UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Uma Arquitetura para Controle e Proteção de Direitos Autorais de Hiperdocumentos na Internet

por

KLAUS PROKOPETZ

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. José Valdeni de Lima
Orientador

Prof. Dr. Raul Fernando Weber
Co-Orientador

Porto Alegre, dezembro de 1999.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Prokopetz, Klaus

Uma arquitetura para controle e proteção de direitos autorais de hiperdocumentos na Internet / por Klaus Prokopetz. - Porto Alegre: PPGC da UFRGS, 1999.

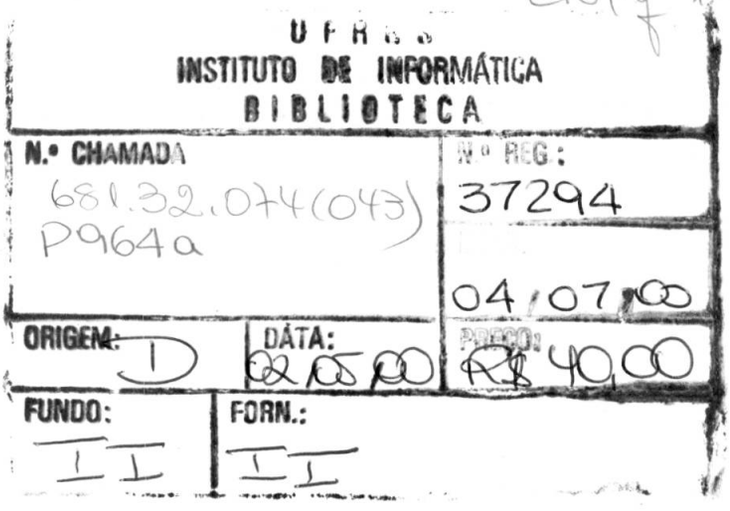
111 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 1999. Orientador: Lima, José Valdeni de. Co-orientador: Weber, Raul Fernando.

1.Hiperdocumento. 2.Direitos Autorais. 3.Esteganografia. 4.*Watermark*. 5.Criptografia. I. Lima, José Valdeni de. II. Weber, Raul Fernando. III Título.

Aplicação dos Computadores - SBU
Automação: Escritórios
Hiperdocumento
Direito autoral: Internet
Marca d'água digital
Criptografia

CDU 1.03.04.00-2



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof[a]. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. Franz Rainer Semmelmann

Diretor do Instituto de Informática: Prof. Philippe Oliver Alexandre Navaux

Coordenadora do PPGC: Prof[a]. Carla Maria Dal Sasso Freitas

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Quero agradecer, compartilhar e dedicar este trabalho a todos aqueles que colaboraram lendo, revisando, questionando, sugerindo, incentivando, ou apenas ouvindo. Agradeço especialmente a:

DEUS, simplesmente pela vida.

CONRAD e ONILDA, meus pais, pelo amor, dedicação e educação; mas principalmente por serem meus pais.

ÉLIDA GRISA, minha amada esposa, pela ternura, carinho, compreensão e paciência inesgotável.

PROF. VALDENI, meu amigo e orientador, pela confiança, atenção, apoio, conselhos e ensinamentos sempre preciosos.

PROF. WEBER, meu co-orientador, pelas explicações, encaminhamentos e, em especial, pela prestatividade com que assumiu minha co-orientação.

RICARDO DILL e JOÃO KRAEMER, meus grandes amigos, pela companheirismo e parceria de tantos anos, que acompanharam com entusiasmo a elaboração deste trabalho.

FÁBIO AUGUSTO DAL CASTEL pelo grande auxílio na implementação do protótipo.

CAPES pelo apoio financeiro.

Obrigado a todos.

Sumário

Lista de Abreviaturas

DCT	*Discrete Cosine Transform*
DES	*Data Encryption Standard*
DVD	*Digital Versatile Disk*
GML	*Generalized Markup Language*
IDEA	*International Data Encryption Algorithm*
ISBN	*International Standard Book Number*
ISSO	*International Standards Organization*
LSB	*Least Significant Bit*
MD5	*Message Digest Algorithm 5*
NBS	*National Bureau of Standards*
NSA	*National Security Agency*
ODA	*Open Document Architecture*
PDF	*Portable Document Format*
PDL	*Page Description Languages*
PS	*PostScript*
RSA	*Rivest-Shamir-Adelman*
SGML	*Standard Generalized Markup Language*
SHA	*Secure Hash Algorithm*
SHS	*Secure Hash Standard*
URL	*Uniform Resource Locator*
WIPO	*World Intellectual Property Organization*
WWW	*World Wide Web*
WYSIWYG	*What You See Is What You Get*

Lista de Figuras

Lista de Tabelas

Resumo

Com o crescimento exponencial da WWW - *World Wide Web*, muitos hiperdocumentos, ou alguns de seus componentes, podem aparecer ilegalmente em algum *site*. O maior impedimento para o uso generalizado da Internet como meio de disseminação de informações tem sido a facilidade de interceptar, copiar e redistribuir hiperdocumentos ou partes destes, exatamente como na sua forma original. Por esta razão, até agora as aplicações na rede têm se destinado, com as devidas exceções, para publicações de documentos gratuitos ou de publicidade comercial ou artística [RUA97].

Devemos considerar que não há e é pouco provável que se obtenha uma maneira absolutamente segura de proteger um hiperdocumento e todos seus componentes do ataque de piratas em um canal inseguro de comunicação como a Internet. No entanto, algumas técnicas podem tornar o hiperdocumento menos vulnerável. Com a certeza de que, dado tempo e recursos necessários, ainda poderá ser pirateado. Todavia, se esta tarefa for onerosa o suficiente a ponto de tornar mais fácil simplesmente adquirir uma cópia legal do hiperdocumento ao invés de pirateá-lo, então podemos considerar que o hiperdocumento estará seguro.

Uma alternativa para dificultar o trabalho dos piratas seria tornar cada cópia do hiperdocumento uma versão única, embutindo algum meio de identificação do autor e do leitor que teve acesso aquela cópia. Assim, se uma cópia ilegal for encontrada, seria possível identificar o leitor que desencadeou, propositadamente ou não, o processo de cópias ilegais e consequentemente rastrear os piratas envolvidos. A idéia é nunca disponibilizar uma cópia desprotegida. O leitor deve ter acesso sempre a cópias marcadas. Estas marcas devem estar embutidas de forma que os piratas não consigam localizá-las, nem retirá-las e, preferencialmente nem desconfiem de sua existência.

Neste enfoque, este trabalho propõe uma arquitetura de controle e proteção de direitos autorais. Esta arquitetura encontra uma solução para o problema da pirataria utilizando as técnicas de criptografia e *watermark*. Para isto, são utilizados dois módulos: um para o autor e outro para o leitor. O primeiro é um aplicativo que a partir da versão original de um hiperdocumento gera uma versão protegida. Esta poderá ser disponibilizada em qualquer *site* da Internet, sem nenhum controle adicional sobre a mesma. O segundo é um *plug-in* para *browser* da Internet, que interpreta a versão protegida, confere a identificação do leitor e, antes de disponibilizar o hiperdocumento, insere neste uma marca de identificação do autor e do leitor. Adicionalmente, o módulo leitor pode comunicar para o autor todas as tentativas de acesso, autorizadas ou não, ao seu hiperdocumento.

A solução utiliza técnicas de criptografia para garantir a segurança do hiperdocumento durante seu armazenamento no *site* do autor, durante sua transferência até o computador do leitor e para identificação do leitor. Depois o controle de acesso e proteção ao hiperdocumento é garantido por técnicas de *watermark*.

Palavras-chave: hiperdocumento, direitos autorais, esteganografia, *watermark*, criptografia.

Title: "An Architecture for Control and Copyright Protection of Hyperdocuments on the Internet"

Abstract

With the fast development of the World Wide Web (WWW), many hyperdocuments - or parts of them - may appear illegally at several sites. The ease with which hyperdocuments can be intercepted, copied and redistributed is the most important obstacle for the use of the Internet as a means to disseminate information. That is why, until now, the WWW has been used mostly (with a few exceptions) for the publication of free documents or for commercial and artistic advertising [RUA97].

Currently, there is no way that is absolutely secure to protect hyperdocuments against hackers – and it is unlikely that there will be one any time soon. However, there are techniques that can make hyperdocuments less vulnerable, even if this protection can also be broken given enough time and effort. Still, if piracy becomes costly enough so that it is easier to simply obtain hyperdocuments by legal means, than we can assume that the hyperdocument is secure.

One option to make piracy harder would be to make each copy of a hyperdocument a unique version, encoding some sort of identification of both the author and the reader having access to that specific copy. This would allow identification of readers who started (deliberately or not) the process of illegally copying a document, and also tracing of the hackers involved. The idea is not to release an unprotected copy ever. Readers must have access to identified copies only. The identification marks should be embedded in such a way that it would not be possibly to either recognize or remove them. Ideally, hyperdocument hackers would not be aware of the existence of such marks.

The present thesis proposes an architecture for control and protection of copyright, utilizing the techniques of cryptography and watermark. For that, two modules are employed, one for the author and another one for the reader. The first module creates a protected version of an original hyperdocument. This version can be published at any Internet site without additional security control. The second is a plug-in module for Internet browsers. It reads the protected version, checks the reader's identification and inserts the author's and the reader's identification mark in the hyperdocument before releasing it. In addition, the second module informs the author of how many attempts (authorized or not) have been made to access the hyperdocument.

The solution presented herein utilizes cryptographic techniques to ensure that a hyperdocument will be secure while stored at the author's site, during download to the reader's computer, and during reader identification. After that, watermark techniques ensure protection and access to the hyperdocument.

Keywords: hyperdocument, copyright, stegnography, watermark, cryptography.

1 Introdução

O aperfeiçoamento e a grande redução de custos das tecnologias de exibição, impressão e armazenamento, tornaram viável o uso do computador para mostrar, imprimir e armazenar versões digitais de documentos das mais diversas mídias. Além disto, novas técnicas de estruturação e acesso a estes documentos digitais potencializaram a eficiência e acurácia da informação representada. Tornou-se possível unir de forma organizada, em um mesmo documento, informações armazenadas em diversos formatos e acessada das mais diversas formas.

Ao mesmo tempo, a tecnologia de comunicação tornou possível a distribuição eletrônica destes novos documentos. Logo que se viu o potencial desta forma de transferência de informações, surgiram as arquiteturas para documentos: SGML - *Standard Generalized Markup Language* e ODA - *Open Document Architecture*. Estas uniformizaram e ditaram padrões de forma a tornar os documentos portáveis e acessáveis da maneira que são atualmente, haja vista a utilização do padrão SGML pelos *browsers* da Internet.

O acesso por rede de computadores, como a Internet, é particularmente interessante, ainda mais se os dados mudam freqüentemente. Porém, contrapondo os benefícios de custos, agilidade e esforço reduzido, a distribuição eletrônica ainda apresenta problemas de pirataria, que também é potencializada pelo formato digital das informações. É muito mais fácil para um indivíduo que obtêm um documento digital repassá-lo para um grupo inestimavelmente grande, do que para outros que recebem cópias físicas do mesmo documento. Além disto, as cópias digitais são mais fidedignas aos originais do que as cópias não-digitais.

A evolução dos documentos físicos para documentos digitais, e depois para modelos estruturados como o hiperdocumento, não é uma surpresa. Isto porque o hiperdocumento é resultado de um desenvolvimento gradual e lógico, onde em cada fase agregou-se um novo aspecto, passando por organização, multimídia, hiperestrutura e acesso. Neste amadurecimento natural parece lógico a grande preocupação atual com os aspectos de proteção aos direitos autorais dos proprietários dos hiperdocumentos.

1.1 Proteção de Direitos Autorais

O direito autoral se origina no momento em que o autor cria uma obra, mediante atividade de espírito pessoal [HAM74]. Isto é, o autor não copia ou executa "mecanicamente" uma tarefa, mas, de seu próprio conhecimento e vontade, cria uma nova obra. Diante deste contexto, fica claro o direito de proteção dos hiperdocumentos pelos institutos reguladores do direito autoral.

O direito autoral está regulamentado internacionalmente pela Convenção Universal para Direito de Autor, ou simplesmente Convenção de Berna, estabelecida em 1886, na Suíça. Como signatários estão quase a totalidade dos países do mundo. Esta

Convenção é atualizada de tempos em tempos pela WIPO - *World Intellectual Property Organization.* A última atualização é o Ato de Paris, ocorrido em 1971.

O Brasil aderiu à Convenção de Berna em 6 de setembro de 1952. Atualmente, o assunto está regulamentado pela Lei 9610/98, a qual confere ao autor as prerrogativas e reivindicações, denominados "direitos morais". Dentre estes estão:

a) direito de reivindicar a paternidade da obra;

b) direito de ter seu nome, pseudônimo ou sinal publicado junto com a obra, como autor;

c) direito de exigir um ressarcimento nos casos em que, por força de lei não possa impedir a utilização por terceiros;

d) direito de reivindicar a cessação da violação, a remoção dos objetos contrafeitos, informações, ressarcimento de dano e apreensão de rendimentos.

Da mesma forma, a Lei estabelece os casos em que as obras caem em domínio público:

a) que tiverem seu prazo de proteção esgotado;

b) que o autor faleceu sem deixar herdeiros;

c) que o autor é desconhecido, caso em que a obra é transmitida pela tradição cultural;

d) que tenha sido publicada em países que não participem de tratados a que tenha aderido o Brasil, e que não confiram aos autores de obras aqui publicadas o mesmo tratamento que dispensam aos autores sob sua jurisdição.

Por força da Convenção de Berna, o direito de autor de obras publicadas no Brasil está assegurado igualmente em todos os demais países contratantes, nos seguintes termos:

> *"Todo o nacional de um Estado contratante e todo aquele que publicar a sua obra primeiramente num destes Estados pode obter em todos os demais Estados contratantes os mesmos direitos para suas obras publicadas, que neste são concedidos aos seus nacionais"*[1].

Um caso diferenciado é o dos Estados Unidos, que se manteve fora da Convenção de Berna durante mais de um século. Neste período, editou uma legislação

---

[1] Artigo 1°, Parágrafo Único da Lei 7646 de 18 de Dezembro de 1987.

própria para regulamentar o direito de autor em seu Estado. Ao aderir à Convenção, o Congresso norte-americano estabeleceu que nenhuma mudança seria feita na sua legislação, já que o conjunto de dispositivos existentes fornecia adequada proteção aos direitos autorais. Por esta razão, sob certos aspectos, sua legislação é muito mais formal do que a legislação dos demais países signatários. Todavia estas formalidades são válidas apenas aos nacionais de seu Estado, e são consideradas satisfeitas aos nacionais de outros Estados, como é o caso do Brasil, se:

> *"...desde a primeira publicação, todos os exemplares da obra publicada com a autorização do autor ou de qualquer outro titular dos seus direitos contiverem o símbolo © acompanhado do nome do titular do direito de autor e da indicação do ano da primeira publicação..."*[2].

No Brasil, optou-se por proteger os direitos de autor sem a necessidade de qualquer formalidade. Nestes termos, a doutrina e os Tribunais têm se manifestado da seguinte forma [HAM74]:

a) o registro é irrelevante para a gênese do direito de autor, é uma simples medida de segurança;

b) o registro facilita a prova;

c) o registro fornece a melhor prova, mas nada mais que uma presunção. Por isto o autor anda bem aconselhado se fizer o registro da obra;

d) a omissão do registro não prejudica a proteção, podendo o direito ser provado de outra forma;

e) o registro não tem força probante absoluta, apenas transfere o ônus da prova e pode ser desvirtuado pela prova em contrário.

Assim, a publicação da obra, com seu autor devidamente identificado, até prova em contrário, é suficiente para presumir a propriedade da obra. Isto é, transfere-se o ônus da prova a quem queira reclamar a propriedade da mesma obra. A este basta apresentar como prova em contrário, uma publicação anterior com seu nome. A Lei considera como publicação "*a comunicação da obra ao público, por qualquer forma ou processo*[3]". Portanto, podemos incluir a Internet dentre as possibilidades de publicação. Resta então apenas a necessidade de provar uma ordem cronológica para as publicações.

No caso específico dos programas de computador, também protegidos por direito de autor, o Brasil editou recentemente a Lei 9609/98, específica para regulamentar o assunto. Anteriormente, era regulado pela Lei 7646/87, ora revogada. A nova lei manteve o mesmo texto do dispositivo anterior para a definição de programa de computador:

[2,3] Artigo 4° Inciso I da Lei 5988 de 14 de Dezembro de 1973.

> *"Programa de computador é a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital, para fazê-los funcionar de modo e para fins determinados"*[3].

Esta definição é imprópria e pode gerar controvérsias ao classificarmos determinados casos, como o hiperdocumento. Pois este, além de seu conteúdo, é composto por uma estrutura lógica e uma estrutura de apresentação. Estas estruturas por si próprias são instruções codificadas, necessárias para tratar a informação contida no conteúdo do documento, e destinam-se a manusear o documento adequadamente. Todavia, não parece ter sido a intenção do legislador incluir casos como dos hiperdocumentos e outros semelhantes nos regulamentos de proteção aos programas de computador. Assim, nos casos de lide que verse sobre o assunto, caberá ao poder judiciário solicitar auxílio aos profissionais de informática como peritos no assunto, à semelhança do que já é feito para outras áreas.

Quanto à extinção do direito de autor, a Lei 5988/73 protege o autor por 60 anos após a morte do mesmo, ou com a morte dos filhos, pais ou cônjuge, quando a estes couber a sucessão. No caso de programas de computador, a Lei 9609/98 concede ao autor o prazo de 50 anos para explorar sua criação. Anteriormente a Lei 7646/87 concedia o prazo de 25 anos.

Como foi visto, os direitos autorais dos hiperdocumentos divulgados através da Internet estão protegidos no Brasil e nos países que mantêm a citada reciprocidade de tratamento legal, desde que contenha o *"símbolo © acompanhado do nome do titular do direito de autor e da indicação do ano da primeira publicação..."*[4]. Resta apenas, em caso de discussão jurídica, provar a paternidade do hiperdocumento, bem como sua data de publicação.

1.2 Objetivos

Atualmente os hiperdocumentos estão sujeitos aos seguintes ataques:

a) interceptação: piratas atuam em canais inseguros de transmissão e capturam o hiperdocumento;

b) pesquisa: piratas acessam locais inseguros de armazenamento e copiam os hiperdocumentos;

---

[3] Artigo 1°, Parágrafo Único da Lei 7646 de 18 de Dezembro de 1987.
[4] Artigo 4° Inciso I da Lei 5988 de 14 de Dezembro de 1973.

c) vazamento: usuários autorizados acessam os hiperdocumentos e os repassam para piratas;

d) mascaragem: piratas obtêm, por meio ilícito, identificação de usuários autorizados e acessam os hiperdocumentos utilizando esta identificação.

A princípio se poderia pensar que o simples uso de sistemas de criptografia convencionais resolveria os problemas de segurança, mas na verdade estes sistemas oferecem uma proteção muito pequena contra a pirataria. Pois, uma vez que estes documentos estejam decodificados, não há maneira de evitar ou marcar sua reprodução ou retransmissão ilícita [COX95].

Da mesma forma, simplesmente inserir uma marca identificando o autor, não seria suficiente para inibir a pirataria do hiperdocumento. Nem tão pouco tornaria possível isolar as fontes de origem das piratarias. Seria necessário que a marca contivesse a identificação tanto do autor como do leitor. Todavia, isto implicaria em individualizar cada cópia do hiperdocumento antes de enviá-la para o usuário. O que acarretaria uma sobrecarga no provedor do hiperdocumento, pois a tarefa de embutir uma marca requer certo nível de processamento. Mesmo assim, as técnicas atuais não utilizam a marca como método de segurança, mas apenas como meio de identificação, permitindo que qualquer um tenha acesso ao hiperdocumento.

Diante da situação era necessário encontrar uma solução efetiva que permitisse autores, editores e provedores de informação divulgar os seus hiperdocumentos na Internet com certo grau de segurança física e de garantia de direitos autorais, mas que também não alterasse em nada a relação dos autores e leitores com seus provedores de acesso. Ou seja, a solução não deveria requerer nenhum controle adicional pelos provedores de acesso, tais como protocolos seguros, códigos e senhas. Em outras palavras, sua utilização deveria ser transparente, sem implicar em qualquer impacto.

Neste contexto foi concebido o presente trabalho. O objetivo específico da pesquisa é propor mecanismos para controle de proteção de direitos autorais de hiperdocumentos, para uma publicação eletrônica segura na Internet.

1.3 Estrutura do Trabalho

No intuito de tornar claro o processo evolutivo que culminou neste trabalho, serão apresentados, nos próximos capítulos, todos os embasamentos e passos percorridos.

Inicialmente serão vistos três capítulos de contextualização, que abordam os temas Hiperdocumento, Criptografia e Esteganografia. Estes capítulos pretendem apenas consolidar conceitos e distinguir técnicas que fundamentarão a solução proposta por este trabalho.

O capítulo de hiperdocumentos fundamenta o objeto de estudo deste trabalho, com ênfase para suas arquiteturas. A arquitetura do SGML merece destaque, pois ainda é o padrão da Internet, ambiente onde se pretende proteger os hiperdocumentos da ação de piratas.

No terceiro capítulo, serão revistos os principais algoritmos de criptografia, tanto simétrico como assimétrico, bem como a técnica de assinatura digital. A ciência da criptografia oferece estes recursos para uma comunicação segura feita através de um canal inseguro, como a Internet. O presente trabalho utiliza estes recursos para efetuar a comunicação entre os diversos elementos que compõem a arquitetura proposta.

Como será demonstrado no transcorrer do texto, através da integração de técnicas *watermark* e assinatura digital será possível identificar autores e leitores dos hiperdocumentos, e por consequencia, as fontes de pirataria. Por isto, o terceiro capítulo contextualiza a esteganografia, ciência que estuda técnicas de comunicação secreta, dando ênfase para a técnica de *watermark* aplicada para cada tipo de mídia: imagem, texto e som.

Em seguida, o capítulo de estado da arte apresenta as principais ferramentas disponíveis na rede, que possuem algum mecanismo para proteção dos direitos de autores e distribuição de documentos digitais na Internet.

Os capítulos 6 e 7 são o centro deste trabalho. No primeiro é detalhada toda arquitetura proposta e seu funcionamento para controlar e proteger os hiperdocumentos na Internet. No segundo, é demonstrada a arquitetura do protótipo, com os detalhes de implementação de cada módulo.

Por fim, seguem-se as conclusões obtidas, enumeram-se as contribuições e sugerem-se possíveis continuações e amadurecimentos em trabalhos futuros.

2 Hiperdocumento

Neste capítulo são revistos conceitos básicos sobre os hiperdocumentos. Inicialmente são revisados os termos utilizados no cotidiano e depois uma breve abordagem dos padrões internacionais: o ODA e o SGML. Para maiores subsídios o assunto, o leitor é convidado a verificar [MAN94], [CHO94], [ISO86], [ISO89] e [ISO93], principais fontes que basearam a redação deste capítulo.

2.1 Terminologia

Hiperdocumento pode ser considerado um conjunto de documentos eletrônicos de qualquer tipo de conteúdo, que se relacionam por qualquer tipo de *link*, organizados sob uma multiestrutura.

O termo hiperdocumento e outros vinculados evoluíram conforme a tecnologia passou a tratar novas mídias, permitindo atender novas necessidades. Na figura 2.1, que demonstra esta evolução, temos a origem dos termos.

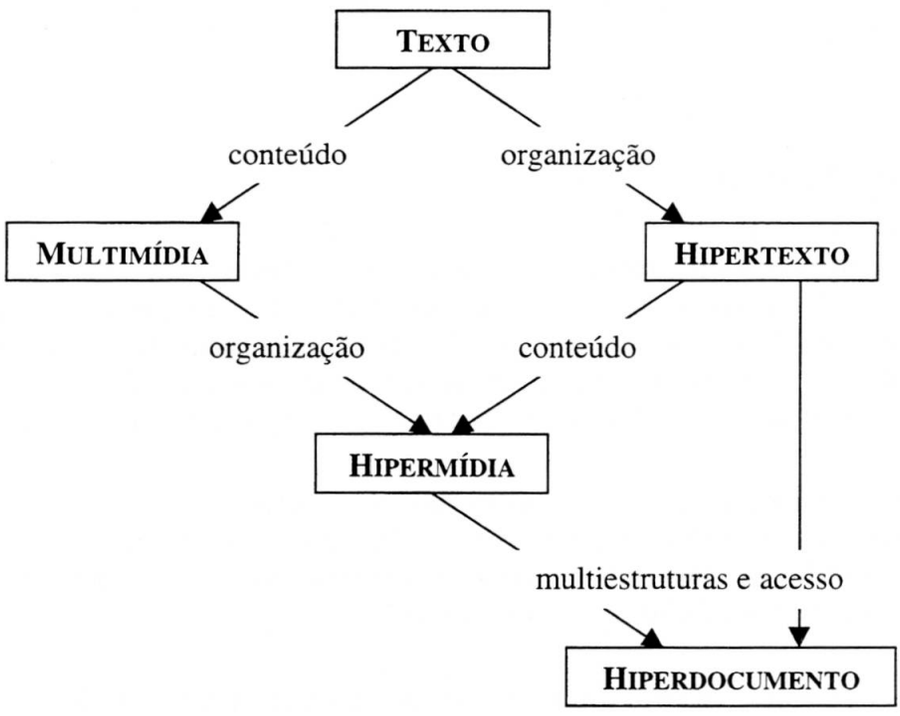


FIGURA 2.1 – Origem dos Termos [LIM 95]

Assim, historicamente os termos vinculados a hiperdocumentos possuíram os seguintes significados:

Documento Eletrônico: é um documento que existe em forma digital dentro de um sistema de computador. Inicialmente, por restrições tecnológicas, o termo

estava vinculado apenas a conteúdo textual. Porém, atualmente devemos considerar qualquer tipo de mídia. Assim, o termo pode ser aplicado para denominar isoladamente tanto um texto, como uma foto, um vídeo ou um som digitalizado;

Hipertexto: representa a adição do aspecto organização de texto a um documento eletrônico, resultante da hiperestrutura proveniente das referências internas e externas que existem normalmente dentro de um texto;

Multimídia: como o próprio nome sugere, se referencia à expansão do conteúdo, que era apenas textual, para adicionar também dados tais como som, imagem e vídeo;

Hipermídia: representa tanto um documento multimídia acrescido de aspectos de organização, como um documento hipertexto acrescido de aspectos multimídia;

Hiperdocumento: é um objeto complexo composto de conteúdo (texto, som imagem), organização (hiper e multiestruturas) e acesso (interativo, cooperativo e distribuído).

Link: denomina as referências internas e externas entre os documentos eletrônicos.

2.2 Arquiteturas

Documentos eletrônicos podem conter informações descrevendo o seu conteúdo ao invés de descrever a formatação da informação. Por exemplo, para conteúdos textuais, em vez de marcar um cabeçalho de seção como negrito, fonte grande, seguido por um espaço vertical de 5mm, é declarado como um cabeçalho de nível 2. A formatação deste nível é conhecida pela ferramenta que apresentará o documento.

Desta forma, o mesmo documento pode ser apresentado de diferentes maneiras, dependendo das preferências do usuário e do dispositivo de saída. Este formato estrutural também encoraja consistência e permite processamento automático, como por exemplo a geração de tabelas de conteúdo ou índices.

Os diferentes formatos estruturais dos documentos eletrônicos são denominados de arquiteturas. ODA e SGML são dois importantes padrões que definem arquiteturas de documentos. Em geral, podemos dizer que o padrão SGML foi desenvolvido para o ambiente de publicação, especificamente a Internet. Enquanto que o padrão ODA foi elaborado para o ambiente de escritório, prevendo a intercâmbio de documentos eletrônicos entre diferentes plataformas e softwares.

2.2.1 ODA - *Open Document Architecture*

ODA é um padrão internacional que define uma arquitetura aberta para documentos eletrônicos. É um padrão auto-contido, elaborado para ser usado na troca de documentos para edição, processamento, armazenamento, impressão e retransmissão entre diferentes sistemas.

Com este padrão, tornou-se possível trocar informações, de uma forma organizada entre diferentes tipos de sistemas de processamento de informações, independente de sua plataforma. Isto permite que sistemas proprietários de processamento de texto e publicação trabalhem juntos, compartilhando um modelo comum de informação.

2.2.1.1 Elementos

A figura 2.1 ilustra os elementos de um documento ODA. Nota-se que há elementos genéricos e específicos. Os primeiros dizem respeito à definição de um documento, e os segundos à instância de cada documento gerado.

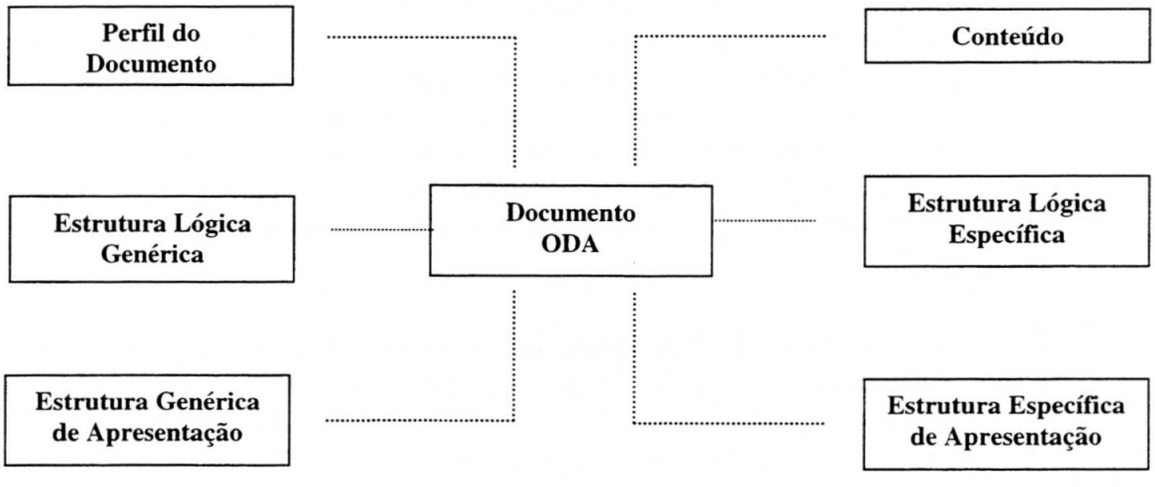


FIGURA 2.2 – Elementos de um Documento ODA

Perfil do Documento: inclui atributos referentes ao documento como um todo. Ele inclui informações gerenciais, tais como título, nome do autor, data de criação, palavras chave e outros. Este tipo de informação pode ser obtido e recuperado automaticamente pelo sistema. Também são incluídas algumas informações técnicas, tais como se o documento tem uma estrutura lógica ou de apresentação, ou ambas;

Estrutura Lógica Genérica: são grupos de regras no próprio documento que podem ser consideradas de definição da classe do documento, como um padrão. Por exemplo, poderia existir uma classe de documentos chamados relatórios. Um relatório poderia ter a seguinte organização: um índice, um cabeçalho de

capítulo no topo da página, e seções com um esquema de numeração. O modelo ODA permite embutir este tipo de informação no próprio documento como estrutura lógica e de apresentação. Esta informação pode ser usada pelo sistema de edição para assegurar que o documento esteja sempre ajustado para sua própria classe;

Estrutura Genérica de Apresentação: é como o documento aparece na página. Esta é determinada por um processo de formatação baseado nas diretivas da estrutura. Normalmente, o processo de criação de uma estrutura de apresentação é chamado de formatação;

Conteúdo: é a informação compartilhada pelas estruturas de apresentação e lógica. Pode consistir de textos, tabelas, equações, imagens, vídeos, sons ou outra forma de representação da informação. O conteúdo de um documento poderia ser visto como uma informação que primeiramente representa marcas perceptíveis pelo ser humano, usados para transmitir significado;

Estrutura Lógica Específica: é a organização de um documento, a maneira que o autor estrutura a informação sem considerar a forma de apresentação. Tipicamente as partes da estrutura lógica são capítulos e seções. Todavia, muitas partes pequenas são possíveis, como por exemplo uma nota de rodapé, uma referência para outra parte do documento e uma citação bibliográfica. As partes de um documento não são isoladas, mas são relacionadas pela sua estrutura, por exemplo, um capítulo é composto de uma seqüência de seções, uma figura é composta de um imagem e uma legenda, e um cabeçalho de seção é composto pelo número da seção e o título. Algumas partes de um documento, tais como índices e bibliografias, podem ser especificados pelo autor, ou então gerados automaticamente pelo sistema para garantir relacionamento entre as partes do documento;

Estrutura Específica de Apresentação: representa a instância de documentos, a individualização e aplicação de uma estrutura genérica para um documento específico. Esta estrutura especificará quais elementos da estrutura genérica aplicam-se ao caso específico de cada documento.

2.2.1.2 Tipos

A arquitetura ODA também definiu três tipos de documentos, os quais são resultado da combinação dos elementos de estrutura lógica, estrutura de apresentação e conteúdo:

Documentos Processáveis: contêm estrutura lógica e conteúdo. São adequados para uso pelas aplicações que necessitam editar o documento, tais como processadores de textos;

Documentos de Apresentação: contêm estrutura de apresentação e conteúdo. São usados pelas aplicações de impressão ou visualização para disponibilizar o conteúdo quando a edição não é necessária;

Documentos Processáveis e de Apresentação: contêm tanto estrutura lógica, como estrutura de apresentação. Podem ser usados por qualquer aplicação, visualização, impressão ou edição.

2.2.2 SGML - *Standard Generalized Markup Language*

SGML é um padrão internacional que define uma linguagem e notação para descrever classes de documentos. Atualmente, SGML é muito mais popular do que ODA, graças ao HTML, uma especialização do SGML, que é adotado pelos *browsers* WWW.

SGML utiliza anotações ou marcas, chamadas de *markups*, no meio do conteúdo para regular a sua formatação, impressão e outros processamentos. As marcas de SGML são usadas para realizar uma interpretação de um texto explícito, separando seus elementos lógicos, como título, capítulo, parágrafo e seção.

O SGML precisa especificar quais marcas são permitidas e onde elas são possíveis, bem como quando uma marca é necessária, e como ela é distinguida do conteúdo do documento. Para isto, as marcas SGML são intercaladas por *tags*. Cada elemento lógico no documento é precedido por um *star-tag* e seguido por um *end-tag*.

