Confidencialidade de Dados para Redes ATM com Criptografia Parcial Compatível com Transmissão Multimídia

Patricia Nattrodt B. Soares *José Marcos S. Nogueira*

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Caixa Postal 702, 30123–970 Belo Horizonte, MG
E-mail: {patty,jmarcos}@dcc.ufmg.br

Resumo

A confidencialidade dos dados, obtida através do uso de criptografia, é um dos serviços mais importantes para garantir sigilo na comunicação em redes abertas. No entanto, a criptografia é um processo que exige grande esforço computacional, tornando-se um problema em redes de alta velocidade. Este trabalho apresenta a especificação, implementação e testes de desempenho de um serviço de confidencialidade de dados para redes ATM com suporte para criptografia parcial. Para a realização da verificação de desempenho e análise da viabilidade, foi desenvolvida e integrada ao serviço uma aplicação de transmissão de vídeo MPEG. Os resultados mostram a viabilidade do uso do serviço por aplicações com requisitos de tempo real, dependendo da disponibilidade dos recursos da rede e do nível de segurança necessário.

Abstract

Data confidentiality, which can be achieved with the use of cryptography, is one of the most important services in open networks. It can assure secrecy in their communication services. However, cryptography is a time-consuming process which, by itself, poses a problem in high speed networks. This paper presents the specification, implementation and performance tests of a service aimed for providing data confidentiality in ATM networks, with the support of partial encryption. In order to obtain a measure for performance verification of the proposed service, a MPEG video transmission application was developed, for this specific purpose. The results show that the service can be used in applications with real time constraints, depending on the available resources of the network and on the security level which is under requirement.

1 Introdução

As redes de computadores estão influenciando enormemente a maneira como as pessoas interagem e se comunicam. Numa sociedade que valoriza cada vez mais o conhecimento, comunicação rápida e eficiente se torna um diferencial importante. Por isso, serviços de informação valiosos vêm sendo oferecidos através das redes.

Apesar das redes mais usadas atualmente suportarem muitas aplicações úteis (como correio eletrônico, transferência de arquivos e WWW), em geral, não tratam adequadamente as necessidades das aplicações multimídia, como requisitos de tempo real, demanda por grande quantidade de largura de banda e por garantia da qualidade do serviço. Como o número de aplicações multimídia em rede vem crescendo rapidamente, é imperativo que as redes de computadores suportem essas exigências.

O modo de transferência assíncrono (ATM) é citado como a tecnologia que permite flexibilidade e eficiência, características necessárias para as futuras redes multimídia, multi-serviço e de alta velocidade. ATM é uma tecnologia de comutação e multiplexação usada para transportar pacotes pequenos de tamanho fixo, denominados células, através de uma rede de alta velocidade. Permite a integração e o transporte de dados na forma de texto, voz e vídeo através da mesma infra-estrutura, suporta diferentes níveis de qualidade de serviço dependendo do tipo de tráfego e funciona de forma semelhante tanto para redes locais quanto para redes de longa distância [1].

Com o crescente número de aplicações distribuídas usando a infra-estrutura de redes públicas e o uso indiscriminado dessas redes, aspectos de segurança se tornam fundamentais [2]. Para que os serviços de informação continuem crescendo como se espera, é preciso oferecer uma rede que, além de eficiente, seja segura.

Uma das fraquezas da tecnologia ATM é a falta de mecanismos para garantir a segurança na rede. Cada vez mais, empresas, instituições financeiras e órgãos governamentais querem migrar para ATM. No entanto, para explorar toda a sua potencialidade, a disponibilidade de serviços de segurança, como autenticação, confidencialidade, não-repudiação, integridade e controle de acesso, se torna um fator indispensável [3].

Alguns trabalhos vêm sendo desenvolvidos no sentido de incorporar mecanismos de segurança à tecnologia ATM [4][5][6]. Um dos serviços mais importantes para garantir uma transmissão com sigilo é a confidencialidade dos dados, obtido através do uso de criptografia. Porém, a criptografia é um processo que exige muito esforço computacional, tornando-se um problema em virtude das grandes velocidades exigidas por algumas aplicações.

Existem alguns trabalhos que consideram esse problema na transmissão de vídeos. Agi e Gong [7] e Li et al. [8] tratam segurança em vídeos MPEG e se baseiam em variações do esquema de criptografia seletiva, que usa o fato do decodificador depender da integridade dos quadros I para a decodificação dos quadros P e B. Em teoria, a cifração dos quadros I torna inútil a informação contida nos quadros P e B.

Já Kunkelman et al. [9] examinam o uso de criptografia parcial nos sistemas de transporte para dados multimídia. Diferente dos trabalhos citados anteriormente, esse esquema é mais abrangente, se aplicando a vídeos codificados com algoritmos baseados em transformada discreta de cosseno (DCT - *Discrete Cosine Transform*), como MPEG, Motion-JPEG, H.261 e H.263. Essa abordagem tira proveito da informação dos coeficientes DCT em cada quadro para otimizar a relação entre a quantidade de informação cifrada e o nível de segurança obtido.

A motivação para este trabalho foi investigar um mecanismo capaz de transmitir dados com segurança em redes ATM possibilitando taxas de transmissão compatíveis com as aplicações multimídia e que pudesse ser utilizado por qualquer aplicação da rede. Para isso, foi implementado um serviço de confidencialidade de dados para redes ATM a partir da arquitetura proposta em [4]. Essa arquitetura, além de confidencialidade, oferece autenticação de origem e destino. O serviço foi adaptado para operar na ausência de um protocolo de distribuição de chaves. Além disso, um mecanismo de criptografia parcial simples foi adicionado com o intuito de melhorar o desempenho do serviço e, assim, viabilizar o seu uso por aplicações multimídia com requisitos de tempo real, mas ainda mantendo o serviço de segurança genérico.

Para fazer a análise de desempenho e verificar o impacto do serviço de segurança e do mecanismo de criptografia parcial em aplicações multimídia, foi desenvolvida e integrada ao serviço de confidencialidade uma aplicação de transmissão de vídeo MPEG via ATM.

Este trabalho está organizado como descrito a seguir. A seção 2 apresenta a descrição do serviço de confidencialidade para redes ATM como proposto em [4]. A seção 3 traz a especificação do serviço de confidencialidade com criptografia parcial. Na seção 4, são apresentados aspectos relacionados com a implementação. A seção 5 apresenta os testes realizados para a verificação de desempenho e a análise de viabilidade. E a seção 6 traz a conclusão do trabalho.

2 O Serviço de Confidencialidade dos Dados

Um dos serviços oferecidos pela arquitetura proposta por Dênio T. Silva [4] é o de confidencialidade dos dados. Os módulos da arquitetura diretamente relacionados com esse

serviço são o Protocolo Específico do Serviço de Segurança (SSSP - *Service Specific Security Protocol*) e o Gerenciador Local de Segurança (LSM - *Local Security Manager*), descritos nesta seção.

