UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

**Criptografia – Uma Implementação
do Protocolo de Micropagamento
*PayWord***

João Carlos Néto

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO GRAU DE MESTRE
EM
CIÊNCIA DA COMPUTAÇÃO

Orientador: Prof. Dr. Routo Terada

– São Paulo, maio de 2002 –

Este exemplar corresponde à redação final
da dissertação devidamente corrigida e
defendida por João Carlos Néto
e aprovado pela Comissão Julgadora.

São Paulo, 07 de maio de 2002.

Comissão Julgadora:

• Prof. Dr. Routo Terada (Orientador)	IME – USP
• Profa. Dra. Ana Cristina Vieira de Melo	IME – USP
• Prof. Dr. Wilson Vicente Ruggiero	EP – USP

Agradecimentos

Ao professor Routo Terada, meu orientador e professor, pelo inestimável apoio na definição desta dissertação, na orientação e confiança depositada no meu estudo e neste trabalho.

Aos professores do Departamento de Ciência da Computação do IME/USP que, de uma maneira direta ou indiretamente, colaboraram na formação acadêmica do meu mestrado.

Aos professores Wilson Ruggiero e Ana Cristina, meus professores e membros da banca de qualificação, pelas orientações, críticas e observações referentes a esta dissertação.

Ao Banespa e aos colegas do Banco que me apoiaram e incentivaram durante todo o período de desenvolvimento deste estudo.

À minha esposa Amália e minhas filhas Milene e Karla cujo amor, carinho e compreensão foram importantes na realização deste trabalho e, também, pelo enorme apoio, auxílio e estímulo que tornaram menos árduos os esforços feitos.

Resumo

Esta dissertação apresenta uma análise dos esquemas de Micropagamento Eletrônico *PayWord*, *MicroMint* e *MilliCent*, que empregam técnicas de criptografia para produzir e controlar a utilização da moeda eletrônica, por um Sistema de Pagamento Eletrônico para transações de pequeno valor monetário. O trabalho descreve em detalhes o Sistema Protótipo de Micropagamento que foi desenvolvido utilizando o protocolo *PayWord*, de forma a não só atender aos requisitos e às características de uma aplicação de comércio eletrônico, como também contemplar operações de venda, pagamento e entrega de produtos e informações pela Internet.

Abstract

This dissertation presents an analysis of the Electronic Micropayments schemes, *PayWord*, *MicroMint* and *MilliCent*, that use cryptography techniques to produce and control the use of the electronic coin, by an Electronic Payment System for low-value transactions. The work describes in detail the Micropayment Prototype System that was developed using the *PayWord* protocol, not only to assist the requirements and characteristics of an electronic commerce application, as well as to contemplate sale operations, payment and delivery of goods and information by Internet.

Sumário

1	Introdução	1
1.1	Objetivo e Motivação	2
1.2	Trabalhos Relacionados	2
1.3	Contribuições	3
1.4	Organização	4
2	Sistemas de Pagamento Eletrônico.....	5
2.1	Visão Geral	5
2.2	Requisitos para Sistemas de Pagamento Eletrônico	6
2.2.1	Atomicidade	6
2.2.2	Segurança	6
2.2.3	Privacidade	7
2.2.4	Escalabilidade	7
2.2.5	Interoperabilidade	7
2.3	Classificação dos Sistemas de Pagamento Eletrônico	8
2.3.1	Operação <i>On-line</i> versus <i>Off-line</i>	8
2.3.2	Macropagamento versus Micropagamento	8
2.3.3	Solução por Hardware versus Software	8
2.4	Protocolos de Segurança e Pagamento	9
2.4.1	Ataques à Segurança de Computadores	9
2.4.2	Requisitos para um Sistema de Pagamento Eletrônico Seguro	10
2.5	Visão Geral dos Sistemas de Pagamento Eletrônico	11
2.5.1	Sistemas Tradicionais de Pagamento Baseados na Internet	11
2.5.2	Dinheiro Eletrônico	15

3	<i>PayWord</i>	16
3.1	Visão Geral	17
3.2	Detalhes do <i>PayWord</i>	18
3.2.1	Certificação do Usuário pelo <i>Broker</i>	18
3.2.2	Compras	18
3.2.3	Compensação dos Pagamentos pelo <i>Broker</i> ao Vendedor	21
3.3	Avaliação de Segurança	21
3.3.1	Prevenção de Falsificação	21
3.3.2	Detecção de Gasto Duplicado	22
3.3.3	Outros Comentários	22
4	<i>MicroMint</i>	24
4.1	Visão Geral	25
4.2	Detalhes do <i>MicroMint</i>	26
4.2.1	Produção de Moedas Eletrônicas	26
4.2.2	Compra de Moedas Eletrônicas Específicas por Usuário	28
4.2.3	Compras	29
4.2.4	Compensação dos Pagamentos pelo <i>Broker</i> ao Vendedor	29
4.3	Avaliação de Segurança	30
4.3.1	Prevenção de Falsificação	30
4.3.2	Detecção de Gasto Duplicado	31
4.3.3	Outros Comentários	31
5	<i>MilliCent</i>	32
5.1	Visão Geral	32
5.2	Detalhes do <i>MilliCent</i>	34
5.2.1	Estrutura da Moeda Eletrônica <i>Scrip</i>	34
5.2.2	<i>Master Scrip Secret</i> e Certificação do <i>Scrip</i>	35
5.2.3	Validação do <i>Scrip</i>	36
5.2.4	<i>Customer Secret</i> e <i>Master Customer Secret</i> do <i>Scrip</i>	37
5.2.5	Protocolos do <i>MilliCent</i>	38
5.3	Interações entre <i>Broker</i> , Vendedor e Consumidor	40

