição de lista de vídeos disponíveis (Figura 8) identificada pelo campo de controle, o servidor irá verificar quais os privilégios do cliente e irá responder com a lista de vídeos. O cliente, então, irá selecionar o vídeo desejado e fazer uma Requisição de Vídeo.

Controle	E_{KR}[tempo]

Figura 8: Formato da Mensagem de Requisição de Lista de Vídeos

Quando o servidor recebe uma requisição de vídeo (Figura 9), este verifica se o usuário possui permissão de acesso ao mesmo. Em caso afirmativo, inicia o processo de transmissão do vídeo MPEG seguro. Caso contrário, envia uma mensagem de erro informando que o acesso ao vídeo não foi permitido.

Controle	Nome do Vídeo	E_{KR}[tempo]

Figura 9: Formato da Mensagem de Requisição de Vídeo

Todas as mensagens trocadas entre o servidor e o cliente possuem um campo com o horário, assinado digitalmente pelo remetente. Desta forma garante-se a autenticidade das mensagens, e evita-se ataques do tipo *reply*.

É importante registrar estas transações de controle de acesso e troca de certificados em um arquivo de log do servidor de vídeo. Com estas informações é possível identificar quais são os usuários do sistema e quem acessa os vídeos disponíveis, permitindo uma posterior auditoria,

seguro de distribuição de vídeo MPEG. A implementação deste mecanismo resultou em:

- um *player* MPEG seguro, o **S/Viewer**,
- na definição do esquema de codificação / criptografia MPEG para o servidor de vídeo, o **S/Server**,
- no protocolo de acesso ao vídeo.

5.1 Requisitos

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

- Controle de Acesso;
- Reprodução de Material Autenticado;
- Diversos Níveis de Segurança.
- Confiabilidade do Protocolo

5.1.1 Controle de Acesso

Os vídeos armazenados no servidor são divididos em dois grupos: vídeos de acesso público e de acesso restrito. Os vídeos de acesso público podem ser reproduzidos por qualquer usuário, fazendo parte dos privilégios de todos os usuários que acessem o servidor de vídeo após a sua identificação. Já os vídeos de acesso restrito, como é o caso daqueles pertencentes aos cursos *online*, possuem associados a eles uma lista com os usuários que tem a autorização para reproduzi-los. A lista de permissão de acesso de um vídeo é criada quando da inserção do mesmo no servidor de vídeo, podendo ser alterada por seu responsável posteriormente.

A identificação do usuário é feita através de seu certificado digital, e os seus privilégios são determinados verificando as listas de acesso dos vídeos.

5.1.2 Reprodução de Material Autenticado

Para iniciar a reprodução de um vídeo MPEG seguro, o S/Viewer solicita o certificado digital do servidor de vídeo, garantindo desta forma a autenticidade do material a ser reproduzido.

Além disto, o *player* também verifica a integridade do vídeo a ser reproduzido, através da verificação do *hash* do mesmo, que está encriptado na assinatura digital.

5.1.3 Níveis de Segurança

Os vídeos disponíveis no servidor possuem diferentes requisitos de segurança. Estes níveis podem ser obtidos através de dois modos:

- utilizando diferentes esquemas de codificação / criptografia MPEG;
- através do uso de diferentes chaves de criptografia para diferentes clientes.

A Tabela 3 (Qiao et al,. 1998[b]) mostra que diferentes esquemas de criptografia oferecem níveis diferentes de segurança e de velocidade. Maior segurança, em geral, resulta em menor velocidade de criptografia.

Desta forma, a escolha do esquema de criptografia deve considerar um compromisso entre atender aos requisitos de segurança e oferecer um desempenho adequado em termos de tempo de codificação / decodificação.

Um nível adicional de segurança, para casos mais críticos, pode ser obtido através do uso de chaves diferentes de criptografia para clientes diferentes terem acesso a um mesmo vídeo.

5.1.4 Confiabilidade do Protocolo

Outro aspecto que deve ser considerado é a confiabilidade do protocolo de comunicação entre o servidor e o *player*. Este não pode ser vulnerável a ataques conhecidos, possuindo a máquina de estados bem definida, sem falhas, tratando todos os possíveis erros.

5.2 Arquitetura do Sistema

A Figura 4 ilustra a arquitetura do sistema proposto e implementado.

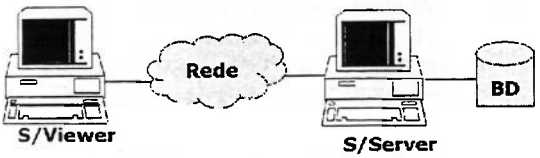

Figura 4: Arquitetura do Sistema

O servidor de vídeo, S/Server, possui uma base de dados associada, onde estão cadastrados os vídeos com as informações necessárias (nível de segurança e usuários que terão acesso aos vídeos) e os usuários com as suas permissões.