Tags são identificadas por seus nomes. Além disto, *tags* identificam elementos lógicos e referências de entidade internas ou externas. Uma referência de entidade é um tipo de macro para incluir um conteúdo - a entidade - em um documento exatamente no lugar da sua referência. Esta entidade pode ser uma tabela, uma figura ou outro elemento qualquer. Os elementos lógicos válidos e as *tags* que devem identificá-los são descritos em um DTD - *Document Type Definition*. Assim, cada classe de documento, como por exemplo um relatório técnico ou um memorando, tem seu próprio DTD. Em outras palavras, o DTD define a gramática que indica quais os tipos de marcas permitidas e requeridas para a classe de documento que está sendo definida. O SGML propriamente não especifica nenhuma classe em particular.

2.2.2.1 Elementos

Diferentemente do padrão ODA, um documento SGML possui apenas as partes de estrutura lógica e conteúdo, omitindo a estrutura de apresentação. Esta fica a cargo das implementações das ferramentas próprias para apresentação de documentos SGML, como por exemplo, os *browsers* WWW. Assim, conforme ilustra a figura 2.3, um documento SGML possui apenas três elementos:

Declaração SGML: define quais caracteres são usados no DTD, no texto do documento e outras características especiais do SGML usadas no documento;

DTD: define a estrutura lógica do documento;

Instância do Documento: contém o texto do documento, incluindo referência para o DTD usado.

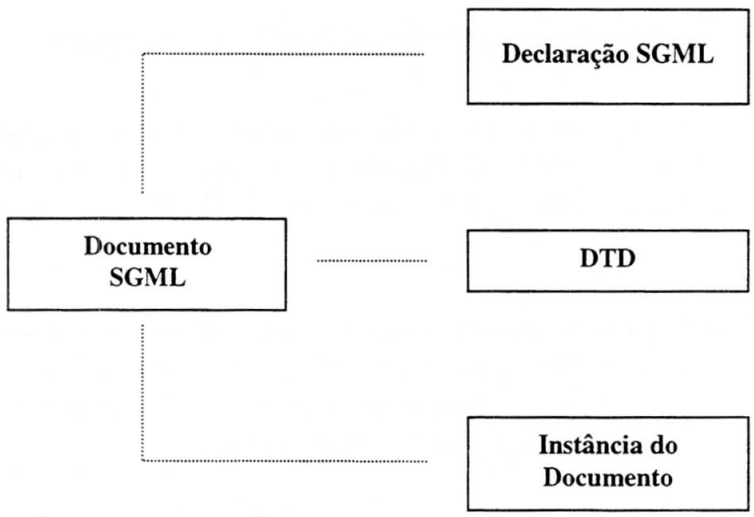


FIGURA 2.3 – Elementos de um Documento SGML [MAN 94]

Atualmente, com o XML, padrão mais recente do SGML, existe uma tendência em também separar a instância do documento em estrutura lógica e estrutura de apresentação.

3 Criptografia

Este capítulo pretende apenas rever conceitos que fundamentarão a solução proposta ao final deste trabalho. A redação deste capítulo baseou-se principalmente em [ARR93] e [WEB98]. Para um estudo mais aprofundado sobre assunto, o leitor é convidado a visitar [RSA 99]. Inicialmente são revistos os termos ligados à criptografia. Depois são apresentados os principais algoritmos de criptografia, tanto assimétricos como simétricos, e as funções *hash*. Por fim, são esboçadas as técnicas de assinatura digital, um dos pilares da solução proposta.

3.1 Visão Geral

A criptografia consiste no estudo de transformações matemáticas reversíveis, utilizadas para proteger informações armazenadas em local inseguro, ou em trânsito em um canal inseguro, tornando as informações ilegíveis a possíveis intrusos e garantindo a legitimidade das partes envolvidas.

A figura 3.1 demonstra um modelo padrão de criptosistema. Como pode ser observado, no centro do modelo há um canal ou local inseguro através do qual se quer transmitir ou armazenar uma informação. Inicialmente o texto claro é submetido a um processo de cifragem que, com base na chave de cifragem, gera um texto cifrado. Este texto, após ser transmitido ao seu receptor, sofre um processo inverso para decifrar o texto cifrado, que com a correta chave para decifragem, gera o texto claro original. As setas indicam uma comunicação unilateral entre os elementos.

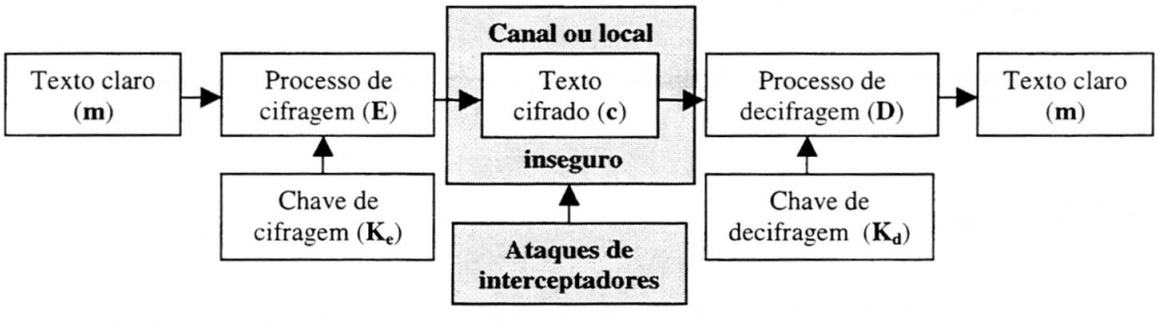


FIGURA 3.1 – Modelo de Criptosistema

As transformações matemáticas efetuadas nos processos de cifragem e decifragem consistem, respectivamente, em:

$$c = E(m, K_e) \tag{3.1}$$

$$m = D(c, K_d) \rightarrow m = D(E(m, K_e), K_d) \tag{3.2}$$

3.2 Terminologia

O estudo da criptografia utiliza a seguinte convenção de termos:

Texto claro: também chamado de texto normal, texto original ou texto compreensível; é a informação original que será criptografada. Esta pode ser de qualquer tipo e formato.

Chave(s): código secreto utilizado como parâmetro nas transformações matemáticas, ocorridas durante os processos para cifrar e decifrar a informação.

Texto cifrado: também chamado de criptograma, é a informação na sua forma ilegível, após passar pelo processo de cifragem.

Cifragem: também chamado de codificar ou criptografar; é o processo que recebe como entrada o texto claro e a chave para cifrar, e gera o texto cifrado.

Decifragem: também chamado de decodificar ou decriptografar; é o processo que recupera o texto claro a partir do texto cifrado e da chave para decifrar.

Criptoanálise: é a arte de 'quebrar' um criptograma. Isto é, recuperar a informação criptografada sem possuir a devida chave.

Criptógrafo: é a pessoa que criptografa a informação.

Criptoanalista: especialista em criptoanálise. Sua tarefa é determinar um método que possa 'quebrar' a codificação.

Interceptador: também chamado de atacante; é um criptoanalista não autorizado.

Ataque: tentativa de criptoanálise feita pelo criptoanalista ou interceptador.

3.3 Algoritmos Básicos

Os primeiros criptosistemas pressupunham, como garantia de segurança de seus métodos, o sigilo do próprio algoritmo utilizado nos processos para cifrar e decifrar. Tais sistemas são apenas de interesse histórico e não atendem as necessidades atuais.

Todos os métodos modernos utilizam chaves para cifrar e decifrar, que devem ser mantidas em segredo. Assim, ainda que os algoritmos sejam públicos ou conhecidos, um criptograma somente será decifrado se for informada a chave correta para o processo.

Existem duas classes de algoritmos baseadas em chaves: os simétricos e os assimétricos. Os primeiros caracterizam-se por utilizar chaves idênticas para cifrar e decifrar. Os outros, por sua vez, utilizam chaves diferentes para cada um destes processos.

Obviamente, é pressuposto intrínseco dos algoritmos simétricos que exista um meio de comunicação seguro, pelo qual o emissor possa transmitir ao receptor a chave secreta do criptograma. Para os algoritmos assimétricos não há esta necessidade. Cada indivíduo possui duas chaves distintas: uma chave pública (K_e) e uma chave secreta (K_d), que serão utilizadas em todos os processos de comunicação criptografada. Estas chaves estão relacionadas entre si e servem para especificar transformações inversas, sendo computacionalmente impraticável deduzir K_d a partir de K_e.

Os métodos modernos de criptografia são impossíveis de serem executados por humanos. Alguns métodos mais robustos são até computacionalmente caros de serem processados. Os métodos atuais são tão seguros que nem mesmo o seu autor é capaz de decifrar o criptograma sem dispor da chave apropriada.

3.3.1 Algoritmos Simétricos

Como já foi dito, os algoritmos simétricos caracterizam-se por utilizar a mesma chave para cifrar e decifrar. Por este motivo, tais algoritmos também são denominados de sistemas de criptografia de chave única.

Estes métodos clássicos utilizam funções elementares, como deslocamentos e substituições, para gerar difusão e confusão na informação cifrada. Os exemplos mais conhecidos de cifradores computacionais simétricos são o DES - *Data Encryption Standard* e o IDEA.

3.3.1.1 Algoritmo DES - *Data Encryption Standard*

O DES é uma modificação desenvolvida pela IBM nos anos 70 a partir do sistema LUCIFER, também da mesma empresa. Em janeiro de 1977, o DES foi adotado como padrão pelo NBS - *National Bureau of Standards* americano para departamentos e agências federais.

O DES opera em bloco de 64 bits, com uma chave de 56 bits, expandida artificialmente para obter uma chave de 64 bits, sendo o último bit de cada byte utilizado para paridade. O algoritmo é o mesmo para cifrar e decifrar.

Alguns cientistas têm alertado que a chave do DES é muito pequena, sendo passível de ataques do tipo força-bruta. O tamanho da chave projetada originalmente foi de 128 bits, o qual ainda hoje, inquestionavelmente, eliminaria qualquer chance de ataque por força-bruta. Todavia, na ocasião o NSA - *National Security Agency* americano solicitou que o tamanho da chave fosse reduzido para o tamanho atual. Os motivos para isto nunca foram divulgados.

Comenta-se que atualmente os equipamentos do NSA americano podem 'quebrar' um criptograma DES em três a quinze minutos. Todavia, o DES ainda pode ser considerado robusto suficiente para garantir contra ataques casuais de

interceptadores comuns. Uma variação mais segura deste algoritmo, o Triple-DES ou 3DES, já foi desenvolvida, mas ainda é pouco difundida.

3.3.1.2 Algoritmo IDEA - *International Data Encryption Algorithm*

O algoritmo IDEA foi desenvolvido em 1992, na Suíça. Atualmente é o algoritmo simétrico de domínio público mais robusto. Opera em blocos de 64 bits com uma chave de 128 bits, característica dos algoritmos mais seguros. O mesmo algoritmo básico é utilizado para cifragem e decifragem. Ainda é um algoritmo recente e até o momento todas as tentativas de criptoanálise fracassaram.

O IDEA é patenteado nos Estados Unidos e em muitos países Europeus. Todavia, o seu uso não comercial é livre.

3.3.2 Algoritmos Assimétricos

Os algoritmos assimétricos utilizam duas chaves distintas, uma para cifrar e outra para decifrar. A primeira é divulgada abertamente, por este motivo denominada de chave pública. A segunda, denominada de chave secreta, ou chave privada, é mantida em segredo pelo seu proprietário. Assim, qualquer um pode utilizar a chave pública para cifrar uma informação, mas apenas aquele que conhecer a respectiva chave privada poderá decifrar o criptograma.

A abordagem de chaves assimétricas, tal como apresentada, foi introduzida em 1976. Até então, aceitava-se que as chaves para cifragem e para decifragem, quer fossem iguais ou diferentes, deveriam ambas ser mantidas secretas. A questão era como distribuir tais chaves por canais inseguros [WEB98].

Os criptosistemas de chaves assimétricas eliminaram o problema da distribuição de chaves, pois as chaves públicas podem circular livremente, uma vez que servem apenas para cifrar as informações. Logo quem a utilizá-la já tem acesso à informação que se quer proteger. Além disto, estes criptosistemas tornam possível a implementação da idéia de assinaturas digitais, versões eletrônicas das assinaturas manuscritas (vide item 3.4, a seguir). Todavia, as vantagens das chaves assimétricas somente são possíveis se satisfeitos os seguintes requisitos:

a) se $c = E\ (\ m,\ K_e)$, então $m = D\ (\ c,\ K_d\ )$, para todos T;

b) é computacionalmente impossível deduzir K_d a partir de K_e;

c) é computacionalmente possível calcular um par K_d e K_e que satisfaça os dois requisitos acima.

Assim, supondo que Alice queira enviar uma informação para Bob, sendo que estes nunca se encontraram, nem se comunicaram antes. Alice somente necessita cifrar sua informação com a chave pública de Bob, obtida em um local qualquer:

$$c = E(m, K_{eBob}) \tag{3.3}$$

Para decifrar o criptograma, Bob simplesmente inverte a transformação, utilizando sua chave privada:

$$m = D(c, K_{dBob}) \tag{3.4}$$

Como a chave K_{dBob} é secreta, ninguém mais consegue decifrar a informação, a não ser o próprio Bob.

Geralmente, os algoritmos simétricos são muito mais rápidos para serem executados em um computador do que os algoritmos assimétricos. Isto fez com que na prática os dois algoritmos fossem utilizados em conjunto. Primeiro, gera-se randomicamente uma chave simétrica, utilizada para criptografar a informação, e com a chave pública criptografa-se a chave simétrica. Assim, utiliza-se o algoritmo assimétrico apenas para distribuir a chave simétrica usada para cifrar uma informação.

Atualmente, o RSA é o algoritmo assimétrico mais conhecido e mais robusto.

3.3.2.1 Algoritmo RSA - *Rivest-Shamir-Adelman*

O RSA, assim denominado pelos seus autores Ron **R**ivest, Adi **S**hamir e Leonard **A**delman, baseia-se nos princípios de que é relativamente fácil encontrar números primos grandes (considera-se assim aqueles números entre 100 a 200 dígitos decimais), enquanto que é computacionalmente inviável fatorar o produto destes, o que na prática seria o melhor caminho para 'quebrar' o criptosistema.

Vale ressaltar que os métodos de fatoração e o poder computacional estão evoluindo rapidamente. Por exemplo, em 1980 era possível fatorar números de até 60 dígitos. Já em 1995, chegou-se a fatorar números de 129 dígitos e em 1996 um de 130 dígitos [HEA99]. Todavia, mesmo que o poder computacional atual aumente um milhão de vezes, tal fatoração ainda consumiria milhões de anos. Mesmo assim, caso seja necessário aumentar o nível de segurança, basta acrescentar alguns dígitos aos números primos envolvidos. [WEB98]. O RSA Data Security recomenda que se utilize, no mínimo, 230 dígitos.

O RSA é considerado seguro quando usado apropriadamente, com chaves suficientemente longas (512 bits é inseguro, 768 bits é modernamente seguro, 1024 é seguro e 2048 bits deve ser seguro por décadas) [SSH99].

Alguns especialistas consideram que o RSA é muito vulnerável a ataques do tipo texto escolhido. Há ainda o novo *timing attack*, que aparentemente pode ser utilizado para quebrar muitas implementações de RSA.

O algoritmo está patenteado apenas nos Estados Unidos (seu registro expira no ano 2000) e é livre em qualquer outro lugar.

3.4 Funções *Hash*

Funções *hash* calculam, a partir de uma grande massa de dados, um valor de tamanho especificado. Originalmente este valor era utilizado para distribuir os dados por toda a área de armazenamento de um banco de dados, assim este usaria todos os recursos efetivamente disponíveis e de forma equilibrada. Matematicamente temos:

$$h = H(m) \tag{3.5}$$

Para criptografia, as funções *hash* devem ainda garantir que:

a) seja rápido e fácil calcular o valor *hash*;

b) seja inviável produzir uma massa de dados que origine um determinado valor *hash*;

c) que seja infinitamente impossível duas massas de dados diferentes produzirem o mesmo valor.

Tipicamente, estas funções produzem valores de 128 bits ou mais. Atualmente este número é grande o suficiente para garantir a maioria dos requisitos da criptografia. Muitos algoritmos de funções *hash* de qualidade estão disponíveis. Dentre os mais conhecidos estão o MD5 - *Message Digest Algorithm 5* e o SHA - *Secure Hash Algorithm*.

3.4.1 MD5 - *Message Digest Algorithm 5*

O MD5, desenvolvido pela *RSA Data Security*, é uma versão melhorada do MD4, com o acréscimo de mais funções de embaralhamento e mais rodadas. É utilizado para gerar um valor *hash* de tamanho escolhido de até 128 bits.

O MD5 é mundialmente usado e é considerado seguro. Todavia, alguns especialistas têm alertado sobre potenciais fragilidades nele. Também já foi reportada a possibilidade de se montar um hardware especializado em 'quebrar' o MD5. Com um custo de cerca de U$ 10 milhões em 1994, este equipamento teoricamente poderia encontrar, em 24 dias, duas massas de dados que gerassem o mesmo valor *hash* [COS99].

3.4.2 SHA - *Secure Hash Algorithm*

O SHA, divulgado pelo governo americano, é nitidamente inspirado no MD5, com a diferença básica de produzir um valor *hash* de até 160 bits. Muitos o consideram completamente seguro. É um algoritmo razoavelmente novo.

3.5 Assinatura Digital

Assinatura digital é a versão eletrônica da assinatura manuscrita. É utilizada para certificar que o documento 'assinado' foi redigido pelo seu signatário, ou que este concorda com seu conteúdo.

Um assinatura digital deve ser capaz de garantir os seguintes requisitos:

a) autenticidade: é possível determinar que a assinatura pertence, ou não, a quem diz representá-la;

b) infalsificável: não há como gerar uma assinatura de outra pessoa sem conhecer seu código secreto;

c) inalterável: é possível determinar se o documento foi alterado após sua assinatura. Tornando sem efeito a assinatura digital;

d) unicidade: a assinatura é única para o documento subscrito pelo seu signatário. Outro documento, ou outro signatário, geraria outra assinatura;

e) inquestionável: o signatário não tem como negar a assinatura, exceto alegando 'quebra' de seu código secreto, utilizado para gerar a assinatura.

Um criptosistema para assinatura digital consiste de uma função matemática S para assinatura e outra V para verificação da mesma:

$$c = S(m, K_d) \tag{3.6}$$

$$m = V(c, K_e) \tag{3.7}$$

Existem muitos métodos para assinar e verificar a assinatura digital. Atualmente, os mais utilizados são algoritmo RSA e Funções *Hash*.

3.5.1 Assinatura Digital com RSA - *Rivest-Shamir-Adelman*

Ainda que alguns criptosistemas simétricos possam ser utilizados para criação e verificação de assinaturas digitais, os assimétricos podem realizar esta tarefa diretamente. Dentre estes, o RSA é o mais conhecido e um dos que combina funções de cifragem e assinatura no mesmo algoritmo.

Assim, invertendo o uso das chaves no RSA, as funções para cifragem e decifragem eqüivalem-se às funções para assinatura e verificação, respectivamente:

$$E(m, K_d) = S(m, K_d) \tag{3.8}$$

$$D(m, K_e) = V(m, K_e) \tag{3.9}$$

Supondo que Alice queira assinar o documento m e enviá-lo a Bob, basta que Alice inverta o uso de suas chaves assimétricas, utilizando sua chave privada K_{dAlice} para cifrar o documento:

$$c = E(m, K_{dAlice}) \tag{3.10}$$

Bob, então, utiliza a chave pública de Alice para recuperar o documento, decifrando o criptograma recebido:

$$m = D(c, K_{eAlice}) \tag{3.11}$$

Como somente a chave pública de Alice pode restaurar um documento criptografado com sua chave privada, e esta apenas Alice conhece, conclui-se que somente Alice poderia ter assinado tal documento. No entanto, vale ressaltar que, como qualquer um tem acesso a chave pública de Alice, não há qualquer privacidade no documento enviado, pois qualquer um pode restaurar o documento. Para adicionar garantia de privacidade, bastaria que Alice cifrasse o documento, então assinado, com a chave pública de Bob:

$$c' = E(c, K_{eBob}) \tag{3.12}$$

Agora, antes de Bob recuperar o documento original, deve decifrar o criptograma com sua chave privada:

$$c = D(c', K_{dBob}) \tag{3.13}$$

$$m = D(c, K_{eAlice}) \tag{3.14}$$

Com isto, Alice tem certeza que apenas Bob conseguirá restaurar o documento por ela assinado. Bob, por sua vez, tem certeza de que apenas Alice pode ter assinado aquele documento.

3.5.2 Assinatura Digital com Funções *Hash*

Na prática, o processamento de algoritmos assimétricos pode ser computacionalmente pesado e demorado, sendo estes fatores diretamente proporcionais ao tamanho do documento assinado. Para minimizar este problema, alguns criptosistemas de assinatura digital têm utilizado funções *hash*.

Assim, Alice gera um valor *hash* do seu documento:

$$h = H (m) \tag{3.15}$$

Então, assina apenas o valor *hash*, que é sensivelmente menor do que o documento e envia o documento original e o *hash* assinado para Bob:

$$a = S (h, K_{dAlice}) \tag{3.16}$$

$$Alice \rightarrow Bob: m, a \tag{3.17}$$

Bob recupera o *hash*, usando a chave pública de Alice, calcula o valor *hash* do documento recebido e compara os dois:

$$h = V (a, K_{eAlice}) \tag{3.18}$$

$$h' = H (m) \tag{3.19}$$

Se os valores (h *e* h') forem iguais, então o documento e a assinatura recebidos estão corretos e válidos.

4 Esteganografia

Neste capítulo será apresentada o estudo de técnicas de comunicação secreta: a esteganografia. Inicialmente, será dada uma visão geral sobre o assunto, abordando em especial aspectos históricos. Em seguida, será detalhada uma destas técnicas: a *watermark*, com demonstrações do uso moderno desta técnica, aplicado, para diversas mídias, entre as quais estão vídeo, som, texto e imagens.

4.1 Visão Geral

Esteganografia, do grego *steganos* (secreto) mais *grafia* (escrita) do latim, significa literalmente "escrita secreta". É utilizada para denominar um conjunto de técnicas de comunicação secreta que escondem a informação transmitida em outra portadora e inofensiva, sem atrair suspeitas.

Da mesma forma que a criptografia, a esteganografia tem origem muito antiga. Ao longo da história, um grande número de técnicas têm sido usadas para esconder informações. Principalmente em períodos de guerra, quando os países utilizavam estas técnicas para comunicar-se secretamente com seus agentes no estrangeiro. David Kahn [KAH67] faz uma excelente narração histórica sobre este tema. Bruce Norman [NOR73] também relata uma numerosa lista de eventos de criptografia e esteganografia durante épocas de guerra em *Secret Warfare: The Battle of Codes and Ciphers*.

Um dos primeiros documentos descrevendo esteganografia pode ser encontrado nas "Histórias" de Herodoto. Na Grécia antiga, os textos eram escritos em tabuletas recobertas com cera. Numa destas estórias, Demeratus tentava avisar Esparta que Xerxes pretendia invadir a Grécia. Para escapar à vigilância dos inimigos, Demeratus raspou a cera de uma das tabuletas e escreveu a mensagem diretamente na madeira. Depois, recobriu-as novamente com cera. As tabuletas pareciam nunca terem sido utilizadas e passaram pela inspeção dos sentinelas sem provocar suspeitas [JOH99].

Outro método engenhoso foi de raspar a cabeça de um mensageiro e tatuar a mensagem ou imagem em sua cabeça. Após seu cabelo crescer novamente, a mensagem seria imperceptível até que sua cabeça fosse raspada novamente [JOH99].

Outra forma comum de escrever secretamente é através do uso de tintas invisíveis. Tais tintas foram usadas com muito sucesso na Segunda Guerra. Uma carta inocente poderia conter uma mensagem muito diferente escrita entre as linhas. Na época, fontes comuns de tintas invisíveis eram leite, vinagre, sucos de fruta e urina. Todos estes desaparecem quando secam. Após, tintas mais sofisticadas foram desenvolvidas, de tal forma que algumas mensagem passavam por vários processos químicos, como o que ocorre durante uma revelação em laboratórios fotográficos [JOH 99].

O objetivo das técnicas de esteganografia era esconder a verdadeira mensagem, de forma que sua existência não atraísse suspeitas para a mensagem portadora. Caso alguém desconfiasse da existência de uma mensagem secreta, o objetivo não teria sido alcançado. Por isto, em muitos casos, as informações não sofriam nenhum tipo de processo de codificação que as tornassem ilegíveis, caso fossem detectadas. Um exemplo é a seguinte mensagem em texto claro, enviada por um espião alemão durante a Segunda Guerra [KAH67]:

> *Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.*

Utilizando a segunda letra de cada palavra obtemos a verdadeira mensagem:

> *Pershing sails from NY June 1.*

Há que se deixar clara a diferença entre criptografia e esteganografia. A criptografia modifica os dados para que não sejam legíveis aos possíveis interceptadores da mensagem. Ou seja, um interceptador está apto detectar a existência da mensagem, interceptá-la e modificá-la. Todavia, não conseguirá decifrar o conteúdo desta mensagem, pois estará matematicamente transformado. Diferente da esteganografia, que simplesmente esconde as informações no meio de outros dados, de forma que sejam imperceptíveis por terceiros. Esteganografia não é um substituto da criptografia. Pelo contrário, as duas se completam: enquanto uma pode garantir que a mensagem original seja decifrada apenas pelo seu destinatário, a segunda pode garantir que a mensagem chegue ao seu destinatário sem levantar suspeitas.

Como não poderia deixar de ser, as técnicas de esteganografia também se adaptaram às novas tecnologias e são atualmente aplicadas ao mundo da informática. Arquivos de computadores contêm áreas de dados insignificantes ou não utilizadas. A esteganografia tira proveito destas áreas, preenchendo-as com informações. Estes arquivos podem ser enviados ou transportados sem que alguém conheça o seu real conteúdo [BEN95]. Uma das técnicas esteganográficas mais discutidas e aplicadas à informática é a *watermark*.

4.2 *Watermark*

Digital watermark é melhor explicada quando comparada com a *watermark* tradicional. Estas são adicionadas em alguns tipos de papéis para oferecer prova de autenticidade. São imperceptíveis, exceto quando o papel é visto contra a luz. Similarmente, a técnica de *digital watermark* consiste em adicionar marcas em documentos eletrônicos de maneira que possam ser detectados apenas pelo programa de computador correto, permanecendo imperceptíveis pelo homem.

O termo *watermark*, ou sua tradução – marca d'água -, já é utilizado por alguns aplicativos comerciais, como é o caso do MS WORD[5] (vide figura 4). No entanto, estes aplicativos utilizam a expressão para referir-se a um recurso que inclui, no documento, um elemento visual com características apenas de apresentação.

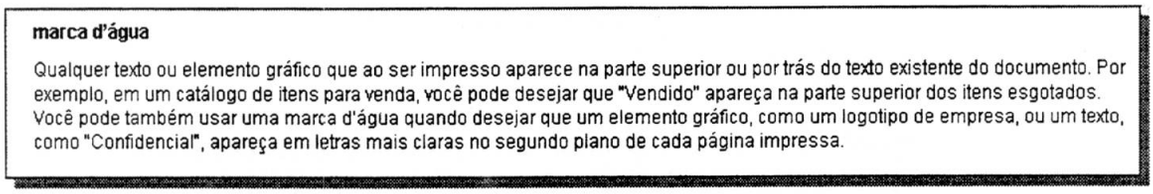


FIGURA 4.1 – Marca d'água - Tela de Ajuda do MS Word 97

Para este trabalho, o termo *watermark* está sendo utilizado para referir-se à tecnologia ou método de incluir alguma informação adicional e escondida em um documento eletrônico, que permita fornecer prova de sua propriedade e identificar a fonte de onde foi obtido o documento [AND96]. Em outras palavras, uma *watermark* é um sinal ou padrão digital inserido em um documento eletrônico[BER97].

Uma *watermark* pode ser idêntica para múltiplas cópias, por exemplo, para identificar a fonte do documento, seu autor exclusivamente. Uma *watermark* pode ainda ser única para cada cópia, por exemplo, para identificar quem é o leitor que adquiriu o direito de uso sobre aquela cópia [BER97].

Watermark envolve a transformação do documento original para outro documento marcado. Isto distingue *watermark* de outra técnica, o *fingerprinting*, onde o arquivo original é mantido intacto, mas outro arquivo é criado para descrever o conteúdo do arquivo original [BER97].

Watermark pode ser comparada com métodos de criptografia, os quais também transformam o documento original em outro. Todavia, com a criptografia o documento se torna irreconhecível até sua decodificação, enquanto que com *watermark* o documento permanece basicamente intacto e reconhecível. Além disto, documentos decodificados são livres de qualquer efeito residual da criptografia, enquanto a *watermark* permanece, ou pretende-se que permaneça, eternamente no documento.

4.2.1 Requisitos

Cox et al [COX95], *Boney* et al [BON96], *Berghel* et al [BER97] e outros, apontam alguns requisitos básicos para uma *watermark* eficiente:

- **Discreta**: deve ser invisível e imperceptível para prevenir a detecção não autorizada e conseqüente tentativa de remoção. E,

---

5 WORD é marca registrada de Microsoft Corporation.

principalmente, sua presença não deve interferir com o conteúdo do documento protegido;

- **Robusta**: deve ser difícil de remover. É claro que, na teoria, qualquer marca pode ser removida com conhecimento suficiente do processo de inserção. Todavia, se somente parte do conhecimento estiver disponível, como por exemplo a localização exata da marca dentro de uma imagem não ser conhecida, então tentativas para remover ou destruir a marca resultarão em severas degradações na fidelidade dos dados após a marca ser retirada. Em particular, a marca deve ser robusta para, por exemplo, suportar operações de compressão, descompressão, *resample*, *cropping*, ataques de piratas e outras.

- **Universal**: a mesma *watermark* deve ser aplicável para todos os tipos de mídias (som, imagens, vídeos). Isto é potencialmente útil em produtos multimídia.

- **Inquestionável**: a marca recuperada não pode suscitar dúvidas quanto ao seu verdadeiro proprietário.

- **Embutida**: a marca deve estar inserida direta e espacejadamentec nos dados, e não em uma porção específica do arquivo, por exemplo, no seu *header*.

- **Múltipla**: deve permitir a inclusão de múltiplas marcas no mesmo documento.

4.2.2 LSB - *Least Significant Bit*

A abordagem mais conhecida para inserir informações é a LSB. Neste caso, o bit menos significativo de cada byte hospedeiro é trocado por um dos bits da informação a ser escondida. Com isto, a cada oito bytes hospedeiros é possível esconder um byte de informação.

Por exemplo, suponha que um arquivo de áudio, vídeo ou imagem contenha, em um local qualquer de seus dados, os seguintes oito bytes de informação:

D	132	134	137	141	121	101	74	38
B	10000100	10000110	10001001	10001101	01111001	01100101	01001010	00100110

Agora suponha que se queira esconder um byte de informação nestes mesmos oito bytes. Digamos que o byte a ser escondido eqüivalha a:

D	213
B	11010101

A transformação, utilizando a técnica de LSB, substituirá o último bit de cada byte de informação, pelo bit equivalente do byte que será escondido.

B	10000100	10000110	10001001	10001101	01111001	01100101	01001010	00100110
B	1	1	0	1	0	1	0	1
B	10000101	10000111	10001000	10001101	01111000	01100101	01001010	00100111

Como resultado obtém-se os seguintes oito bytes de informação:

B	10000101	10000111	10001000	10001101	01111000	01100101	01001010	00100111
D	133	135	136	141	120	101	74	39

Como é possível verificar, cada byte foi alterado em no máximo um bit. Esta modificação não é percebida pelo ouvido ou olho humano, em uma imagem ou som. A informação inserida dificilmente poderá ser recuperada e alterada por piratas que desconheçam o algoritmo de inserção randômica da marca, que escolherá os bytes a serem modificados. Todavia, este método pode ser facilmente corrompido, pois sabendo que o algoritmo afeta apenas os bits menos significativos de uma palavra, então é possível alterá-los novamente, destruindo qualquer código de identificação existente. Assim, da mesma forma que a inserção da marca não é percebida pelo ouvido ou olho humano, sua destruição também não o será.

4.2.2 *Watermark* para Imagens

Em geral, as técnicas de *watermark* para imagens concentram-se em sobrepor um padrão ou sinal digital na imagem a ser protegida, conforme ilustra a figura 4.2.

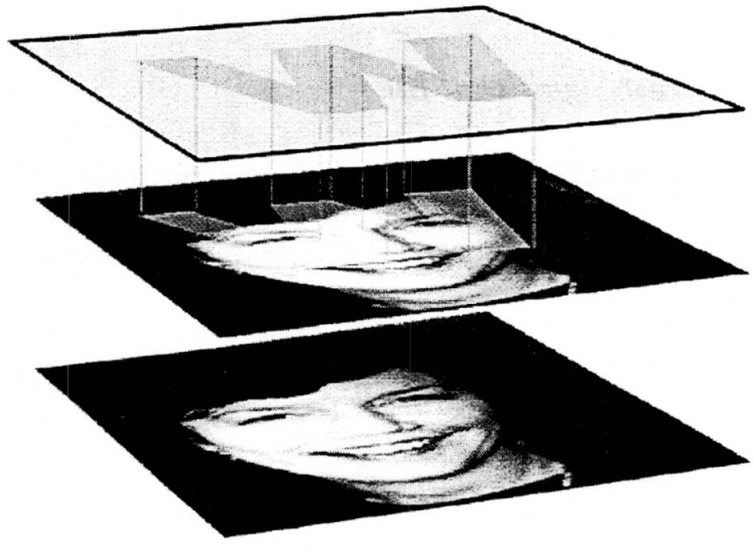


FIGURA 4.2 – *Watermark* em Imagens [ZHA97]

Desta forma, uma seqüência de bits são transformados para um padrão de preenchimento de imagem, semelhante aos exemplos apresentados na figura 4.3. Este padrão deve ser discreto o suficiente, quase transparente, para que ao ser sobreposto à imagem, permaneça imperceptível. Por esta razão, não deve existir qualquer traço regular de preenchimento, como blocos, linhas, colunas e outros. Uma boa técnica é

obter um padrão inicial a partir da própria imagem e efetuar algumas alterações que de alguma forma codifiquem a informação, para então embuti-la na imagem.