2.1 Protocolo Específico do Serviço de Segurança

O SSSP utiliza algoritmo simétrico de bloco na subcamada SSCS (*Service Specific Convergence Sublayer*) da camada de Adaptação ATM tipo 5 (AAL5). Esse protocolo também permite a troca de chave de sessão através do circuito virtual estabelecido para transferência de dados. Para isso, são utilizadas PDUs especiais de gerenciamento. O serviço de confidencialidade é fim-a-fim e é oferecido de forma independente para cada conexão estabelecida. O protocolo é responsável por manter controle dos parâmetros necessários para cada circuito virtual. Ou seja, os dados enviados em um ponto de acesso ao serviço (SAP - *Service Access Point*) da AAL5 são cifrados e decifrados de forma independente dos outros SAPs, utilizando uma chave de sessão e um vetor de inicialização únicos para aquele SAP. O ponto de acesso ao serviço é identificado pelo número do circuito virtual definido no processo de estabelecimento da conexão. O SSSP define como a subcamada SSCS deve cifrar e decifrar os dados; define o formato das PDUs de gerenciamento de chaves e também especifica como as PDUs de gerenciamento do protocolo devem ser tratadas.

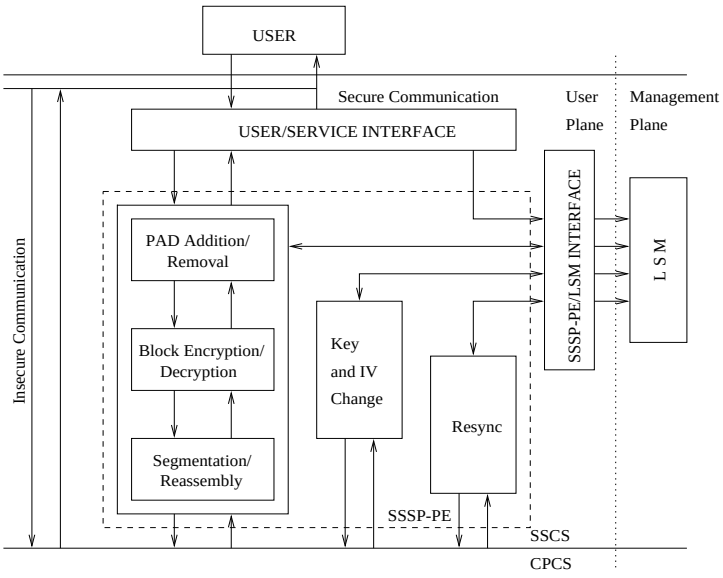

Figura 1: Estrutura do serviço de segurança.

O funcionamento básico do SSSP é dividido em três fases. Na transmissão, ocorre a adição de um campo de preenchimento (*pad addition*), criptografia em bloco (*block encryption*) e segmentação (*segmentation*). Na recepção, o processo é o inverso: é realizada a remontagem (*reassembly*), seguida da decifração em bloco (*block decryption*) e da retirada do campo de preenchimento (*pad removal*). Além dessas fases, o SSSP possui duas funções importantes para o gerenciamento do serviço: troca de chave de sessão e ressincronização. Esses procedimentos e as fases do protocolo são simplificadamente descritos a seguir. O relacionamento do SSSP com o usuário do serviço e com o LSM, incluindo suas fases e os procedimentos, é ilustrado na figura 1.

- ***Pad Addition/Removal***. Na transmissão, a PDU que chega na camada AAL5, que passa a ser uma SSSP-SDU, deve ser acrescida de um campo de preenchimento (PAD)

para que o comprimento da SDU se torne múltiplo de 8 bytes. Desse modo, a SDU pode ser dividida em um número inteiro de blocos de 8 bytes para o cifrador de blocos. Na recepção, após a decifração em bloco, o campo de preenchimento deve ser retirado.

- ***Block Encryption/Decryption***. O cifrador de bloco cifra a SDU em blocos de 8 bytes em modo CBC (*Cypher Block Chainning*) no nó origem. Os blocos são agrupados para gerar a SSSP-PDU. No nó destino, após a remontagem, a PDU é decifrada.
- ***Segmentation/Reassembly***. Em virtude do campo de preenchimento, a PDU do SSSP poderá ficar com tamanho maior que o permitido para a SDU. Neste caso, a PDU deverá ser segmentada em duas partes. No nó destino, a PDU é remontada.
- **Troca de Chave de Sessão e VI**. Quando o procedimento para troca de chave de sessão é invocado, a entidade do protocolo SSSP do nó origem gera a chave de sessão e o vetor de inicialização, passa esses valores para o Gerenciador de Segurança Local, instala os parâmetros no cifrador e manda essa informação para o nó destino. Depois disso, o nó origem espera do nó destino uma PDU idêntica à enviada. Quando essa PDU chegar, a chave de sessão e o vetor de inicialização são instalados no decifrador. Quando uma PDU de troca de chave chega ao nó destino, o protocolo SSSP decifra a chave de sessão com a chave mestra, decifra o vetor de inicialização com a nova chave de sessão e instala os dois parâmetros na decifração. Depois disso, entrega os parâmetros para o Gerenciador de Segurança Local e os instala na cifração. Em seguida, monta e transmite de volta uma mensagem idêntica à que foi recebida.
- **Ressincronização**. O procedimento para ressincronização do protocolo é realizado com a troca do vetor de inicialização. Os passos envolvidos na ressincronização são análogos aos da troca de chave de sessão. A ressincronização é realizada periodicamente e também é controlada pelo nó origem.

2.2 Gerenciador de Segurança Local (LSM)

O Gerenciador de Segurança Local é um módulo que controla todo o procedimento de distribuição de chaves (requisição, geração, envio, recepção, instalação, autenticação e verificação da validade). Além disso, também controla os parâmetros locais de segurança.

3 Confidencialidade com Criptografia Parcial

Como dito anteriormente, a confidencialidade dos dados é uma questão importante quando se trata de comunicação através de redes abertas. O mecanismo mais utilizado para garantir segurança na comunicação é a criptografia. No entanto, o uso de algoritmos criptográficos em todo o fluxo de dados pode representar queda de desempenho inaceitável para aplicações multimídia com requisitos de tempo real em redes ATM.

A criptografia parcial é um mecanismo usado para reduzir o impacto dos algoritmos criptográficos e, com isso, fazer com que as aplicações consigam taxas de transmissão mais elevadas. Existem esquemas específicos para cifrar dados parcialmente, como a criptografia seletiva, que leva em consideração a estrutura de um vídeo com compressão MPEG para minimizar o volume de dados que precisam de cifração. No caso deste trabalho, optamos por implementar a criptografia parcial em dados escolhidos aleatoriamente. Ou seja, não há nenhum tipo de processamento nos dados para selecionar quais trechos serão cifrados. Apesar desse mecanismo poder ser menos eficiente que esquemas específicos, mantém o serviço de segurança genérico, podendo ser utilizado por qualquer aplicação da rede.

O serviço de confidencialidade dos dados com suporte para criptografia parcial foi desenvolvido para redes ATM através da implementação de uma adaptação do SSSP e

do LSM, descritos na seção anterior. Denominamos essa extensão do SSSP de Protocolo Específico do Serviço de Segurança com Criptografia Parcial (SSSP-PE - *Service Specific Security Protocol with Partial Encryption*).

A dinâmica de distribuição inicial das chaves de sessão do SSSP foi modificada, visto que o protocolo de distribuição de chaves não está no escopo deste trabalho. Em virtude disso, também foi definido um mecanismo para realizar a distribuição inicial das chaves de sessão de forma automática, ainda que sem autenticação. A chave mestra é inicializada uma vez e é a mesma para todas as conexões. Essa é uma deficiência imposta pela falta de um protocolo de distribuição de chaves.