5.3.1	Compra de <i>scripVendor</i> pelo <i>Broker</i>	40
5.3.2	Compra de <i>scripBroker</i> pelo Usuário	42
5.3.3	Compra de <i>scripVendor</i> pelo Usuário.....	42
5.3.4	Pagamento com <i>scripVendor</i> pelo Usuário ao Vendedor	42
5.4	Avaliação de Segurança	43
5.4.1	Prevenção de Falsificação	43
5.4.2	Detecção de Gasto Duplicado	43
5.4.3	Outros Comentários	43
6	Comparação dos Protocolos.....	44
6.1	<i>PayWord</i> versus <i>MicroMint</i>	44
6.2	<i>MilliCent</i> versus <i>MicroMint</i>	45
6.3	Comparação de Propriedades.....	47
6.4	Escolha do <i>PayWord</i>	47
7	Arquitetura	49
7.1	Componentes.....	49
7.2	Interações	50
7.3	Infra-Estrutura.....	52
7.4	Bancos de Dados	53
7.4.1	Banco de Dados do Servidor de Certificação/Compensação	54
7.4.2	Banco de Dados da Carteira Eletrônica	56
7.4.3	Banco de Dados do Servidor de Páginas <i>Web</i> e do Servidor de Pagamentos.....	58
7.5	Comunicação.....	60
7.5.1	<i>Browser</i> versus Servidor de Páginas <i>Web</i> e Carteira Eletrônica.....	60
7.5.2	Carteira Eletrônica versus Servidores de Pagamentos e Certificação / Compensação	61
7.5.3	Servidor de Pagamentos versus Servidor de Certificação/Compensação	62
7.6	Criptografia e Certificado Digital	63
7.6.1	Geração de Certificados Digitais	64
7.6.2	Assinatura Digital	67
7.6.3	Verificação de Assinatura Digital	68

7.6.4	Criptografia RSA	70
7.6.5	Função <i>Hash</i>	73
8	Implementação	75
8.1	Servidor de Certificação/Compensação	75
8.1.1	Componentes e Estrutura	76
8.1.2	Monitoração	77
8.1.3	Cadastramento de Novo Usuário/Cliente.....	78
8.1.4	Lista dos Usuários/Clientes.....	79
8.1.5	Lista dos Certificados Digitais.....	79
8.1.6	Lista das Vendas	80
8.2	Carteira Eletrônica	81
8.2.1	Componentes e Estrutura	81
8.2.2	<i>Login</i> do Usuário.....	82
8.2.3	Abertura de Nova Conta	83
8.2.4	Certificação da Nova Conta	84
8.2.5	Monitoração da Certificação.....	85
8.3	Servidor de Pagamentos.....	87
8.3.1	Componentes e Estrutura	87
8.3.2	Monitoração	88
8.3.3	Compromisso e Pagamento.....	89
8.3.4	Compensação	91
8.3.5	Lista das Vendas	92
8.4	Loja Virtual	93
8.4.1	Componentes e Estrutura	93
8.4.2	<i>Home Page</i>	95
8.4.3	Lista de Produtos.....	96
8.4.4	Compra.....	97
8.4.5	Pagamento.....	98
8.4.6	<i>Download</i> da Compra	99
8.4.7	Produtos Adquiridos	100
8.4.8	Registro de Operações	101

8.4.9	<i>Download</i> da Carteira Eletrônica	102
9	Conclusões e Perspectivas	103
	Referências Bibliográficas	106

Capítulo 1

Introdução

Prover conteúdo é uma das principais atrações e benefícios da Internet [34]. Com o seu crescimento nos últimos anos, criou-se um mercado eletrônico para produtos e serviços. Na virada do século XX para o século XXI, mais de 70 milhões de computadores de usuários de microinformática estavam conectados à Internet [28].

O sucesso de *Sites* de comércio eletrônico, como por exemplo, *Amazon.com* [29], mostrou o enorme potencial de negócios na oferta de serviços pela Internet para esse grande número de usuários na venda de produtos pelo uso de seus navegadores (*browsers*).

O comércio de produtos tangíveis ou materiais, entretanto, não é o único negócio nesse mercado virtual. Produtos e serviços intangíveis, como por exemplo, informações, terão igualmente uma importância significativa.

Vender produtos e informações intangíveis pela Internet é um sério problema ainda não totalmente resolvido. A venda de conhecimento, de programas executáveis, de jornais, livros e artigos eletrônicos, de imagens, vídeos, música, bem como de outros produtos intangíveis, de pequeno valor monetário, vislumbra uma nova abordagem de comércio eletrônico onde os compradores desejam imediato acesso ao conteúdo ofertado, facilidade no pagamento e a entrega eletrônica rápida destes produtos.

Hoje, os produtos e serviços intangíveis ofertados na Internet são inviáveis de serem adquiridos pelos sistemas atuais de pagamento eletrônico, como é o caso do cartão de crédito. Tais sistemas de pagamento não admitem transações de pequeno valor monetário, como da ordem de centavos de real, pois foram desenhados para tratar pagamentos de transações de valores acima de dez reais.

Em decorrência desta nova oportunidade no comércio eletrônico via Internet, e levando-se em conta tais dificuldades, surgiu a necessidade de desenvolver novos mecanismos de pagamento eletrônico para esse cenário, como a solução de Micropagamento Eletrônico.

Micropagamento Eletrônico possui, sem dúvida, características e requisitos diferentes dos Sistemas de Pagamentos Eletrônicos tradicionais, exigindo, por conseqüente, não só estudos e análise, como principalmente implementação de solução aceitável.

Nesta dissertação são analisados os seguintes esquemas de Micropagamento: *PayWord* [1], *MicroMint* [1] e *MilliCent* [4].

É descrita inicialmente uma visão geral dos Sistemas de Pagamento Eletrônico, suas propriedades desejáveis, as diferenças entre Macro e Micropagamento, os protocolos de segurança e pagamento.

Em seguida, é apresentado um panorama geral desses protocolos, sua descrição detalhada e partes envolvidas, a estrutura do dinheiro eletrônico, as técnicas de criptografia empregadas, a geração ou produção de moeda eletrônica, as compras e pagamentos, o processo de ressarcimento do dinheiro eletrônico e uma avaliação de segurança envolvendo prevenção de falsificação, detecção de gasto duplicado e outros comentários sobre cada esquema. Da mesma forma é feita uma comparação dos esquemas analisados, resumizando suas similaridades, diferenças e propriedades.

O maior desafio deste estudo foi definir uma arquitetura e posterior implementação de uma solução para um Sistema Protótipo que demonstrasse, através de uma aplicação de Loja Virtual na Internet, um esquema prático e exequível de uso dos mecanismos de Micropagamento Eletrônico para endereçar o problema enunciado.

1.1 Objetivo e Motivação

O principal objetivo desta dissertação é apresentar uma análise de esquemas de Micropagamento Eletrônico e o emprego de técnicas de criptografia, na implementação de uma solução para atender os requisitos e as características a serem aplicadas no comércio eletrônico para transações de pequeno valor monetário, solução essa que contemple a venda, o pagamento e a entrega de produtos/informações realizados através da Internet.