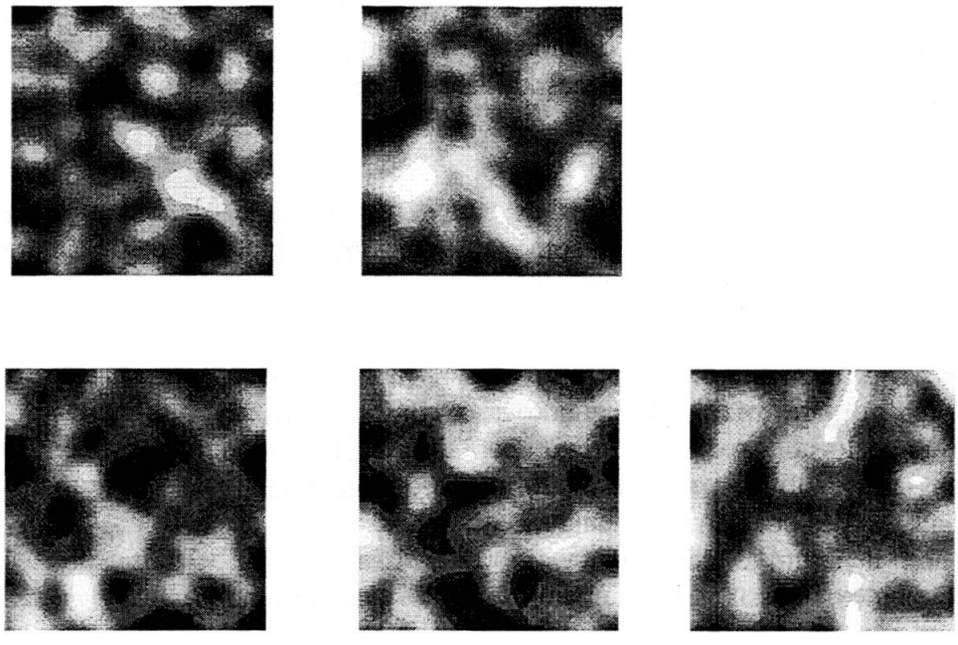


FIGURA 4.3 – Exemplos de *Watermarks* para imagens [FRI97]

4.2.3 *Watermark* para Vídeo

A idéia básica de *watermark* para vídeo é análoga àquela adotada para imagens, no entanto agora aplicada para uma seqüência de quadros de imagens. Ou seja, consiste na adição de um padrão que não pode ser percebido, identificado ou removido sem conhecimento dos parâmetros do algoritmo de *watermark* que o inseriu [HAR97].

A partir de uma seqüência de informações de bits que querem ser escondidas no vídeo, gera-se um sinal. Este é então ampliado por um fator de amplitude α e modulado como uma seqüência binária. O sinal é então adicionado ao vídeo digital original. O propósito da ampliação é adicionar redundância e embutir um bit de informação em muitos *pixels*.

4.2.4 *Watermark* para Áudio

No campo de áudio houveram algumas tentativas de utilizar *watermark*, mas ao que parece nenhuma tecnologia ainda se firmou. As experiências mais conhecidas utilizam também a técnica de LSB para inserir uma *string* de identificação em um sinal de áudio.

4.2.5 *Watermark* Para Textos

Diferente das outras mídias, nesta a *watermark* não pode alterar o conteúdo dos dados, por mais discreta que seja a alteração. Pois sendo os textos uma série de símbolos interpretados individualmente, qualquer alteração seria imediatamente percebida. Neste caso, resta apenas utilizar aspectos de apresentação do texto para embutir a *watermark*.

Brassil [BRA94][BRA95] e outros propõem e defendem técnicas de deslocamento tanto de linhas como de palavras, bem como de alteração das características das letras para embutir uma *watermark* em textos.

No primeiro caso, conforme figura 4.4, as linhas são deslocadas verticalmente para cima ou para baixo. No segundo caso, conforme figura 4.5, algumas palavras do texto são deslocadas para esquerda ou para direita. E, no terceiro caso, confirme figura 4.6, as letras sofrem pequenas deformações, diferenciando-as do normal da fonte utilizada. Ao deslocamento das palavras ou linhas, ou a deformação das letras, se de um ponto ou nenhum, é dado um valor 0 ou 1. E assim, pela interpretação destes deslocamentos é possível obter uma informação, que será a própria *watermark*.

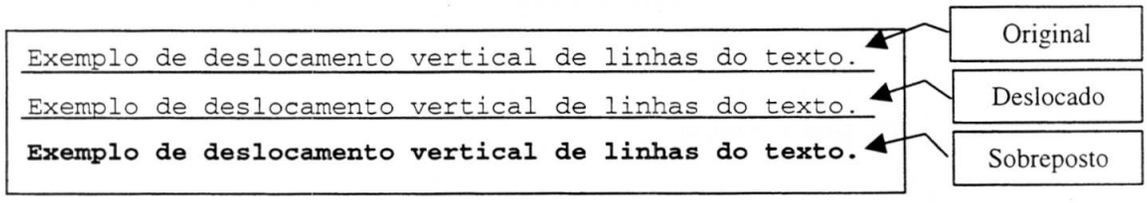


FIGURA 4.4 – *Watermark* para Textos - Deslocamento Vertical

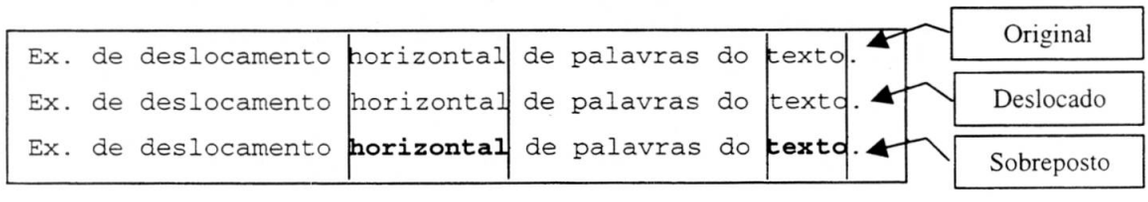


FIGURA 4.5 – *Watermark* para Textos - Deslocamento Horizontal

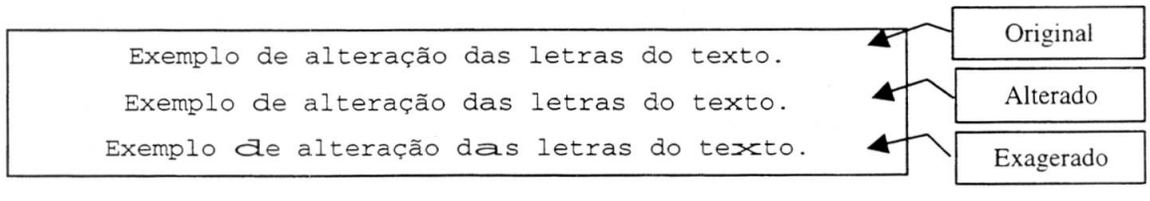


FIGURA 4.6 – *Watermark* para Textos – Alterações de Letras

Johnson [JON99] comenta uma alternativa mais simples de utilização de deslocamento horizontal de palavras para embutir informações em textos. Inicialmente identifica-se em um texto portador (figura 4.7), as palavras da informação que se pretende embutir, observando inclusive sua ordem de ocorrência. Em seguida,

condensa-se em um ponto o espaço que antecede cada palavra identificada, e expande-se em também um ponto o espaço após estas palavras, gerado um novo texto portador alterado, vide figura 4.8.

```
We explore new steganographic and cryptographic
algorithms and techniques throughout the world to
produce wide variety and security in the electronic web
called the Internet.
```

FIGURA 4.7 – Texto Portador Original [JON 99]

```
We explore new steganographic and cryptographic
algorithms and techniques throughout the world to
produce wide variety and security in the electronic web
called the Internet.
```

FIGURA 4.8 – Texto Portador Alterado [JON 99]

```
We explore new steganographic and cryptographic
algorithms and techniques throughout the world to
produce wide variety and security in the electronic web
called the Internet.
```

FIGURA 4.9 – Texto Portador Original e Alterado Sobrepostos [JON 99]

Visualizando tanto o texto original, como o texto alterado em separados, não é possível notar qualquer diferença, nem desconfiar da existência de qualquer informação embutida. Todavia, ao sobrepor estes textos, conforme ilustra a figura 4.9, algumas palavras ficam em negrito, identificando a informação embutida: *explore the world wide web.*

Todas estas técnicas de *watermark* partem do princípio de que ninguém terá acesso para modificar o texto após a marca ser embutida. Senão, esta seria facilmente removida pela simples formatação do texto. Assim, após a *watermark* ser embutida, o texto deve ficar desabilitado para alterações, devendo ser apresentado de forma que não seja permitida sua formatação. Em geral, a alternativa tem sido converter o texto para um *bitmap* no momento da apresentação ao leitor.

Em [BRA95] o autor propõe uma implementação para um sistema de marcas de documentos que justamente tem esta preocupação. O protótipo funciona com documentos em *PostScript* e compreende um codificador e um decodificador.

Como pode ser verificado na figura 4.10, o esquema geral do codificador proposto usa a técnica de deslocamento horizontal de palavras. O codificador lê um arquivo *PostScript*, insere uma marca, informada por um livro de códigos (*codebook*) e gera um documento imagem.

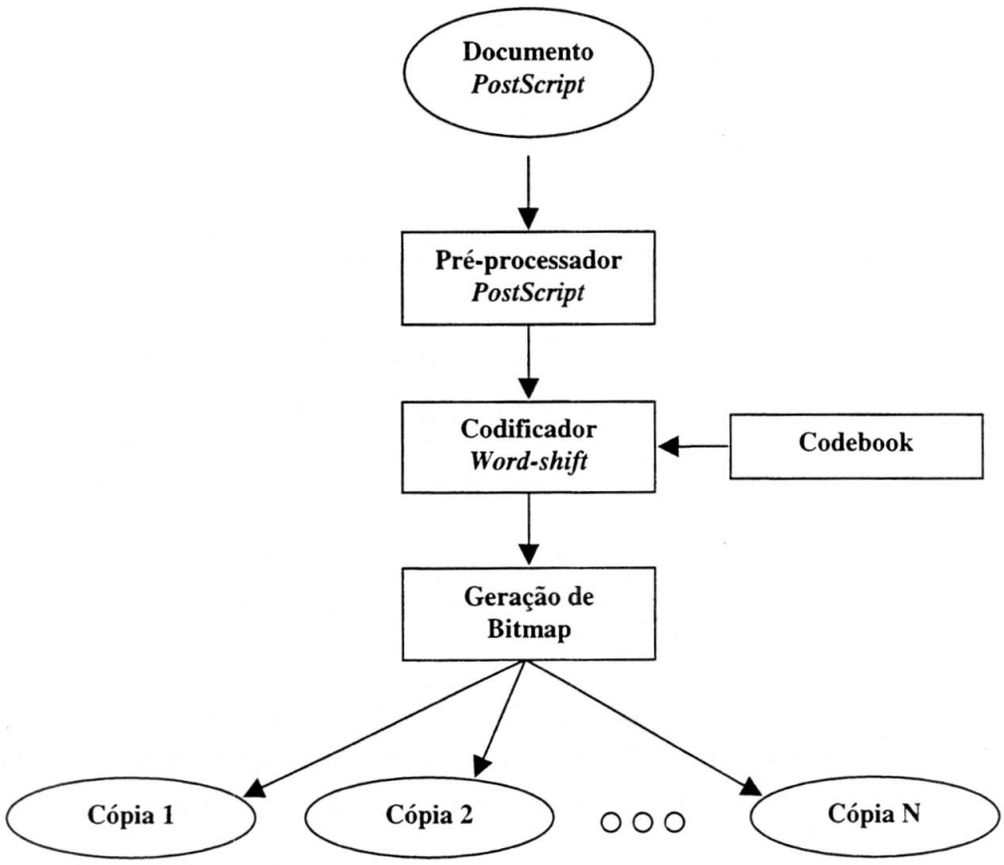


FIGURA 4.10 – Arquitetura Lógica do Codificador Proposto em [BRA95]

O escopo da *watermark* em documentos texto é um tanto diferenciado das demais mídias. Mesmo que o texto seja apresentado para o leitor como uma imagem, não permitindo sua edição ou formatação, o conteúdo ainda pode ser capturado e armazenado através de qualquer software de OCR. Ou ainda, se o texto for de pequenas proporções, poderá até ser totalmente redigitado.

Por isto, *watermarks* em documentos texto não irão impedir que partes dos textos sejam copiados, ou utilizados sem autorização. Por outro lado, a *watermark* pode ser muito bem aplicável naqueles casos em que o documento texto é extenso o suficiente para que sua redigitação ou digitalização via OCR seja inviável.

5 Estado da Arte

Neste capítulo são apresentados as principais ferramentas ligadas a distribuição e proteção dos direitos de autores de documentos digitais. Os destaques são o ACROBAT[6] para distribuição de documentos, o DIGIMARC[7] e SURESIGN[8] para proteção de imagens digitais e o UNZIGN para teste de robustez de *watermark* em imagens. Por fim, é visto o projeto DOCMARK da Universidade de Columbia, para distribuição de documentos com ênfase na proteção dos direitos do autor.

5.1 Comentários Iniciais

A informática avança rapidamente na tarefa de disponibilizar meios que permitam a distribuição eletrônica de documentos digitais, com a preocupação de inibir a pirataria e garantir os direitos autorais. Por esta razão, o levantamento que segue, feito com o objetivo de sustentar e fundamentar o presente trabalho, com certeza deve estar parcialmente ultrapassado até conclusão do mesmo. Todavia é um ótimo termômetro do que está acontecendo no mercado. Três itens são o destaque deste levantamento: o ADOBE PDF, o DIGIMARC WATERMARKING e o UNZIGN.

O ACROBAT, aplicativo da ADOBE para manusear documentos PDF - *Portable Document Format* é uma ótima opção para distribuição de documentos digitais, com um bom controle de permissões ao usuário, onde o autor tem o poder de permitir, ou não, a impressão e a alteração do documento pelo leitor. Porém, a versão testada ainda era falha no item de segurança, pois utilizava apenas um simples sistema de senhas para garantir o acesso ao documento. A recente versão 4.0 do ACROBAT para Windows, que não foi possível analisar, promete suportar assinaturas codificadas e digitais biométricas, que podem ser usadas para autenticar a autoria, controlar alterações e verificar aprovações de documentos. Mesmo assim, ainda não se falou em *watermark* para estes documentos. Assim, uma vez que se tenha acesso ao conteúdo de um documento PDF, seria possível pirateá-lo.

O outro item de destaque, a solução de *watermark* da DIGIMARC, é um sistema profissional para inserir marcas em imagens, principalmente pela sua implementação como filtro para grandes editores gráficos e seu aplicativo complementar, o MARCSPIDER, que vasculha a rede a procura de imagens marcadas, notificando o autor dos locais encontrados.

Por fim, o UNZIGN e STIRMARK, ferramentas que poderiam ser consideradas como facilitadoras da pirataria, mas que atuam em prol da antipirataria. Com estas ferramentas os desenvolvedores podem medir a robustez de suas *watermarks*, pois a

---

[6] ACROBAT é marca registrada de Adobe Systems Incorporated.
[7] DIGIMARC é marca registrada de Digimarc Corporation.
[8] SURESIGN é marca registrada de Signum Technologies.

intenção delas é justamente retirar as marcas dos documentos protegidos, simulando a ação dos piratas autorais.

5.2 Stego e EzStego

Romana Machado desenvolveu STEGO [MAC99], uma aplicação simples, mas pioneira na arte de esconder informações. O programa utilizava a técnica LSB para esconder informações em arquivos de imagem no clássico formato PICT, nativo de computadores MACINTOSH.

Todavia, em função da estrutura da *palette* de cores, esta primeira implementação apresentava problemas ao trabalhar com imagens de 256 cores. Na figura 5.1 é possível perceber muitos *pixels* próximos com cores de níveis diferentes de saturação um dos outros.

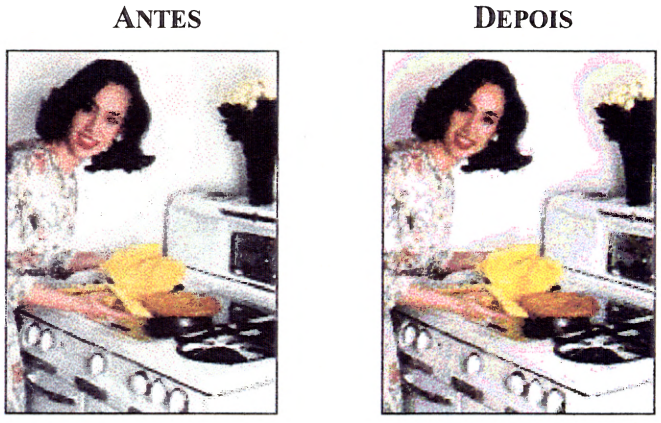


FIGURA 5.1 – Imagens em 256 cores antes e depois da transformação de Stego [MAC 99]

Analisando a tabela de cores (figura 5.2) utilizada na imagens acima, nota-se que muitas cores adjacentes são muito diferente. Como a troca do último bit significante de cada byte acarreta a conseqüente troca do índice na tabela de cores, a transformação produzida pelo STEGO originava uma deformação visual nas imagens.

FIGURA 5.2 – Tabela de cores utilizada na figura 5.1

Para resolver este problema, Romana desenvolveu o EZSTEGO [MAC 99], uma nova versão, que coloca em ordem a tabela de cores da imagem antes de fazer a transformação, colocando cores similares próximas uma das outras. O objetivo é encontrar a melhor ordenação para as cores da imagem, de tal forma que o total da "distância" entre as cores seja minimizado. A "distância" entre duas cores é calculada pela fórmula:

$$\sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2} \qquad (5.1)$$

Não há um método que seja notoriamente reconhecido como a melhor maneira de organizar uma *palette* de cores. Para a implementação de EZSTEGO, Romana optou por um algoritmo básico de organização, pelo qual antes de inserir uma ocorrência na tabela de cores, pesquisa-se sua melhor localização dentre as cores que já foram inseridas na tabela.

5.3 S-Tools

Desenvolvido por Andy Brown, STEGANOGRAPHY TOOLS [BRO99] ou simplesmente S-TOOLS é uma ótima ferramenta para esconder informações em arquivos de áudio e imagem.

A antiga versão para WINDOWS 3.X incluía três programas: o ST-BMP.EXE para arquivos de imagem nos formatos GIF e BMP; o ST-WAV.EXE para arquivos de áudio no formato WAV e o ST-FDD.EXE que escondia informações em áreas "não utilizadas" de disquetes. A nova versão, compatível com WINDOWS 95 e WINDOWS NT, integra em um único aplicativo as funcionalidades dos antecessores ST-BMP e ST-WAV, e deixou simplesmente de dar suporte as características do ST-FDD.

Em ambas versões, as informações são criptografas com algoritmos bastante seguros (IDEA, DES e outros) antes de serem inseridas nos arquivos hospedeiros. A técnica utilizada para inserir as informações continua sendo a LSB.

A interface é baseada em recursos de *drag-and-drop*. Após iniciar o aplicativo, basta arrastar um arquivo dos formatos suportados (BMP, GIF ou WAV) para dentro da janela do S-TOOLS. Uma nova janela será aberta mostrando o arquivo arrastado, como nas figura 5.3 e 5.4 .

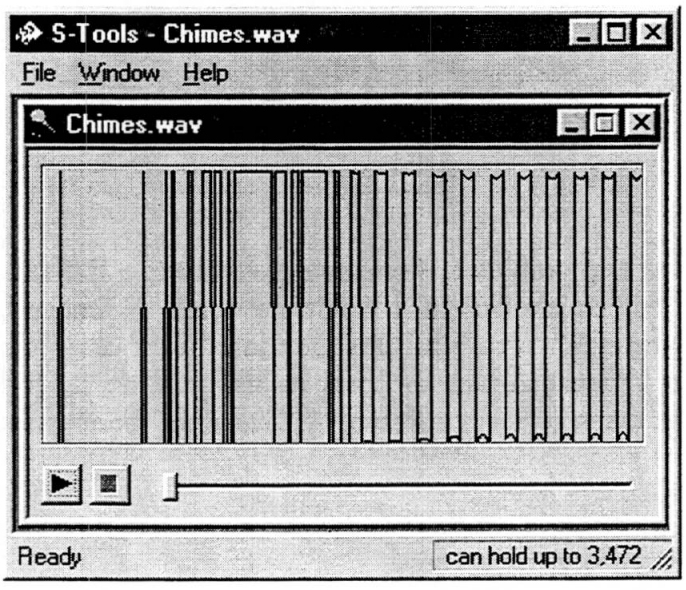


FIGURA 5.3 – Tela do S-Tools exibindo um arquivo WAV

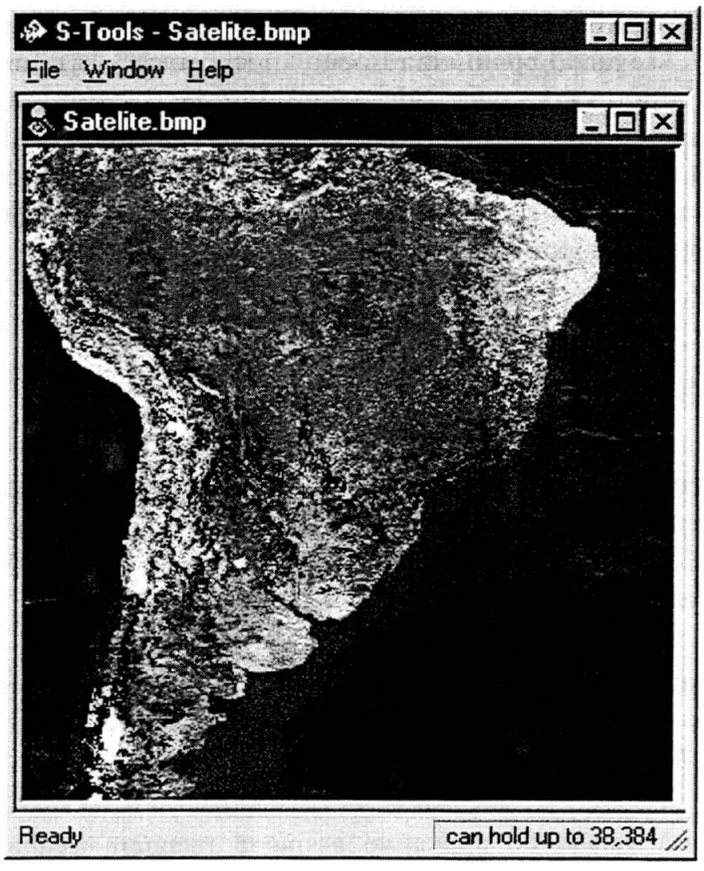


FIGURA 5.4 – Tela do S-Tools exibindo um arquivo BMP

Em seguida, basta arrastar o arquivo contendo as informações que devem ser escondidas para cima da janela da imagem ou do áudio. Isto fará com que o aplicativo abra uma outra janela de diálogo, como a da figura 5.5, onde deve ser informado um código e algoritmo, que serão utilizados para criptografar a informação. Note-se que no título é exibida a quantidade de bytes que serão inseridos. Caso a quantidade de informações a serem inseridas seja grande demais para o arquivo hospedeiro, o aplicativo avisará que não será possível inserir as informações.

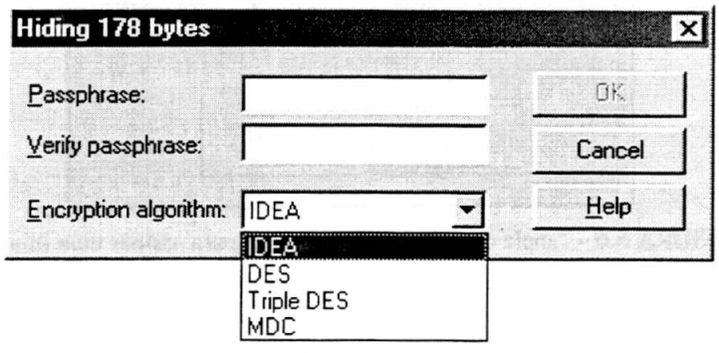


FIGURA 5.5 – Janela de Diálogo do S-Tools com os parâmetro de criptografia

Após inserir as informações, uma nova janela será exibida, permitindo que o original e a nova versão sejam comparados.

Ainda que seja um aplicativo bem desenvolvido, principalmente pela sua interface e facilidade de uso, dois pontos devem ser criticados. O primeiro diz respeito a

técnica utilizada, o LSB como já foi dito não é robusto suficiente para suportar os ataques simples. O segundo ponto diz respeito à recuperação da informação, a idéia de criptografar as informações antes de serem inseridas, acrescenta algum nível de segurança no sistema como um todo. Porém, apenas possuindo o código utilizado para criptografar, será possível recuperar as informações da marca inserida. Logo, terceiros que desconheçam tal código, não terão como identificar o proprietário original da imagem ou áudio protegidos.

5.4 JK_PGS

O projeto da SIGNAL PROCESSING LABORATORY (LTS) do SWISS FEDERAL INSTITUTE OF TECHNOLOGY, Lausanne, Suíça, em colaboração com o LABORATOIRE DE RESEAUX DE COMMUNICATIONS, desenvolveu alguns algoritmos para esconder informações em arquivos de imagens e vídeos digitais.

Para testar e comprovar seus algoritmos foi implementado o aplicativo JK_PGS [SIG99], iniciais dos nomes dos desenvolvedores, Frederic **J**ordan e Martin **K**utter, mais a abreviatura de ***P**retty **G**ood **S**ignature*.

A primeira versão do aplicativo assina e recupera assinaturas em imagens estáticas. O propósito único da assinatura é identificar o proprietário da imagem, seja para uma possível disputa judicial, ou para que os interessados tenham como descobrir o proprietário da imagem, para então contatá-lo.

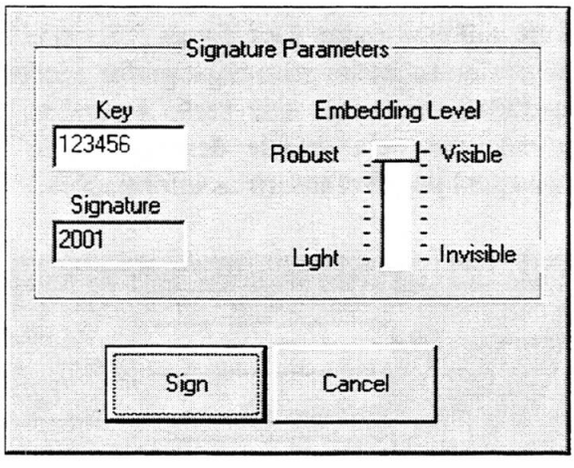


FIGURA 5.6 – Janela de diálogo do JK_PGS para assinar uma imagem

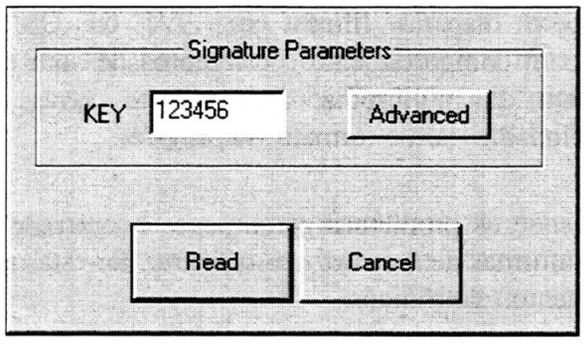


FIGURA 5.7 – Janela de diálogo do JK_PGS para recuperar assinatura

A assinatura, ou também chamado de *Registration Identification Number* (RIN), é um número de 28 bits. De posse deste número, qualquer um pode acessar o *RIN Database*, mantido pelo LTS, e descobrir o nome e e-mail do, teoricamente, proprietário da imagem. Todavia, outro número de 32 bits, chamado de *Personal Identification Number* (PIN), pode ser adicionado a imagem. Funcionando como uma senha, o propósito do PIN é determinar se uma assinatura é pública ou não. Assim, se o PIN de uma imagem for igual a zeros, qualquer está apto a recuperar o RIN.

Atualmente o JK_PGS está disponível para Windows, SGI, SUN e LINUX. Uma nova versão deverá estar disponível no formato de *plug-in* para editores gráficos com PHOTOSHOP, PAINTSHOP e PICTURE PUBLISHER. Todavia espera-se que o algoritmo implementado seja mais robusto, para que resista pelo menos a operações simples como rotação, escala e recorte.

5.5 Adobe PDF

O ADOBE PDF, desenvolvido pela ADOBE SYSTEMS, é um formato de arquivos usado para representar um documento digital de forma independente da aplicação, equipamento e sistema operacional utilizado para criá-lo. Preserva todas as fontes, formatação, cores e elementos gráficos do documento original, sendo rigorosamente fiel a este na materialização da cópia em papel, na visualização na *Web*, em CD ou em multimídia.

Arquivos PDF são compactos e podem ser compartilhados, visualizados, navegados e impressos exatamente da maneira desejada através de um *Reader* gratuito. Qualquer documento pode ser convertido para PDF, até mesmo documentos digitalizados e armazenado na forma de bitmap. Os documentos PDF mantém a aparência e o *layout* de seus originais.

Quanto à navegação, além de permitir embutir os *hyperlinks* comuns, permite também criar *bookmarks*, que funcionam como árvores de hierarquias de informações. Um clique em qualquer dos tópicos relacionados na árvore, ou em um *hyperlink*, remete imediatamente ao assunto procurado, quer esteja ele um outro PDF ou um URL.

Pode-se também disparar filmes em AVI ou QUICKTIME[9], bem como implementar botões com comportamento e elementos de som e navegação que, para muitos, se aproximam da multimídia. Além disto, novas características podem facilmente serem adicionadas para o formato via *plug-in*.

O PDF transcende os problemas geralmente encontrados no compartilhamento de arquivos entre plataformas de *hardware* e *software*. Por esta razão se diz o ideal para distribuição de documentos eletrônicos.

Deve-se observar que o Grupo PDF, um consórcio que representa as gráficas comerciais que atualmente formam um mercado em torno de 10 bilhões de dólares, está pressionando para que o PDF se torne um formato universal. O Grupo PDF foi formado para assessorar a ADOBE no aprimoramento do PDF, para que ofereça soluções para as necessidades extremamente exigentes desse mercado, que incluem a saída de maior qualidade e maior volume [McK98]

Assim como o HTML foi o responsável pelo crescimento da *Web* e o *Postscript* foi a base da publicação eletrônica, o PDF promete ser a âncora para a distribuição eletrônica no futuro.

5.5.1 PDF versus HTML

A Internet já existia há muito tempo antes da *Web*. No entanto, eram necessários sólidos conhecimentos de UNIX para utilizá-la. Com o surgimento do protocolo HTTP e da linguagem HTML, foi possível automatizar todos os procedimentos que permitem a conexão global, tal como conhecemos atualmente.

A HTML, além da funcionalidade crucial dos links de hipertexto, foi desenvolvida como uma linguagem para diagramação de páginas, o que inclui uma ampla variedade de atributos de texto e recursos gráficos. Ao longo do tempo, a HTML foi ampliada com muito mais em termos de conectividade de documentos, do que em termos de apresentação para *Web*.

O PDF, por sua vez, foi elaborado para ter uma portabilidade semelhante ao *postscript*. A diferença é que, enquanto que os dispositivos de saída para o *postscript* são impressoras laser, os dispositivos de saída para PDF são interfaces gráficas com o usuário, seja qual for a plataforma de *hardware* e *software*.

O PDF é perfeitamente integrável ao HTML, podendo ser acessado dentro do *browser* e proporcionando a navegabilidade comum do formato, acrescentando os valiosos recursos de zoom e movimentação da página, além de todos os recursos que tenham sido implementados pelo autor.

---

[9] QUICKTIME é marca registrada de APPLE COMPUTER.

Resumindo, o HTML foi desenvolvido para apresentar informações na tela. Todavia, mesmo as mais recentes extensões HTML ainda não se aproximam dos recursos de design e diagramação de página do PDF. Em termos de impressão, o formato PDF representa uma promessa muito maior, uma vez que o PDF retém todos os recursos do *postscript* para materializar o documento em papel.

De modo geral, pode-se dizer que quando for necessário preservar o *layout* do documento original, exibindo ao usuário final o documento da maneira como seria impresso, a melhor solução é o PDF. Logo, como na distribuição eletrônica, o controle da qualidade de saída, tanto em tela como impressa, é justamente uma das principais preocupações, o formato PDF é muito mais recomendado do que o HTML.

TABELA 5.1 – Comparação entre HTML e PDF [PDF99]

CARACTERÍSTICAS	HTML	PDF
Padronização e Desenvolvimento	Consórcios públicos	Proprietário (Adobe)
Funções de Hipertexto	Excelente	Excelente
Mecanismos de procura	Excelente	Excelente
Criação de novos documentos	Editores HTML	Qualquer um
Conversão de documentos	Trabalhosa e demorada	Simples e rápida
Qualidade de impressão	Pobre	Alta
Possibilidades de zoom na tela	Não	Sim
Aparência determinada pelo	Leitor	Autor

5.5.2 Estilos de Documentos

O formato PDF suporta três estilos de documentos:

PDF Image Only: contém somente uma figura bitmap do documento original, seja texto ou figura. Normalmente estes documentos são produzidos pela digitalização de documentos em papel. Este estilo é ideal para converter e visualizar arquivos imagens.

PDF Normal: contém informações textuais, que podem ser indexadas, pesquisas e copiadas. A formatação de página e imagens gráficas são preservadas. São significativamente menores que *PDF Image Only*, sendo o ideal para distribuição eletrônica.

PDF Original Image with Hidden Text: combina características dos dois anteriores. Ou seja, contém uma figura bitmap do documento original e também seu texto convertido. Isto permite pesquisas avançadas, como procura por expressões ou palavras, enquanto garante que a aparência do documento é idêntica ao original. Este estilo é ideal para aqueles casos em que, por propósitos legais ou documentacionais, é necessário manter o original digitalizado.

5.5.3 Adobe Acrobat

O Adobe Acrobat [ADO99] é um pacote de aplicativos comercializados pela ADOBE para criar e manipular documentos PDF. Seus principais itens são ACROBAT EXCHANGE, ACROBAT READER, ACROBAT DISTILLER, ACROBAT PDFWRITER, ACROBAT CATALOG e o *plug-in* CAPTURE.