Esta seção apresenta o detalhamento da especificação do serviço, descrevendo as primitivas de serviço, o funcionamento do gerenciador local de segurança e a sua interface com o SSSP-PE.

3.1 Primitivas de Serviço do Protocolo SSSP-PE

As funções do serviço podem ser acessadas pelo usuário através de cinco primitivas, descritas a seguir.

- **Inicializar_Parâmetros**. Esta primitiva é utilizada para fazer a inicialização dos parâmetros necessários para o início do serviço de segurança para a conexão que acabou de ser estabelecida. Essa função irá substituir o procedimento de distribuição de chaves que, idealmente, seria realizado automaticamente no estabelecimento da conexão. Esta primitiva chama a função que cria um registro em uma Tabela de Segurança Local com os parâmetros relativos à conexão que foi estabelecida. Esta primitiva deverá ser invocada pelo usuário logo após o procedimento de estabelecimento da conexão. Tem como parâmetro o identificador da conexão.
- **Finalizar_Parâmetros**. Esta primitiva é utilizada para remover os parâmetros relacionados a uma determinada conexão. Para isso, a primitiva chama a função que apaga o registro na Tabela de Segurança Local contendo os parâmetros relativos à conexão que será fechada. Além disso, a primitiva gera uma mensagem que será enviada ao nó destino para sinalizar que os parâmetros relativos à essa conexão devem ser retirados da sua Tabela de Segurança Local. Esta primitiva deverá ser invocada pelo usuário do nó origem antes do procedimento de desconexão. Tem como parâmetro o identificador da conexão.
- **Configurar_Criptografia_Parcial**. Esta primitiva é utilizada para configurar a taxa de dados criptografados. Tem como parâmetros o identificador da conexão e a taxa. Por exemplo, se a taxa for configurada em 80%, 80% das mensagens transmitidas serão criptografadas e as 20% restantes não.
- **Enviar_Mensagem_Cifrada**. Esta primitiva é utilizada para transmitir dados de um usuário local para um usuário remoto com cifração. Tem como parâmetros o identificador da conexão e os dados do usuário.
- **Receber_Mensagem_Cifrada**. Esta primitiva é utilizada para que o usuário local receba dados do usuário remoto decifrados. Tem como parâmetros o identificador da conexão e os dados do usuário.

Do ponto de vista do usuário, o ordenamento natural das primitivas oferecidas está listado a seguir. As primitivas em negrito são oferecidas pelo serviço. As demais são primitivas oferecidas na API do driver da placa ATM usada numa implementação. A primitiva entre colchetes indica que a mesma é opcional. Caso ela não seja invocada, o protocolo assume que a taxa de dados a serem cifrados é 100%.

Abrir_Conexão;
Inicializar_Parâmetros;
[**Configurar_Criptografia_Parcial**;]
Enviar/Receber_Mensagem_Cifrada;
Finalizar_Parâmetros;
Fechar_Conexão.

A primitiva Configurar_Criptografia_Parcial pode ser invocada entre chamadas das primitivas Enviar/Receber_Mensagem_Cifrada, caso a aplicação deseje alterar a taxa de dados que serão cifrados. O diagrama de tempo da figura 2 traz um exemplo de funcionamento do protocolo, ilustrando a ordem de chamada das primitivas do serviço e o procedimento de troca de chave, disparado pelo protocolo.

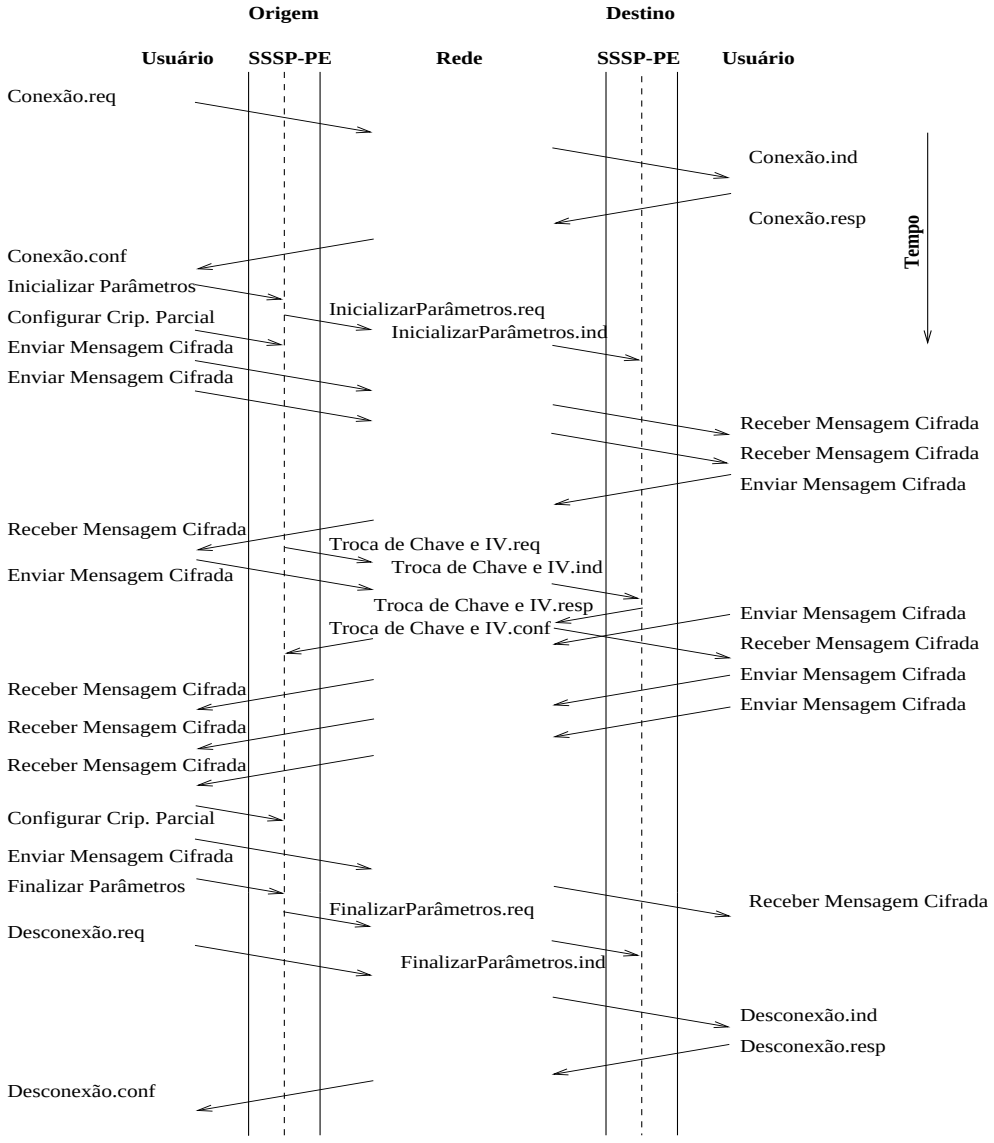

Figura 2: Exemplo de funcionamento do protocolo.

3.2 O Gerenciador de Segurança Local (LSM)

Como proposto por Dênio T. Silva [4], o LSM é um módulo responsável por controlar, além dos parâmetros locais de segurança, todos os processos envolvidos na distribuição

de chaves (seção 2.2). Tal proposta não especifica a estrutura de dados do LSM nem sua interface com o protocolo de segurança (SSSP).