A motivação principal, por sua vez, surgiu do fato de ser um assunto em desenvolvimento, imprescindível para viabilizar novos negócios e produzir modelos de tecnologia os quais constituem requisitos importantes na implementação e para obter sucesso nas soluções de Micropagamento Eletrônico.

1.2 Trabalhos Relacionados

As pesquisas na área de Micropagamento Eletrônico são recentes. Vêm se desenvolvendo com estudos acadêmicos, propostas de esquemas de Micropagamento e implementações para avaliar solução.

Dois esquemas simples de Micropagamento foram apresentados por *Ronald Rivest* e *Adi Shamir* [1] em maio de 1996. São os esquemas de Micropagamento *PayWord* e *MicroMint* que serão apresentados nos capítulos 3 e 4. Um protótipo do esquema *MicroMint* foi apresentado por *Burstein* [30] em maio de 1998.

A *Digital Equipment Corporation*, atualmente parte da empresa *Compaq Computer Corporation*, propôs o *MilliCent*, esquema de Micropagamento, que será analisado no capítulo 5. Um sistema protótipo foi implementado com o protocolo *MilliCent* e encontra-se em *beta tests* na Internet [4].

Outra proposta de esquema de Micropagamento foi da IBM, anteriormente chamado de *MiniPay*. A implementação da IBM é de um Sistema de Pagamento Eletrônico desenvolvido conforme o documento *Common Markup for Micropayment per-fee-links*, do grupo W3C [31]. Atualmente, tal implementação é chamada de *Mpay* e também se encontra em *beta tests* [32].

Nos dias atuais, existem propostas de tecnologia sobre os esquemas de Micropagamento. Uma excelente lista dessas propostas pode ser encontrada na página *Micropayments Overview* do Site W3C [33].

1.3 Contribuições

Como contribuições deste trabalho, espera-se:

- mostrar os principais conceitos, características e requisitos dos Sistemas de Pagamento Eletrônico;
- desenvolver um estudo analisando três propostas de esquemas de Micropagamento Eletrônico;
- apresentar uma comparação dos esquemas de Micropagamento analisados, sintetizando suas similaridades, diferenças e propriedades;
- implementar um Sistema Protótipo de Micropagamento com as funções necessárias para viabilizar uma Loja Virtual na Internet, utilizando o protocolo *PayWord*;
- adaptar e implementar uma Estrutura de Criptografia e de Certificados Digitais para atender a implementação do Sistema Protótipo;
- apresentar uma referência bibliográfica abrangente sobre o assunto Micropagamento Eletrônico.

1.4 Organização

Os assuntos tratados no presente estudo estão organizados da seguinte maneira:

- o capítulo 2 começa com os requisitos para qualquer Sistema de Pagamento pela Internet e dá uma introdução e visão geral sobre os Sistemas de Pagamentos existentes;
- o capítulo 3 provê detalhes do esquema de Micropagamento *PayWord*, proposto por *Ronald Rivest*, do *MIT LCS* e *Adi Shamir*, do *Weizmann Institute of Science*. Esta tese apresenta a Arquitetura e Implementação de um Sistema Protótipo de Micropagamento, utilizando o protocolo *PayWord*;
- o capítulo 4 descreve detalhes do esquema de Micropagamento *MicroMint*, também proposto por *Ronald Rivest* e *Adi Shamir*;
- o capítulo 5 apresenta pormenores do esquema de Micropagamento *MilliCent*, desenvolvido pela Compaq para comércio eletrônico pela Internet;
- o capítulo 6 desenvolve uma análise comparativa dos três esquemas de Micropagamento estudados, resumando suas similaridades, diferenças e propriedades;
- o capítulo 7 descreve as principais características da Arquitetura para implementação do Sistema Protótipo de Micropagamento que foi desenvolvido utilizando-se o protocolo *PayWord*;
- o capítulo 8 expõe com detalhes a Implementação dos componentes do Sistema Protótipo de Micropagamento que foi desenvolvido utilizando-se a Arquitetura apresentada capítulo 7;
- o capítulo 9 apresenta as considerações finais da dissertação.

Capítulo 2

Sistemas de Pagamento Eletrônico

2.1 Visão Geral

O comércio eletrônico começou há mais de vinte anos, com o estabelecimento do padrão de troca eletrônica de documentos (EDI – *Electronic Data Interchange*) [38] entre empresas, como ordens de compra, faturas, entregas e outros. EDI é baseado em um somatório de formatos padronizados que definem conjuntos de transações (ou mensagens) que podem ser usados para enviar dados básicos de negócios de um computador para outro entre empresas. Esses conjuntos de transações substituem os documentos em papel, tais como ordens de compra, faturas ou conhecimentos de embarque.

A partir de 1990, com o avanço dos sistemas de comunicação baseados na Internet, conciliando-se as técnicas de redes de computadores com *hypertext*, nasce a grande rede global (*World Wide Web*), ou simplesmente *Web*.

Em 1994, os navegadores (*browsers*) com interfaces amigáveis foram viabilizados, obtendo-se imediato impacto na propagação da *Web*. As empresas e o público em geral reconheceram o potencial da rede global. Isso fez com que as empresas começassem a disponibilizar seus serviços e informações através da *Web* para clientes em qualquer parte do mundo. Nessa época, as instituições empresariais iniciaram pesquisas para obter receitas através dos serviços e informações fornecidos na *Web*.

O primeiro passo para obter lucro com esse novo negócio foi colocar anúncios nas páginas *Web*. As empresas pagavam para colocar anúncios no formato de pequenas figuras ou animações em *Web Sites* com alto número de acessos. Para aumentar suas receitas, elas estabeleceram lojas virtuais na *Web* vendendo produtos diretamente aos consumidores. Dessa forma, os consumidores puderam pesquisar e encontrar os menores preços de produtos e serviços. Nessa ocasião, contudo, não havia tecnologia para se fazer os pagamentos via Internet.

A primeira abordagem no processo de solução do problema foi atendida com o envio do número do cartão de crédito do consumidor para o vendedor, através de formulários eletrônicos pela Internet, sem o uso de técnicas de criptografia. No caso de transações de valores altos, esse método era muito inseguro. Mas, para transações de valores baixos, como o pagamento de informações contidas em páginas *Web* ou a entrega eletrônica de conteúdos, como jornal, revistas, artigos, etc., tal método não seria aplicável, visto que a informação transmitida poderia ser tão pequena que não compensariam as taxas envolvidas com cartão de crédito.