TABELA 5.2 – Componentes do Acrobat 3.0 por plataforma [ADO 99]

	Reader	*Exchange*	*PDF Writer*	*Distiller*	*Catalog*	*Capture*
Macintosh	✔	✔	✔	✔	✔	
PowerMac	✔	✔	✔	✔	✔	*
Win 3.1	✔	✔	✔	✔	✔	✔
Win95	✔	✔	✔	✔	✔	✔
WinNT 3.5.1	✔	✔	✔	✔	✔	✔
WinNT 4.0	✔	✔		✔	✔	
OS/2	✔					
SunOS	✔	✔		✔		
Solaris	✔	✔		✔		
HP-UX	✔	✔		✔		
AIX	✔	✔		✔		
IRIS (SGI)	✔					
LINUX	✔					

O **ACROBAT READER** é um aplicativo necessário para visualizar, navegar e imprimir um arquivo PDF. Este aplicativo está disponível gratuitamente na *home page* da Adobe. A versão 3.0 do Acrobat Reader está disponível também como um *plug-in* para o *browser* Netscape Navigator e para o Internet Explorer. Assim, é possível escolher entre fazer o download do PDF para leitura off-line, o que é mais recomendável, ou configurar um *browser* para automaticamente acionar o *viewer* do Acrobat Reader como um *helper application*.

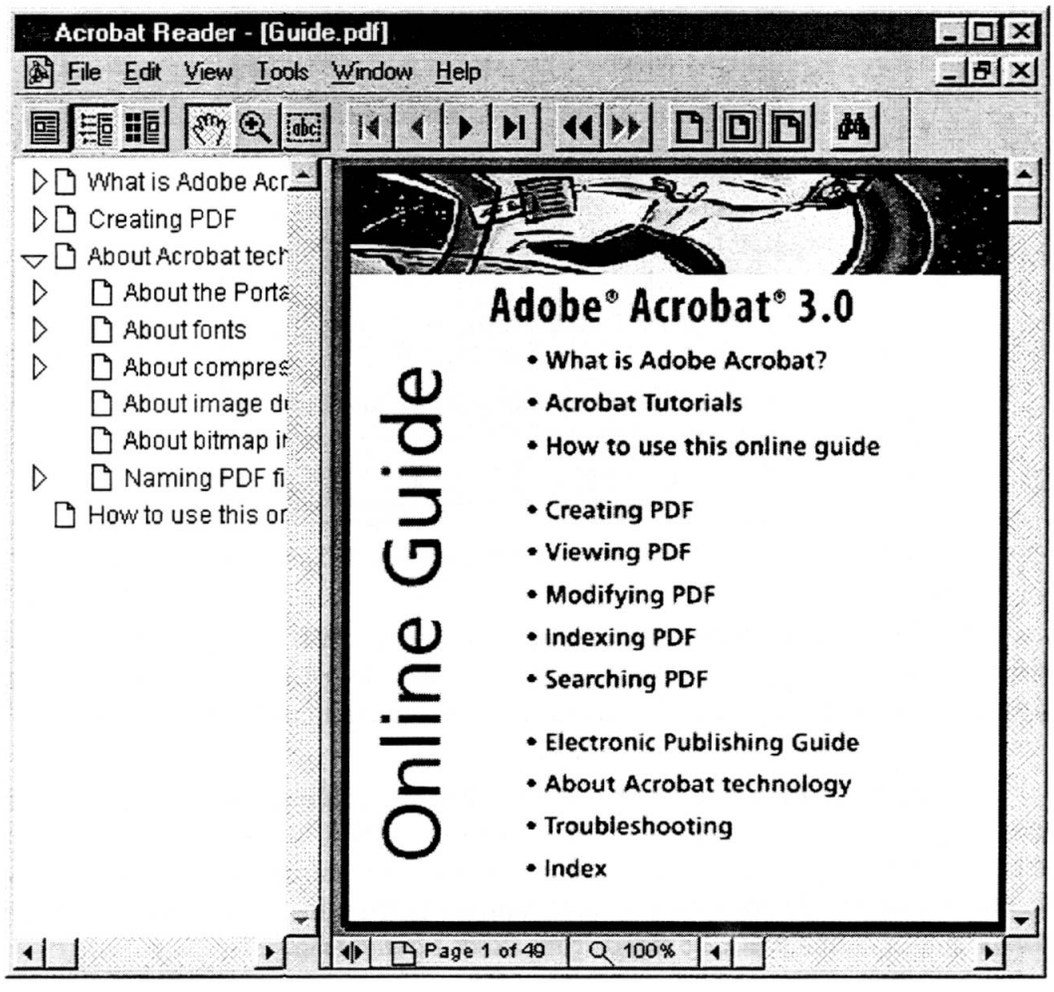


FIGURA 5.8 – Tela de navegação de documentos PDF do Acrobat Reader [ADO 99]

O **ACROBAT EXCHANGE** é o elemento principal do ACROBAT, é usado para implementar todas as características possíveis do PDF, como inserir *hyperlinks*, formulários, filmes, sons e outros. Também é no EXCHANGE que são definidos os atributos de segurança e restrições, tais como evitar que o leitor imprima ou modifique o arquivo, selecione texto e imagens, ou crie e modifique anotações eletrônicas e campos do formulário, vide figura 5.9.

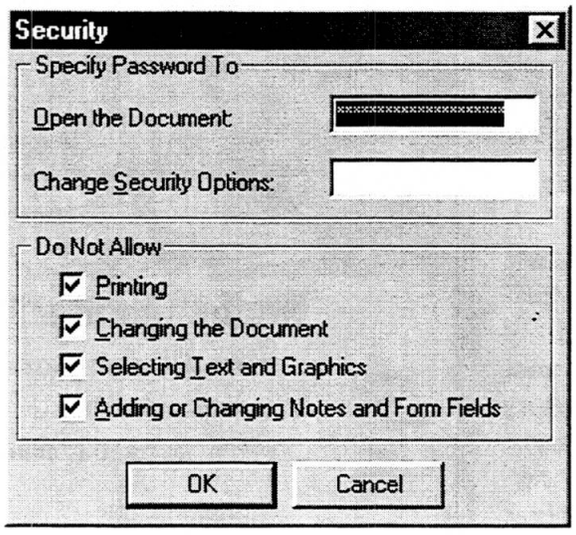


FIGURA 5.9 – Tela de atributos de egurança de um PDF no Acrobat Exchange

O **ACROBAT DISTILLER** cria um documento PDF a partir de um arquivo *postscript*, permitindo uma customização precisa de parâmetros como: resolução, inclusão de fontes, preservação de paletas de cores, tamanho da página e outros.

O **ACROBAT PDFWRITER** funciona como uma impressora virtual. Assim, para converter o documento original para o formato PDF basta "imprimi-lo" para o PDFWRITER. Este processo não é tão preciso e não permite uma customização tão refinada dos ajustes de conversão quanto aqueles permitidos pelo ACROBAT DISTILLER. Todavia, é uma forma prática e rápida para fazer a conversão.

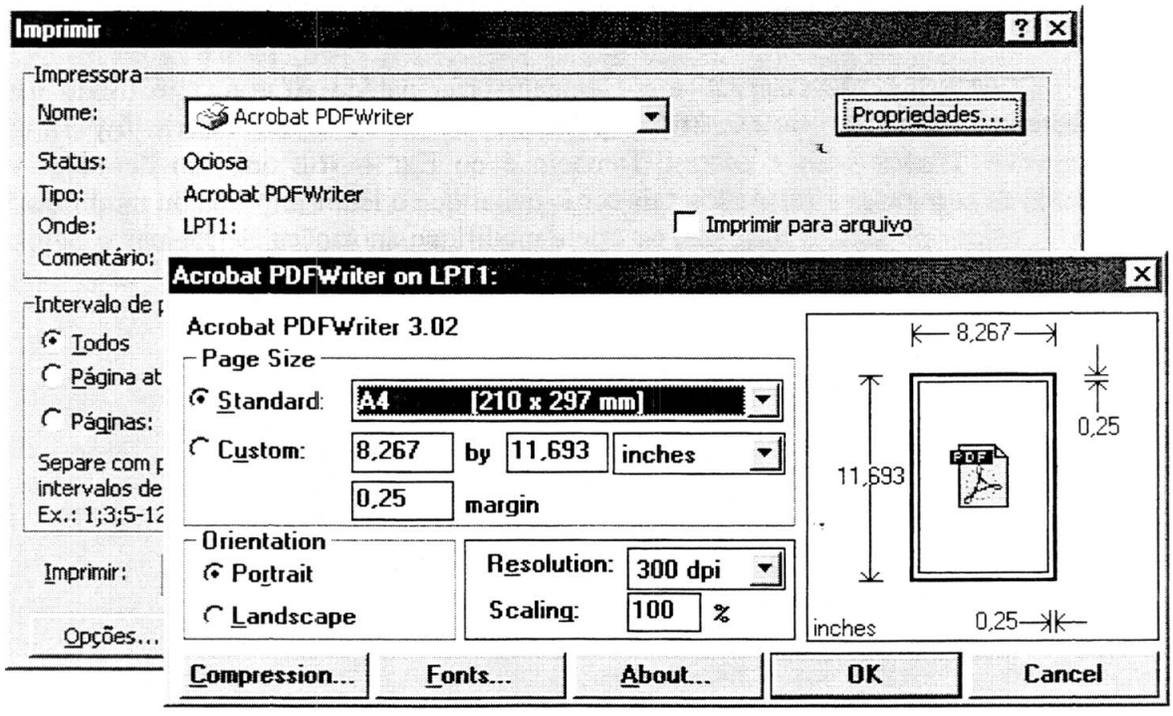


FIGURA 5.10 – Tela de Impressão do Word para o Acrobat PDFWriter

O **ACROBAT CATALOG** é utilizado para indexar arquivos PDF, em um ou mais diretórios, criando um banco de dados que realizará uma consulta integral nos PDF. Pode também ser programado para "varrer" periodicamente diretórios atualizando automaticamente o banco de dados de consulta.

O **ADOBE CAPTURE** é um importante *plug-in* para o **ADOBE EXCHANGE**, que usa reconhecimento óptico de caracteres para converter documentos *PDF Image Only* em *PDF Normal* ou *PDF Original Image with Hidden Text*. O reconhecimento é executado em vários níveis, convertendo não apenas o conteúdo do texto, mas também a aparência dele. Esse processo envolve a substituição de fontes e de seus atributos para que correspondam ao documento de origem, bem como reconstruir o documento nas dimensões exatas da página original.

5.6 Digimarc Watermarking

Originalmente criado por Geoffrey Rhoads para monitorar a movimentação e utilização de suas fotos na Internet, o sistema da DIGIMARC [DIG 99] é a melhor solução disponível no mercado em *watermark* para imagens. O sistema é oferecido como um filtro para os principais editores gráficos, como o PHOTOSHOP[10] e o PHOTO-PAINT [11]. Através deste filtro é possível inserir e recuperar de forma rápida e fácil, uma *watermark* em imagens.

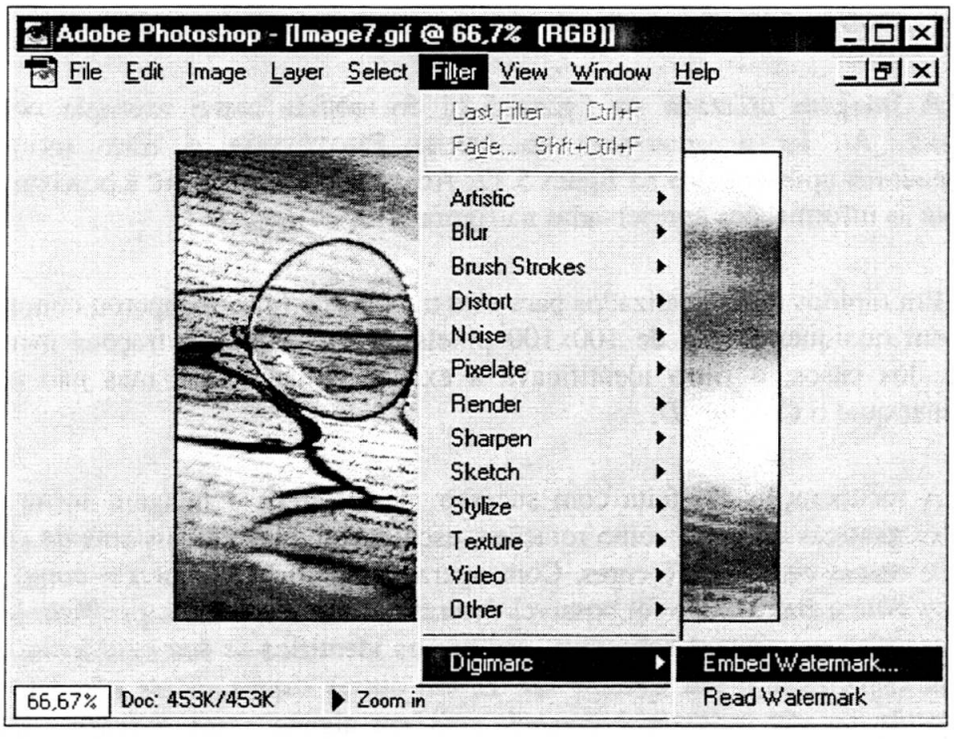


FIGURA 5.11 – Tela do Adobe Photoshop mostrando acesso ao filtro da Digimarc

---

[10] PHOTOSHOP é marca registrada Adobe Systems Incorporated.

[11] COREL DRAW é marca registrada de Corel Systems Corporation.

A *watermark* da DIGIMARC contém o identificador do proprietário, *Creator ID*, o identificador da Imagem, *Image ID*, e os atributos de *copyright*. Conforme demonstra a figura 5.12, ao recuperar a *watermark*, o filtro apresenta estes dados ao usuário e permite obter maiores informações sobre a imagem e seu proprietário, através do botão WEBLOOKUP ou pelo endereço URL que aparecem na tela. Ambas opções direcionam para o DIGIMARC LOCATOR SERVICE, um serviço que armazena, no *site* da DIGIMARC, informações detalhadas sobre o autor e sobre a imagem marcada.

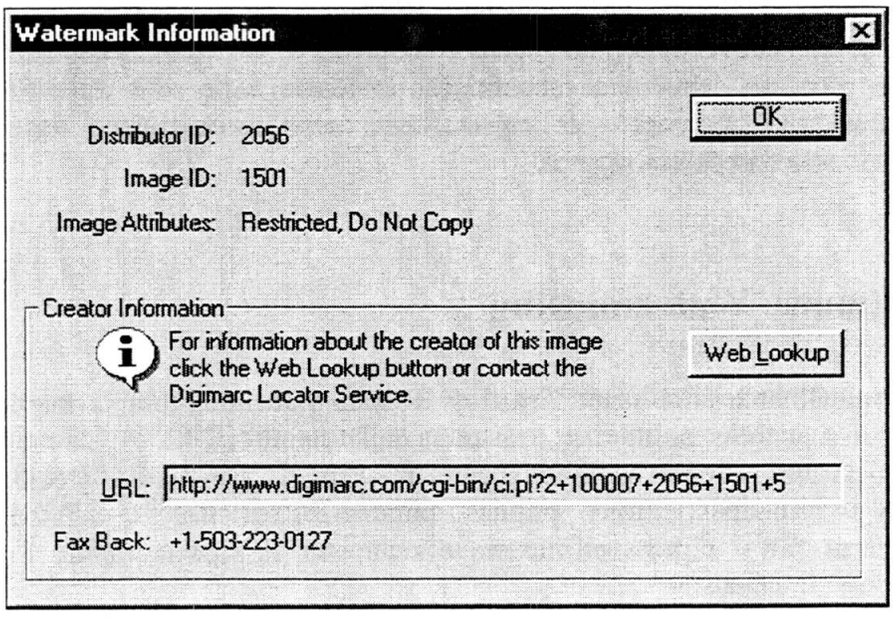


FIGURA 5.12 – Tela do filtro da Digimarc exibindo Informações da *Wartermark*

A imagem utilizada na figura 5.11 foi obtida como exemplo no *site* da DIGIMARC. Ao ler a *watermark* no ADOBE PHOTOSHOP, o filtro recuperou os identificadores apresentados na figura 5.12. Acessando o DIGIMARC LOCATOR SERVICE obtêm-se as informações apresentadas na figura 5.13.

Em rápidos testes realizados para este trabalho, o filtro recuperou com sucesso a marca em qualquer fração de 100x100 pixels da imagem. Em frações menores, na maioria dos casos, o filtro identificava a existência da marca, mas não conseguia determinar qual o *Creator ID*.

A recuperação era feita com sucesso mesmo após a imagem sofrer algumas operações gráficas comuns, como rotações, escala, conversão do sistema de cores para escala de cinzas ou para 256 cores. Com operações gráficas complexas como *Emboss*, *Gaussian Blur* e outros, não foi possível determinar um resultado específico. Isto é, em alguns casos, a *watermark* sobrevive, em outros identifica-se sua existência, mas não consegue especificar o seu *Creator ID*. E, em outras simplesmente não é localizada. Estes resultados são bastante coerentes, pois em alguns casos a própria imagem é deformada de tal maneira que não se torna mais identificável, logo a marca não teria como, nem porquê, permanecer, pois trata-se de uma nova imagem, ainda que criada a partir de outra.

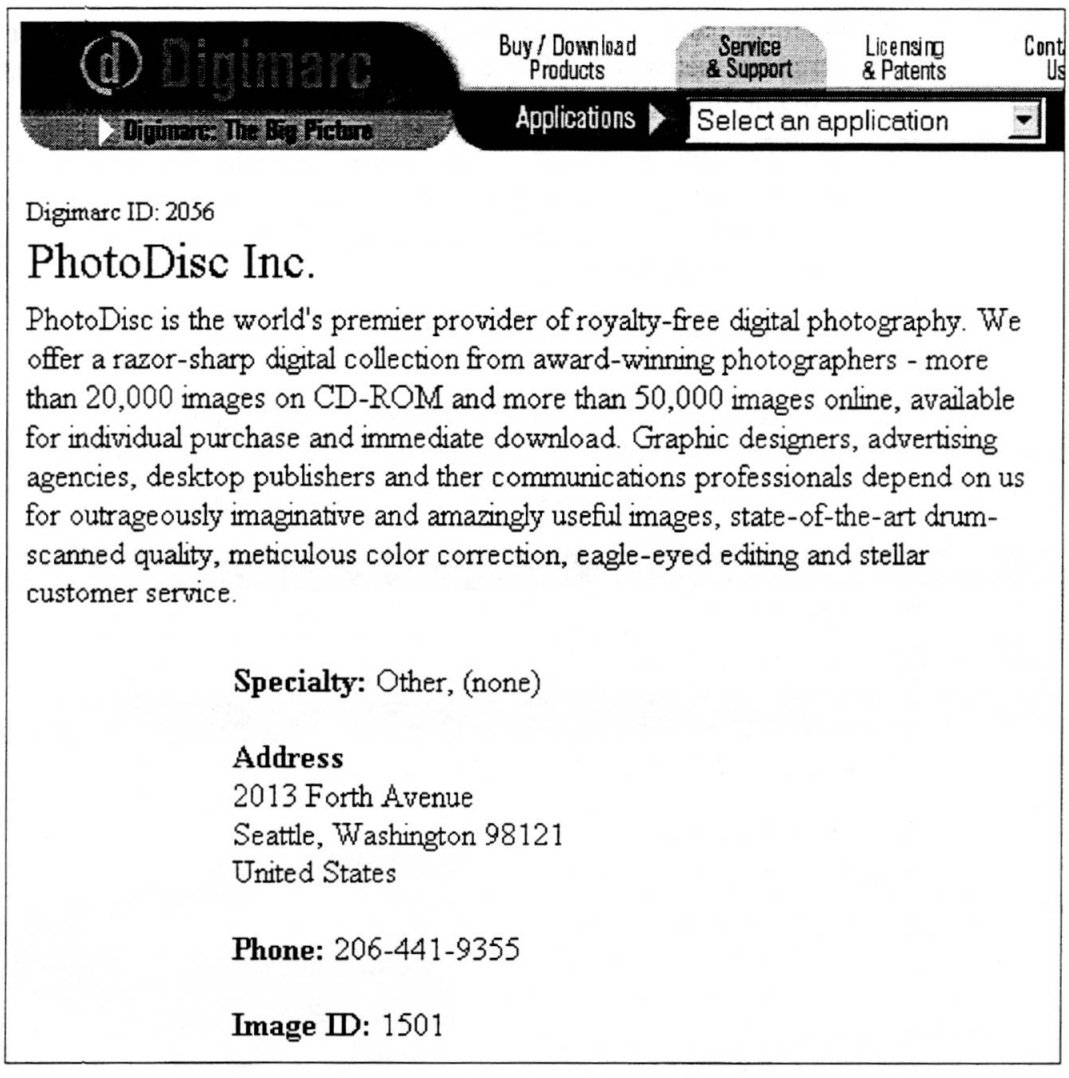


FIGURA 5.13 – Tela do *site* da Digimarc com dados do proprietário da imagem [DIG 99]

Após inserir a marca, quando um terceiro abrir a imagem em um dos editores gráficos suportados, será automaticamente avisado de que aquela imagem é protegida por direitos autorais e não deve ser usada de forma ilícita. Caso este terceiro abra a imagem em um outro aplicativo gráfico, não suportado pelo PICTUREMARC, ou uma versão antiga daqueles, nenhuma mensagem de aviso será emitida. Todavia, a marca permanecerá escondida na imagem e não será detectada nem pelo editor, nem pelo terceiro.

Além do filtro, a DIGIMARC oferece também dois aplicativos independentes de editores gráficos: o PICTUREMARC e o READMARC.

O PICTUREMARC[12] é oferecido para usuários registrados e permite inserir e recuperar uma *watermark* sem a utilização de um editor gráfico. Suas principais características são:

---

[12] PICTUREMARC é marca registrada de Digimarc Corporation.

- funciona para os principais formatos de arquivo de imagens do mercado, inclusive JPEG e GIF, padrões onde conhecidamente há perda de informações, em função da compressão dos dados;
- permite medir a "força" da *watermark* inserida. Ou seja, através de um medidor próprio é possível verificar qual a probabilidade da *watermark* inserida pelo aplicativo "sobreviver" a edição e compressão da imagem;
- suporta pequenas imagens, a partir de 100 x 100 *pixels*.

O READMARC é uma aplicativo gratuito disponível a qualquer usuário na rede. É oferecido como alternativa para que os usuários não registrados possam confirmar a existência ou não de uma *watermark* da DIGIMARC em qualquer imagem.

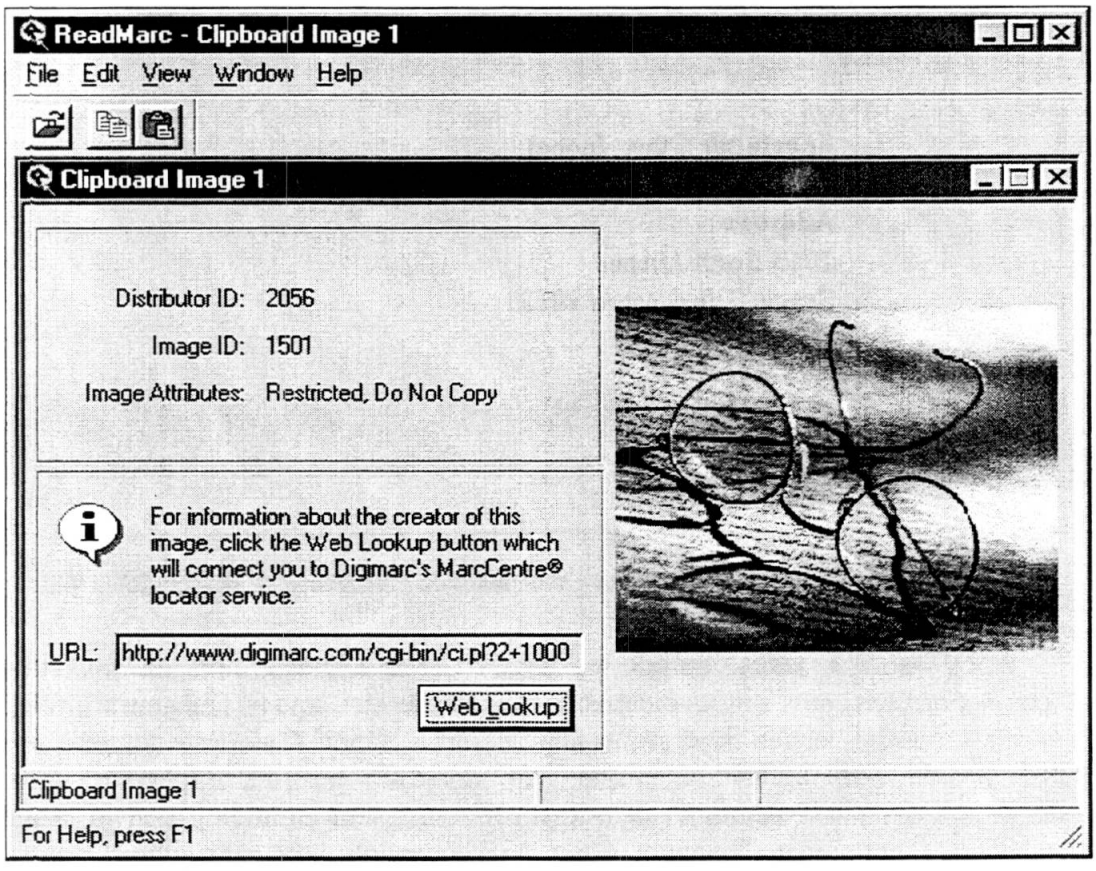


FIGURA 5.14 – Tela do Readmarc exibindo informações da *watermark*

Por fim, a DIGIMARC possui um outro aplicativo muito importante: o MARCSPIDER. Disponível apenas para os usuários registrados, este aplicativo vasculha a *Web* procurando por imagens marcadas com *watermarks* da DIGIMARC e relata os resultados. Com isto, os proprietários de imagens podem identificar locais que estão usando suas imagens legalmente ou ilegalmente na rede.

A solução da DIGIMARC é particularmente interessante, pois permite ao proprietário descobrir exatamente onde suas imagens estão sendo utilizados na *Web*. E qualquer um na Internet pode descobrir quem é o proprietário de determinada imagem

divulgada. Neste sentido, a sistemática da DIGIMARC permite uma comunicação de duas vias:

- ao autor, quando procura por pessoas que estão utilizando sua imagem, legalmente ou ilegalmente;
- aos interessados ou piratas, ao serem informados pelos aplicativo de que aquela imagem está protegida por uma *watermark* da DIGIMARC.

5.7 Suresign

SIGNUM TECHNOLOGIES [SIN 99] oferece um grupo de ferramentas para inserir e recuperar marcas em imagens. A solução, muito semelhante àquela oferecida pela DIGIMARC, inclui um *plug-in* para ADOBE PHOTOSHOP (SURESIGN WRITER/DETECTOR), uma aplicação de processamento *batch* (SURESIGN SERVER) e um kit para desenvolvedores (SURESIGN SDK).

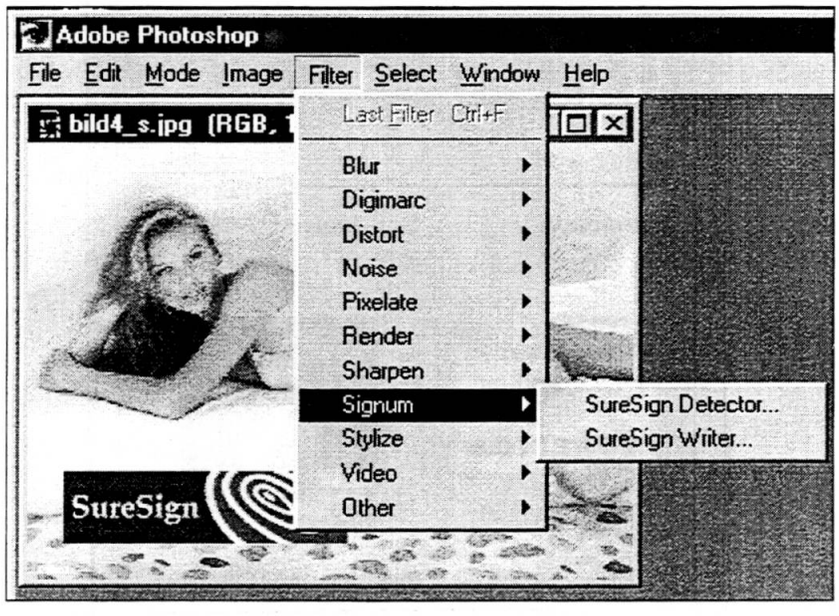


FIGURA 5.15 – Tela do Adobe Photoshop mostrando acesso ao filtro da SureSign

Para utilizar o *plug-in* SURESIGN WRITER é necessário adquirir um código único de identificação de proprietário de imagem (*Fingerprint ID*) da SIGNUM, que permitirá seu uso durante um ano. Enquanto não for adquirido este código, o WRITER será executado em modo de demonstração e estará habilitado a embutir apenas um código padrão. Isto é útil durante a fase de avaliação do software, mas não permite qualquer proteção real, pois este código padrão não identifica o proprietário da imagem, trata-se apenas de uma marca da própria SURESIGN.

Além do *FingerPrint ID*, o WRITER requer um identificador único para imagem: o *Image ID*. E, conforme pode ser verificado na figura 5.17, o WRITER tem ainda a opção de escolher e controlar o nível de transparência de um *bitmap* a ser colocado como logotipo na imagem.

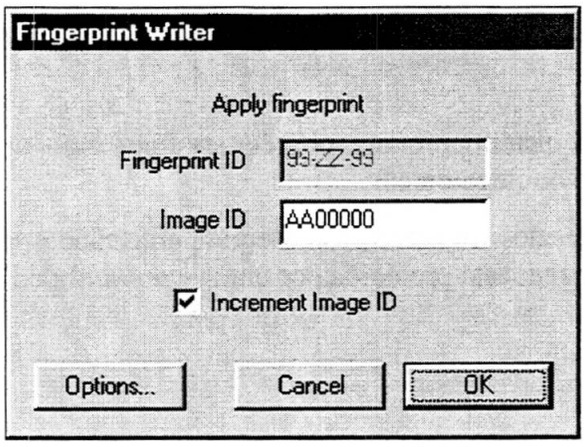


FIGURA 5.16 – Tela do SureSign Writer

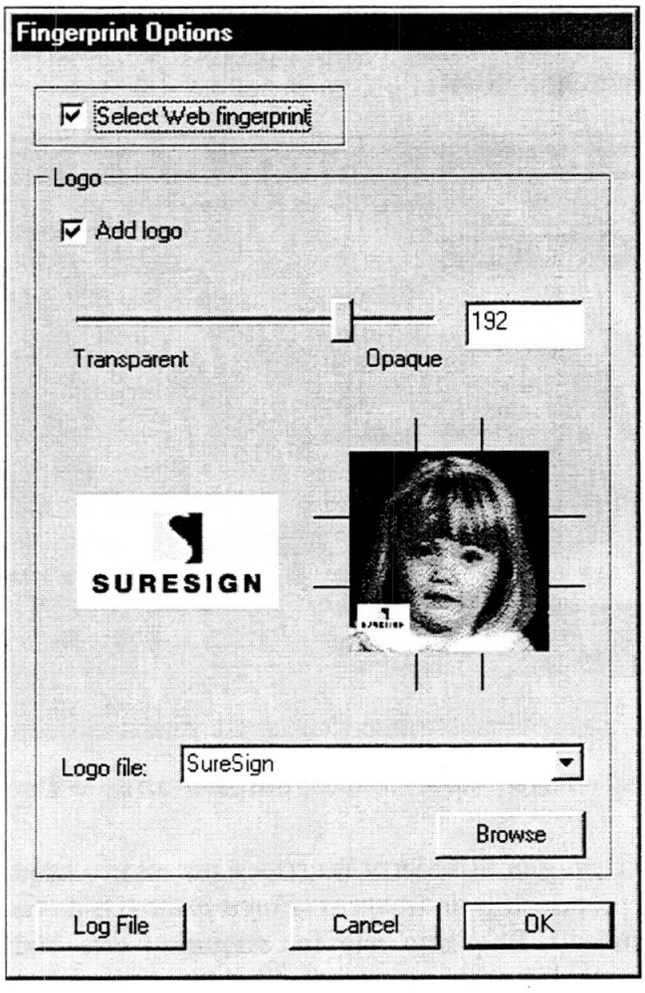


FIGURA 5.17 – Tela de opções do SureSign Writer

O *plug-in* SURESIGN DETECTOR, distribuído gratuitamente, recupera a marca inserida pelo WRITER nas imagens e fornece todos os dados para acesso direto ao REGISTRY. Este é uma base de dados interativa, acessada através do *site* da SIGNUM e que contém dados atualizados de contato dos proprietários de imagens marcadas com o SURESIGN. O acesso ao REGISTRY está disponível gratuitamente, 24 horas por dia. Desta

forma, qualquer um pode examinar as imagens de seu interesse com o DETECTOR e, se estiverem marcadas, localizar seus proprietários através do REGISTRY.

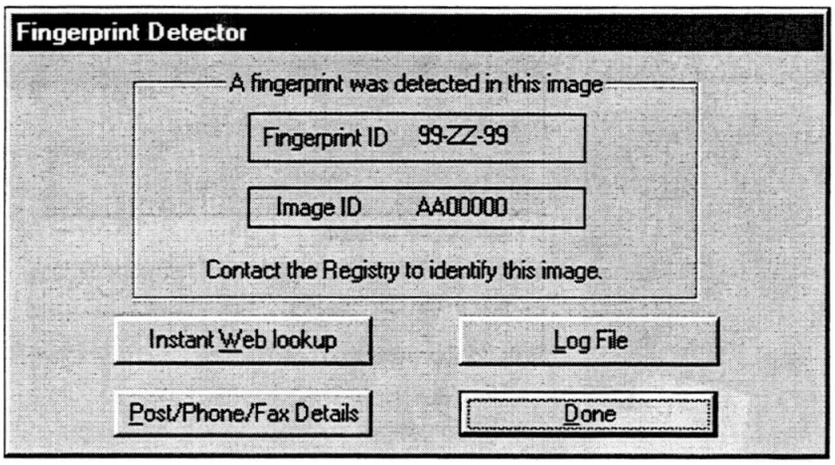


FIGURA 5.18 – Tela do SureSign Detector

Na figura 5.18 é o botão **INSTANT WEB LOOKUP** que abre o navegador, conecta com o *site* da SIGNUM, acessa o REGISTRY e traz uma página com os dados do proprietário da imagem marcada, conforme a figura 5.19.