O S/Viewer permite ao usuário assistir vídeos MPEG padrão ou seguros (vms - vídeo MPEG seguro), além de poder escolher entre assistir um vídeo gravado localmente ou acessar um servidor de vídeo.

O cliente se comunica com o servidor através do protocolo TCP, utilizando uma porta entre as disponíveis. A implementação do servidor suporta múltiplas conexões, através da criação de novos processos.

5.3 Especificação

A especificação do mecanismo de distribuição segura de vídeo MPEG consta de três partes: do protocolo de acesso, do servidor de vídeo (**S/Server**) e do *player* (**S/Viewer**). As próximas seções tratam desta especificação.

5.3.1 Protocolo de Acesso

Para acessar um vídeo disponível no **S/Server** (servidor de vídeo), o **S/Viewer** (o *player* MPEG seguro proposto) irá estabelecer uma conexão com o

cobrança, ou até mesmo uma simples verificação dos vídeos mais acessados.

5.3.2 *O servidor de Vídeo: S/Server*

O servidor de vídeo é responsável pela criptografia e pela transmissão do vídeo MPEG. Uma vez recebida a requisição do vídeo, o servidor deve cumprir as seguintes etapas, conforme ilustrado na Figura 10:

- criptografar o vídeo MPEG, ou localizar um já criptografado;
- montar o cabeçalho MPEG seguro;
- montar o arquivo a ser transmitido;
- transmitir o arquivo.

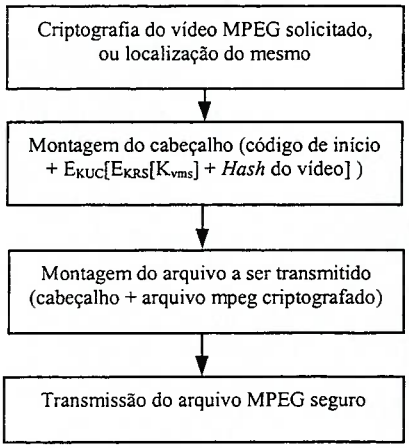

Figura 10: Fluxograma da Criptografia e Transmissão do Vídeo Seguro

A criptografia do vídeo MPEG é feita utilizando um dos esquemas apresentados anteriormente, conforme o nível de segurança:

- escolhe-se o algoritmo correspondente ao nível de segurança e desempenho desejados, determinado quando o vídeo é cadastrado no servidor.
- em geral, utiliza-se o mais eficiente, tomando como base a comparação mostrada na Tabela 3, que implica na utilização do VEA (Qiao *et al*, 1998).

A criptografia do arquivo MPEG pode ser realizada antes da transmissão do mesmo, ou somente quando o usuário solicita o vídeo (criptografia *on the fly*). Caso a criptografia seja anterior à requisição, esta será atendida mais rapidamente, porém o nível de segurança é menor já que diversos usuários receberão vídeos MPEG seguros criptografados com a mesma chave. A criptografia no instante da requisição permite que diferentes usuários recebam arquivos MPEG seguros e com diferentes chaves de criptografia.

O cabeçalho MPEG seguro é necessário para que o visualizador (**S/Viewer**) possa reproduzir o vídeo MPEG. Este cabeçalho é composto pelo código de início de seqüência de vídeo VMS (vídeo MPEG seguro), pela chave de criptografia do vídeo VMS (K_{VMS}) e pelo *hash* do vídeo VMS.

Para garantir a confidencialidade da chave de criptografia do vídeo (K_{VMS}) e do *hash* do vídeo VMS, estas informações são criptografadas com a chave pública do usuário (K_{UC}). Além disto, para garantir a autenticidade da chave de criptografia do vídeo (K_{VMS}), esta é criptografada com a chave privada do servidor (K_{RS}). Assim o cabeçalho fica conforme observa-se na Figura 11.

Código de início	$E_{KUC}[E_{KRS}[K_{vms}]$ + *Hash* do vídeo]

Figura 11: Cabeçalho do Arquivo MPEG seguro (VMS)

Após montar o cabeçalho, este é acrescentado ao arquivo MPEG criptografado, e a transmissão do mesmo é iniciada. A transmissão do arquivo pode ocorrer de duas formas: o arquivo completo ou por *streaming*.

5.3.3 *Reprodução do Vídeo MPEG: S/Viewer*

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.