Dáí resultou ser necessário estabelecer diferentes protocolos de pagamento para cada modelo de negócio.

2.2 Requisitos para Sistemas de Pagamento Eletrônico

As soluções de Pagamento Eletrônico devem possuir as propriedades descritas a seguir.

2.2.1 Atomicidade

As transações devem ser completas ou ser desfeitas em caso de falha. Quando uma transação falhar, é necessário retornar ao último estado consistente, de forma que não haja prejuízo ao usuário. Isso deve garantir que o processo de compra seja completado ou retornado ao último estado consistente anterior concluído com sucesso. Essa propriedade é similar a sistemas de bancos de dados transacionais que implementam os procedimentos de *commit*¹ e *rollback*².

2.2.2 Segurança

As transações precisam ser seguras de forma a não permitir fraudes com o dinheiro eletrônico. A segurança é o obstáculo mais importante para a aceitação geral do Sistema de Pagamento Eletrônico. A falsificação do dinheiro em papel ou em moedas é difícil, visto serem necessários equipamentos especiais e profissionais peritos. Por outro lado, dinheiro de circulação eletrônica é apenas um conjunto de dados binários que podem ser copiados facilmente. A cópia ou a rerepresentação desse tipo de dinheiro deve ser prevenida e existir mecanismo de detecção. O ideal é que seja incondicional e computacionalmente impossível criar, copiar ou reutilizar ilegalmente o dinheiro eletrônico. As transações de pagamento eletrônicas devem ser seguras contra interceptação, modificação ou fabricação.

¹ Completa a transação e assegura que todas alterações realizadas no banco de dados serão permanentes.

² Abandona a transação e desfaz qualquer alteração realizada no banco de dados pela transação.

2.2.3 Privacidade

Não permitir rastrear o fluxo do dinheiro de forma a violar a privacidade do usuário. A identidade do indivíduo que utiliza o dinheiro em circulação eletrônica não pode ser revelada. Os usuários desejam que os Sistemas de Pagamento Eletrônico tenham o mesmo grau de privacidade que prevalece na utilização do dinheiro em espécie. Todos sabem que uma compra feita com dinheiro em papel não é rastreável. É impossível que um vendedor saiba quem comprou um determinado item de sua loja apenas olhando para o dinheiro que está no seu caixa. Nem o banco sabe o que o cliente comprou com o dinheiro que ele sacou. Existem, porém, alguns esquemas de pagamento que ignoram a questão da privacidade e permitem rastrear os pagamentos dos usuários. Outros protocolos são incondicionalmente não rastreáveis, onde os gastos de um indivíduo não são determinados, mesmo que as partes estejam em conluio. Para algumas transações pode existir uma forma mais fraca de privacidade, mas o rastreamento é dificultado o suficiente a ponto de o custo de obter determinada informação ser maior do que o benefício pretendido.

2.2.4 Escalabilidade

Os Sistemas de Pagamento Eletrônico devem possuir mecanismos em condições de admitir o crescimento do número de usuários e de absorver a sobrecarga de acessos em certos períodos de pico, sem provocar impacto no desempenho de utilização do sistema pelo usuário final. Isto exige que os equipamentos onde as transações do sistema são processadas sejam escaláveis, facilitando a adição de novos equipamentos quando necessário.

2.2.5 Interoperabilidade

Para alcançar a escalabilidade, os sistemas devem admitir múltiplos servidores onde vão ser processadas as transações dos usuários. Nem sempre, contudo, os usuários são clientes do mesmo ambiente de servidores em que está hospedado o Sistema de Pagamento Eletrônico. Decorre disso que o dinheiro de circulação eletrônica, produzido para um certo sistema, deverá ser aceito e conferido mundialmente. Sem a aceitação mútua, ele só poderia ser usado entre partes que compartilhassem um mesmo dinheiro de circulação eletrônica. Quando o dinheiro de circulação eletrônica é cunhado em um sistema, este deve possibilitar seu intercâmbio automático entre outros que possuam o mesmo protocolo de pagamento.

2.3 Classificação dos Sistemas de Pagamento Eletrônico

Os Sistemas de Pagamento Eletrônico podem ser classificados pelos seguintes critérios:

2.3.1 Operação *On-line* versus *Off-line*

As transações de pagamento podem ser executadas *on-line*, quando participa da operação um servidor para autenticação e autorização em cada pagamento, ou *off-line*, quando não há participação de uma terceira parte durante o pagamento entre o comprador e o vendedor.

2.3.2 Macropagamento versus Micropagamento

Muitas das atuais soluções de Comércio Eletrônico são focadas em Sistemas de Macropagamento, como por exemplo, *CyberCash* [39, 40], *Mondex* [41] e *NetCash* [42]. Tais sistemas possuem flexibilidade nas compras pela Internet de produtos e serviços com grau de segurança e confiabilidade e estão desenhados para compras no montante de R\$ 10,00 ou superior.

Como característica, os Sistemas de Macropagamento trabalham de duas maneiras: o cliente paga a compra com cartão de crédito, ou o cliente estabelece uma conta junto ao vendedor. Se o montante do pagamento for significativo, as transações desse sistema são tipicamente lentas, decorrente da necessidade de verificação e autenticação do número do cartão de crédito ou da conta do cliente para autorizar o valor da compra. Também são caras em função das taxas envolvidas de risco e de administração. Obviamente essa abordagem não é praticável para sistemas de pagamento onde tipicamente o montante da compra é da ordem de centavos ou poucos reais.

Os Sistemas de Micropagamento [33] são soluções para compras de produtos e serviços de valor da ordem de centavos ou de poucos reais e possuem flexibilidade nas operações. São também sistemas providos de velocidade suficiente na transação de pagamento, para que o usuário não tenha que esperar na entrega do produto ou serviço adquirido. Por exemplo, se um usuário está pagando R\$ 0,01 para ler um artigo *on-line*, não é admissível a espera de alguns minutos para fazer o pagamento, antes da entrega do artigo para leitura *on-line*.

2.3.3 Solução por Hardware versus Software

Alguns sistemas de pagamento admitem um *hardware* para ampliar a segurança da solução de *software*, como na autenticação do usuário, no controle de acesso ao sistema, nos processos de criptografia, etc. Um exemplo é o uso de *smart cards*; eles provêm um armazenamento resistente a ataques às informações confidenciais do usuário, como a chave privada do processo de criptografia e informações pessoais.