User ID: **99-ZZ-99**

Contact name: **SureSign Demonstration Software User**

SURESIGN

Address:
SureSign Demonstration Software User
Anything
Anyone's Organisation
Anyone's Building
Anyone's Street
Anyone's Town
Anyone's Zip Code
Anywhere in the World

Phone: **+44 (0) 1242 580555**
Fax: **+44 (0) 1242 251600**

Email: signum@signumtech.com

URL: http://www.signumtech.com

FIGURA 5.19 – Tela do *site* da SureSign com dados do proprietário da imagem [SIN 99]

5.8 Steganos

Em termos de proteção de direitos autorais, a mais recente novidade comercial na Internet é o STEGANOS [DEM 99], desenvolvido pela DEMCOM para a plataforma Windows 95/98/NT.

Com uma interface muito amigável (figura 5.20 e 5.21), no estilo *Wizard*, o aplicativo permite criptografar e esconder qualquer tipo de informações em arquivos nos formatos BMP, DIB, WAV, VOC, TXT ou HTML.

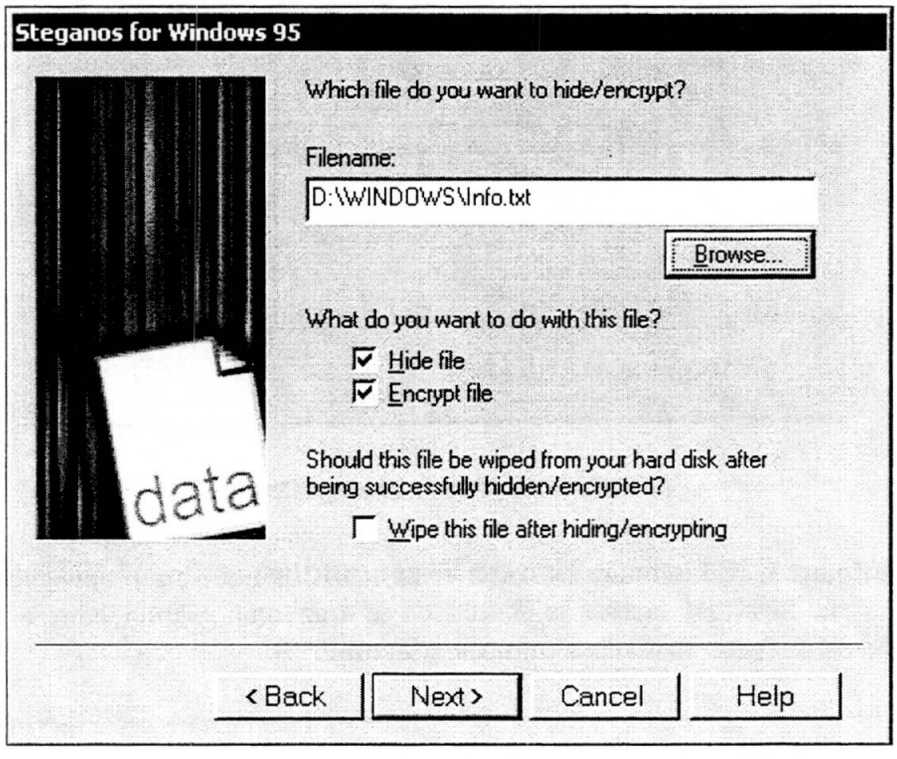


FIGURA 5.20 – Tela do Steganos com informações a serem escondidas

Em termos de criptografia, STEGANOS utiliza o algoritmo HWY1, o qual é compatível com RC4. Para esconder informações são utilizados dois diferentes métodos. Em arquivos de áudio e imagem utiliza-se o conhecido método LSB. Já para arquivos textos utiliza-se o método *word-and-line-shifting*.

Antes de esconder as informações, o aplicativo também compacta-as, utilizando as rotinas de *zlib,* desenvolvido por JEAN-LOUP GAILLY e MARK ADLER. Estes são membros do INFO-ZIP GROUP e mantém o código para o popular programa WINZIP.

Alternativamente, o aplicativo permite que as informações sejam apenas criptografadas, gerando um arquivo SEF (*Steganos Encrypted File*), ou apenas escondidas, no seu formato natural, sem qualquer transformação criptográfica.

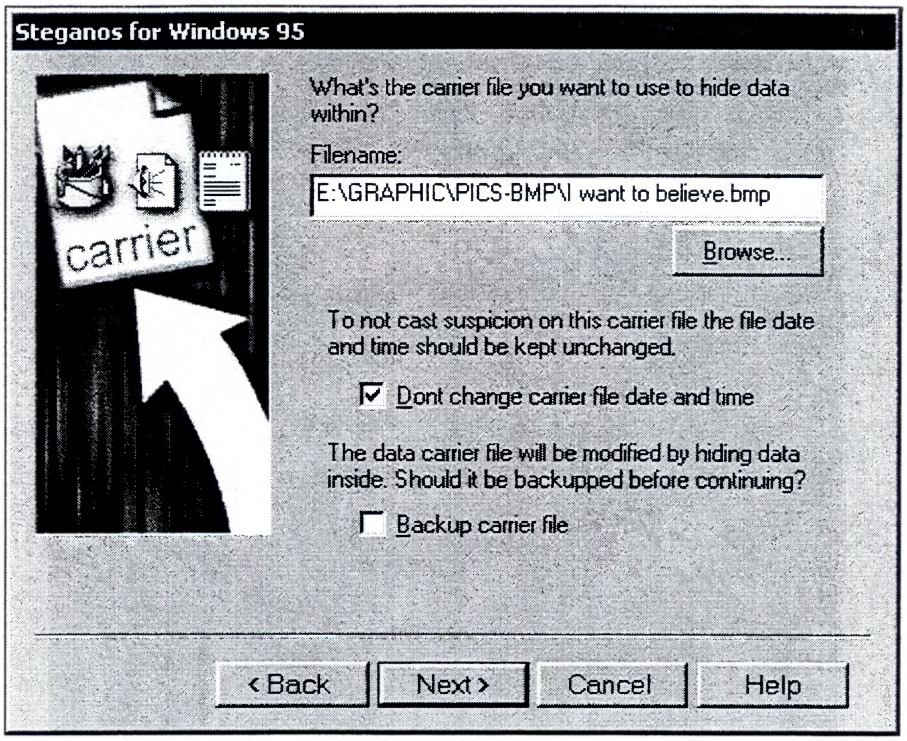


FIGURA 5.21 – Tela do Steganos com informações do arquivo hospedeiro

5.9 Unzign e Stirmark

UNZIGN [UNZ99] e STIRMARK [KUH99] são dois aplicativos "*dewatermarkers*". Ou seja, aplicativos que testam a robustez das marcas inseridas em arquivos de imagem. A estratégia é tentar localizar e recuperar a marca inserida, ou simplesmente destruir qualquer vestígio da marca. A maioria dos ataques detalhados em [KUT99A] estão implementados nestes aplicativos.

Assim, a partir de uma imagem marcada, estes aplicativos aplicam as transformações possíveis com vários parâmetros, gerando uma série de novos arquivos da mesma imagem. Estas então devem ser testadas com os programas originais de extração da marca e caso a marca não seja localizada e identificada em algum dos arquivos, a destruição da marca foi feita com sucesso. O processo completo pode ser automatizado usando um simples arquivo *batch*.

PETITCOLAS [PET99C] realizou vários testes de avaliação de ambos aplicativos, com imagens marcadas pelos mais conhecidos sistemas de *watermark*. Um estudo pormenorizado com o STIRMARK está disponível em [PET99D]. Como pode ser verificado pelas figuras 5.22 a 5.25, a destruição das marcas, por qualquer um dos dois aplicativos, é imperceptível aos olhos humanos, não comprometendo em nada a qualidade da imagem.

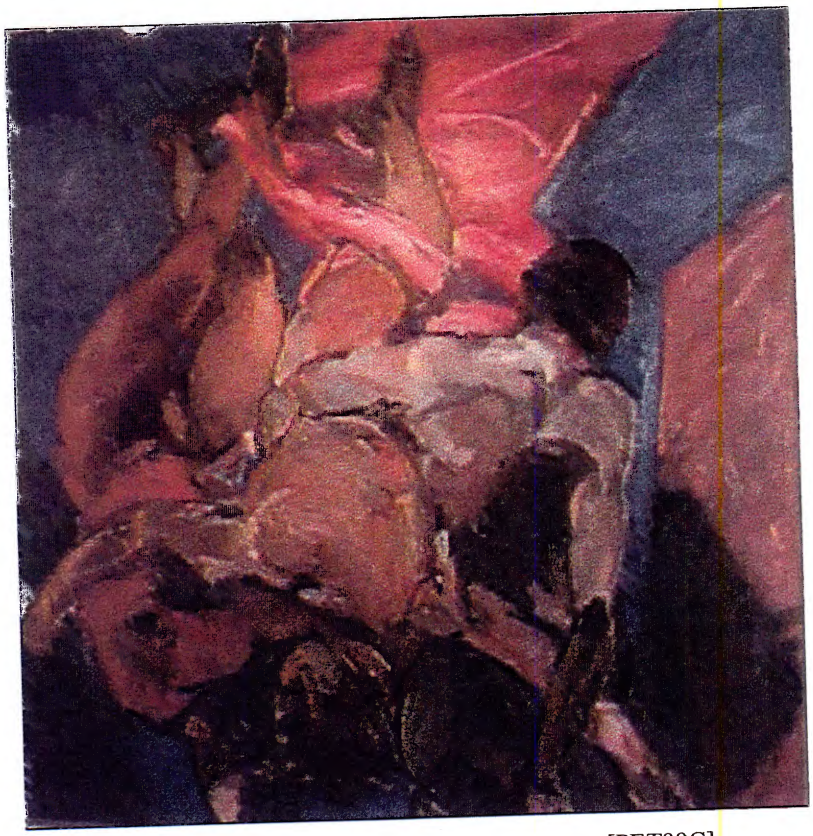


FIGURA 5.22 – Imagem original sem marca [PET99C]

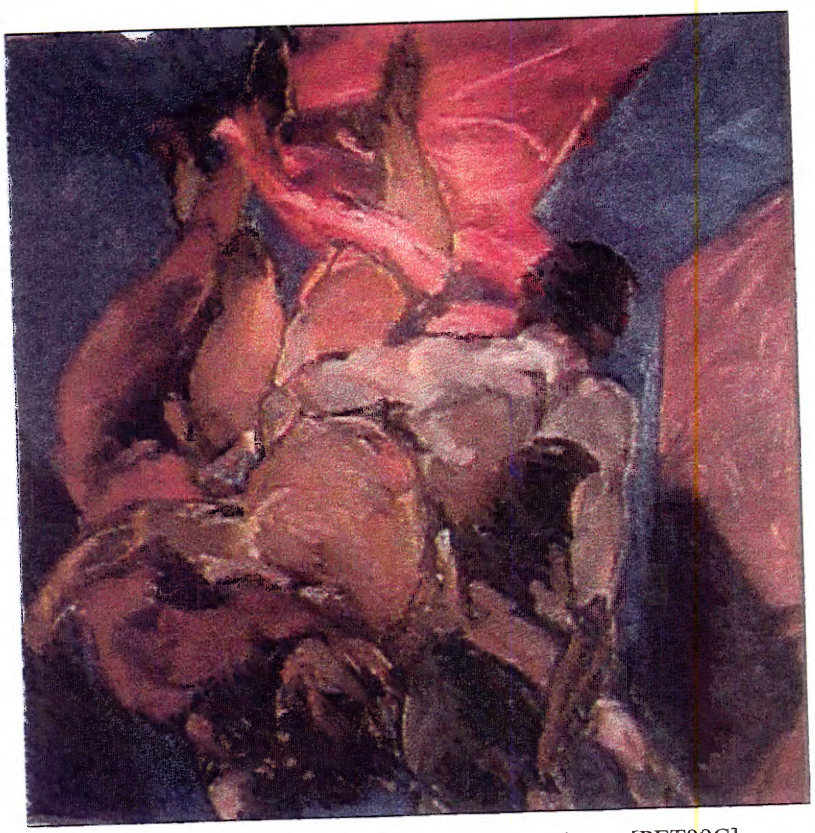


FIGURA 5.23 – Imagem marcada com Digimarc [PET99C]

FIGURA 5.24 – Imagem marcada apos destruição da marca com Stirmark [PET99C]

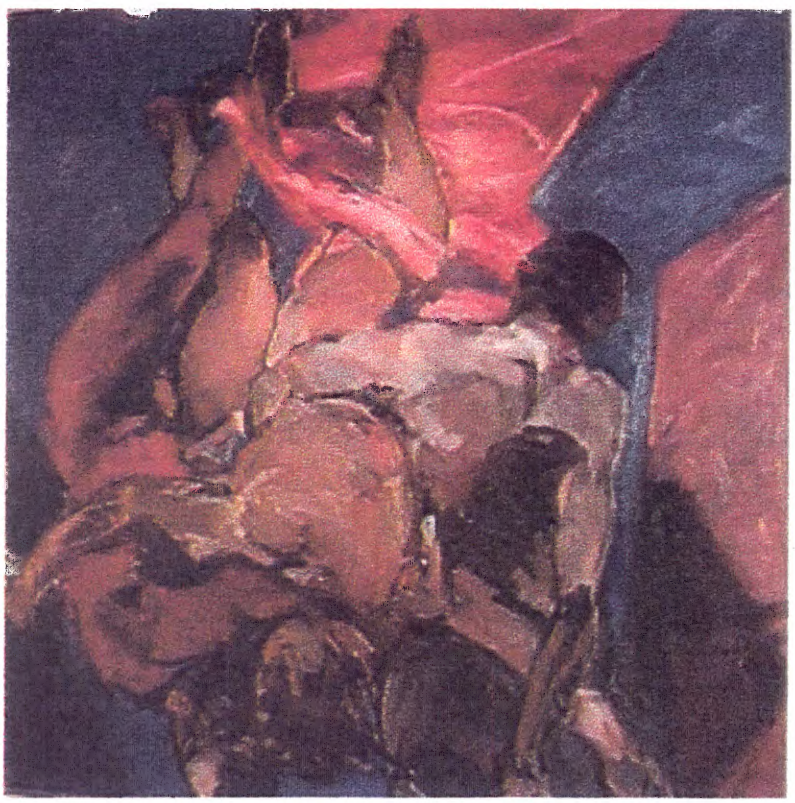


FIGURA 5.25 – Imagem marcada apos destruição da marca com Unzign [PET99C]

5.10 DocMark

Resultado de um projeto na Universidade de Columbia, DOCMARK [SON98] é um sistema de distribuição de documentos baseado na arquitetura proposta em [CHO94]. O objetivo do projeto foi, a partir da arquitetura básica de [CHO94], levar para os computadores dos clientes, a pesada tarefa de inserir uma marca nos documentos.

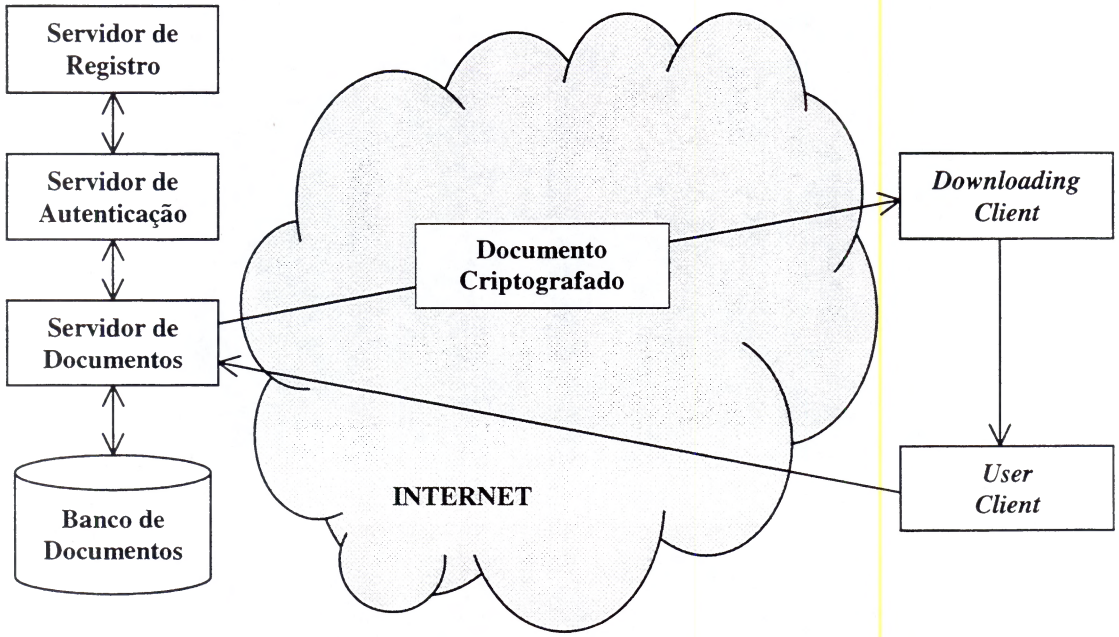


FIGURA 5.26 – Arquitetura de DocMark proposta em [SON98]

SERVIDOR DE DOCUMENTOS

Este módulo é a interface para o banco de dados de documentos. As principais tarefas deste módulo são:

a) receber os pedidos enviados pelos usuários;

b) encaminhar a identificação do usuário ao SERVIDOR DE AUTENTICAÇÃO;

c) construir dinamicamente o módulo *DONWLOADING CLIENT* para suportar especificamente os documentos solicitados;

d) criptografar os documentos a serem enviados;

e) transferir o módulo *DONWLOADING CLIENT* e os documentos ao cliente.

SERVIDOR DE AUTENTICAÇÃO

Este módulo é responsável por autenticar o usuário antes de ocorrer qualquer transferência de dados para o mesmo. Se a autenticação falhar, o SERVIDOR DE DOCUMENTOS interrompe a transação. O processo de autenticação é feito através de usuário e senha. Após cada autenticação, este módulo encaminha informações para o SERVIDOR DE REGISTRO.

SERVIDOR DE REGISTRO

Este módulo é responsável por todo o controle contábil e financeiro da conta do usuário.

USER CLIENT

Este módulo inicia toda a transação encaminhando os pedidos ao *SERVIDOR DE DOCUMENTOS*. Cabe também a este módulo receber o *DOWNLOADING CLIENT* para o computador do usuário.

DOWNLOADING CLIENT

Este módulo recebe do SERVIDOR DE DOCUMENTOS todos os documentos solicitados, ainda no formato criptografado. Em seguida, decifra os documentos, insere a *watermark* e converte o documento para o formato de visualização (*bitmap*).

A grande inovação da arquitetura de DOCMARK, para aquela proposta por [CHO94], é repassar para o computador do cliente toda a pesada tarefa de personalização da cópia, momento em que é inserida a *watermark* identificando autor e leitor.

Para enviar o documento a salvo até o computador do cliente, o sistema criptografa a versão original do documento, utilizando um criptosistema simétrico, onde a chave é o número do cartão de crédito do usuário, obtido do SERVIDOR DE REGISTRO. Os autores baseiam-se no fato de que os clientes ficariam relutantes em repassar este número para terceiros. Todavia, os próprios autores admitem a fragilidade desta técnica e sugerem que poderia ser utilizado um criptosistema assimétrico.

A documentação não é clara a respeito do conteúdo da *watermark* que é inserida nos documentos; presume-se que seja o *identificador do usuário*. Também não é fornecido nenhuma ferramenta que recupere e mostre a *watermark* de um documento qualquer, já no formato de visualização.

5.10.1 Proposta de Choudhury

Em [CHO94] o autor propõe duas arquiteturas para a distribuição eletrônica de documentos: uma baseada na utilização por hardware e outro por software. Para o

presente trabalho é oportuno analisar apenas a arquitetura baseada em software, que é composta por quatro elementos:

a) ***COPYRIGHT SERVER***: autentica os pedidos dos leitores

b) ***DOCUMENT SERVER***: marca, criptografa e comprime o documento solicitado

c) ***DISPLAY CLIENT***: decifra e apresenta os documentos recebidos

d) ***PRINTING CLIENT***: decifra e imprime os documentos recebidos

Inicialmente, o documento desejado é marcado e então criptografado com uma chave simétrica aleatória. Em seguida, esta chave simétrica é criptografada com a chave pública do leitor. Ambos arquivos são enviados para o leitor, que informa para o módulo *display* ou *printing* sua chave privada, para então então reverter todo processo de criptografia e permitir o uso do documento.

5.10.2 Limitações

Apesar de resolver o problema da sobrecarga no servidor de documentos, responsável por marcar, criptografar e comprimir todo documento desejado, o DOCMARK ainda mantém algumas importantes limitações da arquitetura proposta por [CHO94], a saber:

a) inexistência de controle de acesso: uma vez que o documento chegou ao leitor, não há qualquer controle do autor sobre as operações efetuadas (impressão, cópia e outros) com o hiperdocumento enviado;

b) desconhecimento de tentativas de acesso não autorizado: como o sistema é baseado na identificação de usuário e senha, o autor desconhece qualquer tentativa de corromper o sistema de segurança;

c) não são consideradas situações especiais como Hiperdocumento Público, Leitor Anônimo, Autor Anônimo, Controle de Distribuição, Hiperdocumento Nômade, conforme detalhadas no capítulo 6.

6 Proposta de Solução

Neste capítulo será apresentada uma proposta de arquitetura para controlar e proteger os hiperdocumentos na Internet. Inicialmente são descritas as premissas desta proposta. Em seqüência, é apresentada a arquitetura e sua dinâmica de funcionamento. Depois são detalhadas as permissões de uso do hiperdocumento, o arquivo de chaves de leitores autorizados e as mensagens de controle de uso do hiperdocumento. Também são discutidas situações importantes como as de leitores anônimos, hiperdocumentos públicos, autores anônimos e hiperdocumentos nômades. Por fim, são descritas as limitações conhecidas.

6.1 Premissas

Este trabalho pretende colaborar com a pesquisa de proteção de direitos autorais, propondo um nova arquitetura baseada no DocMark (vide item 5.9) para controlar e proteger os direitos de autores de hiperdocumentos. As premissas básicas desta proposta são:

- a arquitetura deve ser de fácil implantação e utilização, de forma que possa ser amplamente utilizada na Internet. A solução não deve representar nenhum recurso ou controle de segurança novo para os *sites* de publicação dos hiperdocumentos. O acesso deve ser tão simples quanto o que está atualmente em uso;

- o leitor deve ter acesso apenas a hiperdocumentos marcados. Nenhum hiperdocumento pode estar no *site* de publicação de forma desprotegida, como no seu formato original;

- a partir de qualquer cópia do hiperdocumento deve ser possível identificar o autor do mesmo. Ou se este desejar permanecer no anonimato, devem ser fornecidos meios adicionais que permitam o autor provar sua autoria a qualquer tempo. A identificação deve ser uma informação relativamente pública, que seja de alguma forma disponível a qualquer interessado;

- para rastrear e identificar fontes de pirataria, toda cópia do hiperdocumento deve ser individualizada, contendo informações do leitor que teve acesso ao hiperdocumento;

- o processo de individualização das cópias não deve onerar o processamento do servidor do *site* de publicação. Por este motivo, a individualização deve ser feita no computador do leitor;

- o autor deve ser comunicado de todas as operações efetuadas com seus hiperdocumentos: individualização, violação de segurança, cópia, impressão

e outras, conforme o que o autor especificar quando da disponibilização do hiperdocumento no *site* de publicação;

- a cada acesso ao hiperdocumento, o leitor deverá ser devidamente identificado e autenticado.

6.2 Arquitetura

A figura 6.1 apresenta a arquitetura proposta. A Internet, por não ser possível delimitar sua abrangência, é representada por uma grande nuvem. O autor e o leitor aparecem em extremos diferentes da Internet, justamente para ilustrar que estão em locais físicos diferentes e comunicando-se através da Internet. O Módulo Leitor e o módulo autor, que suportam essa arquitetura, aparecem juntos com os seus respectivos usuários. O *site* de publicação e o servidor de chaves públicas aparecem dentro da Internet, demonstrando que não são únicos e não estão juntos do autor nem do leitor. As setas contínuas representam o fluxo de mensagens obrigatórias entre os elementos da arquitetura. As setas pontilhadas representam uma comunicação opcional, que ocorrerá apenas em determinadas situações.

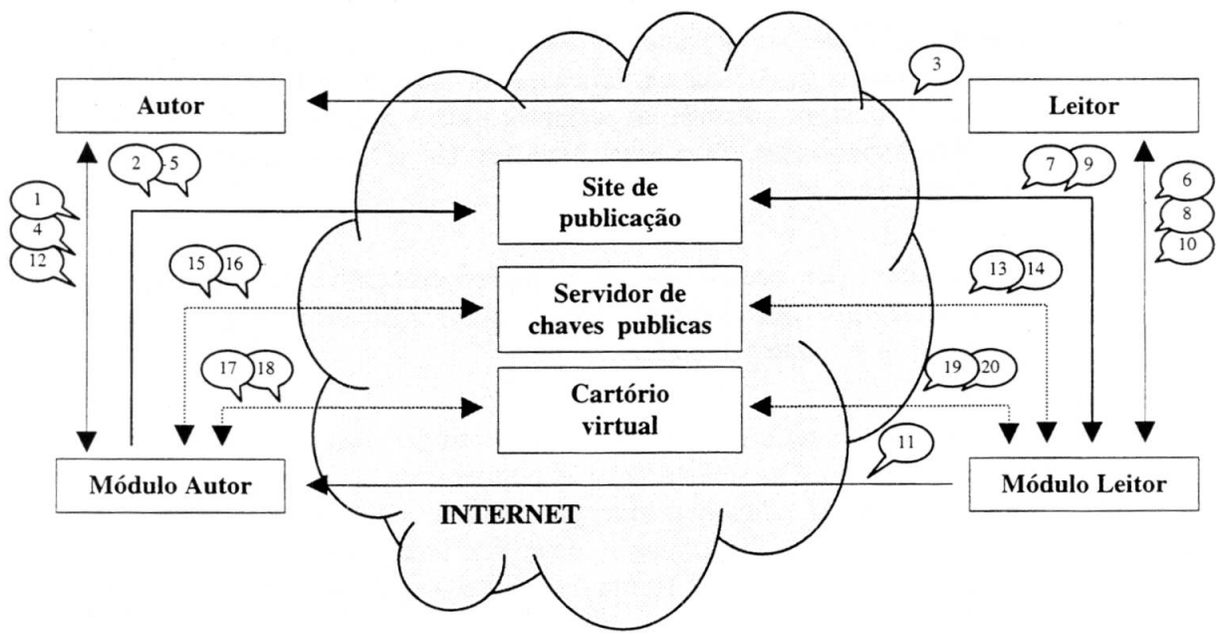


FIGURA 6.1 – Arquitetura proposta

Conforme ilustra a figura 6.1, a arquitetura está baseada nos seguintes elementos chave:

AUTOR

Pessoa proprietária dos direitos autorais do hiperdocumento.

LEITOR

Pessoa interessada em ter acesso ao hiperdocumento, seja para leitura, impressão, edição ou outro.

MÓDULO AUTOR

É um criptosistema e analisador de hiperdocumento. Utilizado exclusivamente pelos autores de hiperdocumentos, atua como interface entre o autor e a arquitetura proposta. Após identificar a árvore de conteúdo do hiperdocumento, criptografa cada um dos itens com a chave única do sistema. Cabe ainda a este módulo gerar o arquivo criptografado de chaves de leitores, contendo as chaves públicas e permissões de cada leitor autorizado, bem como receber as mensagens enviadas pelo Módulo Leitor e, após filtrar aquelas desejadas, exibi-las para o autor.

MÓDULO LEITOR

Trata-se de um criptosistema decodificador e de inserção de *watermark*. É responsável por decodificar os arquivos gerados pelo Módulo Autor e autenticar a identificação do leitor, verificando se sua chave privada confere com a chave pública dos leitores autorizados, ou com a chave pública do autor (também inserida na *watermark*). Ainda lhe cabe gerar a cópia marcada do hiperdocumento, contendo a identificação e chave pública do leitor, além de comunicar todas as operações deste ao autor. Posteriormente, este módulo deverá autenticar o leitor a cada acesso, baseando-se na marca existente no hiperdocumento, e comunicar as operações ao autor.

SITE

Um servidor qualquer conectado na Internet, onde o hiperdocumento será publicado. Isto é, onde o hiperdocumento ficará disponível para acesso pelos leitores. Não é necessário nenhum recurso adicional de *hardware* ou *software* para este *site*, além dos normalmente utilizados na disponibilização de uma *home page* qualquer.

SERVIDOR DE CHAVES PÚBLICAS

Servidor na Internet que cadastra usuários e gera um par de chaves assimétricas, permitindo acesso público ao seu cadastro de usuários e respectivas chaves públicas.

CARTÓRIO VIRTUAL

Agente independente na Internet que recebe hiperdocumentos para registro e, opcionalmente, as mensagens de controle de acesso para distribuição entre os interessados autorizados, como por exemplo o autor e o editor. Atua como mediador entre as partes interessadas na distribuição eletrônica de hiperdocumentos, certificando os dados e partes envolvidas.

A tabela 6.1 descreve o conteúdo e elementos envolvidos em cada mensagem representada pelos balões de diálogos na figura 6.1

TABELA 6.1 – Quadro de interação da arquitetura

	De	Para	Mensagem
1	Autor	Módulo Autor	Informa hiperdocumento original, sua chave pública e o identificador único do hiperdocumento, para gerar hiperdocumento mestre. Escolhe também as mensagens de controle de acesso que deseja receber.
2	Módulo Autor	*Site* Internet	Envia hiperdocumento mestre
3	Leitor	Autor	Envia pedido de acesso ao hiperdocumento
4	Autor	Módulo Autor	Informa chave e permissões do leitor
5	Módulo Autor	*Site* Internet	Envia arquivo de chaves e permissões de leitores
6	Leitor	Módulo Leitor	Solicita acesso ao hiperdocumento
7	Módulo Leitor	*Site* Internet	Baixa arquivo de chaves e permissões de leitores
8	Módulo Leitor	Leitor	Solicita pedido de chave privada e autentica leitor
9	*Site* Internet	Módulo Leitor	Baixa hiperdocumento mestre e gera cópia marcada
10	Módulo Leitor	Leitor	Apresenta hiperdocumento marcado
11	Módulo Leitor	Módulo Autor	Operações com o hiperdocumento: acessos, cópias, tentativas de quebra de segurança
12	Módulo Autor	Autor	Filtra as operações recebidas e repassa para autor aquelas especificadas no item 1, acima
13	Módulo Leitor	Serv. chaves públicas	No caso de leitor anônimo ou hiperdocumento público, solicita chave pública de leitor
14	Serv.chaves públicas	Módulo Leitor	Envia chave pública do leitor solicitado
15	Módulo Autor	Serv.chaves públicas	Solicita a chave pública de determinado leitor
16	Serv.chaves públicas	Módulo Autor	Envia a chave pública do leitor solicitado
17	Módulo Autor	Cartório Virtual	Envia cópia do hiperdocumento para registro de autoria
18	Cartório Virtual	Módulo Autor	Envia mensagens recebidas do módulo autor
19	Módulo Leitor	Cartório Virtual	Envia mensagens de controle de uso do hiperdocumento distribuído por editora
20	Cartório Virtual	Módulo Leitor	Envia dados de registro da cópia distribuida por editora

6.3 Dinâmica de Funcionamento

A seguir é detalhado a dinâmica de funcionamento da arquitetura. A descrição segue o mesmo formalismo utilizado pelos autores [WEB98] e [ARR93] para detalhar o funcionamento dos algoritmos de criptografia.

O funcionamento inicia com o autor, que através de uma função insere uma *watermark* em seu hiperdocumento original, gerando uma cópia mestre, marcada com sua chave pública e o identificador único do hiperdocumento.

$$h_m = W(k_e^{autor}, h, h_{ID}) \tag{6.3.1}$$

onde: h_m = hiperdocumento mestre

W = função esteganográfica de inserção de *watermark*

k_e^{autor} = chave pública do autor

h = hiperdocumento original

h_{ID} = identificador único do hiperdocumento. Este idenficador é um número qualquer para o sistema, fica a cargo do autor gerenciar a integridade deste número com o respectivo hiperdocumento.