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 decriptografar as informações criptografadas utilizando para tanto a chave privada do cliente ($D_{KRC}[E_{KRS}[K_{VMS}]$ + *Hash* do vídeo], resultando em $E_{KRS}[K_{VMS}]$ e no *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. Na distribuição de vídeo por *streaming*, o método de cálculo do *hash* do arquivo MPEG é aplicado nas partes do arquivo MPEG.

O próximo passo é obter a chave K_{VMS} (através de $D_{US}[K_{VMS}]$) para, então, iniciar a reprodução do vídeo.

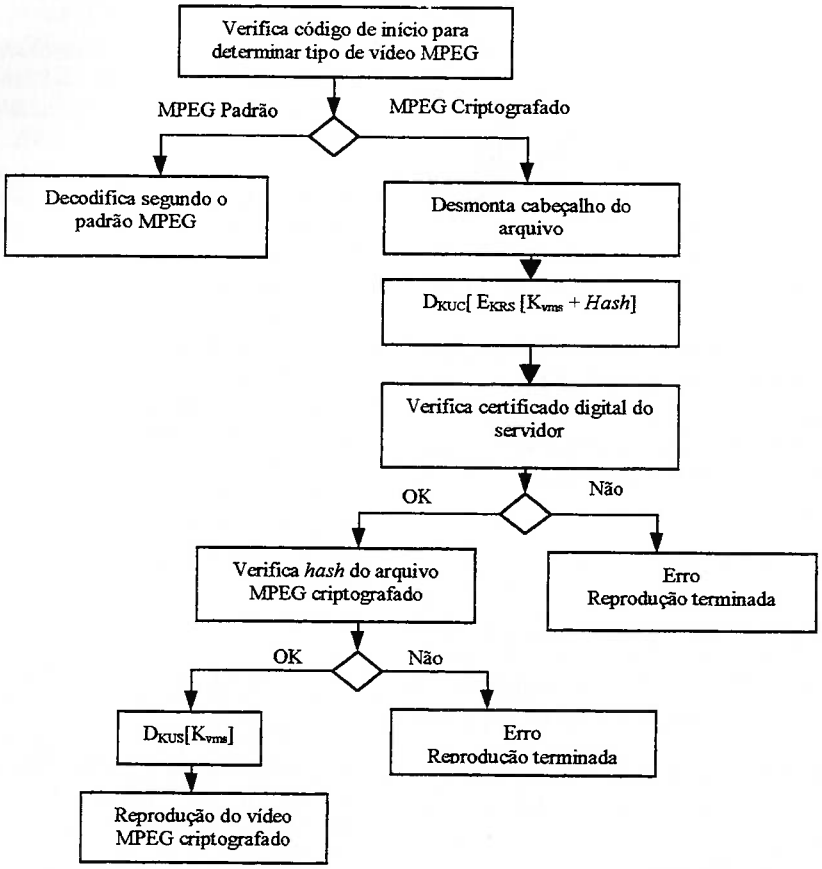

Figura 12: Fluxograma do Mecanismo de reprodução de vídeo MPEG criptografado

A Figura 12 mostra as etapas do processo de reprodução de um vídeo MPEG pelo *player* proposto.

6 Implementação e Resultados

A implementação do protótipo do S/Server e do S/Viewer permitiu demonstrar a viabilidade do mecanismo proposto e colher alguns resultados. Esta seção apresenta o processo de implementação e o resultados obtidos.

6.1 Implementação

O *player* MPEG seguro (S/Viewer) e o servidor (S/Server) propostos utilizam o software MPEG implementado na parte 5 do padrão ISO/IEC 13818-5 e 11172-5. Este programa foi desenvolvido em linguagem C, que é a mesma linguagem adotada para o desenvolvimento do servidor e do *player* seguro.

O decodificador MPEG utilizado gera a saída do vídeo em um display gráfico executável no ambiente Unix. Assim, este Mecanismo de Distribuição Segura de Vídeo foi implementado no ambiente Unix (Linux Red Hat 6.1 com Kernel 2.20-12).

No desenvolvimento da Interface Gráfica do S/Viewer, utilizou-se o aplicativo Glade (Glade). Este aplicativo permite que sejam montadas janelas com botões e menus de maneira amigável. É possível utilizar a interface padrão GTk, ou a Gnome para gerar as janelas, gerando o código correspondente em uma linguagem de programação escolhida. A Figura 13 mostra a tela inicial do S/Viewer.

Figura 13: Tela Inicial do S/Viewer

Tanto o S/Viewer como o S/Server manipulam certificados digitais, utilizam funções de Hash e diversos algoritmos de criptografia padronizados. No

desenvolvimento do S/Server e do S/Viewer foi utilizada a biblioteca OpenSSL (OpenSSL), que implementa os padrões de diversos algoritmos e certificados digitais, possui código aberto e documentação disponível.