2.4 Protocolos de Segurança e Pagamento

Os Sistemas de Pagamento Eletrônico contam com um conjunto de técnicas de criptografia para atingir a transmissão segura dos dados e o intercâmbio de mensagens autenticadas. Mensagens confidenciais trocadas entre partes por uma rede de comunicação insegura como a Internet, não podem ser lidas por uma atacante que poderá usá-las de modo indevido. Isto precisa ser assegurado num Sistema de Pagamento Eletrônico onde os dados não possam ser lidos, alterados, excluídos ou modificados indevidamente na comunicação entre as partes. Um intruso deve ser impedido de ganhar o controle do sistema e se passar por um usuário autorizado.

2.4.1 Ataques à Segurança de Computadores

Estas são as categorias de ataques à Segurança de Computadores aqui consideradas:

Vazamento: a privacidade do usuário é afetada pela aquisição de suas informações confidenciais por receptor não autorizado.

Falsificação: a modificação não autorizada de informações resulta em perda da integridade ou confidencialidade de dados.

Roubo de recurso: o uso das facilidades do sistema (processamento, memória, espaço em disco, conexão à rede de comunicação, etc.) por indivíduos ou sistemas não autorizados, podem resultar na negação dos serviços do sistema (DoS – *Denial of Service*) a seus usuários autorizados.

Vandalismo: neste ataque, o intruso interfere no sistema para adulterar os dados do usuário ou entidade e não para obter lucro financeiro.

Riscos adicionais devem também ser considerados, quando se usa uma rede de comunicação insegura como a Internet, para o envio de informações de um pagamento:

- mudança do conteúdo de uma mensagem (integridade);
- interceptação de uma mensagem e leitura de seu conteúdo (confidencialidade);
- envio de uma mensagem assinada em nome de outra pessoa (autenticação);
- ausência de prova irrefutável que alguém enviou uma certa informação (não repúdio).

2.4.2 Requisitos para um Sistema de Pagamento Eletrônico Seguro

Para evitar os riscos considerados acima, um Sistema de Pagamento Eletrônico deveria tratar dos seguintes requisitos:

2.4.2.1 *Prevenção de Espionagem*

Requisito que deve assegurar que somente o receptor pode ler uma mensagem confidencial em claro. Este problema pode ser resolvido aplicando-se técnicas de criptografia simétrica ou assimétrica nas mensagens de pagamento entre as partes envolvidas. Assim, um espião não acessará o conteúdo das mensagens de pagamento em tempo suficiente para tirar proveito financeiro.

2.4.2.2 *Prevenção de Disfarce*

A assinatura digital previne que um oponente se disfarce como sendo a outra parte no sistema. Para tanto, a assinatura digital é utilizada para autenticar as mensagens. Ela é baseada em algoritmos criptográficos que atestam não só a integridade, como também a autoria da mensagem. Se uma mensagem for alterada, poderá ser detectada. Assim, a integridade da mensagem é assegurada e a assinatura prova que a mensagem foi emitida pelo pretendido emissor. Qualquer receptor de uma mensagem assinada pode verificar a identidade da pessoa que a assinou.

2.4.2.3 *Prevenção de Mensagem Falsificada*

As mensagens enviadas entre as partes em um Sistema de Pagamento Eletrônico devem ser protegidas contra alterações indevidas. Ao se aplicar sumarizações na mensagem (*message digests*) e assinatura digital, fica assegurado que os dados recebidos de um emissor não foram falsificados por uma terceira parte; ou seja, previne o ataque denominado “*man in the middle*”. Assim, por exemplo, o número do cartão de crédito de um pagamento em uma mensagem não pode ser alterado por um intruso.

2.4.2.4 *Prevenção de Substituição*

Um atacante pode interceptar e copiar uma mensagem de resposta legítima de pagamento e reutilizá-la em um contexto ilegal. Um exemplo dessa possibilidade seria a realização de múltiplos saques não autorizados em uma conta bancária; alguém copia uma mensagem legítima de saque de um sistema de pagamento e substitui várias mensagens de saques com mensagens não legítimas. Para evitar o problema, é incluído um número sequencial nas mensagens e, também, a data e horário de geração do pagamento.

2.5 Visão Geral dos Sistemas de Pagamento Eletrônico

Os Sistemas de Pagamento Eletrônico foram desenvolvidos através do uso da infra-estrutura instalada nos bancos ou empresas de cartão de crédito, criando um sistema de comunicação eletrônica entre o vendedor, o consumidor e a instituição financeira.

Esses sistemas têm sido desenvolvidos com esta abordagem, visando tanto reduzir o custo das transações quanto possuir agilidade e privacidade na realização das operações.

2.5.1 Sistemas Tradicionais de Pagamento Baseados na Internet

2.5.1.1 Sistemas Baseados em Cartão de Crédito

Possuem um mecanismo de pagamento através da infra-estrutura já existente de pagamento por cartão de crédito. Nas transações com esse tipo de cartão, informações confidenciais são trocadas entre as partes. Por exemplo, o cliente fornece o seu número de cartão de crédito para o vendedor, através de um formulário eletrônico, o qual envia eletronicamente para empresa de cartão de crédito; esta confirma se o cartão é válido para autorizar a operação. Em seguida, é feito o ajuste da transferência monetária da compra entre a conta do vendedor e a empresa de cartão de crédito.

Muitos desses sistemas utilizam criptografia para proteger as informações confidenciais do cliente quando da transmissão *on-line* na realização do pagamento. Há implementação de técnicas de criptografia e protocolos de segurança, como por exemplo, SSL – *Secure Socket Layer* [11] ou TLS – *Transport Layer Security* [12].

Os sistemas de pagamento via Internet baseados em cartão de crédito não são desenhados para transações de pequeno valor monetário. Na verdade, eles visam compras de montantes acima de dez reais; abaixo desse limite, o custo para processar a transação poderia ficar acima do preço do produto ou do serviço comprado.

O SET – *Secure Electronic Transmission* [13] é um exemplo de protocolo de pagamento pela Internet baseado em cartão de crédito. Esse protocolo é o resultado de um padrão desenvolvido pela Visa Internacional em aliança com a MasterCard, junto com várias empresas, como a Microsoft, IBM, Netscape, GTE, RSA, Verisign e outras. O sistema usa certificado digital para autenticar todas as partes envolvidas e assegura a integridade das informações do pagamento.