A cópia mestre é então transformada através de uma função de criptografia com a chave única do sistema e enviada para o *site* de publicação:

$$h_c = E(k_e^{sistema}, h_m) \tag{6.3.2}$$

onde: h_c = hiperdocumento criptografado

E = função criptográfica para cifragem de dados

$k_e^{sistema}$ = chave pública interna do sistema

h_m = hiperdocumento mestre

Os leitores enviam mensagens ao autor, possivelmente por *e-mail*, com assinatura digital, requisitando acesso ao hiperdocumento:

$$m_a = E(k_d^{leitor}, m) \tag{6.3.3}$$

onde: m_a = mensagem assinada

E = função criptográfica para cifragem de dados

k_d^{leitor} = chave privada do leitor

m = mensagem do leitor

Além de assinada, a mensagem também pode ser criptografada com a chave pública do autor, garantido que somente este possa ler a mensagem:

$$m_c = E(k_e^{autor}, m_a) \tag{6.3.4}$$

onde: m_c = mensagem criptografada

E = função criptográfica para cifragem de dados

k_e^{autor} = chave pública do autor

m_a = mensagem assinada

Inicialmente o autor decifra o pedido com a sua chave privada, tendo acesso à mensagem assinada enviada pelo leitor:

$$m_a = D(k_d^{autor}, m_c) \tag{6.3.5}$$

onde: m_a = mensagem assinada

k_d^{autor} = chave privada do autor

D = função criptográfica para decifragem de dados

m_c = mensagem criptograda

Em seguida o autor decifra a mensagem assinada com a chave pública do leitor, garantindo a identidade do mesmo:

$$m = D(k_e^{leitor}, m_a)) \tag{6.3.6}$$

onde: m = mensagem do leitor

k_e^{leitor} = chave pública do leitor

D = função criptográfica para *de*cifragem de dados

m_a = mensagem assinada

O autor inclui as informações sobre o leitor, a chave pública e permissões no arquivo de chaves autorizadas:

$$a = \left((UserID), (UserName), (S, R, C, P, E), (k_e^{leitor})\right) \tag{6.3.7}$$

onde: a = arquivo de chaves e permissões

UserID = identificador único do leitor

UserName = nome do leitor

S = permissão para armazenamento

R = permissão para leitura

C = permissão para cópia

P = permissão para impressão

E = permissão para edição

k_e^{leitor} = chave pública do leitor

O novo arquivo de chaves e permissões é criptografado com a chave única do sistema e disponibilizado juntamente com os demais arquivos no *site* de publicação:

$$a_c = E(k_e^{sistema}, a) \tag{6.3.8}$$

onde: a_c = arquivo de chaves e permissões criptografado

E = função criptográfica para cifragem de dados

$k_e^{sistema}$ = chave pública interna do sistema

a = arquivo de chaves e permissões

O leitor então acessa o hiperdocumento, ativando o Módulo Leitor, que solicita o identificador e a chave privada do leitor. Efetua o *download* do arquivo de chaves e permissões, decifrando-o com a chave única do sistema:

$$a = D(k_d^{sistema}, a_c) \tag{6.3.9}$$

onde: a = arquivo de chaves e permissões

D = função criptográfica para decifragem de dados

$k_d^{sistema}$ = chave privada interna do sistema

a_c = arquivo de chaves e permissões criptografado

Em seguida o Módulo Leitor inicia o processo de autenticação do leitor. Verifica se o identificador do leitor consta na lista de chaves e permissões. Caso não conste, nega o acesso ao hiperdocumento, comunicando o autor da tentativa de acesso não autorizado. Caso a chave do leitor conste no arquivo de chaves e permissões, segue-se o processo de autenticação. Primeiro, gera internamente um texto aleatório para teste. Em seguida, criptografa este texto de teste com a chave pública do leitor, obtida no arquivo de chaves e permissões, decifrado anteriormente:

$$t_c = E(k_e^{leitor}, t) \tag{6.3.10}$$

onde: t_c = texto de teste criptografado

E = função criptográfica para cifragem de dados

k_e^{leitor} = chave pública do leitor, obtido em arquivo de chaves e permissões decifrado

t = texto de teste gerado aleatoriamente pelo sistema

Continuando o processo de autenticação, o Módulo Leitor decifra o texto de teste com a chave privada informada pelo leitor:

$$t' = D(k_d^{leitor}, t_c) \tag{6.3.11}$$

onde: t' = texto de teste resultado da decifragem

D = função criptográfica para decifragem de dados

k_d^{leitor} = chave privada do leitor, obtida em arquivo de chaves e permissões decifrado

t_c = texto de teste criptografado

Agora, comparando o texto de teste original (*t*) com o texto de teste resultante da decifragem (*t'*), caso os textos forem exatamente iguais, o processo de criptografia por chaves assimétricas foi bem executado. Logo, o leitor está autenticado e será autorizado a acessar o hiperdocumento conforme as permissões registradas no arquivo de chaves e permissões. Caso o leitor não seja autenticado, isto é, os arquivos de testes (*t* e *t'*) não são iguais, o Módulo Leitor nega o acesso ao hiperdocumento e comunica o autor de tentativa de acesso não autorizado.

Dando prosseguimento ao processo, no caso de o leitor ser autorizado ou de o autor ter permitido a leitura pública, o Módulo Leitor realiza o acesso aos demais arquivos que compõem o hiperdocumento. Da mesma forma, o Módulo Leitor decifra os arquivos com a chave única do sistema:

$$h_m = D(K_d^{sistema}, h_c) \tag{6.3.12}$$

onde: h_m = *hiperdocumento mestrel*

D = função criptográfica para decifragem de dados

$k_d^{sistema}$ = chave privada interna do sistema

h_c = *hiperdocumento criptografado*

Módulo Leitor insere uma *watermark* em cada um dos documentos eletrônicos que compõem o hiperdocumento, gerando uma cópia individualizada. A marca é gerada a partir da chave pública do leitor.

$$h_w = W(K_e^{leitor}, h_m) \tag{6.3.13}$$

onde: h_m = hiperdocumento mestre

W = função estegnográfica de inserção de *watermark*

k_e^{autor} = chave pública do autor

h_w = cópia individualizada do hiperdocumento

O Modulo Leitor apresenta então a cópia individualizada do hiperdocumento para o leitor e comunica todas as operações deste para o autor.

6.4 Permissões de Acesso

Inicialmente, os hiperdocumentos possuem cinco tipos de permissões de acesso diferentes. Alguns destes tipos compreendem outros tipos. Por exemplo, a permissão para realizar cópias pressupõe uma permissão de leitura. Todavia esta permissão de leitura não precisa ser explicitamente atribuída.

ARMAZENAMENTO

O leitor tem permissão apenas para manter armazenada uma cópia do hiperdocumento. Esta opção é particularmente importante para o caso de centros

de registros tornarem-se uma realidade, tomando a posição de "cartórios virtuais". Assim, cópias dos hiperdocumentos poderiam ser enviadas para serem armazenadas nestes centros, mas sem ter seus conteúdos publicados. No caso de litígio, estes centros trariam ao processo suas cópias, confirmando o registro por determinada parte, em determinada data. A parte que efetuou o registro recuperaria o conteúdo do hiperdocumento armazenado, informando sua identificação e chave pública, que seria autenticada com a chave de autor contida na *watermark* do hiperdocumento.

LEITURA

O leitor tem o acesso mais simples: visualização e leitura do hiperdocumento. Inclui-se a permissão de executar sons, vídeos e animações.

CÓPIA

O leitor pode selecionar e copiar partes do hiperdocumento. Este controle pode utilizar os recursos de Copiar & Colar, ou através de uma opção de Exportar. Poderia ainda ser acrescentado algum nível de controle sobre o "quanto" pode ser copiado. Este nível de controle poderia ser definido pelo autor na criação da cópia mestre. Por exemplo, o autor poderia definir que os leitores podem apenas copiar 40% do seu conteúdo, ou podem realizar 10 operações de cópias. Ou ainda o leitor pode realizar 10 operações de cópia de, no máximo, 40% do seu conteúdo. Esta quantidade poderia ainda ser definida de forma especifica para cada LEITOR, usando-se o arquivo de chaves e permissões. Caberia ao Módulo Leitor manter este controle, escondendo esta informação na cópia do hiperdocumento.

IMPRESSÃO

O leitor pode imprimir o hiperdocumento. Um acesso interessante pode ser composto permitindo que o leitor apenas possa imprimir o hiperdocumento. Sabe-se que ainda hoje uma cópia em papel e uma cópia digital oferecem possibilidades diferentes de manuseio.

EDIÇÃO

O leitor pode editar o hiperdocumento. Neste caso, seria necessário utilizar uma arquitetura fortemente estruturada para o hiperdocumento com a adaptação de uma ferramenta de editoração, seria possível controlar a edição por partes do hiperdocumento, permitindo a edição compartilhada do mesmo. A complexidade deste controle cresce conforme a sua granularidade, se por documento, capítulo, parágrafo ou outro item de nível mais detalhado.

6.5 Arquivo de Chaves Autorizadas

A arquitetura propõe um DTD para o arquivo de chaves e permissões. A figura 6.2 apresenta a definição deste DTD.

```
<!-- DTD Identificação e Permissões de Leitores -->
<!ENTITY % doctype "Readers" >

<!--                Definição dos Elementos              -->
<!--      Elementos        Tags       Conteúdo -->
<!--      ---------------- ---- ------------------------ -->
<!ELEMENT Reader           - -  (RID,Rname,RPerm,RKey)>
<!ELEMENT RID              - -  (#PCDATA)>
<!ELEMENT RName            - -  (#PCDATA)>
<!ELEMENT RPerm            - -  (S|R|C|P|E)+>
<!ELEMENT RKey             - -  (#PCDATA)>

<!ENTITY SGML     "Standard Generalized Markup Language">
```

FIGURA 6.2 – DTD para o arquivo de Chaves e Permissões

Conforme o padrão SGML para DTD, a sintaxe de definição de um novo elemento segue o seguinte padrão de *markup*:

```
<!ELEMENT   element-name   omittag-info   content-model>
```

onde:	*<! ELEMENT*	*inicia uma definição de elemento*
	element-name	*nome do elemento*
	omittag-info	*composto por dois bytes, indicando quando o start-tag e end-tag da instância do elemento são obrigatórios ou opcionais. O valor '-' indica é que o tag é obrigatório e 'O' indica que o tag é opcional.*
	content-model	*descreve o tipo de dados ou markups adicionais.*
	>	*encerra a definição do elemento*

Para definição de tipos novos de atributos, o SGML utiliza a seguinte sintaxe:

```
<!ATTLIST element-name attribute-name values default-value>
```

onde:	*<! ATTLIST*	*inicia uma definição de atributo*
	element-name	*nome do elemento que conterá o atributo*
	attribut-name	*nome do atributo*
	values	*valores permitidos para o atributo*
	default-value	*valor default para o atributo*
	>	*encerra a definição do atributo*

Como pode ser visto na figura 6.2, o DTD proposto define cinco novos tipos de *markups*:

READER

Define uma instância de leitor. É composto por outros quatro tipos de *markups*: *RID*, *RName*, *RPerm* e *RKey*. O conector ',' (vírgula) define que todos os elementos devem ser informados e na mesma ordem que a definição.

RID

De tamanho livre, é o identificador único (*UserID*) do leitor, determinado pelo autor ou obtido em um servidor de chaves públicas por ocasião de seu registro. Deve iniciar com o *start-tag* <RID> e terminar com o *end-tag* </RID>.

RNAME

Também de tamanho livre, é o próprio nome ou apelido (*UserName*) do leitor. Deve iniciar com o *start-tag* <RName> e terminar com o *end-tag* </RName>.

RPERM

De pelo menos uma e até cinco ocorrências de permissões do leitor, sendo 'S' (*Store*) para Armazenamento, 'R' (*Read*) para Leitura, 'C' (*Copy*) para Cópia, 'P' (*Print*) para Impressão e 'E' para Edição (*Edit*). Deve iniciar com o *start-tag* <RPerm> e terminar com o *end-tag* </RPerm>. O conector '+' (sinal de adição) define que devem existir uma ou mais ocorrências do elemento.

RKEY

É a chave pública (*PublicKey*) de um par de chaves assimétricas do leitor. Apesar de ter tamanho livre, a possibilidade de inseri-la ou não depende do tamanho do hiperdocumento. Todos documentos eletrônicos (vídeo, som, imagem, texto e outros) que compõem o hiperdocumento devem ser marcados, mas tudo depende do tamanho da chave pública. Deve iniciar com o *start-tag* <RKey> e terminar com o *end-tag* </RKey>.

A figura 6.3 apresenta uma ocorrência do arquivo de chaves autorizadas, conforme o DTD da figura 6.2.

```
</Reader>
<RID>1234567890</RID>
<RName>Alice</RName>
<RPerm>S</RPerm>
<RPerm>R</RPerm>
<RPerm>C</RPerm>
<RPerm>P</RPerm>
<RPerm>E</RPerm>
<RKey>ABC123JKL123LKDGDUOIERUTRTOETIUERTEROITUE02439890243853
4IUOIUWERTIUWER092348508234059823 4IEROUOTIWUERTIOUWERTIOU4230
9582340985230948 52IUEROIU3420598FGJDFIGUSDIOGUDFOIU4095823409
58234098ERTIUWETROOTU304958230948502394852340 98EROIUTOWERIUTW
EORIUTIOWEU349058</RKey>
</Reader>
```

FIGURA 6.3 – Exemplo de Arquivo de Chaves e Permissões

6.5.1 Múltiplos Arquivos de Chaves Autorizadas

O crescimento do arquivo de chaves e permissões pode tornar inviável a manutenção de um único arquivo com esta função. Neste caso, as chaves autorizadas podem ser mantidas em múltiplos arquivos, organizados por faixas numéricas de identificação de leitores.

Desta forma, o Módulo Leitor, ao solicitar o *ReaderID* para um leitor, identificaria o arquivo de chaves autorizadas necessário, conforme a faixa numérica deste identificador, e procederia a recuperação apenas do arquivo pertinente.

6.6 Mensagens de controle de acesso

O Módulo Leitor comunica todas as operações efetuadas com o hiperdocumento, sejam elas de acesso efetuados e autorizadas ou de tentativas de acessos. Parâmetros diferentes são enviados junto com cada mensagem, conforme o seu tipo. A tabela 6.2 demonstra os tipos de mensagens inicialmente previstos, com seus respectivos parâmetros de informações.

TABELA 6.2 – Mensagens de Controle de Acesso

TIPO	MENSAGEM	PARÂMETROS
1	Tentativa de acesso remoto não autorizado	Identificador do hiperdocumento Endereço de rede Identificador e chave pública do leitor que tentou o acesso (dados informados na tela)
2	Tentativa de acesso local não autorizado	Identificador do hiperdocumento Endereço de rede Identificador e chave pública do leitor que tentou o acesso (dados informados na tela) Identificador do leitor que efetuou a individualização da cópia (recuperado na *watermark*)
3	Hiperdocumento aberto	Identificador do hiperdocumento Endereço de rede Identificador do leitor
4	Hiperdocumento impresso	Identificador do hiperdocumento Endereço de rede Identificador do leitor Ponto inicial e final de impressão
5	Hiperdocumento copiado	Identificador do hiperdocumento Endereço de rede Identificador do leitor Ponto inicial e final de cópia
6	Hiperdocumento armazenado	Identificador do hiperdocumento Endereço de rede Identificador do leitor
7	Hiperdocumento editado	Identificador do hiperdocumento Endereço de rede Identificador do leitor

6.7 Leitor Anônimo

A arquitetura proposta contempla também a situação em que o autor determina acesso para leitores anônimos. Ou seja, permissões públicas para seu hiperdocumento. Este recurso é interessante, por exemplo, quando o autor deseja permitir a leitura para todos, mas deseja também que apenas alguns possam copiar e outros possam imprimir. Para atender esta situação, a arquitetura necessita adicionar na estrutura do arquivo de chaves e permissões um controle interno, onde possam ser determinadas as permissões publicadas.

Todavia, ao permitir a existência de leitores anônimos cria-se um problema para o controle de acesso, pois a princípio não haveria como identificar os leitores. Neste caso, há duas alternativas. Na primeira alternativa, o autor pode simplesmente optar por não individualizar as cópias, ou seja, ele não deseja identificar nem controlar quem está acessando seu hiperdocumento. Para esta situação, a cópia apresentada para o leitor será sempre idêntica ao hiperdocumento mestre. Isto manterá a garantia de autoria do hiperdocumento, pois a versão que o leitor terá contato possuirá sempre a identificação do autor.

Na outra situação, onde o autor deseja o controle de acesso, é obrigatória a identificação do leitor para a individualização das cópias. Neste caso, o Módulo Leitor será responsável pela identificação do leitor. Para tanto, a arquitetura necessita uma complementação para interagir com os servidores de chaves públicas na Internet. Esta interação depende de firmar parcerias e estabelecer padrões de segurança e *layout* de informações para troca de mensagens. Com esta interação, o Módulo Leitor pode solicitar ao leitor sua identificação e o servidor de chaves públicas onde o mesmo está registrado. Após confirmar a existência do identificador, o Módulo Leitor recupera a chave pública e prossegue normalmente sua rotina.

6.8 Hiperdocumento Público

Ao manter todas as permissões como públicas, teremos uma situação em que o autor torna o hiperdocumento público, onde todos podem realizar qualquer ação com o hiperdocumento. Todavia, esta situação não representa que o autor abdicou de seu direito autoral. Pelo contrário, ele apenas permitiu o livre acesso e uso ao hiperdocumento, mas permanece com a notoriedade de sua autoria. Isto é, deseja ser reconhecido como autor daquele hiperdocumento.

A arquitetura atende plenamente estes casos ao gerar o hiperdocumento mestre, será inserido o identificador do autor no hiperdocumento, garantido o reconhecimento de sua autoria a qualquer tempo.

Da mesma forma que ocorre nos casos de leitores Anônimos, mesmo para hiperdocumentos públicos, o autor ainda pode optar, ou não, por manter a individualização das cópias e respectivos controles de acesso. Para isto deve-se apenas

complementar a arquitetura conforme descrito no final do item 6.6, para que seja possível interagir com servidores de chaves públicas.

Em virtude da possibilidade de um hiperdocumento públicos gerar um número grande de mensagens de controle, cabe ao autor avaliar bem a situação antes de ativar o controle de individualização e a comunicação de mensagens.

6.9 Autor Anônimo

Um autor pode, por algum motivo qualquer, optar por não declarar sua autoria sobre determinado hiperdocumento, mas pode adicionalmente desejar que ninguém se aproprie indevidamente da obra de sua autoria.

A arquitetura auxilia também nesta situação, bastando que seja omitida a identificação do autor ao gerar o hiperdocumento mestre. Assim, ao individualizar a cópia, a *watermark* inserida conterá apenas as informações sobre o leitor.

Supondo agora a situação em que algum impasse judicial seja criado por um leitor que acessou o hiperdocumento e, verificando que não havia autor identificado, resolveu declarar-se como autor. Tal situação pode ser resolvida se o legítimo autor se pronunciar e apresentar sua versão original. Assim, ao analisar as duas versões, aquela do falso autor deverá conter uma marca com sua identificação aparecendo como leitor. A versão que o legítimo autor apresentar não deverá conter qualquer marca. Logo, a versão do falso autor foi baseada na versão original do legítimo autor.

A falta de identificação do autor a princípio não permite a manutenção das características de controle de acesso, feito pelas mensagens do Módulo Leitor ao Módulo Autor. O Módulo Leitor também não teria condições de identificar para onde enviar as mensagens. Todavia, esta situação pode ser contornada, inserindo-se no lugar do identificador do autor algum endereço para envio das mensagens. A comunicação com este endereço deve ser mantido por algum protocolo de segurança e o conteúdo das mensagens conteria o autor como anônimo, mas teria a identificação do hiperdocumento e do leitor que o está acessando. Isto permite que o autor mantenha o controle de acessos através do identificador do hiperdocumento.

6.10 Hiperdocumento Nômade

Hiperdocumento nômade é a denominação proposta neste trabalho para os hiperdocumentos que não podem ser copiados, mas que podem circular entre os leitores. Desta forma, existirá uma única cópia do hiperdocumento, que ora estará com um determinado leitor, ora estará com outro leitor. Esta situação reflete o que ocorre com a original em papel de um documento, onde apenas um leitor tem acesso em determinado momento.

É importante ressaltar que durante o processo de inserção da *watermark*, uma porção pequena de informações é trocada pela marca. No caso de textos, estes dados representam formatações. No caso de outras mídias, estes dados representam conteúdo. A retirada desta *watermark*, ainda que se saiba exatamente o algoritmo de inserção, implica na degradação daquela cópia. Os dados perdidos não podem ser recuperados exatamente como eram no original, antes da inserção da *watermark*. Novos dados serão gerados aleatoriamente para o preenchimento da lacuna que se criará. Este processo de inserção e retirada da *watermark*, quando executado uma única vez, ou poucas vezes, permanece imperceptível aos humanos. Porém, ao executar seguidas vezes, o processo pode gerar uma degradação perceptível no hiperdocumento.

Por este motivo, não é possível fazer circular uma cópia do hiperdocumento e trocar sua *watermark* sem que haja uma degradação cumulativa, que ao final de vários ciclos acabará sendo perceptível pelos humanos. Para atender esta necessidade, a arquitetura proposta necessita algumas adaptações.

Inicialmente um processo complementar de controle de acesso ao original deve permanecer ativo no *site* de publicação. O Módulo Leitor, antes de acessar o hiperdocumento mestre, comunica-se com este processo e verifica se o hiperdocumento está disponível. Se estiver, o Módulo Leitor solicita o acesso e ganha um prazo para usar e controlar o hiperdocumento. A partir deste momento o hiperdocumento passa a estar indisponível no *site* de publicação. O Módulo Leitor, então, prossegue normalmente seus procedimentos, efetuando a individualização da cópia e emitindo as mensagens de controle de acesso.

Em um outro momento, o leitor que está com a cópia individualizada procede a liberação do hiperdocumento. Neste instante, o Módulo Leitor invalida a cópia individualizada e comunica o fato ao módulo de controle do original, que está ativo no *site* de publicação. Da mesma forma, ao esgotar o prazo de validade, o Módulo Leitor pode invalidar a cópia individualizada, requerendo a ação do leitor para sua revalidação junto ao *site* de publicação.

Durante o período em que o hiperdocumento estiver indisponível, o módulo de controle do original pode ser consultado para saber qual leitor está com o hiperdocumento.

A questão de hiperdocumentos nômades é bastante complexa e requer uma série de requisitos de segurança que, talvez, possam ser implementados com garantia de sucesso apenas em redes fechadas como as Intranets. Este fator, mesmo assim, não invalida a conveniência de seu estudo.

6.11 Controle de Distribuição

A distribuição eletrônica na Internet fará surgir elementos semelhantes aos existentes na distribuição física de documentos, que podem ser aprimorados. Por exemplo, as editoras são responsáveis por produzir e comercializar um lote de número

determinado de cópias de livros. Todavia, o autor não detém muitos controles sobre esta produção e comercialização. Nem sequer as cópia são numeradas, para que se possa realizar alguma forma qualquer de controle. Na maioria dos casos, resta ao autor apenas confiar que a editora cumprirá sua parte do acordo.

A arquitetura proposta pode alterar esta realidade para a distribuição eletrônica. Controles adicionais podem ser incorporados para a arquitetura, de forma que seja possível controlar um código seqüencial que identifica o número da cópia. Este código deve ser fornecido por um módulo de controle no *site* de publicação, e inserido junto com os demais dados na *watermark*. As mensagens de controle de acesso devem ser encaminhadas para um mediador, talvez o próprio cartório virtual previsto na arquitetura, que deve registrar tais mensagens e repassá-las para ambas as partes (autor e editora). Em caso de litígio entre o autor e a editora, o mediador pode pesquisar a verdade dos fatos em seus próprios registros, dispensando a necessidade de provas das partes.

6.12 Limitações Conhecidas

A principal limitação conhecida na arquitetura proposta é a proteção e sigilo das chaves internas do sistema. Estas chaves são utilizadas para manter a comunicação segura com o Módulo Leitor, seja para receber o hiperdocumento mestre e então proceder a individualização da cópia, seja para enviar as mensagens do controle de acesso ao Módulo Autor.

Durante a idealização da arquitetura proposta questionou-se a utilização de algum esquema de troca periódica destas chaves internas, onde o próprio sistema autenticaria os módulos leitores e trocaria sua chave de acesso. Porém, isto não acrescentaria um nível muito maior de segurança: uma vez quebrada a chave do sistema, o pirata estaria apto também a receber a troca desta chave. Todavia, este problema é conhecido e intrínseco ao estudo da criptografia, e não é foco central deste trabalho.

No próximo capítulo será apresentado o protótipo desenvolvido. Este protótipo implementa muitas das idéias proposta neste capítulo. Com isto se pretende demonstrar a viabilidade das soluções elaboradas.

7 Protótipo: HiperMark

Neste capítulo são tratados os aspectos de implementação do protótipo desenvolvido. Inicialmente é demonstrada a arquitetura do protótipo, depois são vistos os detalhes de implementação de cada módulo.

7.1 Arquitetura do Protótipo

Para demonstrar a viabilidade da arquitetura proposta, foi implementado o protótipo denominado HIPERMARK. O objetivo é exemplificar o uso da arquitetura com hiperdocumentos no formato HTML. Embora não seja totalmente funcional, o protótipo atende a muitos princípios da arquitetura proposta e permite que seja usado como núcleo de um sistema real.

Referindo-se à dinâmica de funcionamento da plataforma, o envio de mensagens com assinatura digital, conforme especificado nos itens 6.3.4 a 6.3.6, não foram implementados por serem hipoteticamente absorvidos por algum aplicativo de gerenciamento de chaves públicas (por exemplo, o PGP) em conjunto com alguma aplicação de correio eletrônico (por exemplo, o MICROSOFT OUTLOOK ou NETSCAPE MESSAGER).

Da mesma forma, as mensagens de acesso e tentativa de violação (itens 6.3.12 e 6.3.16), bem como a inserção da *watermark* (item 6.3.14) não foram implementadas no protótipo por não estarem diretamente relacionados com a proposição da arquitetura, mas como um aspecto de implementação e utilização real.

Todavia, a ausência destes itens não invalida a elaboração dos testes desejados. Pois, como foi dito, o principal objetivo do protótipo é a exemplificação da arquitetura proposta.

A figura 7.1 ilustra como ficou a arquitetura implementada no protótipo. De acordo com o mesmo padrão utilizado na figura 6.1, a abrangência da Internet é representada por uma grande nuvem, com autor e leitor em extremos diferentes. Apenas o *site* de publicação foi considerado para efeitos de protótipo, abstraindo todos os aspectos ligados aos servidores de chaves públicas e aos cartórios virtuais. O módulo autor possui um sub-módulo denominado *Engine*. Apenas para efeito de prototipação foi desenvolvido um utilitário denominado *KeyGen*. A finalidade deste utilitário é gerar chaves assimétricas aleatórias, suprindo assim a falta da interface com os servidores de chaves públicas. Devido as vantagens de implementação, o módulo leitor foi desenvolvido como um *plug-in* para Netscape Navigator. Aparecendo, assim, como um sub-módulo de um *browser* Internet

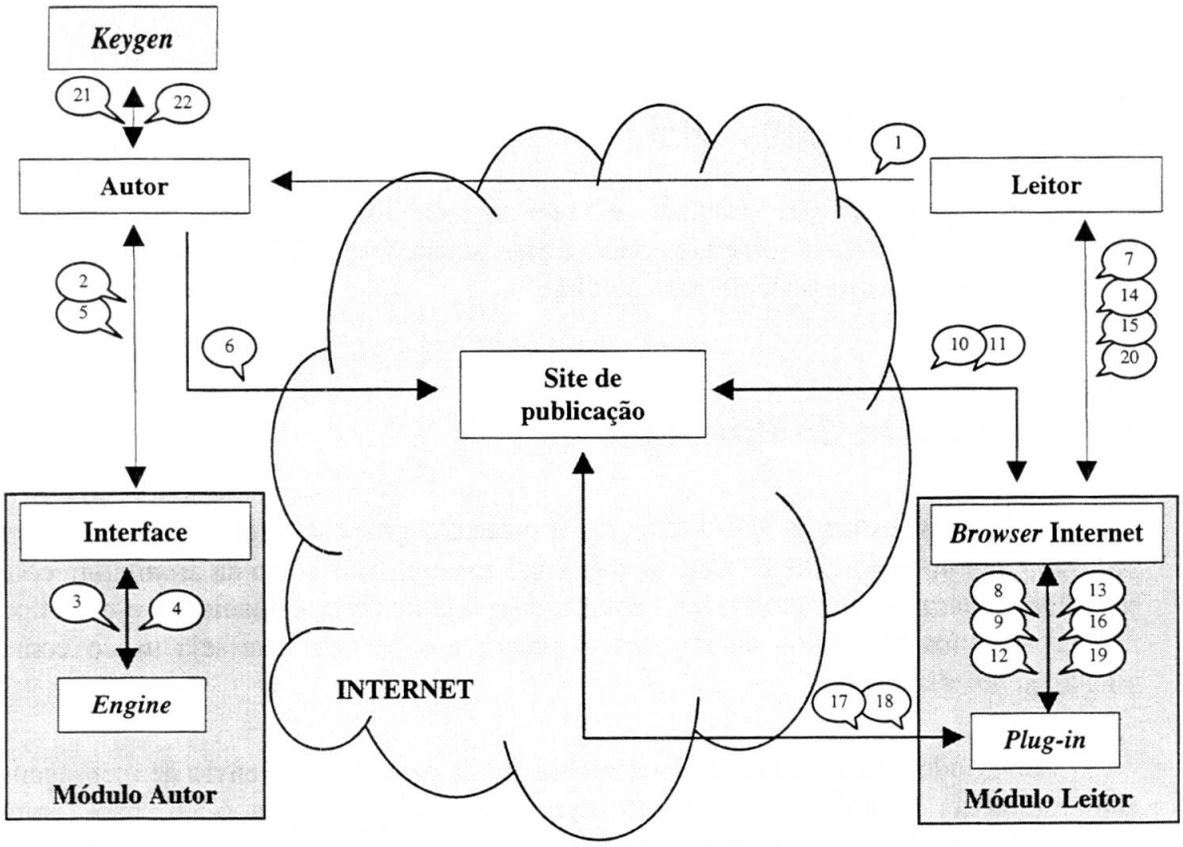


FIGURA 7.1 – Arquitetura do Protótipo

A tabela 7.1 descreve o conteúdo e elementos envolvidos em cada operação, representada pelas setas e balões de diálogos na figura 7.1

TABELA 7.1 – Quadro de Interação do Protótipo

	De	Para	Operação
1	Leitor	Autor	Solicita acesso ao hiperdocumento desejado.
2	Autor	Interface	Informa hiperdocumento, dados do leitor e permissões.
3	Interface	*Engine*	Informa o hiperdocumento e o arquivo de chaves e permissões.
4	*Engine*	Interface	Criptografa o hiperdocumento e o arquivo de chaves e permissões.
5	Interface	Autor	Avisa sobre geração dos arquivos criptografados.
6	Autor	*Site* Internet	Envia arquivos criptografados.
7	Leitor	*Browser*	Pede para abrir o hiperdocumento criptografado.
8	*Browser*	*Plug-in*	Encaminha pedido do leitor.
9	*Plug-in*	*Browser*	Solicita o arquivo de chaves e permissões criptografado.

	De	Para	Operação
10	*Browser*	*Site* Internet	Encaminha pedido do *plug-in*.
11	*Site* Internet	*Browser*	Envia o arquivo de chaves e permissões criptografado.
12	*Browser*	*Plug-in*	Encaminha o arquivo recebido.
13	*Plug-in*	*Browser*	Solicita a chave privada do leitor.
14	*Browser*	Leitor	Encaminha o pedido da chave privada.
15	Leitor	*Browser*	Informa sua chave privada.
16	*Browser*	*Plug-in*	Encaminha a chave privada do leitor.
17	*Plug-in*	*Site* Internet	Após autenticar o leitor, prossegue com a recuperação do hiperdocumento.
18	*Site* Internet	*Plug-in*	Envia o hiperdocumento criptografado.
19	*Plug-in*	*Browser*	Após decifrar o hiperdocumento e inserir a marca, encaminha a cópia individualizada.
20	*Browser*	Leitor	Apresenta a cópia individualizada.
21	Autor	*Keygen*	Solicita chaves hipotéticas para testes
22	*Keygen*	Autor	Devolve chaves hipotéticas criadas dinamicamente

7.2 Utilitário *Keygen*

O *KEYGEN* é um utilitário gerador de chaves utilizado para criar chaves hipotéticas para serem utilizados nos testes do protótipo. Desenvolvido em linguagem C, é um programa por linha de comando que utiliza as mesmas rotinas do *ENGINE* e do *plug-in* (Módulo Leitor).

A figura 7.2 demonstra a tela de ajuda de sintaxe do *KEYGEN*. Como pode ser visto, o programa espera quatro parâmetros: dois *strings*, que são utilizados como base para a geração das chaves; e dois nomes de arquivos destino, onde serão gravadas as chaves geradas. Por limitações das rotinas utilizadas na implementação, os *strings* informados devem ter no máximo 80 *bytes*. Consequentemente as chaves geradas também terão no máximo 80 *bytes*. Para utilizar números maiores, as rotinas de tratamento de números longos precisam ser revisadas.

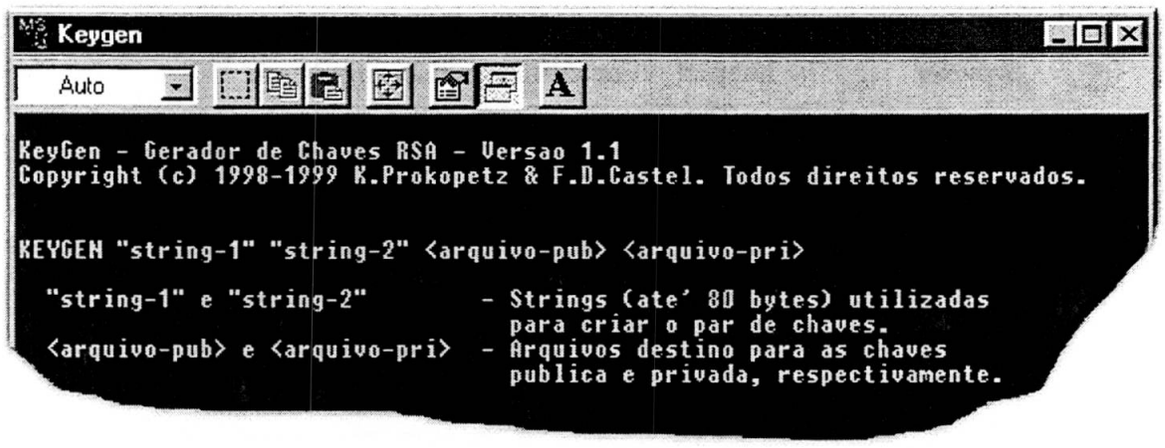


FIGURA 7.2 – Tela de sintaxe do KEYGEN

7.3 Módulo Autor

7.3.2 *Engine*

O *ENGINE* é um sub-módulo do módulo autor, onde encontra-se o código de criptografia e *watermark*, comuns ao Módulo Leitor. Da mesma forma que o *KEYGEN*, é um programa de linha comando, desenvolvido também em linguagem C.

A figura 7.3 apresenta a tela de ajuda de sintaxe do *ENGINE*. O aplicativo recebe dois parâmetros. O primeiro é opcional e diz respeito ao modo de execução. Se o parâmetro for informado, será ativado o modo silencioso, fazendo com que nenhuma mensagem de resultado de processamento seja ecoado na tela. O segundo parâmetro é o nome do arquivo HTML a ser protegido pelo *HIPERMARK*. Todos os links contidos nele serão analisados. Se o link apontar para uma mídia local, esta também será protegida.

Ao final da execução, existirá uma cópia de cada elemento protegido, com a extensão *.HYP. Estes arquivos estarão criptografados pelo sistema e só poderão ser abertos através do Módulo Leitor.