A implementação do protótipo do S/server e do S/Viewer foi dividida nas seguinte etapas:

- compilação do código MPEG do comitê da ISO (MPEG Software Simulation Group, 1994);
- criação da interface do cliente;

desenvolvimento da infra-estrutura de rede do cliente e do servidor;

- implementação do protocolo de acesso;
- identificação do tipo de vídeo;
- criptografia de vídeo.

A criptografia de vídeo implementada foi a criptografia pura, utilizando o algoritmo 3DES-CBC (Stallings, 1998).

6.2 Resultados Obtidos

Com a implementação do protótipo do MPEG *player* e do servidor de vídeo seguro, foi possível verificar a viabilidade da implementação do "Mecanismo de Distribuição Segura de Vídeo MPEG".

Porém, uma vez especificado e implementado o mecanismo, como assegurar que a sua utilização soluciona problemas de outros protocolos, ou apresenta uma melhor solução?

Um outro protocolo que poderia ser utilizado para distribuição de vídeo, bem como de outros materiais multimídia (imagens, texto), é o SSL (Shostack, 1994), ou TLS (Dierks *et al*, 1999). O item 6.2.1 faz uma comparação do SSL com o mecanismo proposto neste trabalho.

Outro aspecto a ser discutido após a implementação é sobre o atraso inserido pelo uso de criptografia quando da transmissão e reprodução de um arquivo de vídeo. O item 6.2.2 trata deste tópico.

6.2.1 Comparação com SSL

O SSL provê autenticação do cliente e do servidor através de certificados digitais, assim como o mecanismo proposto neste trabalho. Ambos calculam o *hash* da mensagem transmitida permitindo a verificação da integridade. Porém, o "Mecanismo de Distribuição Segura de Vídeo MPEG" (MDSVM) provê alguns serviços, que geram diferenças significativas entre ambos:

- a criptografia adequada para o vídeo;
- a confidencialidade garantida mesmo quando o arquivo de vídeo está armazenado no cliente;
- possibilidade de identificar distribuições indevidas, que violem direitos autorais.

O SSL provê apenas a criptografia convencional para dados (DES, IDEA, RC4, entre outros), enquanto o MDSVM possui a capacidade de utilizar diversas técnicas de criptografia específicas para vídeo MPEG. Assim, o sigilo do vídeo é obtido mais eficientemente no MDSVM.

O MDSVM garante a confidencialidade do arquivo de vídeo, mesmo quando está armazenado na máquina do usuário, pois a decriptografia só pode ser realizada no S/Viewer. No caso do SSL, o sigilo é garantido apenas durante a transmissão e, uma vez na máquina do usuário, o arquivo não está encriptado. Estas diferenças ocorrem porque o MDSVM atua na camada de aplicação, e o SSL abaixo dela.

Como para obter a chave para abertura do vídeo encriptado é necessária a chave privada do cliente, é possível identificar para qual usuário válido o arquivo de vídeo foi entregue. E se alguma cópia ilegal for encontrada, é possível identificar quem a distribuiu, através da chave pública e dos arquivos de registro do S/Server.

Por estas razões, conclui-se que o "Mecanismo para Distribuição Segura de Vídeo MPEG", além de ter a implementação viável, apresenta uma boa solução para os problemas encontrados na distribuição de vídeo.

6.2.2 Efeitos do Atraso

Os arquivos de vídeo seguros encriptados *on the fly* (quando são requisitados) introduzem um atraso na transmissão de vídeo. Da mesma forma, a decriptografia do vídeo para a sua execução também gera atraso.

Os arquivos de vídeo utilizados para teste do protótipo do "Mecanismo de Distribuição Segura de Vídeo MPEG" são aqueles recomendados como referência pelo comitê MPEG. O nome dos vídeos, sua duração e tamanho são observados na Tabela 4.

Tabela 4: Vídeos de Teste

Vídeo	Tamanho (bytes)	Duração (seg)
bike.mpg	642.590	3,801
flowers.mpg	2.819.836	4,389
siegel.mpg	2.078.802	3,053
tennis.mpg	1.246.001	3,429

Com a implementação da criptografia de vídeo pura, foi possível quantificar o atraso introduzido. A Tabela 5 mostra os atrasos obtidos com o uso da criptografia simples, algoritmo 3DES-CBC.

Tabela 5: Atraso Introduzido pela Criptografia do Vídeo

Vídeo	Atraso na Transmissão (ms)	Atraso na Reprodução (ms)
bike.mpg	327	136
flowers.mpg	1.647	1.088
siegel.mpg	1.248	44
tennis.mpg	725	263

O computador utilizado para efetuar os testes é um Intel Pentium III 650Mhz, com 128 MB de memória RAM, e Linux 7.0 instalado. O servidor e o cliente

foram executados simultaneamente no computador referido.