No caso deste trabalho, o Gerenciador de Segurança Local é responsável por gerenciar a chave de sessão e o vetor de inicialização para cada circuito virtual, controlar a validade dos mesmos e manter parâmetros para possibilitar a implementação de criptografia parcial. Para isso, definimos uma Tabela de Segurança Local (LST - *Local Security Table*), que possui os dados necessários para a manutenção desse serviço. Essa tabela possui os seguintes campos: identificador da conexão, chave de sessão para cifração, vetor de inicialização para cifração, chave de sessão para decifração, vetor de inicialização para decifração, data de criação da chave de sessão para cifração, papel do nó na comunicação (origem ou destino), taxa de dados a serem cifrados, número total de PDUs enviadas e número de PDUs cifradas.

Durante a operação normal do protocolo, as chaves de sessão e os vetores de inicialização são iguais para cifração e decifração. No entanto, durante o procedimento para troca de chaves, esses valores serão diferentes. Para que a troca de dados de usuário não seja interrompida durante esse procedimento, é necessário armazenar os parâmetros tanto para cifração quanto para decifração de forma independente.

A data de criação só é armazenada para a chave de cifração, pois o procedimento para troca de chaves gera uma nova chave de sessão e um novo vetor de inicialização, que são utilizados nos dois sentidos (cifração e decifração).

O papel do nó na comunicação é armazenado já que é preciso saber qual nó detém o controle sobre a validade das chaves. Esse controle é feito pelo nó origem.

A taxa de dados a serem cifrados também deve ser mantida para cada conexão. O número total de PDUs enviadas e o número de PDUs cifradas são armazenados para que possa ser feito o controle de cifração de PDUs no caso da utilização de criptografia parcial.

O protocolo SSSP-PE possui uma interface com o gerenciamento da subcamada SSCS. Com essa interface, o SSSP-PE tem acesso aos parâmetros locais de segurança e às funções do LSM. A interface SSSP-PE/LSM é feita através das primitivas descritas a seguir.

- **Solicita_Chave_e_VI**. Esta primitiva solicita os parâmetros de segurança ao gerenciador local. Tem como parâmetros o identificador do circuito virtual e a direção (cifração ou decifração). Retorna a chave de sessão e o vetor de inicialização correspondentes ao circuito virtual.
- **Atualiza_Chave_e_VI**. Esta primitiva atualiza os valores dos parâmetros de segurança na Tabela de Segurança Local. É invocada durante o procedimento de troca de chave e de vetor de inicialização. Tem como parâmetros o identificador do circuito virtual, a chave de sessão, o vetor de inicialização e a direção.
- **Adiciona_Entrada_TSL**. Esta primitiva cria uma nova entrada na Tabela de Segurança Local. Será invocada assim que a conexão for estabelecida. Tem como parâmetros o identificador do circuito virtual, a chave de sessão, o vetor de inicialização, a data de criação da chave de sessão e o papel do nó na comunicação.
- **Remove_Entrada_TSL**. Esta primitiva apaga uma entrada na Tabela de Segurança Local correspondente ao identificador do circuito virtual. Será invocada pelo nó origem antes do procedimento de desconexão. O nó destino invoca esta primitiva ao receber uma PDU de finalizar parâmetros. Tem como parâmetro o identificador do circuito virtual.
- **Configura_Taxa_Criptografia**. Esta primitiva configura a taxa de criptografia parcial. Tem como parâmetros o identificador do circuito virtual e a taxa a ser utilizada.
- **Deve_Cifrar**. Esta primitiva informa se a próxima PDU com dados do usuário a ser enviada deve ser cifrada ou não. Essa decisão é baseada na taxa de criptografia, no

número de PDUs enviadas e no número de PDUs cifradas. A regra possibilita uma distribuição uniforme das PDUs cifradas ao longo da transmissão. Por exemplo, caso a taxa de cifração seja de 50%, teremos PDUs cifradas e não cifradas alternadamente. Tem como parâmetro o identificador do circuito virtual e retorna verdadeiro ou falso.

4 Implementação

Os módulos SSSP-PE e LSM foram implementados na linguagem de programação C. O ambiente utilizado para a implementação deste trabalho foi a rede ATM do Laboratório de Redes de Alta Velocidade (DCC-UFMG). Os equipamentos utilizados foram:

- Comutador ATM LightStream 100 (A100 HyperSwitch) da Cisco, com suporte para 16 linhas ATM a 155Mbps, totalizando um *throughput* agregado de 2.5Gbps.
- Duas estações de trabalho Sun Ultra 1 com placas Sun ATM (155Mbps). As placas são ligadas ao comutador ATM através de cabo par trançado categoria 5. As máquinas possuem 96 MB de memória RAM, processador Sun UltraSPARC de 143 MHz, 1 GB de disco rígido e rodam sistema operacional Solaris 2.5.1.

As placas Sun ATM e o comutador seguem as recomendações do padrão UNI 3.0 do ATM Forum.

Para a cifração dos dados, foi utilizado o IDEA (*International Data Encryption Algorithm*), algoritmo simétrico que opera em blocos de 64 bits e utiliza chaves de 128 bits. Segundo Schneier [10], o IDEA é o melhor e mais seguro algoritmo de bloco disponível para o público atualmente.

O MD5 (*Message Digest 5*) produz um conjunto de 4 blocos de 32 bits, que são concatenados para formar um único valor hash de 128 bits, tamanho da chave do IDEA. Como o MD5 é considerado um algoritmo seguro, ele foi utilizado para a geração das chaves simétricas do IDEA.

O pacote de criptografia escolhido para a implementação, chamado SSLeay, foi desenvolvido na Austrália por Eric Young. O SSLeay foi escolhido pois não sofre restrições de exportação, encontra-se disponível na Internet, implementa os algoritmos necessários e utiliza a linguagem de programação C [11].

5 Testes de Desempenho e Análise de Viabilidade

Para verificar o desempenho do serviço de segurança, foi necessário implementar uma aplicação usuária. A aplicação de transmissão de vídeo MPEG com segurança foi escolhida por duas razões. A primeira motivação foi o serviço de transmissão de vídeo para redes ATM com negociação dinâmica da qualidade de serviço [12], desenvolvido no Laboratório de Redes de Alta Velocidade (DCC-UFMG) e que serviu de base para essa aplicação. A segunda motivação partiu dos trabalhos relacionados com criptografia parcial específicos para vídeos MPEG, referenciados na introdução (seção 1). Dessa forma, foi possível ter algum parâmetro de comparação nesse aspecto. Vale ressaltar que a aplicação usuária tem por objetivo servir de ambiente de testes para o serviço de confidencialidade e, por isso, não tem a pretensão de ser um sistema com recursos sofisticados ou completos para o usuário. A aplicação segue o modelo cliente/servidor e utiliza as primitivas do serviço, definidas na seção 3.1.

Esta seção apresenta os testes de desempenho realizados no serviço de segurança através da aplicação de transmissão de vídeo, abordando os objetivos, o planejamento, a execução desses testes e a análise dos resultados obtidos.

5.1 Objetivos e Planejamento dos Testes

Os objetivos dos testes com a aplicação de transmissão segura de vídeo são: avaliar a funcionalidade do mecanismo de criptografia parcial na transmissão de vídeo; analisar o impacto em termos de tempo do serviço de segurança na aplicação de transmissão de vídeo; verificar o impacto dos procedimentos de troca de chave e ressincronização no tempo de transmissão; medir o *overhead* do protocolo em termos do tráfego gerado; e analisar a viabilidade do serviço de segurança.