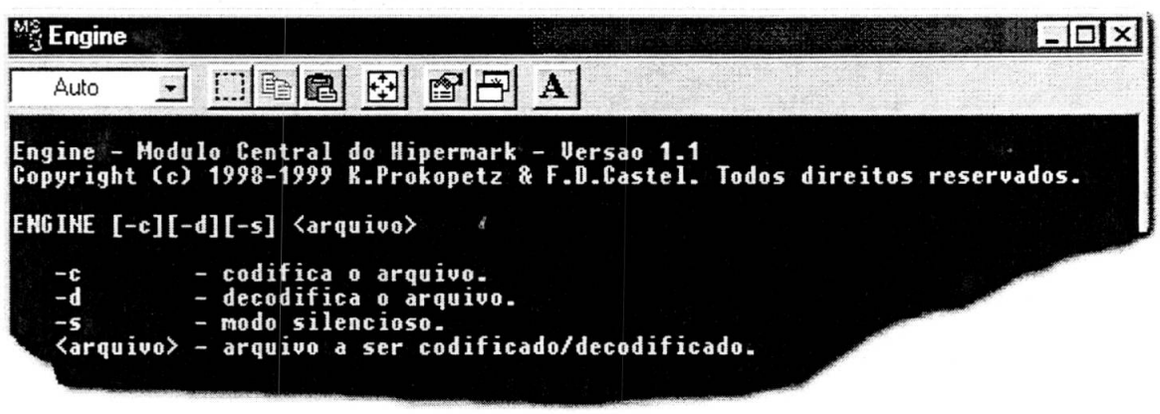


FIGURA 7.3 – Tela de sintaxe do ENGINE

7.3.3 Interface

O INTERFACE é o sistema de diálogo do usuário autor com o *ENGINE*. Numa implementação completa da arquitetura, seria desenvolvida uma versão específica para cada ambiente de execução, com recursos, necessidades e restrições próprias. No protótipo o INTERFACE foi desenvolvido em DELPHI e está disponível apenas para ambiente WINDOWS.

A figura 7.4 mostra um detalhe da tela principal do INTERFACE. A tabela *CONTEÚDO* apresenta a descrição do conteúdo do hiperdocumento, sendo basicamente uma análise a partir de seus links. A tabela *LEITORES* mostra a listagem dos leitores autorizados e suas permissões para o hiperdocumento. A tabela *AUTOR* possui os dados de identificação do autor.

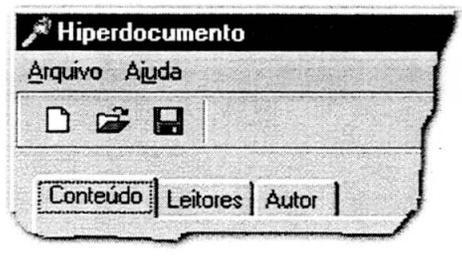


FIGURA 7.4 – Detalhe da tela principal da Interface

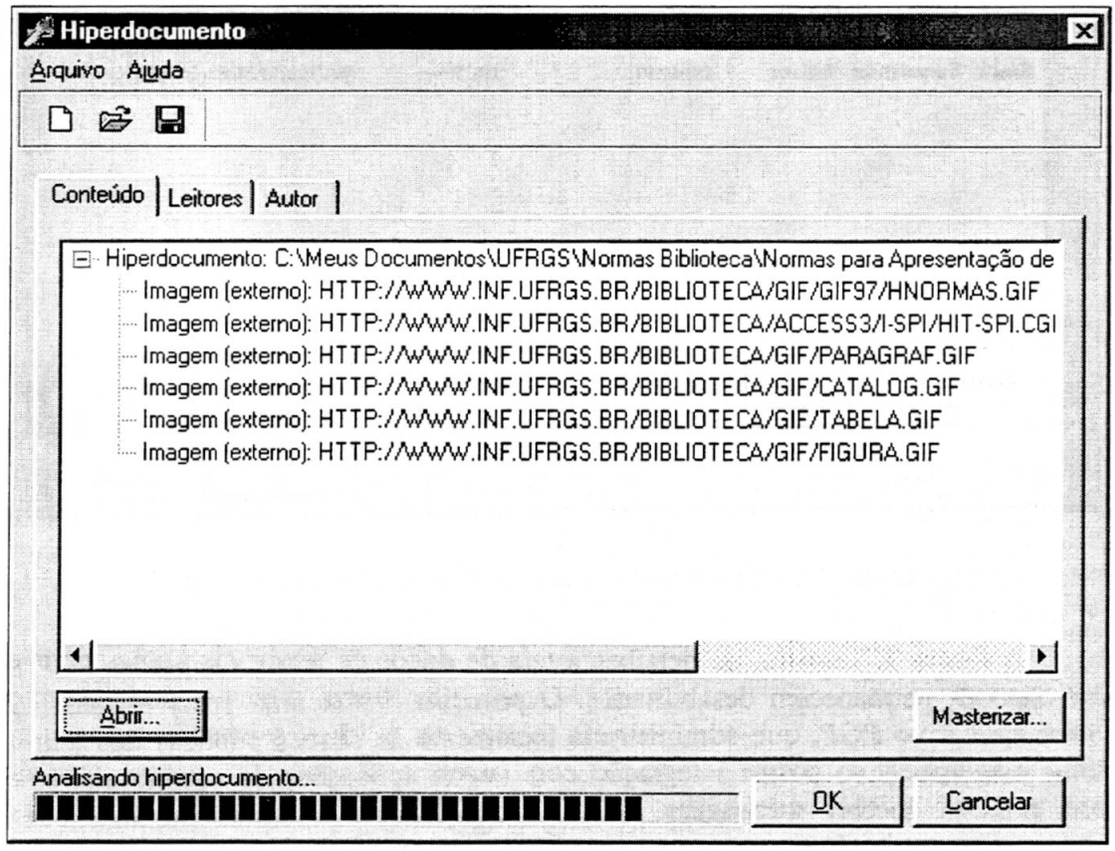


FIGURA 7.5 – Tela de descrição do conteúdo do hiperdocumento

A figura 7.5 mostra a tela de descrição do conteúdo do hiperdocumento. Este conteúdo é apresentado após o usuário selecionar algum documento no formato HTML, através do botão *ABRIR*. Após uma análise detalhada do hiperdocumento, a aplicação apresenta os documentos referenciados pelo mesmo, identificando sua mídia e distinguindo-os entre locais e externos.

Numa versão completa, este módulo deveria expandir recursivamente os hiperdocumentos locais. Deveria também habilitar a seleção de documentos na lista. Permitindo a proteção de documentos específicos, enquanto os demais itens que compõem o hiperdocumento poderiam permanecer com acesso livre.

A figura 7.6 mostra a tela de autorização de leitores para o hiperdocumento. Além de adicionar, modificar e eliminar itens desta lista, o INTERFACE permite que seja importado todo um arquivo de chaves de leitores previamente cadastrados, ou que então seja criado a partir da lista em uso.

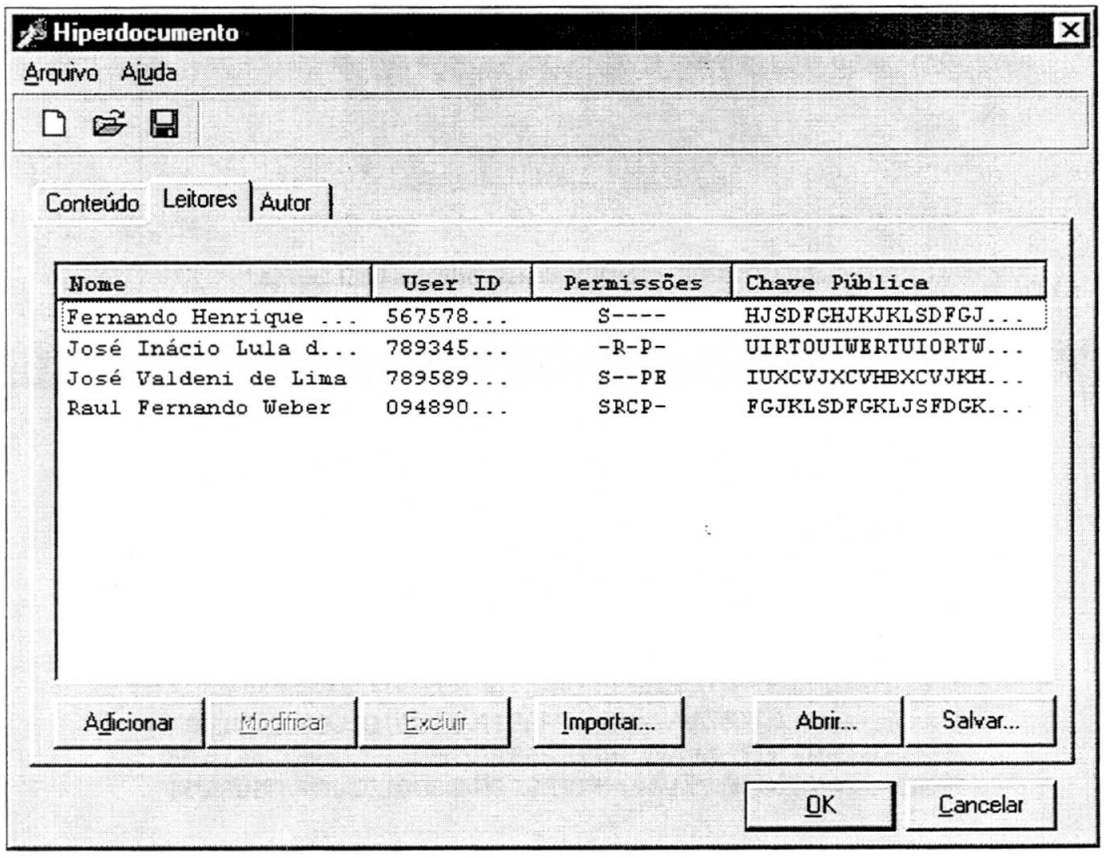


FIGURA 7.6 – Tela de autorização de Leitores para o hiperdocumento

A figura 7.7 mostra em detalhes a tela de dados de leitor. Os botões PGP e o KEY SERVER permanecem desabilitados. O primeiro ilustra uma possível integração com o aplicativo *PGP*, que administraria localmente as chaves públicas dos leitores. Como este aplicativo possui integração com outras aplicações de correio eletrônico, seria possível receber mensagens assinadas dos leitores e armazenar suas chaves públicas. O outro botão supostamente permitiria a integração com os diversos servidores

de chaves públicas, de onde o autor poderia obter o identificador e a chave pública dos leitores que seriam autorizados.

No protótipo a integração com o PGP e o KEYSERVER não foi implementada. Assim, para os testes, o autor receberia o *e-mail* com a chave pública do leitor e copiaria para o campo de chave pública conforme visto na figura 7.7. Para um uso diário da arquitetura esta implementação seria desgastante, por isto, numa versão completa esta tarefa deveria ser mais automatizada, no mínimo implementando as características destes botões desabilitados.

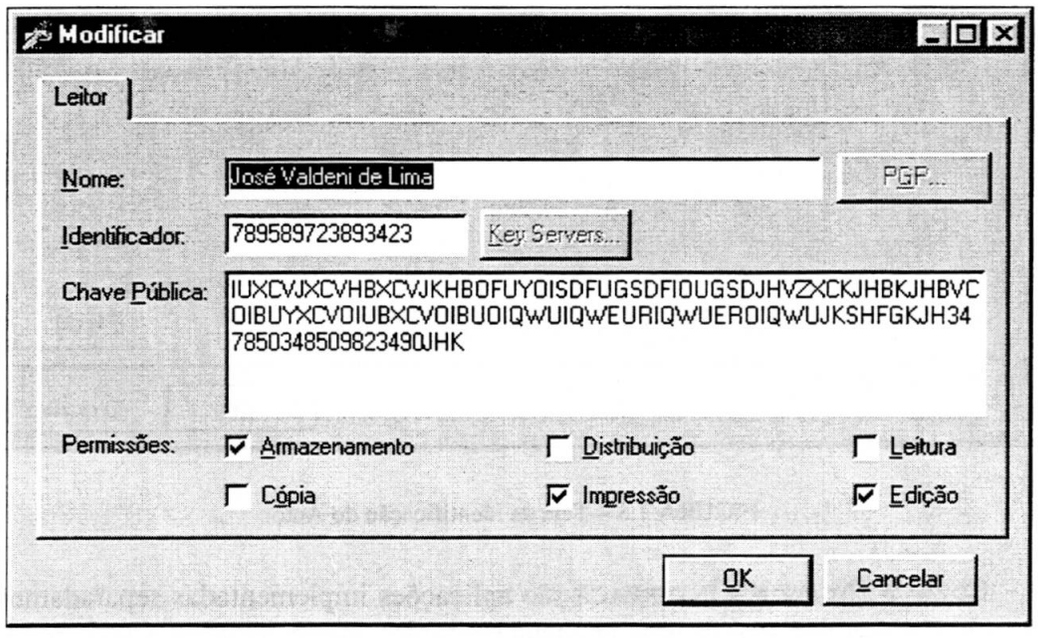


FIGURA 7.7 – Tela de Permissões de Leitor para o hiperdocumento

Para criação das cópias mestres, mencionadas no item 6.3.1, é preciso identificar o autor. A figura 7.8 apresenta como o protótipo requer a identificação do autor.

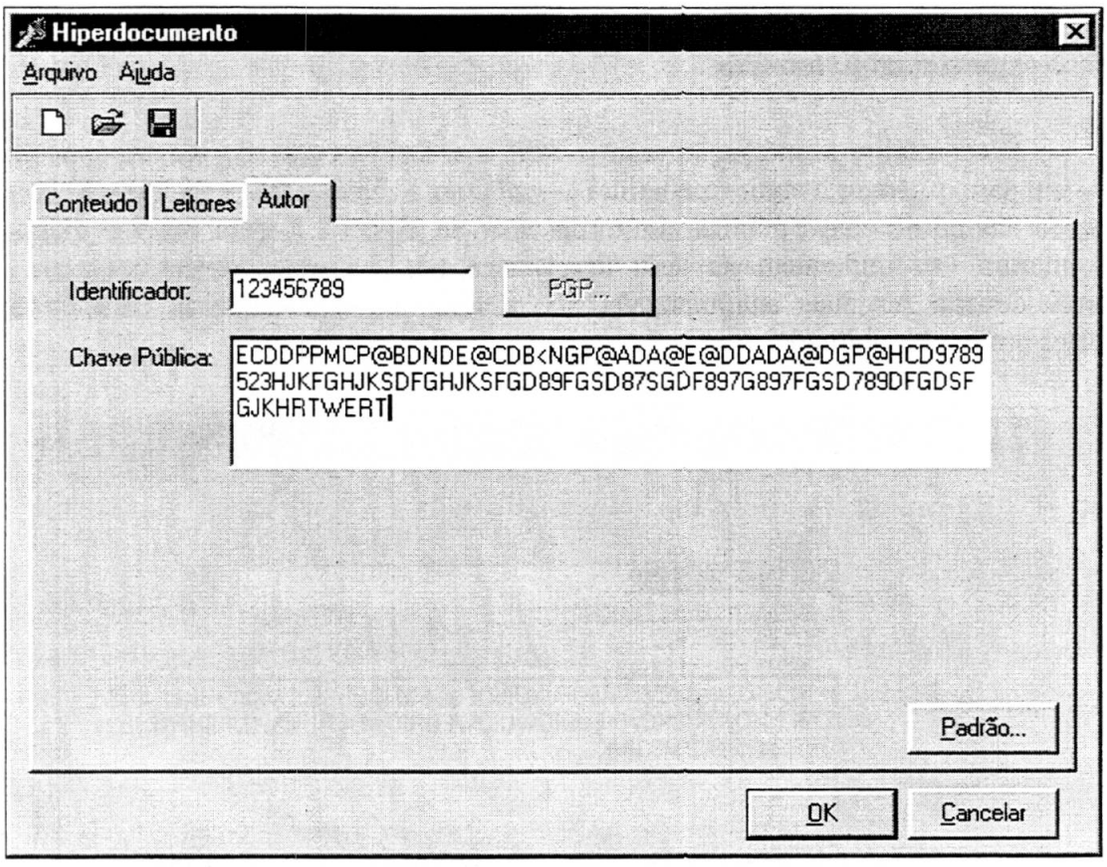


FIGURA 7.8 – Tela de identificação do Autor

Como o *ENGINE* e a INTERFACE são aplicações implementadas separadamente, a integração entre ambas requer uma pequena e simples parametrização. Conforme figura 7.9, o INTERFACE deve conhecer a localização do ENGINE, permitindo ainda a escolha da forma de execução da aplicação: escondida ou não.

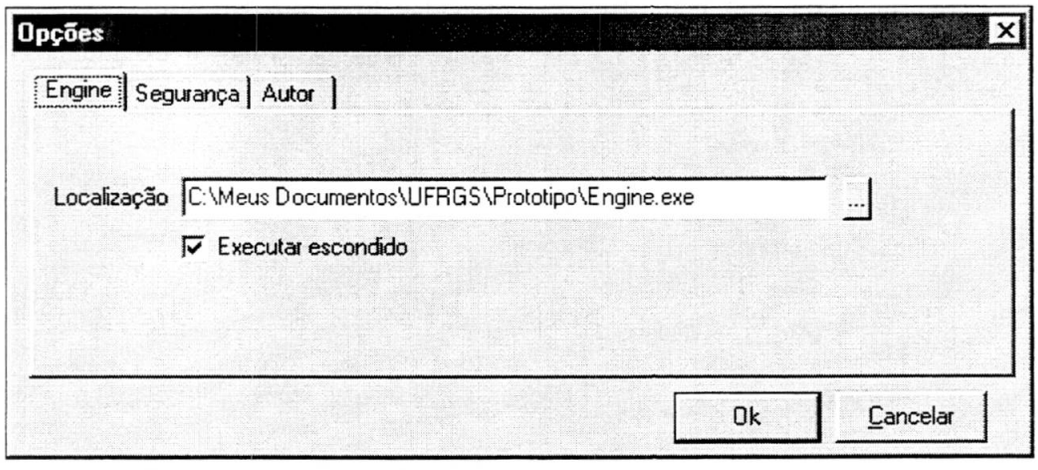


FIGURA 7.9 – Tela de customização da interface

No caso do protótipo, apenas para realizar alguns testes de segurança na comunicação com o Módulo Leitor, foi permitido que fosse customizado também a senha do sistema. Esta característica, implementada na tabela SEGURANÇA, conforme

ilustra a figura 7.10, não estaria disponível na versão completa, pelo menos não desta forma.

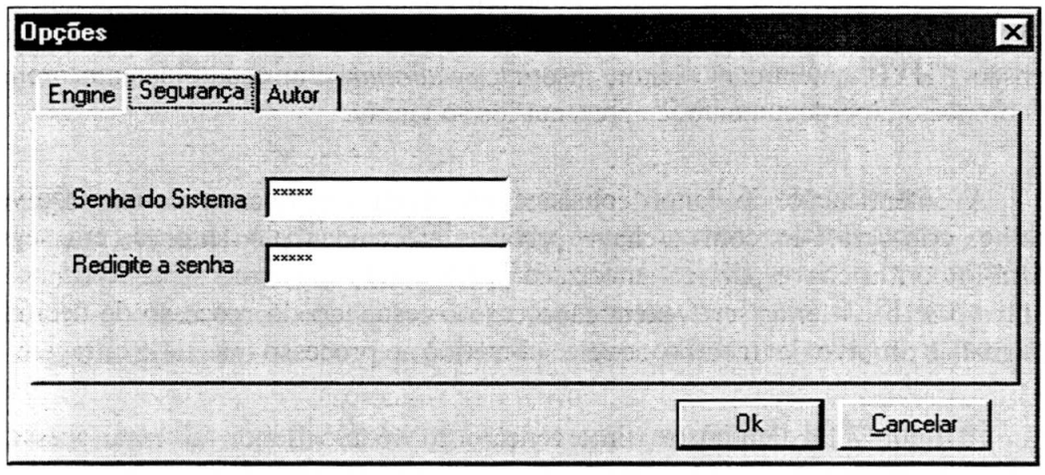


FIGURA 7.10 – Tela de customização de segurança

A figura 7.11 mostra uma pequena facilidade de implementação. Ao invés de sempre informar a identificação do autor para cada hiperdocumento tratado pelo HIPERMARK, foi dada a opção de manter esta identificação armazenada pelo INTERFACE. Desta forma, a identificação do autor na figura 7.8 sempre virá preenchida e necessitará mudança apenas quando mudar autor que estiver utilizando o aplicativo. E, ainda na tela da figura 7.8, pressionando o botão PADRÃO, a identificação informada será assumida como padrão para o INTERFACE e passará então a ser apresentada na tela da figura 7.11.

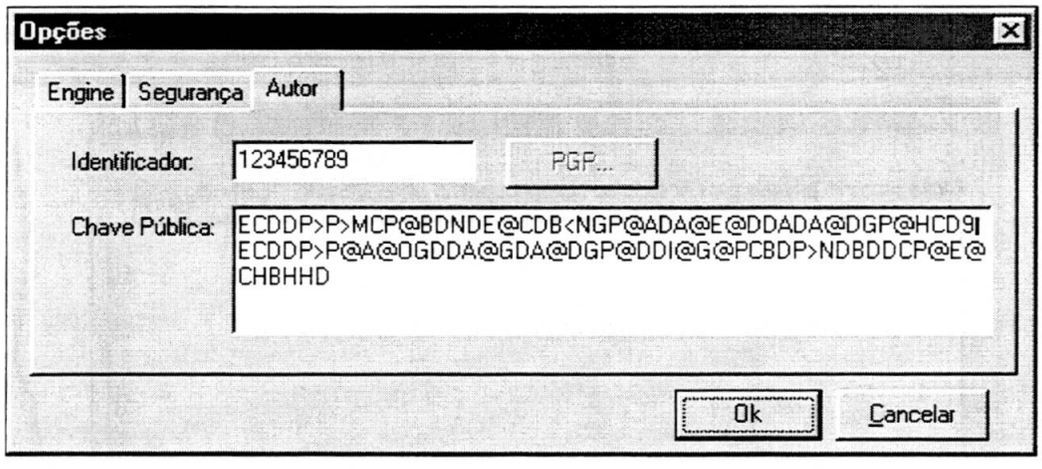


FIGURA 7.11 – Tela de customização de Autor

Ao final da operação (análise do hiperdocumento, especificação de leitores e autor), ao pressionar o botão MASTERIZAR na tabela CONTEÚDO, vide figura 7.5, o INTERFACE monta as informações e passa para o ENGINE realizar as tarefas especificadas no item 6.3.1 e 6.3.2.

7.4 Módulo Leitor

O Modulo Leitor foi desenvolvido em linguagem C, no formato de um *plug-in* para NETSCAPE. O objetivo do *plug-in* é interceptar os pedidos para arquivos de extensão *.HYP, autenticar o leitor, inserir a *watermark* em cada elemento protegido que compõe o hiperdocumento, e apresentá-los ao leitor.

A autenticação do leitor consiste em gerar dinamicamente um arquivo de trabalho, criptografá-lo com a chave privada informada pelo leitor e, em seguida, decifrá-lo com a chave pública autorizada pelo autor, conforme especificado no item 6.3.11 a 6.3.15. O leitor será autenticado caso o resultado do processo de decifragem seja igual ao arquivo de trabalho, aquele submetido ao processo inicial de cifragem.

A figura 7.11 demonstra a intervenção do Módulo Leitor ao tentar acessar um arquivo de extensão *.HYP. Isto é feito de forma automática pelo *browser*. O mecanismo de MIME associa uma extensão de arquivo a um dos *plug-in* instalados. Assim, toda vez que é solicitado um tipo de arquivo diferente de HTML, o browser verifica o *plug-in* ou *helper* associado e executá-lo. Esta associação é determinada em tempo de desenvolvimento do *plug-in*. Para instalar o *plug-in* basta colocá-lo no diretório PLUGINS, subordinado ao diretório de instalação do *browser*.

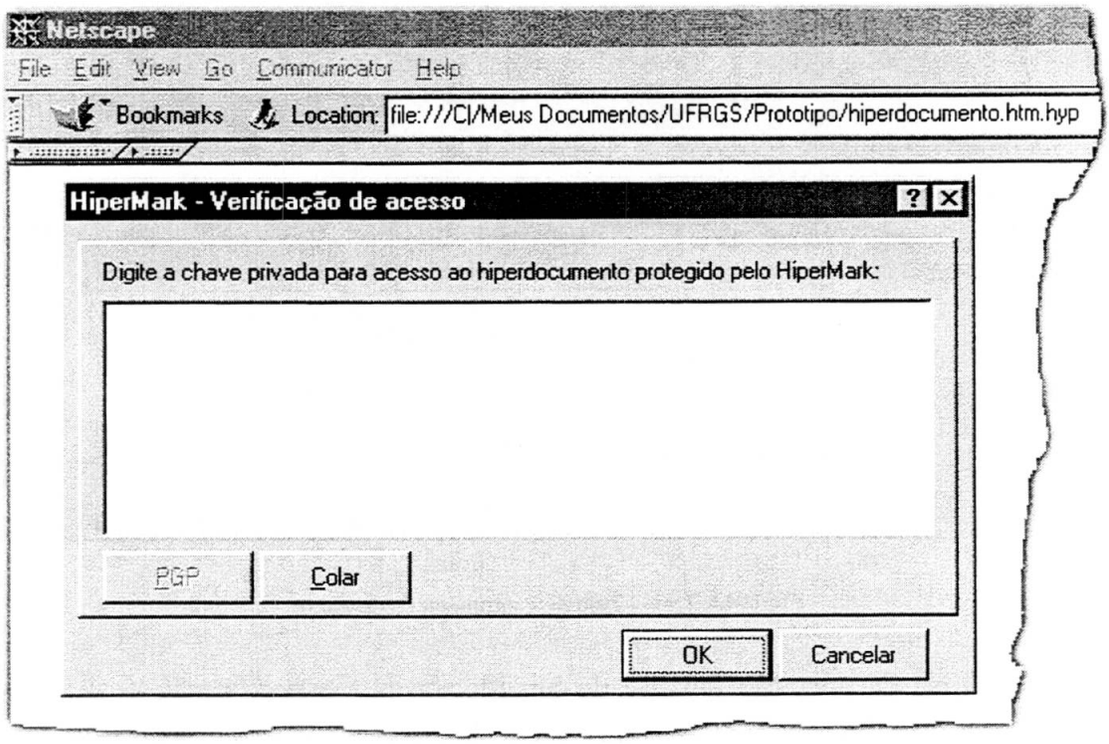


FIGURA 7.12 – Tela de intervenção do *plug-in* (Módulo Leitor)

7.4.1 Integração do Módulo Leitor com os *Browsers*

O Módulo Leitor poderia ser implementado de duas formas diferentes para interagir com o *browser*. Poderia ser um *helper application* ou um *plug-in*. A opção por um ou pelo outro depende do tipo de interação e segurança desejada.

As *helper applications* são pequenos programas independentes do *browser*. Podem utilizar *sockets*, DDE, OLE, ou a combinação destes para interagir com o *browser*. Alguns *helpers* desconhecem isto e usam *sockets* diretamente, enquanto outros usam apenas DDE ou OLE.

Os *plug-ins*, por sua vez, raramente requerem o uso direto de *sockets*, DDE ou OLE, pois são integrados ao *browser* através de uma API. Os *plug-ins* são como bibliotecas de funções utilizadas pelo *browser* para o tratamento de determinado tipo de arquivo.

Ambos são iniciados baseados em uma extensão de arquivos e um tipo de MIME. Assim, quando o *browser* recebe um pedido para abrir determinado tipo de arquivo, verifica em sua lista qual aplicação deve ser chamada.

Diante das diferenças, optou-se por implementar o Módulo Leitor como um *plug-in* pela flexibilidade de integração com o *browser* e pelo tratamento interno das informações, sem que houvesse o trânsito de informações entre duas aplicações (*browser* e Módulo Leitor, como um *helper application*).

8 Conclusão

A pirataria é uma preocupação atual para informática. O formato digital das informações trouxe facilidades inestimáveis para publicação eletrônica, mas também potencializou os recursos da pirataria.

Pesquisadores em criptografia já definiram, apropriadamente, que não há e é pouco provável que se venha a obter, um método absolutamente seguro de proteção de dados para transmissão em um canal inseguro, como a Internet. Todavia, se a tarefa de piratear for onerosa o suficiente a ponto de tornar mais fácil e barato simplesmente adquirir uma cópia legal, devemos considerar que os dados estão seguros.

A Internet é uma realidade e provavelmente o meio de desenvolvimento nos próximos anos e décadas para uma grande variedade de áreas. A ênfase atual está voltada para a publicação eletrônica, que promete uma grande redução de custos, maior abrangência de público, facilidades de pesquisa e organização, e sob alguns pontos de vista, a gradual redução do uso de papel.

Todavia, quando se pensa um uma rede mundial, a publicação eletrônica ainda gera polêmica na questão do direito autoral:

Quais os direitos do autor?

Quais fronteiras limitam este direito?

Como provar a autoria?

Como controlar o uso?

Quais critérios utilizar para cobrança?

8.1 Fundamentação

Como foi visto, o direito de autor independe da mídia de publicação da obra ser digital, ou não. Assim, os direitos do autor de um texto digital ou em papel são os mesmos. Também foi dito que a Internet popularizou os hiperdocumentos, que, aliados ao avanço tecnológico de impressão, armazenamento e comunicação, tornaram viável a publicação eletrônica.

A publicação eletrônica de documentos é mais rápida, menos cara, e requer menos esforços do que fazer cópias papel e então distribuí-las fisicamente. Além disto, devem ser consideradas as inúmeras vantagens de usar um computador, tais como o armazenamento, pesquisa e edição.

A publicação eletrônica possui muitas vantagens, porém muitos autores ainda têm o receio de publicar suas obras como hiperdocumentos para Internet. Por ser uma publicação de abrangência internacional e pela própria natureza eletrônica de sua forma, acreditam erroneamente que suas obras estariam desprotegidas além das fronteiras de

seu país, ou até mesmo nele, podendo ser copiadas indiscriminadamente, sem qualquer controle.

No entanto, este entendimento trata-se de um grande equivoco, pois, por força de tratados internacionais, a proteção ao direito de autor estende-se automaticamente aos países membros destes tratados, que atualmente representam quase a totalidade dos países existentes. Por este motivo, não deve haver receio na publicação de obras na Internet, pois será protegida em todos os países, da mesma forma que o seria se fosse publicado em seu país de origem, na sua mídia não digital.

Porém, ainda resta um único senão. No caso de litígio judicial, onde há disputa da autoria da obra, seria necessário meios para provar a autoria. Para os casos convencionais, ou seja, para obras não digitais, bastaria verificar qual das partes que primeiro tornou pública sua obra, seja qual for a forma física de sua publicação. Para as mídias digitais, não há maneira exata de especificar e garantir o espaço temporal da sua publicação. Assim, na publicação eletrônica torna-se necessário acrescentar facilitadores para provar a autoria da obra.

Inicialmente, pensou-se na criptografia por tratar-se de uma técnica de comunicação sigilosa largamente utilizada. Apesar desta comunicação servir para transmitir um obra que seja protegida por direito de autor, a criptografia por si própria não protege o autor, uma vez que qualquer das partes envolvidas na comunicação pode repassar a obra para terceiros após decifrá-la. Isto porque a criptografia não deixa qualquer vestígio no documento após a decifragem, serve única e exclusivamente para uma comunicação privada em um canal inseguro.

Depois foi analisada a esteganografia, ciência que estuda meios para esconder e transmitir informações sigilosas, sem o conhecimento de sua existência. Já fazem alguns anos que estudiosos desta ciência tem trabalhado com a técnica de *watermark*, aprimorada especificamente para proteção de direitos autorais.

Uma boa *watermark* é aquela que individualiza a cópia da obra. Isto é, além de permitir identificar o autor da obra, especifica também qual o leitor autorizado para cópia em questão. Porém, a inserção de uma marca tem um custo pesado de processamento se feito em grande escala. Por esta razão a maioria das soluções atuais não utiliza a individualização das cópias, isto é, a *watermark* contém apenas informações sobre o autor, ignorando o leitor. Assim, basta inserir uma única vez a marca na obra, ao invés de inseri-la para cada cópia.

Com a individualização das cópias seria possível rastrear piratas autorais. Neste sentido já foi desenvolvido o projeto DocMark (vide item 5.9) na Universidade de Columbia, baseado na proposta de *Choudhury* [CHO94]. No restante, todas as soluções contemplam apenas algumas mídias em separado. E não permitem mais do que simplesmente identificar o autor da mídia protegida e, nas implementações mais avançadas, localizar onde esta mídia estava sendo utilizada.

Todavia, o DocMark não é muito eficaz na combinação de criptografia e esteganografia. O projeto utiliza os recursos de criptografia apenas para carregar suas

informações a salvo até a máquina do leitor. O uso de técnicas de assinatura digital e a possibilidade de informar ao autor sobre o uso de seu hiperdocumento foram simplesmente ignoradas no projeto DocMark.

8.2 Arquitetura proposta versus Requisitos de Choudhury

A arquitetura proposta atende a grande maioria dos requisitos que *Choudhury* [CHO94] apontou para uma publicação eletrônica viável:

MATERIALIZAÇÃO

Os leitores podem visualizar repetidas vezes o hiperdocumento, bastando que informem sua chave privada. Como a cópia armazenada na máquina do leitor conterá a chave pública deste, o *plug-in* sempre poderá conferir e autenticar o leitor antes de reapresentar o hiperdocumento;

AUTENTICIDADE

Como o hiperdocumento original é mantido criptografado com uma chave secreta do sistema, suficientemente robusta, o leitor terá sempre a certeza de que esta realmente lendo o documento original e não uma falsificação;

NÍVEIS DE SEGURANÇA

A arquitetura permite que sejam mesclados documentos protegidos com documentos abertos. Assim, o autor pode utilizar o sistema apenas para proteger determinados elementos de seu hiperdocumento, aqueles de relevante valor. Estas medidas não acarretam nenhuma perda de qualidade para os leitores idôneos. As vantagens são a economia de espaço tempo de processamento.

ENGENHARIA REVERSA

Pela adaptação do uso do sistema de assinatura digital, a arquitetura garante um nível considerável de segurança contra engenharia reversa. Mesma que seja descoberto uma técnica para revelar a marca contida em uma cópia do hiperdocumento, esta técnica não terá efeito sobre outra cópia do mesmo hiperdocumento. Isto porque o conteúdo, além de criptografado, será diferente para cada leitor, pois reflete o uso de chave de chave pública e privada. Desta forma, o custo de desenvolver uma técnica de engenharia reversa para revelar o conteúdo das marcas inseridas é maior do que elaborar a mesma técnica para assinatura digital;

INDEPENDÊNCIA DE AUTOR

A arquitetura permite que o hiperdocumento contenha *links* para documentos de autores diferentes. Na verdade, a questão de hipertexto é transparente: se o link apontar para um ponto desprotegido, será desviado normalmente; se apontar para um *link* protegido, iniciará uma nova instância da arquitetura, mantendo todas as características de segurança;

DOCUMENTOS ESTÁTICOS E DINÂMICOS

Tradicionalmente, os documentos tem sido considerados estáticos, gerados uma vez e então armazenados por longos períodos de tempo. Todavia, isto não tem fundamento porque documentos gerados dinamicamente, tais como resultados de consultas a banco de dados ou os saídas de uma computação, podem não ser distribuídos em uma forma similar. Isto já esta acontecendo através da integração de vários banco de dados e servidores WWW. O usuário não pode prontamente distinguir se um documento existe como lhe foi entregue ou se foi criado pela sua demanda. Este requisito implica que qualquer processamento especial necessário para proteção de direitos autorais deve ter um custo computacional razoável, de forma que seja possível ser feito dinamicamente. Como a arquitetura desvia o processamento pesado de inserção de *watermark* para o computador do leitor, resta apenas um pequeno processamento de criptografar com a chave do sistema o documento dinâmico gerado.