A criptografia do vídeo introduziu um atraso maior do que a decriptografia, devido às operações que o servidor executa no processo. É possível que a própria implementação do algoritmo de criptografia e decriptografia utilizado possua diferentes desempenho.

O atraso inserido mostrou-se alto, porém conforme descrito por (Qiao *et al*, 1998) e apresentado no capítulo 4, a criptografia pura é o processo mais lento.

É importante identificar o tempo que o usuário tolera esperar pelo arquivo de vídeo, já que com a criptografia pura a espera pode representar até 60% da duração do vídeo. Uma vez conhecida a tolerância do usuário, é necessário definir qual o melhor mecanismo de criptografia de vídeo para cada aplicação, minimizando o atraso no início da reprodução do vídeo.

7 Considerações Finais

O presente trabalho trata de um mecanismo para distribuição segura de vídeo MPEG, sendo discutidos níveis de segurança específicos para este tipo de vídeo. Os níveis de segurança dependem das diversas técnicas de criptografia de vídeo MPEG apresentadas neste trabalho. Porém o mecanismo desenvolvido pode ser facilmente adaptado para outros tipos de arquivos de vídeo, e até mesmo para áudio, havendo alterações apenas nos níveis de segurança, e suas respectivas técnicas de criptografia.

Uma questão que pode ser formulada é por que não utilizar um protocolo já padronizado, como o SSL, para a distribuição de vídeo. Tanto o SSL como o mecanismo proposto provêm a autenticação do servidor e do cliente (opcional no SSL) através de certificados digitais, e calculam o *hash* da mensagem transmitida permitindo a verificação da integridade. Porém, o "Mecanismo de Distribuição Segura de Vídeo MPEG" provê alguns serviços, que geram diferenças significativas entre ambos:

- a criptografia adequada para o vídeo;
- a confidencialidade garantida mesmo quando o arquivo de vídeo está armazenado no cliente;
- possibilidade de identificar distribuições indevidas, que violem direitos autorais.

Estes fatos demonstram a validade da proposta do trabalho, dado que os protocolos existentes ainda não contemplam todos os aspectos de segurança necessários.

Outro resultado obtido no trabalho mostra como é importante definir qual o custo aceito pelo usuário do sistema de distribuição segura de vídeo, ou seja, quanto tempo o usuário tolera esperar pelo arquivo de vídeo, já que com a criptografia pura a espera pode representar até 60% da duração do vídeo. Além de definir a tolerância do usuário, é necessário definir qual o melhor mecanismo de criptografia de vídeo para cada aplicação, minimizando o atraso no início na reprodução do vídeo.

Referências Bibliográficas

AGI, I.; GONG, L. "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.

DIERKS, T.; ALLEN, C.; "The TLS Protocol, Version 1.0", *Request for Comments: 2246*, Category: Standards Track, Jan 1999.

FEGHHI, J; FEGHHI, J; WILLIAMS, P. "Digital Certificates, Applied Internet Security". Addison Wesley, 1999.

GLADE; "Glade: GTK+ User Interface Builder". Disponível em: http://glade.pn.org. Acessado em 26/11/2000.

LEGALL, D.J. "MPEG: A Video Compression Standard for Multimedia Applications". Communications of the ACM, Vol. 34, nº 4, April 1991.

LI, Y.; CHEN, Z.; TAN, S. & CAMPBELL, R.H. "Security Enhanced MPEG Player". In *proceedings of the First International Workshop on Multimedia Software Development (MMSD '96)*, Berlin, Germany, March 1996.

MEYER, J; GADEGAST, F. "Sicherheitsmechanismen für Multimedia-Daten am Beispiel MPEG-I Video". Projektbericht, TU Berlim, 1995. http://www.mpeg1.de

MITCHEL, J.L.; PENNEBAKER, W.B.; FOGG, C.E; LEGALL, D.J. "MPEG Video Compression Standard". Chapman and Hall, 1996.

MPEG SOFTWARE SIMULATION GROUP; "MPEG-2 Video Encoder / Decoder, Version 1.1", Jun 1994. MPEG-L@netcom.com

OPENSSL; "OpenSSL: The Open Source Toolkit for SSL/TLS". Disponível em: http://www.openssl.org. Acessado em: 26/11/2000.

QIAO, L.; NAHRSTEDT, K. "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.

QIAO, L.; NAHRSTEDT, K. "Comparison of MPEG Encryption Algorithms". In *Computer & Graphics*, vol. 22, nº 4, 1998.