Para analisar o impacto do serviço de segurança na aplicação, são comparados os tempos de transmissão de vídeos utilizando o protocolo implementado com os tempos de transmissão sem a utilização das primitivas do serviço de confidencialidade.

Foram feitos testes transmitindo vídeos com várias taxas de cifração, variando de 10% a 100%. Comparando os tempos obtidos, é possível avaliar a vantagem de se utilizar o mecanismo de criptografia parcial para diminuir o impacto do serviço de segurança no tempo de transmissão.

O impacto em termos de tempo dos procedimentos de troca de chave e de ressincronização também é analisado. Para isso, a frequência da chamada desses procedimentos é variada.

O *overhead* de tráfego gerado pelo protocolo foi medido através da contagem do número de PDUs de dados de usuário e de gerenciamento (inicialização/finalização de parâmetros, troca de chave e ressincronização) enviadas durante a transmissão dos vídeos.

A viabilidade do serviço de confidencialidade é analisada considerando o tempo do protocolo e o tempo de transmissão típico em redes metropolitanas.

5.2 Execução de Testes e Medições

Alguns vídeos foram escolhidos na Internet para a realização dos testes. A tabela 1 apresenta algumas características desses vídeos. Os valores da taxa média de transmissão e a duração de cada vídeo foram obtidos através do mpeg_stat, um programa que fornece estatísticas sobre vídeos MPEG desenvolvido na Universidade de Berkeley. Os vídeos poderão ser identificados pelo número atribuído na tabela 1. Por exemplo, o vídeo museum3.mpg poderá ser referenciado por vídeo (6).

Vídeo	**Tamanho**	**Taxa Média de Transmissão**	**Duração**
(1) beardE.mpg	542.8 KB	1.15 Mbps	3.87 s
(2) trap.mpg	1.05 MB	1 Mbps	8.70 s
(3) bartbus.mpg	1.88 MB	692 Kbps	22.73 s
(4) Blazer.mpg	5.27 MB	140 Kbps	313.97 s
(5) DG-5.mpg	15.11 MB	1.12 Mbps	103.90 s
(6) museum3.mpg	25.3 MB	1.55 Mbps	130.30 s

Tabela 1: Características dos vídeos utilizados.

O tempo necessário para a transmissão dos dados (tempo da rede) foi desconsiderado em todos os resultados apresentados nesta seção. Isso faz com que os resultados se apliquem a qualquer tipo de rede, seja LAN, MAN ou WAN.

Cada PDU de dados transporta 5KB de vídeo. Os testes mostraram que esse valor não influencia nos tempos do protocolo. Isso ocorre pois o tempo de cifração é diretamente proporcional ao tamanho do bloco. Assim, aumentando o tamanho do bloco de vídeo a ser transportado em cada PDU, o número de PDUs diminui, porém o tempo gasto com cifração aumenta e o tempo final se mantém constante.

Na aplicação usuária, o servidor envia um vídeo com segurança para um cliente. O servidor e o cliente executam em máquinas diferentes, que se comunicam através de uma

rede. No caso deste trabalho, a rede possui apenas um comutador ATM (seção 4). As medições dos tempos do protocolo foram feitas com os tempos do servidor. Os tempos do servidor e do cliente são semelhantes pois ambos executam as mesmas fases do protocolo em ordem inversa.

Criptografia Parcial

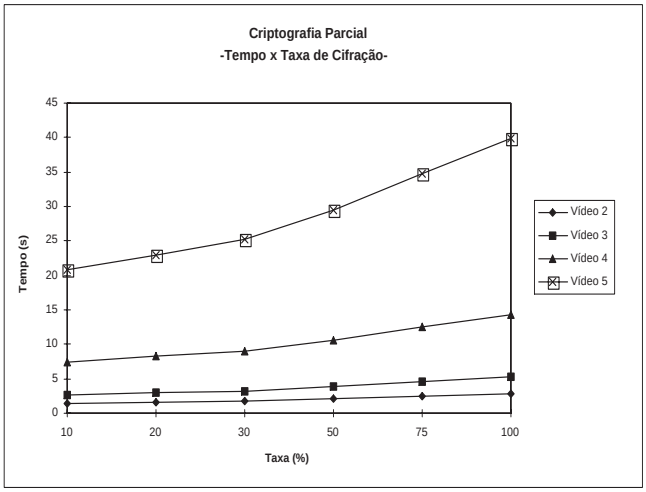

Figura 3: Impacto da criptografia no tempo de transmissão.

Este teste visa avaliar o mecanismo de criptografia parcial do protocolo implementado. Os testes foram realizados nos vídeos 2, 3, 4 e 5. A taxa foi variada, assumindo os valores: 10%, 20%, 30%, 50%, 75% e 100%. O tempo de execução do protocolo no servidor foi medido para cada caso.

Os valores da taxa de cifração estão mais concentrados no intervalo de 10% a 30% com base nos dados de [9]. Segundo esse estudo, a maioria das aplicações com transmissão de vídeos MPEG aceita o nível de segurança obtido caso 5% a 30% dos dados do vídeo estejam cifrados. Aplicações com restrições mais fortes com relação a segurança podem precisar de taxas de cifração mais altas.

O gráfico da figura 3 ilustra o efeito da taxa de criptografia no tempo de execução do protocolo para cada vídeo.

O coeficiente de correlação entre o tempo de cifração para cada taxa considerada e o tempo total de transmissão foi calculado usando o programa de estatística Minitab. O valor obtido foi 0.906, indicando que as variáveis consideradas têm um forte grau de associação linear. Ou seja, a taxa de cifração tem grande influência no tempo do protocolo.

Vídeo	Mín.(10%)	Máx.(100%)	Variação
(2) trap.mpg	1.49s	2.85s	91%
(3) bartbus.mpg	2.71s	5.21s	92%
(4) Blazer.mpg	7.47s	14.38s	93%
(5) DG-5.mpg	20.76s	39.90s	92%

Tabela 2: Tempos do Protocolo.

Os dados da tabela 2 são coerentes com esse resultado. O aumento do tempo de execução do protocolo considerando o caso mais rápido (10% de cifração) e o mais lento (100% de cifração) chega a ser de 93%.

Dessa forma, é possível afirmar que o ganho obtido com a utilização do mecanismo de criptografia parcial é significativo. A configuração da taxa de criptografia é responsabilidade da aplicação e pode variar em função do tipo da aplicação e do nível de segurança necessário.

Impacto do Serviço de Segurança

O objetivo deste teste é verificar o impacto, em termos de atraso inserido, do serviço de confidencialidade numa transmissão. Os testes foram realizados com os vídeos 4, 5 e 6. A banda alocada foi mantida constante. As taxas de cifração utilizadas foram: 30%, um valor considerado bom para transmissão de vídeo com segurança [9], e 100%, o pior

caso. Os resultados obtidos estão ilustrados no gráfico da figura 4.