ARMAZENAMENTO

O hiperdocumento criptografado pode ser armazenado em qualquer *site* na Internet, permitindo a redução de tempo de acesso e custos de transmissão. Com isto oferece uma oportunidade para provedores de serviços de distribuição de informação para servir como um centro de armazenamento.

PRIVACIDADE

Leitores podem ter bons motivos para não revelar a terceiros quais tipos de documentos estão sendo pesquisados. Em uma rede de computadores com *links* fisicamente instáveis, a criptografia durante o trânsito do hiperdocumento até a máquina do leitor garante este requisito, e para abrir uma cópia individualizada do hiperdocumento, será necessário informar a chave privada do leitor.

COBRANÇA FLEXÍVEL

Adicionando mecanismos de mensagens de controle e acesso ao Módulo Leitor, a arquitetura permite diferentes formas de cobrança pelo acesso aos hiperdocumentos:

COBRANÇA POR ASSINATURA

Um *site* de publicação de hiperdocumentos pode, por exemplo, cobrar seus clientes por um plano de preços, indiferente da quantidade e tipo de hiperdocumentos recuperados. Este modelo de cobrança reflete em um baixo custo incremental de acesso documentos, encoraja o uso, evita encargos contestados e reembolsos, e incorre em uma baixa sobrecarga de cobrança;

COBRANÇA POR DOCUMENTO

Somente é cobrada a primeira vez que o documento é recuperado, desencorajando o armazenamento particular de documentos. Neste caso, o servidor de documento deverá armazenar uma identificação de quem recuperou cada documento. Assim, um segundo pedido de um mesmo

usuário pode ser reconhecido e não cobrado. Alternativamente, o servidor pode distribuir uma comprovante para o usuário, provando que ele realmente já pagou pelo documento;

COBRANÇA POR ACESSO

Uma cobrança é gerada cada vez que um documento é visualizado ou impresso. Isto tem a desvantagem de encorajar, se for possível, o armazenamento local do documento, mas por outro lado reduz o montante de informações armazenadas no servidor.

8.3 Contribuições

A principal contribuição deste trabalho é propor mecanismos que garantam a proteção de direitos autorais de hiperdocumentos, através de uma arquitetura para publicação eletrônica na Internet. A arquitetura proposta está baseada na integração de técnicas de *watermark* e assinatura digital, ambas fundamentadas na questão de segurança. Vale ressaltar que o presente trabalho não propõe, nem testa a robustez de algoritmos de *watermark*.

A combinação de esteganografia e criptografia, conforme proposto pela arquitetura, possui grandes vantagens. Em especial, a possibilidade de o conteúdo da *watermark* conter a chave pública do leitor que teve acesso autorizado ao hiperdocumento. Com isto, é possível garantir um bom nível de segurança na individualização da cópia e autenticação do leitor, que deverá informar sua chave privada para efetivar os acessos ao hiperdocumento.

Outro ponto de destaque da arquitetura proposta é que o autor disponibilizará uma única cópia do seu hiperdocumento, mas todas as cópias acessadas serão personalizadas. Isto permitirá que a *watermark*, além de identificar o autor da obra, também identifique o leitor que teve o acesso ao hiperdocumento. Com isto, além de criar meios para provar a autoria dos hiperdocumentos, a arquitetura contribui também para a identificação das fontes de pirataria.

A possibilidade de controlar a distribuição de cópias e efetuar uma cobrança flexível constitui-se em outra importantíssima contribuição, que vai ao encontro das necessidade do mercado extremamente exigente das gráficas comerciais.

A arquitetura contribui também para solucionar situações complexas na publicação eletrônica, como por exemplo os leitores anônimos, o hiperdocumento público, o autor anônimo e o hiperdocumento nômade.

O controle descentralizado da arquitetura permite sua implementação sem sobrecarga para Internet. Como a individualização das cópias é proposta para ser feita localmente, na máquina do leitor, o servidor do autor é isento de qualquer processamento e armazenamento extra. Da mesma forma, os riscos do tráfego de senhas pela rede também são evitados.

Através do controle de uso proposto é possível implementar diferentes níveis de segurança sobre os hiperdocumentos. A medida que é possível parametrizar previamente a necessidade das mensagens de controle de uso, evita-se o tráfego de informações desnecessárias pela rede, bem como seu armazenamento por processos de registro de *logs*.

8.4 Aplicações

8.4.1 Publicação e Distribuição de Documentos

O ambiente ideal para uso da arquitetura proposta são aplicações de publicação e distribuição de documentos, tais como o ADOBE ACROBAT. Atualmente, a segurança deste aplicativo baseia-se na obscuridade de seu formato PDF, casado com um sistema corriqueiro de senhas.

A adoção da arquitetura proposta, ou uma adaptação dela, traria grandes vantagens ao aplicativo, principalmente pela individualização das cópias em uma distribuição de larga escala. Atualmente, se algum usuário do ACROBAT quiser obter uma segurança semelhante, deveria gerar manualmente uma cópia para cada leitor autorizado, variando sua senha de acesso. Mesmo assim, seria praticamente impossível identificar se alguém pirateou a cópia deste leitor.

O caso do ADOBE ACROBAT é particularmente interessante para arquitetura pois além de ser multiplataforma, já possui recursos de apresentação de documentos textos como um *bitmap* para o leitor. Isto permite a implementação das técnicas de *word-and-line-shifting* como sugerido por *Brassil* [BRA95], e que atualmente é a melhor solução para *watermark* em textos.

8.4.2 Ensino a Distância

Aplicações de ensino a distância, quando tiverem a necessidade de identificar o aluno/leitor, podem considerar o uso da arquitetura. Particularmente, este trabalho não incentiva a identificação do aluno como forma de cobrar sua presença nos cursos virtuais, mas como forma de identificar dificuldades e carências específicas de cada aluno. De forma que o orientador do curso possa interagir para que cada aluno tenha o melhor aproveitamento.

No caso do ensino a distância, que já é objeto de pesquisas em outros trabalhos publicados, vale considerar o uso da presente arquitetura combinada com aquela proposta em [PRO97B]. Naquela proposta havia uma base de conhecimento que mantinha informações didáticas dos alunos, além do próprio hiperdocumento. Após a identificação do aluno, um interface solicitava para a base de conhecimento o hiperdocumento a ser apresentado. Tendo em vista que, uma grande maioria de

aplicações hipermídias voltadas ao processo ensino-aprendizagem apresenta o conteúdo de forma repetitiva e sem um acompanhamento de uso, e muito menos levando em conta este acompanhamento de uso, se tentou enfatizar os aspectos de não repetição e de navegação personalizada para cada aluno. Isto evita que a navegação seja repetitiva e permite que o aluno tenha a sua disposição um número grande de possibilidades, o que facilita a avaliação através da compreensão do conteúdo apresentado, independentemente da capacidade de memorização do aluno. A presente arquitetura poderia trazer segurança na identificação e controle de acessos destes alunos aos hiperdocumentos selecionados pela base de conhecimento.

8.5 Trabalhos Futuros

8.5.1 Conclusão do Protótipo

O objetivo do protótipo foi exemplificar o uso da arquitetura proposta. Ainda que atenda a muitos princípios da arquitetura, o protótipo não é totalmente funcional. Sua conclusão poderia contribuir no amadurecimento da proposta, na medida em que pudesse ser utilizado em situações reais, mesmo que fosse restrito a um ambiente acadêmico qualquer.

8.5.2 Sessão de Uso

Durante o processo de autenticação, o Módulo Leitor solicita a chave privada do leitor. Esta característica garante os fundamentos de segurança propostos. De forma que, a cada acesso a um hiperdocumento protegido, a autenticação do leitor seja revalidada. Esta tarefa pode ser muito entendiante. Por isto, sugere-se um estudo de sessão de uso, semelhante ao utilizado para *sites* seguros. Onde o leitor informaria sua chave privada apenas na primeira interação com algum hiperdocumento protegido. Para os demais acessos, o Módulo Leitor faria automaticamente a autenticação com a chave privada do leitor, sem efetuar uma nova solicitação ao leitor. Após encerrada a sessão, seja qual for o critério a ser utilizado, o leitor deveria informar novamente sua chave para estabelecer uma nova sessão.

8.5.3 Envio de Mensagens

A arquitetura não detalhou como o Módulo Leitor identifica o local para envio das mensagens de controle de acesso. Inicialmente seria uma informação contida junto da *watermark*. Todavia, como a proposta evoluiu para contemplar as situações de cartórios virtuais, distribuidoras e autor anônimo, a questão agregou maior complexidade. Uma pesquisa complementar pode detalhar este aspecto, sendo recomendado que um estudo sobre o arquivo de chaves e permissões.

8.5.4 Segurança

Afim de aprimorar o presente trabalho, pesquisadores especializados em segurança podem adicionar características extras para arquitetura proposta, de forma que seja possível rastrear os piratas. Este rastreamento poderia iniciar pela identificação de nodos de rede de onde surgem mensagens de tentativa de violação.

Ainda na questão segurança, poderia ser estudado a questão da incubação do hiperdocumento em uma rede fechada, procurando solucionar o envio das mensagem, mesmo que *a posteriore*. Caso em que poderia se pensar em como embutir fragmentos vitais destas mensagem na própria cópia do documento. Para que, se de alguma forma esta cópia escapar desta rede fechada, possa então se comunicar com o autor, enviando alguma informação estratégica de seu período de aprisionamento. Ou talvez aplicando-se alguma técnica adotada pelos vírus, que consiga manter as características de comunicação imperceptível até que a mesma possa ser restabelecida normalmente com o autor.

Também deve ser estuda a questão da propagação hereditária dos direitos de acessos utilizados durante a construção dos *links* no hiperdocumento. Este estudo deve procurar resolver conflitos gerados quando o autor tem o direito de acesso a um documento, como por exemplo uma fotografia, e deseja utilizá-lo como um link interno de seu hiperdocumento, mas o leitor não tem a permissão necessária para acessar o documento.

8.5.5 Cartórios Virtuais

Outro enfoque muito interessante que poderia ser melhor explorado são os Centros de Autenticação, que atuariam como cartórios virtuais, comprovando a autenticidade de determinadas cópia, ou até mesmo comprovando a conduta de seus autores, garantindo que as mensagens não contenham vírus ou dados clandestinos sobre as informações armazenadas em sua máquina.

8.5.6 Agentes para Internet

Resta ainda um meio do autor identificar elementos de seu hiperdocumento que foram abertos com o *plug-in*, capturados de alguma forma e armazenados em outro formato. É o caso, por exemplo, de um usuário que teve acesso autorizado ao hiperdocumento, informando corretamente seu identificador e chave de acesso, mas que capturou a tela quando foi visualizado uma imagem qualquer contida no hiperdocumento. Ao editar esta imagem em um aplicativo gráfico, salvando em um formato qualquer, a imagem poderá ser utilizada por qualquer aplicativo, sem necessitar do *plug-in*. Porém, a *watermark* estará presente e imperceptível. Resta, portanto, desenvolver agentes para Internet, que vasculhem o material publicado a procura de *watermark* produzidas no uso da arquitetura proposta e que identificarão o autor e leitor.

8.5.7 Ferramenta de Mineração

As mensagens de acesso podem ser personalizadas em dois momentos. O primeiro deles é durante a definição das permissões dos leitores. O autor pode definir que deseja receber apenas determinados tipos de mensagens sobre o uso de seu hiperdocumento. Um segundo momento é quando define para o módulo autor que filtre as mensagens recebidas e exiba apenas as desejadas.

Apesar disto, informações importantes podem estar passando desapercebidas pelo autor. Uma ferramenta de mineração poderia inferir algumas conclusões importantes.

8.5.8 Edição Cooperativa

Ainda hoje a edição cooperativa é um desafio para os pesquisadores. A arquitetura proposta prevê a permissão de acesso para edição do hiperdocumento. Todavia, para explorar este recurso seria necessário a utilização de uma ferramenta de edição.

Como trabalhos futuros nesta linha de pesquisa, sugere-se a integração da arquitetura com alguma ferramenta de edição, principalmente alguma fortemente estruturada. De forma que seja possível não apenas editar o conteúdo do hiperdocumento, mas também a sua própria estrutura. A complexidade deste controle cresce conforme a sua granularidade, se por capítulo, parágrafo ou outro item de nível mais detalhado.

Bibliografia

[ADO 99] ADOBE SYSTEMS. **Acrobat**. Disponível por WWW em http://www.adobe.com/products/acrobat/main.html (dez.1999).

[AND 96] ANDERSON, C.; BURNS, C.; KLEMME, M. Watermarking for the HyperWave Hypermedia System. In: THE WORLD CONFERENCE OF THE WEB SOCIETY, WebNet, 1996, San Francisco. **Proceedings...** Disponível por WWW em http://hyperg.uni-paderborn.de/0x83ea6001_0x0004a6d9 (dez.1997).

[ARR 93] ARRUDA, J.R.C.; ARRUDA, C.M.M. **Segurança de Dados e Criptografia**. Rio de Janeiro: Centro de Produção da UERJ, 1993.

[BEN 95] BENDER, W. et al. Techniques for Data Hiding. In: SPIE, 1995, San Jose. **Proceedings...** Disponível por WWW em http://www.almaden.ibm.com/journal/sj/mit/sectiona/bender.html (dez.1997).

[BER 97] BERGHEL, H. & O'GORMAN, L. **Digital Watermarking**. Disponível por WWW em http://www.acm.org./~hlb/publications/dig_wtr/dig_watr.html (dez.1997)

[BHA 98] BHATTACHARJEE, S. K.; KUTTER, M. Compression Tolerant Image Authentication. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP, 1998, Chicago. **Proceedings...** Disponível por WWW em http://ltswww.epfl.ch:1248/kutter/watermarking/publications/icip98ps.gz (dez.1998)

[BON 96] BONEY, L.; TEWFIK, A.H.; HAMDY, K.N. Digital Watermarks for Audio Signals. In: EUROPEAN SIGNAL PROCESSING CONFERENCE, EUSIPCO, 8.,1996. **Proceedings...** Disponível por ftp em ftp.ee.umn.edu/pub/tewfik/1996/multimedia/eufinal.ps.Z (nov.1997).

[BOR 96] BORS, A.; PITAS, I. Image Watermarking Using DCT Domain Constraints. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP, 1996, Lausanne, Switzerland. **Proceedings...** Disponível por WWW em http://poseidon.csd.auth.gr/papers/PUBLISHED/ CONFERENCE/124/Bors96d.ps.Z (dez.1997).

[BRA 94] BRASSIL, J. et al. Electronic Marking and Identification Techniques to Discourage Document Copying. In: IEEE THE CONFERENCE ON COMPUTER COMMUNICATIONS, INFOCOM, 1994, Toronto, Canada. **Proceedings...** Disponível por ftp em ftp.research.att.com/dist/brassil/ 1994/infocom94a.ps.Z (dez.1997).

[BRA 95] BRASSIL, J. et al. Hiding Information in Document Images. In: ANNUAL CONFERENCE ON INFORMATION SCIENCES AND SYSTEMS, 29., 1995. **Proceedings...** [S.l]: Johns Hopkins University, 1995. P.484-489. Disponível por ftp em ftp.research.att.com/dist~brassil/1995/ciss95.ps.Z (dez.1997).

[BRO 99] BROWN, A. **S-Tools**. Disponível por WWW em ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/s-tools4.zip (dez.1999).

[CHO 94] CHOUDHURY, A.K. et al. **Copyright Protection for Eletronic Publishing over Computer Networks**. Murray Hill: AT&T Bell Laboratories, 1994. Disponível por ftp em ftp.research.att.com/dist/anoncc/copyright.epub.ps.Z (out.1997).

[COO 99] COOPER, O. **Introduction to Cryptography**. Disponível por WWW em http://www.geocities.com/HotSprings/Resort/1698/intro_crypt.html (fev.1999).

[COP 99] COOPER, O. **An introduction to modern cryptography**. Disponível por WWW em http://www.cs.bris.ac.uk/~cooper/crypto.html (fev.1999).

[COS 99] COSTAS, C.; **Cryptography - An Overview**. Disponível por WWW em http://www.hack.gr/users/dij/crypto (fev.1999).

[COX 95] COX, I.J. et al. **Secure Spread Spectrum Watermarking for Multimedia**. NEC Technical Report 95-10. Princeton: NEC Research Institute, 1995. Disponível por WWW em http://www.neci.nj.nec.com/tr/neci-abstract-95-10.html (nov.1997).

[COX 96] COX, I.J. et al. Secure Spread Spectrum Watermarking for Images, Audio and Vídeo. In: INTERNATIONL CONFERENCE ON IMAGE PROCESSING, ICIP, 1996. **Proceedings...** [S.l]. Disponível por ftp em ftp.nj.nec.com/pub/ingemar/papers/icip96.zip (dez.1997)

[COX 97] COX, I.J.; LINNARTZ, J.M.G. Public Watermarks and Resistance to Tampering. In: INTERNACIONAL CONFERENCE ON IMAGE PROCESSING, ICIP, 1997. **Proceedings...** [S.l]. Disponível por e-mail em ingemar@research.nj.nec.com (dez.1997).

[COX 97] COX, I.J.; MILLER, M.T. A review of watermarking and the importance of perceptual modeling. In: ELECTRONIC IMAGING, 1997. **Proceedings...** [S.l]. Disponível por ftp em ftp.nj.nec.com/pub/ingemar/papers/ei97.zip (nov.1997).

[CRA 96] CRAVER, S. et al. **Can Invisible Watermarks Resolve Rightful Ownerships?**. Research Report RC 20509. Disponível por WWW em http://www.watson.ibm.com:8080/main-cgi-bin/search_paper.pl/entry_ids=8214 (nov.1997).

[DEM 99] DEMCOM. **Steganos**. Disponível por WWW em http://www.demcom.com/english/steganos/ (dez.1999).

[DIG 97] DIGIMARC CORPORATION. **Digimarc Watermarking Guide**. Portland: Digimarc Corporation, 1997. Disponível por WWW em http://www. digimarc.com/dloadfile/ReadMarc15Setup.exe (dez.1997).

[DIG 99] DIGIMARC CORPORATION. **Digimarc**. Disponível por WWW em http://www.digimarc.com (dez.1999).

[FRI 97] FRIDRICH, J. **Methods for Data Hiding**. Disponível por WWW em http://ssie.binghamton.edu/~jirif/Research/paper2/hiding.html (dez.1997)

[HAM 88] HAMMES, B.J. **Elementos Básicos do Direito de Autor Brasileiro**. São Leopoldo: Unisinos, 1988. Tese de Doutorado na Faculdade de Direito da Ldwig-Maximilians-Universität, Munique, Alemanha. Tese defendida em 1974.

[HAR 96] HARTUNG, F.; GIROD, B. Digital Watermarking of Raw and Compressed Video. In: EUROPEAN EOS/SPIE SYMPOSIUM ON ADVANCED IMAGING AND NETWORK TECHNOLOGIES, 1996, Berlin, Germany. **Proceeding...** Disponível por WWW em http://www.nt. e-technik.uni-erlangen.de/~hartung/publications/berlin.ps. gz (dez.1997).

[HAR 97] HARREL, W. **Adobe Acrobat for Dummies**. Foster City CA: IDG Books, 1987.

[HAR 97A] HARTUNG, F.; GIROD, B. Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP, 1997, Santa Barbara. **Proceedings...** Disponível por WWW em http://www.nt. e-technik.uni-erlangen.de/~hartung/ publications/ecmast97.ps.gz (dez.1997).

[HAR 97B] HARTUNG, F.; GIROD, B. Fast Public-Key Watermarking of Compressed Video. In: INTERNACIONAL CONFERENCE ON IMAGE PROCESSING, 1997, Santa Barbara. **Proceedings...** Disponível por WWW em http://www.nt.e-technik.uni-erlangen.de/ ~hartung/publications/icip97.ps.gz (dez.1997).

[HAR 99] HARTUNG, F.; KUTTER, M. Multimedia Watermarking Techniques. In: IEEE SPECIAL ISSUE ON IDENTIFICATION AND PROTECTION OF MULTIMEDIA INFORMATION, 1999. **Proceedings...** [S.l:s.n], 1999.

[HEA 99] HEATH, J. **How electronic encryption works and how it will change your business**. Disponível por WWW em http://www. iinet.net.au /~heath/crypto.html/#laws (fev.1999).

[ISO 86] INTERNATIONAL STANDARDS ORGANIZATION. **Standard Generalized Markup Language (SGML)**, ISO 8879, Geneva, Switzerland, 1986.

[ISO 89] INTERNATIONAL STANDARDS ORGANIZATION. **Office Document Architecture (ODA)**, ISO 8813, Geneva, Switzerland, 1989.

[ISO 93] INTERNATIONAL STANDARDS ORGANIZATION, **Hypermedia/Time-based Structuring Language**, ISO 10744, Geneva, Switzerland, 1993.

[JOH 99] JOHHNSON, N.J. **Steganography**. Disponível por WWW em http://www.ise.gmu.edu/~njohnson/stegdoc (mar.1999).

[KOC 95] KOCH, E.; ZHAO, J. Towards Robust and Hidden Image Copyright Labeling. In: WORKSHOP ON NONLINEAR SIGNAL AND IMAGE PROCESSINGM, 1995, Halkidiki, Greece. **Proceedings...** Disponível por WWW em http://www.crcg.edu/~jzhao/jzhao/ pubs/IEEE_Hidden.ps (dez.1997)

[KAH 67] KAHN, D. **The Codebreakers**. New York, NY: The Macmillan Company, 1967.

[KUH 99] KUHN, M. **Stirmark**. Disponível por WWW em http://www.cl.cam.ac.uk/~mgk25/stirmark.html (dez.1999).

[KUT 99A] KUTTER, M.; PETITCOLAS, F. A fair benchmark for image watermarking systems. In: SECURITY AND WATERMARKING OF MULTIMEDIA CONTENTS, SPIE, 1999, San Jose, California. **Proceedings...** Disponível por WWW em http://ltswww.epfl.ch:1248/kutter/watermarking/publications/ei99.ps.gz (mar.1999)

[KUT 99B] KUTTER, M. Watermarking resisting to translation, rotation and scaling, In: MULTIMEDIA SYSTEMS AND APPLICATIONS, SPIE, 1998, Boston, USA. **Proceedings...** Disponível por WWW em http://ltswww.epfl.ch:1248/kutter/watermarking/publications/ScaleRot.ps.gz (dez.1998)

[LAN 97] LANGELAAR, G.C.; LUBBE, J.C.A.; BIEMOND, J. **Copy Protection for Multimedia Data based on Labelling Techniques**. Disponível por WWW em http://www-it.et.tudelft.nl/pda/smash/public/benelux_cr.html (nov.1997).

[LIM 95] LIMA, J.V. Hiperdocumento: Multimídia, Estruturação e Acesso Interativo. In: Escola Regional de Informática, SBC-RS, 3., 1995, Caxias do Sul. **Proceedings...**

[LOW 88] LOW, S.H.; MAXEMCHUK, N.F. **Performance Comparison of Two Text Marking Methods**. Disponível por WWW em http://www.ee.mu.OZ.AU/staff/slow/papers/comparison.ps (nov.1997).

[LOW 95] LOW, S.H. et al. Document Marking and Identification using Both Line and Word Shifting. In: THE CONFERENCE ON COMPUTER COMMUNICATIONS, Infocom, 1995, Boston. **Proceedings...** Disponível por ftp em ftp.research.att.com/dist/brassil/1995/infocom95.ps.Z (nov.1997).

[LOW 97] LOW, S.H.; MAXEMCHUK, N. F.; LAPONE, A. **Document Identification for Copyright Protection using Centroid Detection**. Disponível por WWW em http://www.ee.mu.oz.au/staff/slow/papers/centroid_det.ps (dez.1997).

[MAC 99] MACHADO, R. **EzStego**. Disponível por WWW em http://www.stego.com (dez.1999).

[MAN 94] MANCINO, P. **Can the Open Document Architecture (ODA) standard change the world of Information Technology?** Ericsson Telecom report, 1994. Disponível por WWW em http://www.ericsson.nl/ Library/Report-ODA.ps.gz (nov.1997).

[MAX 94] MAXEMCHUK, N.F. Eletronic Document Distribution. **ATT Technical Journal**, [S.l], p.73-80, Sept.1994. Disponível por ftp em ftp.research.att.com/dist/anoncc/epub.ps.Z (dez.1997).

[McK 98] McKINLEY, T. **Do Papel até a Web**. São Paulo: Quark Books, 1998.

[NIK 96] NIKOLAIDIS, N.; PITAS, I. Copyright protection of images using robust digital signatures. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH A ND SIGNAL PROCESSING, ICASSP, 1996, **Proceedings...** Disponível por WWW em http://poseidon.csd.auth.gr/papers/PUBLISHED/CONFERENCE/134/icassp96.ps.Z (dez.1997).

[NOR 73] Norman, B. **Secret Warfare**. Washington: Acropolis Books, 1973.

[NYC 93] NYCUM, S.H.; MCKENZIE, B. Protecting Intellectual Property Rights in Software. In: CONFERENCE TRI-ADA, 1993, Seattle, Washington. **Proceedings...** Disponível por WWW em http://www.acm.org/pubs/articles/proceedings/ada/170657/p410-nycum/p410-nycum.pdf (nov.1997).

[PDF 99] PDF.COM.BR. **PDF e HTML: Integração.** Disponível por WWW em http://www.pdf.com.br/introducao/pdf_html.html (mai.1999).

[PET 99A] PETITCOLAS, F.A.P.; ANDERSON, R.J. Evaluation of copyright marking systems. In: IEEE MULTIMEDIA SYSTEMS, ICMCS, 1999, Florence, Italy. **Proceedings...** [S.l:s.n.], 1999.

[PET 99B] PETITCOLAS, F.A.P.; ANDERSON, R.J.; KUHN, M.G. Attacks on copyright marking systems. In: INTERNATIONAL WORKSHOP OF INFORMATION HIDING, IH, 2., 1998, Portland, Oregon. **Proceedings...** Disponível por WWW em http://www.cl.cam.ac.uk/~fapp2/papers/ih98-attacks (mai.1999)

[PET 99C] PETITCOLAS, F.A.P. **Weakness of existing watermarking schemes - StirMark vs unZign.** Disponível por WWW em http://www.cl.cam.ac.uk/~fapp2/stirmark_unzign.htm (mar.1999).

[PET 99D] PETITCOLAS, F.A.P. **Stirmark Benchmark**. Disponível em http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/experiments/ResulttableII.xls (mar.1999)

[PIR 99] PIRON, L. et al. OCTALIS benchmarking: Comparison of four watermarking techniques. In: SECURITY AND WATERMARKING OF MULTIMEDIA CONTENTS, SPIE, 1999, San Jose, California. **Proceedings**... Disponível por WWW em http://ltswww.epfl.ch:1248/kutter/watermarking/publications/ei99oc.pdf.gz

[POL 97] PODILCHUK, C.I.; ZENG, W. **Perceptual Watermarking of Still Images**. In: WORKSHOP ON MULDIMEDIA SIGNAL PROCESSING, 1997, Princeton, New Jersey. Disponível por WWW em http://www.ee.princeton.edu/~wzeng/wm_workshop.html (nov.1997).

[PRO 97] PROKOPETZ, K.; LIMA, J.V. **Watermark em Documentos Eletrônicos para Proteção de Direitos de Autor**: trabalho individual. Porto Alegre: CPGCC da UFRGS, 1997. Disponível por WWW em http://www.inf.ufrgs.br/~klaus.

[PRO 97A] PROKOPETZ, K.; LIMA, J.V. Navegação Personalizada de Hiperdocumentos em Ambientes de Ensino-Aprendizagem. In: CONGRESSO INTERNACIONAL DE INFORMÁTICA EDUCATIVA, 1997, Buenos Aires, Argentina. **Anais...** Disponível por WWW em http://www.inf. ufrgs.br/~klaus.

[RSA 99] RSA Security. Disponível por WWW em http://www.rsa.com (out.1999).

[RUA 96] RUANAIDH Ó et al. Watermarking Digital Images for Copyright Protection. In: ELECTRONIC IMAGING AND THE VISUAL ARTS, 1996, Florence, Italy. **Proceedings...** Disponível por WWW em http://cuiwww.unige.ch/õrvanaid/eva_pap.html.

[RUA 97] RUANAIDH Ó et al. **Watermarking Digital Images for Copyright Protection**. Disponível em http://cuiwww.unige.ch/~oruanaid/ieejnl.ps.gz (dez.1997).

[SCH 96] SCHNEIER, B. **Applied Cryptography Second Edition:** protocols, algorithms, and source code in C. New York: John Wiley & Sons, 1996.

[SIG 99] SIGNAL PROCESSING LABORATORY. **JK_PGS**. Disponível por WWW em http://ltswww.epfl.ch/~kutter/watermarking/JK_PGS.html (dez.1999).

[SIN 99] SIGNUM TECHNOLOGIES. **Suresign**. Disponível por WWW em http://www.signumtech.com/suresign/index.html (dez.1999).

[SMI 96] SMITH, J.R.; COMISKEY, B.O. Modulation and Information Hiding in Images. In: WORKSHOP ON INFORMATION HIDING, Isaac Newton Institute, 1996, University of Cambridge, UK. **Proceedings...** Disponível por WWW em http://physics.www.media.mit.edu/~jrs/hiding.ps (nov.1997).

[SON 98] SONGERWALA, M.; LEVY, E.; TSELEPIS, S. **DocMark:** Secure Documento Marking and Distribution. Disponível por WWW em http://www.ctr.columbia.edu/~maz/netsec.html (mai.1998).

[SSH 99] SSH COMMUNICATIONS SECURITY. **Cryptography A-2-Z**. San Jose, CA, USA: SSH Communications Security, 1998. Disponível por WWW em http://www.ssh.fi/tech/ crypto/intro.html(fev.1999).

[UNZ 99] UNZIGN. Disponível por WWW em http://www.altern.org/watermark (dez.1999).

[VOY 96] VOYATSIS, G.; PITAS, I. Applications of Toral Automorphisms in Image Watermarking. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP, 1996, Lausanne, Switzerland. **Proceedings...** Disponível por WWW em http://poseidon.csd.auth.gr/papers/PUBLISHED/CONFERENCE/148/Voyatzis96b.ps.Z (nov.1997).

[VOY 96] VOYATSIS, G.; PITAS, I. Chaotic Mixing of Digital Images and Applications to Watermarking. In: EUROPEAN CONFERENCE ON MULTIMEDIA APPLICATIONS, SERVICES AND TECHNIQUES, ECMAST, 1996, Louvain-la-Neuve, Belgium. **Proceedings...** Disponível por WWW em http://poseidon.csd.auth.gr/papers/PUBLISHED/CONFERENCE/147/Voyatzis96a.ps.Z (nov.1997).

[VOY 97] VOYATSIS, G.; PITAS, I. Embedding Robust Watermarks by Chaotic Mixing. In: INTERNATIONAL CONFERENCE ON DIGITAL SIGNAL PROCESSING, DSP, 13., 1997, Santorini, Greece. **Proceedings**... Disponível por WWW em http://poseidon.csd.auth.gr/papers/PUBLISHED/CONFERENCE/173/Voyatzis97a.ps.Z (nov.1997).

[WAY 97] WAYNER, P. **Digital Copyright Protection**. Chestnut Hill: AP Professional, 1997.

[WEB 98] WEBER, R.F. **Criptografia Contemporânea**. Disponível por ftp em ftp://caracol.inf.ufrgs.br/pub/crypt/Cripto.doc (abr.1998).

[ZEN 97] ZENG, W.; LIU,B. On Resolving Rightful Ownerships of Digital Images by Invisible Watermarks. In: INTERNACIONAL

CONFERENCE ON IMAGE PROCESSING, ICIP, 1997. **Proceedings...** Disponível por ftp em ftp://ee.princeton.edu/pub/wzeng/icip97_wm.ps.gz (dez.1997).

[ZHA 96] ZHAO, J. A WWW Service to Embed and Prove Digital Copyright Watermarks. In: EUROPEAN CONFERENCE ON MULTIMEDIA APPLICATIONS, Services and Techniques, 1996, Louvain-La-Neuve, Belgium. **Proceedings...** Disponível por WWW em http://www. crcg.edu/~jzhao/jzhao/pubs/ecmast96.ps (dez.1997).

[ZHA 96] ZHAO, J.; KOCH, E. A Digital Watermarking System for Multimedia Copyright Protection. In: INTERNATIONAL MULTIMEDIA CONFERENCE, ACM Multimedia, 4.; 1996, Boston. **Proceedings...** Disponível por WWW em http://www.acm.org/pubs/articles/proceedings/multimedia/244130/p443-zhao/p443-zhao.pdf (nov.1997).

[ZHA 97] ZHAO, J. **Look, It's Not There:** Digital Watermarking is the best way to protect intellectual property from illicit copying. Disponível por WWW em http://www.byte.com/art/9701/sec18/art1.htm (out.1997).

Informática  **UFRGS**

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

"Uma Arquitetura para Controle e Proteção de Direitos Autorais de Hiperdocumentos na Internet"

por

Klaus Prokopetz

Dissertação apresentada aos Senhores:

Profa. Dra. Carla Maria Dal Sasso Freitas

Prof. Dr. Carlos Alberto Heuser

Profa. Dra. Karin Becker (PUCRS)

Vista e permitida a impressão.
Porto Alegre, 23 / 02 / 2000.

Prof. Dr. José Valdeni de Lima,
Orientador.

Prof. Dr. Raul Fernando Weber,
Co-orientador.

Profa. Carla Maria Dal Sasso Freitas
Coordenadora do Programa de Pós-Graduação
em Computação - PPGC
Instituto de Informática - UFRGS