Resumidamente, um pagamento eletrônico que utiliza o SET, é iniciado pelo consumidor que possui um cartão de crédito para pagar um produto ou um serviço adquirido de uma loja virtual na *Web*. O sistema dessa loja virtual comunica-se com um *gateway* de pagamento, chamado *Acquirer*, que estabelece o relacionamento de pagamento entre o vendedor e o consumidor possuidor do cartão de crédito. O *gateway* de pagamento verifica junto à empresa de cartão de crédito se a operação é autorizada. Havendo a autorização da operação, a compra do produto ou do serviço é entregue ao comprador. Depois, o vendedor solicita o ressarcimento financeiro junto à instituição financeira do consumidor, com o auxílio do *gateway* de pagamento.

A Figura 2.1 mostra de forma simplificada a arquitetura do SET.

Figura 2.1 – Arquitetura simplificada do SET

O protocolo SET emprega uma técnica de criptografia chamada assinatura dual, que provê um vínculo das mensagens entre as partes e uma identificação sem a necessidade de ver o conteúdo da mensagem. Uma mensagem de pedido de compra compõe-se de duas partes: uma para o vendedor (informações da compra), e outra para a instituição financeira do consumidor (informações de pagamento). Cada parte é criptografada com a chave pública do dono da mensagem. O pedido de compra também contém sumarizações na mensagem (*message digests*) para cada uma das partes e uma assinatura digital. A assinatura é construída pela geração de uma *message digest* da concatenação dessas duas *message digests* e assinando-a com a chave privada do consumidor (assinatura dual) pelo algoritmo de criptografia assimétrica *RSA* [26]. Esta assinatura dual permite que cada parte possa ler e validar a assinatura da outra parte do pedido de compra, sem decryptografar informações da terceira parte do processo. A Figura 2.2 mostra a construção da assinatura dual [36].

Figura 2.2 – Construção da Assinatura Dual

Assim sendo, o protocolo SET provê tanto a confidencialidade das informações de pagamento, como das informações da compra realizada, transmitidas junto com as informações de pagamento. Isso porque as informações da compra são criptografadas com a chave pública do vendedor, como também as informações de pagamento são criptografadas com a chave pública da instituição financeira do consumidor. Do mesmo modo assegura a integridade de todo dado transmitido e garante a autenticidade do consumidor possuidor do cartão de crédito, como sendo o usuário legítimo que forneceu o número de cartão de crédito para o pagamento.

2.5.1.2 Cheques Eletrônicos

Um cheque em papel é um instrumento de pagamento; sua validade e aceitação exigem a assinatura do correntista. O portador de um cheque legítimo assinado tem a promessa de ressarcimento monetário pelo banco onde o correntista possui a conta corrente.

Os cheques eletrônicos funcionam de forma equivalente. Possuem assinatura e certificado digital para autorizar o pagamento, igual a um cheque em papel. O correntista usa um talão eletrônico de cheques baseado em alguma forma segura de *hardware* ou *software*. Este dispositivo ou programa garante o armazenamento seguro da chave privada e dos certificados. Um comprador pode emitir um cheque eletrônico contendo uma instrução para fazer um pagamento de um montante específico para um destinatário identificado. O cheque é enviado em um envelope seguro, por um correio eletrônico seguro, para o destinatário que encaminha eletronicamente para seu banco. O banco efetua a compensação do cheque e transfere o dinheiro do banco do comprador para a conta do destinatário.

O *NetCheque* [35] é um projeto do *Information Sciences Institute* (ISI) da *University of Southern California*. Funciona de forma semelhante ao cheque em papel. O sistema é baseado no sistema de segurança *Kerberos* [37], que usa um *ticket* especial para criar eletronicamente a assinatura e o endosso do cheque.

Um pagamento começa quando um correntista emite um documento eletrônico que inclui o seu nome e o número da conta corrente, o nome de sua instituição financeira, o nome do destinatário e o montante do cheque. Usando o sistema de segurança *Kerberos*, uma assinatura eletrônica é incluída ao cheque eletrônico que autentica o cheque. À maneira de um cheque em papel, é necessário ser certificado pelo pagador com o auxílio de uma outra assinatura do sistema *Kerberos*, antes que o cheque possa ser pago. Finalmente, o cheque assinado e validado é eletronicamente trocado entre instituições financeiras através do sistema de compensação.

2.5.1.3 Smart Cards

Um *smart card* é um pequeno cartão plástico com um microprocessador integrado (*chip*), que garante o armazenamento e o processamento de informações de forma segura. Existe uma variedade de *chips*; uns com características para armazenar informações financeiras para uso em máquinas automáticas de venda; outros, para armazenar informações e aplicações que exigem segurança para operar, como por exemplo, registros de paciente no atendimento médico. A maior utilização dos *smart cards* é na indústria financeira, a exemplo da Visa, que está substituindo os cartões com trilha magnética por essa tecnologia, buscando segurança e implementação de novas aplicações financeiras.

2.5.2 Dinheiro Eletrônico

Dinheiro digital, dinheiro eletrônico ou moeda eletrônica é, basicamente, número serial criptografado que tem a representatividade monetária como o dinheiro em espécie. Os usuários sacam o dinheiro eletrônico de suas contas administradas por uma instituição financeira. A quantia é armazenada em *smart cards* ou em uma carteira eletrônica mantida no disco rígido do microcomputador de seu proprietário e pode ser gasto como o dinheiro tradicional. Para transações simples de valor monetário pequeno, o dinheiro de circulação eletrônica preserva a privacidade do usuário de forma semelhante ao da utilização do dinheiro em espécie. Como o dinheiro de circulação eletrônica pode ser facilmente copiado, ele deve possuir mecanismos para prevenir sua reapresentação em duplicidade. O dinheiro eletrônico é muito flexível e é utilizado nos protocolos de Macropagamento e Micropagamento.

2.5.2.1 PayWord

Como mencionado em páginas anteriores, o protocolo *PayWord* foi proposto por *Ronald Rivest*, do *MIT LCS* e *Adi Shamir*, do *Weizmann Institute of Science*. O *PayWord* é baseado no esquema de crédito em que participam o consumidor, o vendedor e um agente financeiro (*broker*) no Sistema de Micropagamento para pagamentos de compras pela Internet. Uma descrição detalhada é dada no capítulo 3.

2.5.2.2 MicroMint

O protocolo *MicroMint* foi proposto junto com o *PayWord*. Baseia-se no esquema de débito para o consumidor e um corretor (*broker*) e no esquema de crédito para o vendedor. O *MicroMint* serve-se da dificuldade de cunhar moeda eletrônica pelo *broker*. Uma descrição detalhada é dada no capítulo 4.