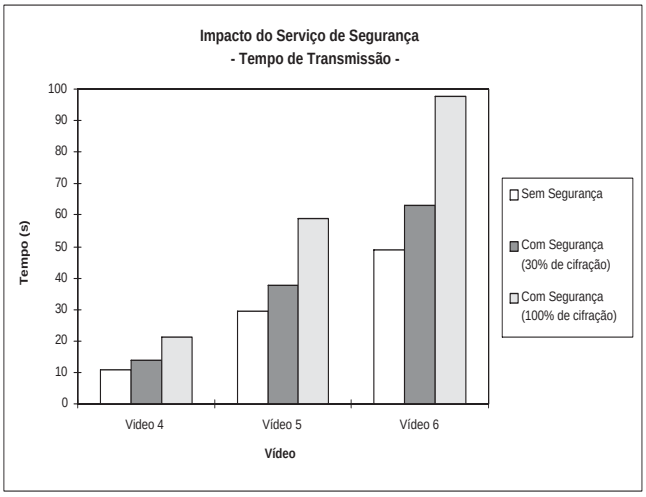

Figura 4: Impacto do serviço de segurança - Tempo de transmissão.

A tabelas 3 e 4 ilustram o impacto da confidencialidade em cada PDU de vídeo para 30% e 100% de cifração, respectivamente.

Vídeo	No. de PDUs	Atraso Total	Atraso/PDU
(4)	1080	3.15s	2.92ms
(5)	2952	8.49s	2.88ms
(6)	4946	14.44s	2.92ms

Tabela 3: Atraso por PDU - 30% de cifração.

O atraso total foi obtido subtraindo o tempo de execução sem segurança do tempo de execução com segurança, ilustrados nas tabelas 5 e 6. Dividindo esse tempo pelo número de PDUs transmitidas, obtemos o atraso inserido em cada PDU.

Vídeo	No. de PDUs	Atraso Total	Atraso/PDU
(4)	1080	10.52s	9.74ms
(5)	2952	29.54s	10.01ms
(6)	4946	48.81s	9.87ms

Tabela 4: Atraso por PDU - 100% de cifração.

Pode-se observar que o impacto do serviço de segurança é considerável, chegando a dobrar o tempo para a transmissão de um vídeo com 100% de cifração em relação à transmissão insegura.

Vídeo	S/ Segurança	C/ Segurança	Acréscimo
(4)	10.69s	13.84s	29%
(5)	29.23s	37.72s	29%
(6)	48.97s	63.41s	29%

Tabela 5: Impacto do serviço de segurança - 30% de cifração.

O tempo gasto apenas com a operação de cifração é em torno de 7ms para uma PDU de vídeo. Comparando esse tempo com o atraso obtido por PDU no caso de 100% de cifração (≈10ms - tabela 4), verificamos que o tempo gasto com cifração é dominante no protocolo.

Vídeo	S/ Segurança	C/ Segurança	Acréscimo
(4)	10.69s	21.21s	98%
(5)	29.23s	58.77s	101%
(6)	48.97s	97.78s	100%

Tabela 6: Impacto do serviço de segurança - 100% de cifração.

Troca de Chave e Ressincronização

Este teste verifica a influência do procedimento de troca de chave e de vetor de inicialização no tempo de execução do servidor. A frequência de chamada do procedimento foi variada em função do número de PDUs enviadas. Para cada vídeo, foram realizados testes considerando 4 casos: invocando o procedimento a cada 350 PDUs (1); a cada 250 PDUs (2); a cada 150 PDUs (3); e a cada 50 PDUs (4). Esses valores foram escolhidos por cobrirem vários casos (frequências baixas e altas) para os vídeos considerados. Cada teste foi executado três vezes.

Vídeo	**Sem Troca**	**350 PDUs**		**250 PDUs**		**150 PDUs**		**50 PDUs**	
		Freq.	**T (s)**	**Freq.**	**T (s)**	**Freq.**	**T (s)**	**Freq.**	**T (s)**
(3)	4.93s	1	4.95s	1	4.97s	2	5.01s	7	5.20s
(4)	13.76s	3	13.83s	4	13.88s	7	13.89s	21	14.35s
(5)	37.77s	8	37.89s	11	37.85s	19	38.32s	59	39.17s

Tabela 7: Frequência da troca de chave e tempo de transmissão.

A tabela 7 mostra os tempos de execução sem troca de chave, as médias dos valores obtidos para cada caso testado e o número de vezes que o procedimento foi disparado. Por exemplo, no vídeo (5), foi obtido o tempo de 37.77s sem troca de chave. Quando o procedimento de troca de chave foi disparado a cada 350 PDUs (8 vezes), o tempo obtido foi 37.89s. Ao disparar o procedimento a cada 50 PDUs (59 vezes), o tempo de execução foi 39.17s. O acréscimo do pior caso (a cada 50 PDUs) em relação à execução sem troca de chave foi apenas de 4%.

Vale ressaltar que os valores da frequência de troca de chave estão superestimados. Na prática, conexões com duração de segundos, minutos ou até mesmo de algumas horas não precisam realizar troca de chave de sessão. Uma avaliação mais precisa da necessidade de trocar chaves de sessão precisa levar em consideração o nível de segurança que a aplicação exige, o algoritmo criptográfico utilizado e o tamanho da chave de sessão.

Observando os dados da tabela 7, verifica-se que o impacto desse procedimento é praticamente nulo. Isso se deve à maneira como esse procedimento é executado. Como o procedimento é realizado em duas etapas, como descrito na seção 2.1, o protocolo não bloqueia a transmissão de dados do usuário entre as duas fases do procedimento.

Como o procedimento de ressincronização é análogo ao procedimento de troca de chave, o resultado se aplica igualmente.

Tráfego Gerado

O protocolo implementado aumenta o tráfego na rede em virtude dos cabeçalhos das PDUs com dados de usuário, das PDUs de troca de chave e de ressincronização e das PDUs de inicialização e de finalização dos parâmetros. O objetivo deste teste é medir o tráfego gerado.

É importante lembrar que a taxa de criptografia não influi no volume de dados. Por isso, os testes foram realizados com uma taxa de cifração fixa. Como os dados medidos são constantes, cada teste foi realizado apenas uma vez. Os resultados obtidos estão ilustrados no gráfico da figura 5. Como o eixo y desse gráfico representa o acréscimo no volume de dados transmitidos, o ponto (0,0) representa o tamanho original de cada vídeo (acréscimo zero).

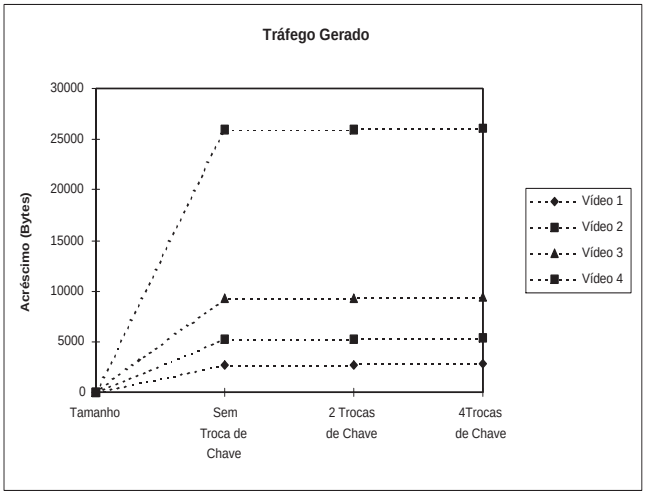

Figura 5: Overhead do protocolo.