SHOSTACK, A.; "An Overview of SSL (version 2)". Disponível em: http://www.homeport.org/~adam/ssl.html, Maio 1995. Acessado em: 26/11/2000.

SPANOS, G.A; MAPLES, T.B. "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.

SPANOS, G.A; MAPLES, T.B. "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.

STALLINGS, W. "Criptography and Network Security, Principles and Practice", Prentice Hall, 2nd Edition, 1998.

TANG, L. "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.

BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

BT/PCS/9301 - Interligação de Processadores através de Chaves Ômicron - GERALDO LINO DE CAMPOS, DEMI GETSCHKO

BT/PCS/9302 - Implementação de Transparência em Sistema Distribuído - LUÍSA YUMIKO AKAO, JOÃO JOSÉ NETO

BT/PCS/9303 - Desenvolvimento de Sistemas Especificados em SDL - SIDNEI H. TANO, SELMA S. S. MELNIKOFF

BT/PCS/9304 - Um Modelo Formal para Sistemas Digitais à Nível de Transferência de Registradores - JOSÉ EDUARDO MOREIRA, WILSON VICENTE RUGGIERO

BT/PCS/9305 - Uma Ferramenta para o Desenvolvimento de Protótipos de Programas Concorrentes - JORGE KINOSHITA, JOÃO JOSÉ NETO

BT/PCS/9306 - Uma Ferramenta de Monitoração para um Núcleo de Resolução Distribuída de Problemas Orientado a Objetos - JAIME SIMÃO SICHMAN, ELERI CARDOSO

BT/PCS/9307 - Uma Análise das Técnicas Reversíveis de Compressão de Dados - MÁRIO CESAR GOMES SEGURA, EDIT GRASSIANI LINO DE CAMPOS

BT/PCS/9308 - Proposta de Rede Digital de Sistemas Integrados para Navio - CESAR DE ALVARENGA JACOBY, MOACYR MARTUCCI JR.

BT/PCS/9309 - Sistemas UNIX para Tempo Real - PAULO CESAR CORIGLIANO, JOÃO JOSÉ NETO

BT/PCS/9310 - Projeto de uma Unidade de Matching Store baseada em Memória Paginada para uma Máquina Fluxo de Dados Distribuído - EDUARDO MARQUES, CLAUDIO KIRNER

BT/PCS/9401 - Implementação de Arquiteturas Abertas: Uma Aplicação na Automação da Manufatura - JORGE LUIS RISCO BECERRA, MOACYR MARTUCCI JR.

BT/PCS/9402 - Modelamento Geométrico usando do Operadores Topológicos de Euler - GERALDO MACIEL DA FONSECA, MARIA ALICE GRIGAS VARELLA FERREIRA

BT/PCS/9403 - Segmentação de Imagens aplicada a Reconhecimento Automático de Alvos - LEONCIO CLARO DE BARROS NETO, ANTONIO MARCOS DE AGUIRRA MASSOLA

BT/PCS/9404 - Metodologia e Ambiente para Reutilização de Software Baseado em Composição - LEONARDO PUJATTI, MARIA ALICE GRIGAS VARELLA FERREIRA

BT/PCS/9405 - Desenvolvimento de uma Solução para a Supervisão e Integração de Células de Manufatura Discreta - JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI

BT/PCS/9406 - Método de Teste de Sincronização para Programas em ADA - EDUARDO T. MATSUDA, SELMA SHIN SHIMIZU MELNIKOFF

BT/PCS/9407 - Um Compilador Paralelizante com Detecção de Paralelismo na Linguagem Intermediária - HSUEH TSUNG HSIANG, LÍRIA MATSUMOTO SAITO

BT/PCS/9408 - Modelamento de Sistemas com Redes de Petri Interpretadas - CARLOS ALBERTO SANGIORGIO, WILSON V. RUGGIERO

BT/PCS/9501 - Sintese de Voz com Qualidade - EVANDRO BACCI GOUVÊA, GERALDO LINO DE CAMPOS

BT/PCS/9502 - Um Simulador de Arquiteturas de Computadores "A Computer Architecture Simulator" - CLAUDIO A. PRADO, WILSON V. RUGGIERO

BT/PCS/9503 - Simulador para Avaliação da Confiabilidade de Sistemas Redundantes com Reparo - ANDRÉA LUCIA BRAGA, FRANCISCO JOSÉ DE OLIVEIRA DIAS

BT/PCS/9504 - Projeto Conceitual e Projeto Básico do Nível de Coordenação de um Sistema Aberto de Automação, Utilizando Conceitos de Orientação a Objetos - NELSON TANOMARU, MOACYR MARTUCCI JUNIOR

BT/PCS/9505 - Uma Experiência no Gerenciamento da Produção de Software - RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO

BT/PCS/9506 - MétodOO - Método de Desenvolvimento de Sistemas Orientado a Objetos: Uma Abordagem Integrada à Análise Estruturada e Redes de Petri - KECHI HIRAMA, SELMA SHIN SHIMIZU MELNIKOFF

BT/PCS/9601 - MOOPP: Uma Metodologia Orientada a Objetos para Desenvolvimento de Software para Processamento Paralelo - ELISA HATSUE MORIYA HUZITA, LÍRIA MATSUMOTO SATO

BT/PCS/9602 - Estudo do Espalhamento Brillouin Estimulado em Fibras Ópticas Monomodo - LUIS MEREGE SANCHES, CHARLES ARTUR SANTOS DE OLIVEIRA

BT/PCS/9603 - Programação Paralela com Variáveis Compartilhadas para Sistemas Distribuídos - LUCIANA BEZERRA ARANTES, LIRIA MATSUMOTO SATO

BT/PCS/9604 - Uma Metodologia de Projeto de Redes Locais - TEREZA CRISTINA MELO DE BRITO CARVALHO, WILSON VICENTE RUGGIERO

BT/PCS/9605 - Desenvolvimento de Sistema para Conversão de Textos em Fonemas no Idioma Português - DIMAS TREVIZAN CHBANE, GERALDO LINO DE CAMPOS

BT/PCS/9606 - Sincronização de Fluxos Multimídia em um Sistema de Videoconferência - EDUARDO S. C. TAKAHASHI, STEFANIA STIUBIENER

BT/PCS/9607 - A importância da Completeza na Especificação de Sistemas de Segurança - JOÃO BATISTA CAMARGO JÚNIOR, BENÍCIO JOSÉ DE SOUZA

BT/PCS/9608 - Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança - BRÁULIO COELHO ÁVILA, MÁRCIO RILLO

BT/PCS/9609 - Implementação de Engenharia Simultânea - MARCIO MOREIRA DA SILVA, MOACYR MARTUCCI JÚNIOR

BT/PCS/9610 - Statecharts Adaptativos - Um Exemplo de Aplicação do STAD - JORGE RADY DE ALMEIDA JUNIOR, JOÃO JOSÉ NETO

BT/PCS/9611 - Um Meta-Editor Dirigido por Sintaxe - MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO

BT/PCS/9612 - Reutilização em Software Orientado a Objetos: Um Estudo Empírico para Analisar a Dificuldade de Localização e Entendimento de Classes - SELMA SHIN SHIMIZU MELNIKOFF, PEDRO ALEXANDRE DE OLIVEIRA GIOVANI

BT/PCS/9613 - Representação de Estruturas de Conhecimento em Sistemas de Banco de Dados - JUDITH PAVÓN MENDONZA, EDIT GRASSIANI LINO DE CAMPOS

BT/PCS/9701 - Uma Experiência na Construção de um Tradutor Inglês - Português - JORGE KINOSHITA, JOÃO JOSÉ NETO

BT/PCS/9702 - Combinando Análise de "Wavelet" e Análise Entrópica para Avaliar os Fenômenos de Difusão e Correlação - RUI CHUO HUEI CHIOU, MARIA ALICE G. V. FERREIRA

BT/PCS/9703 - Um Método para Desenvolvimento de Sistemas de Computacionais de Apoio a Projetos de Engenharia - JOSÉ EDUARDO ZINDEL DEBONI, JOSÉ SIDNEI COLOMBO MARTINI

BT/PCS/9704 - O Sistema de Posicionamento Global (GPS) e suas Aplicações - SÉRGIO MIRANDA PAZ, CARLOS EDUARDO CUGNASCA

BT/PCS/9705 - METAMBI-OO - Um Ambiente de Apoio ao Aprendizado da Técnica Orientada a Objetos - JOÃO UMBERTO FURQUIM DE SOUZA, SELMA S. S. MELNIKOFF

BT/PCS/9706 - Um Ambiente Interativo para Visualização do Comportamento Dinâmico de Algoritmos - IZAURA CRISTINA ARAÚJO, JOÃO JOSÉ NETO

BT/PCS/9707 - Metodologia Orientada a Objetos e sua Aplicação em Sistemas de CAD Baseado em "Features" - CARLOS CÉSAR TANAKA, MARIA ALICE GRIGAS VARELLA FERREIRA

BT/PCS/9708 - Um Tutor Inteligente para Análise Orientada a Objetos - MARIA EMÍLIA GOMES SOBRAL, MARIA ALICE GRIGAS VARELLA FERREIRA

BT/PCS/9709 - Metodologia para Seleção de Solução de Sistema de Aquisição de Dados para Aplicações de Pequeno Porte - MARCELO FINGUERMAN, JOSÉ SIDNEI COLOMBO MARTINI

BT/PCS/9801 - Conexões Virtuais em Redes ATM e Escalabilidade de Sistemas de Transmissão de Dados sem Conexão - WAGNER LUIZ ZUCCHI, WILSON VICENTE RUGGIERO

BT/PCS/9802 - Estudo Comparativo dos Sistemas da Qualidade - EDISON SPINA, MOACYR MARTUCCI JR.