2.5.2.3 MilliCent

O *MilliCent* é um protocolo de segurança para comércio eletrônico pela Internet que foi desenvolvido pela Compaq. Usa uma forma de dinheiro de circulação eletrônica chamado *scrip*. O *MilliCent* foi planejado para transações de montante de menos de um centavo de real até no máximo cerca de R\$ 10,00. No capítulo 5 é dada uma descrição detalhada do *MilliCent*.

Capítulo 3

PayWord

O *PayWord* [1] é um esquema de Micropagamento proposto por *Ronald Rivest*, do *MIT LCS* e *Adi Shamir*, do *Weizmann Institute of Science*, baseado no esquema de crédito³ para transações de pequeno valor.

O esquema compõe-se de três partes: o usuário/consumidor (U), que faz compras e efetua o pagamento, um vendedor (V), que vende produtos/informações e coleta os pagamentos, e um *broker* – agente financeiro (B), que administra a conta do consumidor e do vendedor.

Figura 3.1 – Relacionamento entre *Broker*, Usuário e Vendedor

³ O Usuário não necessita pagar antecipadamente pela moeda eletrônica. Sua conta é debitada somente quando o Vendedor for ressarcido do pagamento pelo *Broker*.

O *PayWord* usa criptografia de chave pública, por exemplo RSA, com um expoente público pequeno. As chaves públicas do usuário U , do vendedor V e do *broker* B , são denotadas respectivamente por PK_U , PK_V e PK_B . Suas chaves secretas, conhecidas também como chaves privadas, são denotadas por SK_U , SK_V e SK_B . Uma mensagem M com assinatura digital produzida por uma chave secreta SK é denotada por $\{M\}_{SK}$. A mensagem pode ser verificada usando a correspondente chave pública PK .

Da mesma forma utiliza-se uma função de espalhamento, ou função *hash*, denotada por h e criptograficamente forte, tal como MD5 [2] ou SHA [3]. A propriedade importante dessa função h é o fato de ser de mão única (*one-way function*), ou seja, é rápida de se calcular o resultado da saída, dada uma entrada; contudo, é necessária uma imensa quantidade de pesquisa computacional para encontrar a entrada que produziu uma dada saída, ou encontrar duas entradas para a mesma saída.

3.1 Visão Geral

Inicialmente o usuário U estabelece uma conta com um *broker* B que emite ao usuário um certificado digital *PayWord*. O certificado digital possui uma data de vencimento e deve ser renovado periodicamente pelo *broker* que verifica a validade da conta do usuário. Com este certificado, o usuário U está autorizado a produzir uma série encadeada de *PayWords*, que representa dinheiro de circulação eletrônica e que será utilizado pelo usuário U no pagamento de compras junto ao vendedor V , o qual resgatará o valor monetário das transações com o *broker* B .

Pode-se supor que cada *PayWord* corresponde ao valor monetário de um centavo e será utilizado no pagamento de produtos/informações em uma aplicação típica de loja virtual na *Web*.

O primeiro passo do usuário U junto a um vendedor V é calcular e assinar um compromisso para uma série encadeada específica de *PayWords* $w_1 \dots w_n$. O usuário produz uma série encadeada de *PayWords* na ordem reversa, tomando um número aleatório para produzir o *PayWord* w_n , que é a semente para produzir os demais *PayWords*, como segue:

$$w_i = h(w_{i+1}) \quad \text{para } i = n-1, n-2, \dots, 0.$$

Aqui o w_0 é chamado de raiz e não é utilizado como um *PayWord*. O usuário U envia um compromisso para o vendedor que verifica sua assinatura. O i -ésimo pagamento do usuário U para o vendedor V consiste de um par (w_i, i) que o vendedor V pode verificar pelo w_0 recebido no compromisso.

No final de cada dia ou de um certo período de tempo, o vendedor V relaciona o último pagamento (w_u, u) e o correspondente compromisso de cada usuário U e os envia para o *broker* B . O *broker* B , então, resgata de cada conta do usuário U os u s centavos e efetua o ressarcimento monetário ao vendedor V .

3.2 Detalhes do *PayWord*

3.2.1 Certificação do Usuário pelo *Broker*

O usuário U deve primeiro estabelecer um relacionamento com um *broker* B , abrindo uma conta e obtendo um certificado *PayWord*. Por um canal seguro, o usuário U fornece ao *broker* B suas informações detalhadas, incluindo número de cartão de crédito, sua chave pública PK_U e o endereço de entrega A_U .

O *broker* B , então, envia para usuário U um certificado C_U assinado com sua chave secreta SK_B , com as seguintes informações:

$$C_U = \{B, U, A_U, PK_U, E, I_U\}_{SK_B}$$

B identifica o *broker*, U identifica o usuário, E é a data de vencimento do certificado, I_U é qualquer informação adicional do *broker*.

Figura 3.2 – Certificação do usuário pelo *broker*

O certificado *PayWord* é uma declaração do *broker* B a todo vendedor V que possua conta em B , garantindo o ressarcimento de todos *PayWords* autênticos produzidos pelo usuário U e apresentados pelo vendedor V ao *broker* B , antes da data de vencimento do certificado.

3.2.2 Compras

3.2.2.1 Compromisso de pagamento

A relação entre o usuário U e o vendedor V é transitório. Isso significa que um usuário U pode visitar uma página da loja virtual na *Web* do vendedor V e comprar produtos/informações desse *site* e, em seguida, mover-se para outro *site* na Internet. Quando o usuário U estabelece um contato com um novo vendedor V , ele calcula uma nova série encadeada de *Paywords* $w_1, \dots, w_n$, com raiz w_0 . A série encadeada de *PayWords* pode ser de qualquer tamanho.

O usuário U , por sua vez, constrói seu compromisso para a série encadeada de *PayWords* com as seguintes informações:

$$M_U = \{V, C_U, w_0, D, I_M\}_{SK_U}$$

V identifica o vendedor, C_U é o certificado *PayWord*, w_0 é a raiz da série encadeada de *PayWords*, D é uma data de vencimento de M , e I_M é qualquer informação adicional que pode ser necessária em M_U , tal como o tamanho n de *PayWords*. A mensagem M_U é assinada pelo usuário U e enviada ao vendedor V . Esse compromisso autoriza o *broker* B a pagar ao vendedor V , por qualquer *PayWord* $w_1 \dots w_n$, que o vendedor V apresente ao *broker* B antes da data de vencimento D .