Vídeo	Tamanho Original	Acréscimo	
		Absoluto	Percentual
(1)	542.8 KB	2656 Bytes	0.48%
(2)	1.05 MB	5200 Bytes	0.47%
(3)	1.88 MB	9280 Bytes	0.47%
(4)	5.27 MB	25960 Bytes	0.47%

Tabela 8: Overhead do Protocolo.

Os valores obtidos para a execução do protocolo sem troca de chave (representados no gráfico da figura 5) são apresentados na tabela 8 em termos absolutos e percentuais. Por exemplo, para transmitir o vídeo (1), o protocolo gera 2656B, o que corresponde a um acréscimo de 0.48% em relação ao tamanho do vídeo (542.8KB). Esses resultados mostram que o tráfego gerado pelo protocolo não é significativo. O gráfico da figura 5 também mostra que o tráfego gerado pelo procedimento de troca de chave é desprezível.

5.3 Análise de Viabilidade

Os testes apresentados na seção anterior ilustram dados relativos ao protocolo implementado desconsiderando tempo de rede. Esta seção apresenta uma análise do serviço considerando o tempo de transmissão dos dados para o caso de uma rede metropolitana.

O atraso fim-a-fim da aplicação pode ser calculado somando o tempo da aplicação/protocolo, o tempo de transmissão (tempo para pôr a PDU no meio físico - dependente da banda alocada) e o atraso de trânsito (tempo que o primeiro bit gasta para chegar ao destino - dependente da distância entre origem e destino).

Banda Passante	Tempo Total	Taxa de Transmissão
1Mbps	$4946 \times (39ms + 10ms) + 0.66ms \approx 242s$	0.83Mbps
2Mbps	$4946 \times (19ms + 10ms) + 0.66ms \approx 143s$	1.4Mbps
3Mbps	$4946 \times (13ms + 10ms) + 0.66ms \approx 113s$	1.77Mbps
4Mbps	$4946 \times (9.7ms + 10ms) + 0.66ms \approx 97s$	2.06Mbps

Tabela 9: Tempo total - Vídeo (6) - 4946 PDUs - 100% de cifração.

O tempo de transmissão para uma PDU de vídeo foi calculada para os valores 1Mbps (39ms), 2Mbps (19ms), 3Mbps (13ms), 4Mbps (9.7ms) e 5Mbps (7.8ms) de banda passante. Esse cálculo desconsidera o atraso inserido pela função de transmissão devido à limitação do seu buffer. Ou seja, assumimos buffer de capacidade infinita.

Os cálculos são realizados considerando uma MAN com 10 comutadores e distância média de 20km entre os comutadores. Em um caso pessimista, a distância entre origem e destino é em torno de 200km, implicando em 0.66ms de atraso de trânsito. Esses resultados consideram transmissão em fibra ótica à velocidade da luz (3×10^8m/s). O tempo de processamento no interior dos comutadores foi desprezado.

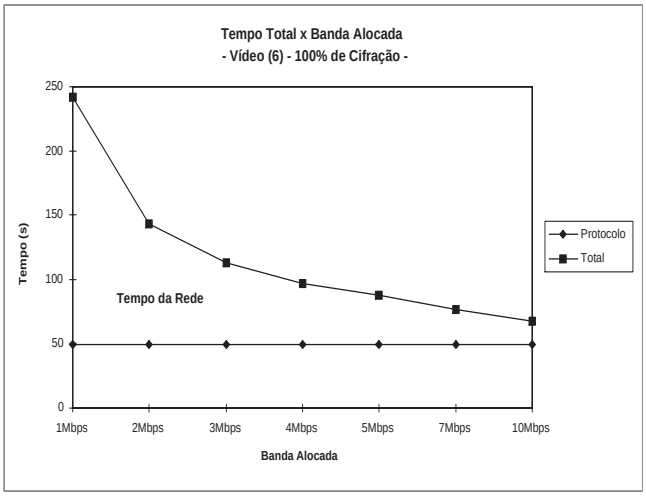

Figura 6: Influência do tempo de rede.

Os tempos da aplicação/protocolo foram obtidos nos testes da seção 5.2. Para 30% de cifração, o atraso obtido por PDU é na faixa de 3ms. No caso de cifração total, o atraso é de 10ms por PDU, aproximadamente.

Com esses dados, é possível estimar o atraso fim-a-fim da aplicação. A tabela 9 apresenta o atraso total e a taxa média de transmissão para cada valor de banda alocada para o vídeo (6) com cifração total.

Para que o vídeo (6) possa ser exibido em tempo real, a taxa de transmissão precisa ser de 1.55Mbps, no mínimo (tabela 1). Com cifração total, essa condição é satisfeita quando a banda alocada é de, no mínimo, 3Mbps (tabela 9). Fazendo os cálculos para 30% de cifração, vimos que 2Mbps de banda já são suficientes para a exibição em tempo real. Esse resultado confirma a influência da taxa de cifração no tempo total de transmissão.

O gráfico da figura 6 ilustra a influência do tempo de transmissão (tempo da rede) no desempenho do serviço. Quanto maior a largura de banda disponível, menor o atraso fim-a-fim. Vale salientar que, ao atingir a taxa média de transmissão necessária para a exibição do vídeo em tempo real, não adianta aumentar o valor da banda, pois os dados não poderão ser consumidos na velocidade em que são recebidos, exigindo da aplicação buffers de recepção maiores. Para o caso de 30% de cifração, o gráfico é equivalente.

5.4 Impacto Visual da Criptografia Parcial

O nível de segurança exigido pela aplicação é um parâmetro decisivo para a escolha da taxa de criptografia para a transmissão. Uma aplicação de vídeo sob demanda, por exemplo, não precisa de altos níveis de segurança, já que o vídeo não possui informação sigilosa. Como a proteção deve ser no sentido de evitar que indivíduos tenham acesso ao serviço de forma ilegal, é suficiente usar mecanismos capazes de tornar incômodo o ato de assistir ao vídeo. Para o caso de vídeo-conferências, a exigência por mecanismos fortes de segurança é grande pois, em geral, envolve dados confidenciais. Nesses casos, os métodos e as taxas de criptografia parcial devem ser analisados cuidadosamente.

Para ilustrar o nível de segurança obtido com criptografia parcial, esta seção traz exemplos de quadros de dois vídeos sem cifração, com 10% e com 30% de cifração. Para a exibição de vídeos parcialmente cifrados, usamos uma modificação do mpeg_player, feito por Thomas Kunkelmann [9].

Observamos que vídeos com muito movimento precisam de taxas de cifração maiores. Isso se deve à maneira como o vídeo é codificado. As figuras 7 e 8 ilustram esse fato. O vídeo flower.mpg (figura 7) não apresenta muita variação de um quadro para outro. Por isso, com 10% de cifração, obtemos um nível de segurança razoável. Já no caso do vídeo

beardE.mpg (figura 8), com 10% de cifração percebemos quadros razoavelmente nítidos. Com 30%, já temos mais dificuldade para reconhecer a imagem.

Figura 7: flower.mpg - sem cifração, 10% e 30% de cifração.