BT/PCS/9803 – The VIBRA Multi-Agent Architecture: Integrating Purposive Vision With Deliberative and Reactive Planning – REINALDO A. C. BIANCHI , ANNA H. REALI C. RILLO, LELIANE N. BARROS

BT/PCS/9901 – Metodologia ODP para o Desenvolvimento de Sistemas Abertos de Automação – JORGE LUIS RISCO BECCERRA, MOACYR MARTUCCI JUNIOR

BT/PCS/9902 – Especificação de Um Modelo de Dados Bitemporal Orientado a Objetos – SOLANGE NICE ALVES DE SOUZA, EDIT GRASSIANI LINO DE CAMPOS

BT/PCS/9903 – Implementação Paralela Distribuída da Dissecação Cartesiana Aninhada – HILTON GARCIA FERNANDES, LIRIA MATSUMOTO SATO

BT/PCS/9904 – Metodologia para Especificação e Implementação de Solução de Gerenciamento – SERGIO CLEMENTE, TEREZA CRISTINA MELO DE BRITO CARVALHO

BT/PCS/9905 – Modelagem de Ferramenta Hipermídia Aberta para a Produção de Tutoriais Interativos – LEILA HYODO, ROMERO TORI

BT/PCS/9906 – Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores-LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos – JOÃO I. DA SILVA FILHO, JAIR MINORO ABE

BT/PCS/9907 – Modelo Nebuloso de Confiabilidade Baseado no Modelo de Markov – PAULO SÉRGIO CUGNASCA, MARCO TÚLIO CARVALHO DE ANDRADE

BT/PCS/9908 – Uma Análise Comparativa do Fluxo de Mensagens entre os Modelos da Rede Contractual (RC) e Colisões Baseada em Dependências (CBD) – MÁRCIA ITO, JAIME SIMÃO SICHMAN

BT/PCS/9909 – Otimização de Processo de Inserção Automática de Componentes Eletrônicos Empregando a Técnica de Times Assíncronos – CESAR SCARPINI RABAK, JAIME SIMÃO SICHMAN

BT/PCS/9910 – MIISA – Uma Metodologia para Integração da Informação em Sistemas Abertos – HILDA CARVALHO DE OLIVEIRA, SELMA S. S. MELNICOFF

BT/PCS/9911 – Metodologia para Utilização de Componentes de Software: um estudo de Caso – KAZUTOSI TAKATA, SELMA S. S. MELNIKOFF

BT/PCS/0001 – Método para Engenharia de Requisitos Norteado por Necessidades de Informação – ARISTIDES NOVELLI FILHO, MARIA ALICE GRIGAS VARELLA FERREIRA

BT/PCS/0002 – Um Método de Escolha Automática de Soluções Usando Tecnologia Adaptativa – RICARDO LUIS DE AZEVEDO DA ROCHA, JOÃO JOSÉ NETO

BT/PCS/0101 – Gerenciamento Hierárquico de Falhas – JAMIL KALIL NAUFAL JR., JOÃO BATISTA CAMARGO JR.

BT/PCS/0102 – Um Método para a Construção de Analisadores Morfológicos, Aplicado à Língua Portuguesa, Baseado em Autômatos Adaptativos – CARLOS EDUARDO DANTAS DE MENEZES, JOÃO JOSÉ NETO

BT/PCS/0103 – Educação pela Web: Metodologia e Ferramenta de Elaboração de Cursos com Navegação Dinâmica – LUISA ALEYDA GARCIA GONZÁLEZ, WILSON VICENTE RUGGIERO

BT/PCS/0104 – O Desenvolvimento de Sistemas Baseados em Componentes a Partir da Visão de Objetos – RENATA EVANGELISTA ROMARIZ RECCO, JOÃO BATISTA CAMARGO JÚNIOR

BT/PCS/0105 – Introdução às Gramáticas Adaptativas – MARGARETE KEIKO IWAI, JOÃO JOSÉ NETO

BT/PCS/0106 – Automação dos Processos de Controle de Qualidade da Água e Esgoto em Laboratório de Controle Sanitário – JOSÉ BENEDITO DE ALMEIDA, JOSÉ SIDNEI COLOMBO MARTINI