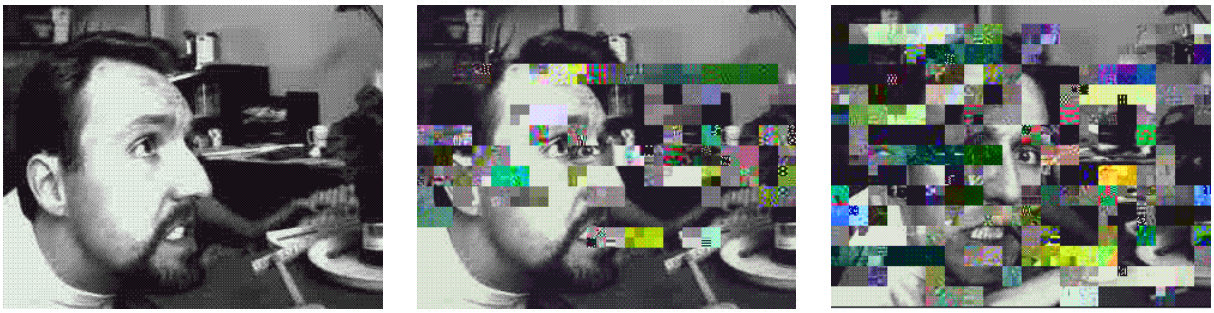

Figura 8: beardE.mpg - sem cifração, 10% e 30% de cifração.

Em [8], trabalho que usa o mecanismo de criptografia seletiva, são apresentados resultados focalizando a questão de desempenho. A métrica utilizada nesses casos foi de quadros por segundo (fps), o que dificultou a comparação de resultados. Agi e Gong [7] analisam variações do esquema de criptografia seletiva. Os resultados desse trabalho mostram que cifrar somente quadros I nem sempre resulta em um nível de segurança forte. Para atingir níveis mais altos, é proposto aumentar a frequência de quadros I no processo de compressão ou cifrar, além de quadros I, blocos I nos quadros P e B. Essas variações são mais seguras, mas há degradação de taxa de compressão (aumenta o tamanho do vídeo) e/ou de desempenho (aumenta a quantidade de dados a serem cifrados). A quantidade de quadros de referência é um parâmetro muito variável nos vídeos. Por isso, não foi possível determinar as taxas de cifração utilizadas nesse trabalho, também comprometendo a comparação de resultados.

Em [9], são apresentados exemplos de vídeos com 1% e 5% de dados cifrados, resultando em quadros com níveis de segurança satisfatórios. Esse trabalho consegue resultados melhores dos que os apresentados aqui no que diz respeito à quantidade de informação cifrada para obter determinado nível de segurança, pois a sua abordagem tira proveito de propriedades dos dados que estão sendo cifrados.

Apesar de apresentarem resultados relativamente bons, esses métodos são de uso restrito. O serviço apresentado neste artigo penaliza a eficiência para manter o serviço de segurança genérico ao realizar a criptografia parcial em dados escolhidos aleatoriamente.

6 Conclusão

A confidencialidade de dados, obtida através do uso de criptografia, é um dos serviços mais importantes para garantir comunicação com sigilo em redes abertas. No entanto,

os algoritmos criptográficos exigem grande esforço computacional e, por isso, tornam-se um problema em redes de alta velocidade. Este trabalho apresentou a especificação, implementação e testes de desempenho de um serviço de confidencialidade dos dados para redes ATM com suporte para criptografia parcial, que visa diminuir o impacto dos algoritmos criptográficos no tempo de transmissão.

Vimos que o esforço necessário para realizar a criptografia parcial cresce linearmente com a taxa de cifração. Também foi observado que o acréscimo imposto pelo protocolo é muito pequeno comparado à complexidade de tempo do algoritmo de cifração.

Observamos que o mecanismo de criptografia parcial utilizado pode não ser tão eficiente quanto aqueles específicos para vídeos MPEG e vídeos codificados com algoritmos baseados em DCT. Esse resultado era esperado, já que a criptografia em dados escolhidos aleatoriamente não tira proveito da sua estrutura. Ainda assim, o ganho obtido usando o serviço com taxas mais elevadas (30%) é significativo em relação à cifração total.

A banda alocada é outro parâmetro que influencia o desempenho do serviço de confidencialidade. Dessa forma, a aplicação precisa considerar o compromisso entre os recursos disponíveis na rede, a taxa efetiva de transmissão e a taxa de cifração. Ou seja, para a aplicação transmitir um vídeo que precisa de uma taxa efetiva de transmissão de 1.55Mbps com 30% de cifração, a rede precisa disponibilizar 2Mbps de banda, no mínimo, para essa conexão. Caso seja necessário cifração total, a rede precisa oferecer no mínimo 3Mbps. Assim, caso a rede possua recursos limitados, é possível utilizar o serviço com taxas de cifração pequenas. Já em redes mais robustas, o serviço pode ser utilizado com níveis mais altos de segurança.

Existem alguns trabalhos em andamento relacionados com segurança em ATM. Como o interesse por essa área é recente, a maior parte dos trabalhos está em fase de especificação ou em fase inicial de implementação. Por isso, não foram encontrados resultados que pudessem ser utilizados para comparação com os dados obtidos neste trabalho.

Algumas idéias de trabalhos futuros são: implementar um protocolo de distribuição de chaves e integrá-lo ao serviço de confidencialidade; implementar um protocolo de negociação de algoritmos do contexto de segurança; implementar métodos de criptografia parcial para determinados tipos de aplicação; e fazer uma análise detalhada do nível de segurança obtido em relação à taxa de cifração para diferentes tipos de vídeos.

Referências

[1] G.C. Sacket e C.Y. Metz. *ATM and Multiprotocol Networking*. McGraw-Hill, first edition, 1996.

[2] P.W. Dowd e J.T. McHenry. Network security: It's time to take it seriously. *IEEE Comp.*, 31(9), 1998.

[3] M. Peyravian e T.D. Tarman. Asynchronous transfer mode security. *IEEE Network*, 11(3), 1997.

[4] D.T. Silva. Arquitetura de um sistema de segurança em redes ATM com gerenciamento de chaves distribuído. Dissertação de mestrado, UFMG, Março 1997. Disp. em *www.sis.dcc.ufmg.br/sis2/apresentacoes.html*.

[5] M. Laurent e P. Rolin. ATM security state of the art. In *Proc. of ATM Development*, France, 1998.

[6] The ATM Forum Technical Committee. ATM Security Specification Version 1.0. STR-SEC-01.02 - Straw Ballot, April 1998.

[7] I. Agi e L. Gong. An empirical study of secure MPEG video transmissions. In *Proc. of the Internet Society Symposium on Network and Distributed System Security*, California, 1996.

[8] Y. Li, Z. Chen, S. Tan, e R. Campbell. Security enhanced MPEG player. In *Proc. of International Workshop on Multimedia Software Development*, Germany, 1996.

[9] T. Kunkelman, H. Vogler, M.-L. Moschgath e L. Wolf. Scalable security mechanisms in transport systems for enhanced multimedia services. In *Proc. of 3rd European Conference on Multimedia Applications, Services and Techniques*, 1998.

[10] B. Schneier. *Applied Cryptography: Protocols, Algoritmhs, and Source Code in C*. 2nd ed., 1996.

[11] T.J. Hudson e E.A. Young. SSLeay and SSLaps FAQ. Disp. em *http://www.psy.uq.oz.au/~ftp/Crypto/*, 1998.

[12] F. F. Pereira. Transmissão de vídeo MPEG com negociação dinâmica de banda passante em redes ATM. Dissertação de mestrado, UFMG, Setembro 1998.