O vendedor V , então, verifica a assinatura do usuário U em M , com a chave pública do usuário PK_U e, a assinatura do *broker* B no certificado C_U . Também verifica sua identificação V e a data de vencimento do compromisso D . A assinatura do usuário U é verificada pelo certificado do usuário C_U . Assim, um oponente não pode falsificar um compromisso de pagamento, assinando com uma chave falsa e nem fingir o usuário U sem conhecer a chave privada PK_U .

Figura 3.3 – Estabelecendo compromisso de pagamento

Se a mensagem M_U for verificada, o vendedor V deve armazená-la até que venha expirar sua data de validade no final do dia. Essa medida previne um possível ataque por substituição (*replay*).

3.2.2.2 Pagamentos

O usuário e o vendedor necessitam estabelecer um acordo quanto ao custo do produto/informações na operação de pagamento. Na pretendida aplicação típica de loja virtual na *Web*, com esquema de Micropagamento, o preço do produto/informações é da ordem de centavos.

Um pagamento P_U de um usuário U para um vendedor V consiste de um *PayWord* e seu índice correspondente, como segue:

$$P_U = (w_i, i) \text{ para } i=1, \dots, n.$$

O pagamento P_U não é assinado pelo usuário U , pois este é facilmente autenticado. Para verificar a validade de um pagamento $P_U = (w_i, i)$, o vendedor V calcula i vezes a função *hash* sobre último *PayWord* w_u recebido do usuário U (i.e., o w_0 da mensagem M_U ou o w_i do último pagamento P_U). O resultado da i -ésima aplicação da função $h(w_u)$ é comparado com w_i e, sendo iguais esses valores, fica assegurada a autenticidade do pagamento P_U .

Com tal processo, o vendedor V necessita apenas armazenar por usuário U o último par (w_i, i) , para validar o próximo pagamento desse usuário U ou, no final do dia, pelo vendedor V para apresentar ao *broker* B o gasto do usuário U e o correspondente ressarcimento monetário.

O usuário gasta o *PayWord* w_1 , depois o w_2 , e assim por diante. Supondo que cada *PayWord* é do valor monetário de um centavo, em cada pagamento realizado pelo usuário, é necessário que ele envie ao vendedor o apropriado *PayWord* correspondente ao pagamento do produto/informações da transação.

Se a compra de um item custa mais que o valor de um *PayWord*, o usuário U pode pagar saltando os *PayWords* para o valor monetário correspondente do item comprado. Por exemplo, na hipótese de que o próximo *PayWord* não gasto é o w_{i+1} , cada *PayWord* é de valor de um centavo e o item custa 5 centavos. Nesse caso, o usuário salta os quatros *PayWords* não gastos (i.e., w_{i+1} , w_{i+2} , w_{i+3} e w_{i+4}), e envia ao vendedor V a mensagem de pagamento $P_U = (w_{i+5}, 5)$.

Figura 3.4 – Fazendo pagamento

3.2.3 Compensação dos Pagamentos pelo *Broker* ao Vendedor

No final do dia ou do período, o vendedor V necessita enviar ao *broker* B as mensagens de compromisso de pagamento M_U de cada usuário U de B , que estabeleceram o compromisso de pagamento, e o último *Payword* $P_U = (w_u, u)$, para que obtenha os ressarcimentos dos valores monetários correspondentes aos us centavos por usuário do *broker* B .

O *broker* B verifica cada mensagem de compromisso M_U , assinada pelo usuário U e assegurada a validade da assinatura pelo certificado do usuário C_U que ele próprio assinou. Da mesma forma são validadas as datas de vencimento e outras informações contidas na mensagem. O *broker* assegura a autenticidade do último *Payword* $P_U = (w_u, u)$, aplicando-se u vezes a função *hash* sobre w_o contida na mensagem M_U . Pode-se supor que o *broker* B honra cada pedido de ressarcimento autêntico.

Figura 3.5 – Compensação dos pagamentos

O vendedor V necessita autenticar os certificados assinados pelo *broker* B . Para tanto, ele precisa receber do *broker* B , por um canal seguro, sua chave pública PK_B .

Não está estabelecido no esquema de Micropagamento *PayWord* a forma que o *broker* B paga o vendedor V . Isto está fora do sistema *PayWord* [1].

3.3 Avaliação de Segurança

3.3.1 Prevenção de Falsificação

Os *PayWords* apresentados nos pagamentos são resultados da função *hash*, de mão única, sobre o valor de um *PayWord* não gasto de uma mesma série encadeada. Portanto, um oponente conhecendo um *PayWord* que foi gasto em um pagamento não pode calcular o valor de um *PayWord* que ainda não foi gasto.

Uma série encadeada de *PayWords* é autenticada por uma mensagem de compromisso e esta mensagem é assinada pelo usuário. A identidade do usuário é assegurada pelo seu certificado que é assinado pelo próprio *broker*.

3.3.2 Detecção de Gasto Duplicado

Um pagamento no protocolo *PayWord* consiste de um compromisso e seu correspondente conjunto de *PayWords*. Dessa forma, o gasto duplicado de *PayWords* significa rerepresentar os mesmos *PayWords* com o mesmo compromisso.

Um vendedor ou um *broker* pode manter a trilha de todos os *PayWords* gastos de um compromisso, mantendo armazenado em uma base de dados apenas o último *PayWord* gasto e o *PayWord* raiz w_0 .

A rerepresentação de um compromisso válido pode ser facilmente encontrada pesquisando-se na base de dados (ver passos 6 e 12 nas Figuras 3.3 e 3.5, respectivamente). Assim sendo, tanto o vendedor como o *broker*, até o vencimento dos compromissos armazenados, previnem a rerepresentação de um compromisso válido, quer por um usuário ao vendedor, quer pelo vendedor ao *broker*.

3.3.3 Outros Comentários

Algumas características do protocolo *PayWord*:

- *O usuário não necessita pagar antecipadamente* – O *PayWord* é um esquema baseado em crédito. Em vez de pagamento antecipado, um usuário *PayWord* gera sua própria série encadeada de *PayWords* (dinheiro eletrônico); e sua conta é debitada somente quando o vendedor for ressarcido do pagamento pelo *broker*.
- *O usuário gera seu próprio PayWord* – Esta característica provê algumas flexibilidades. Por exemplo, um usuário pode gerar os *PayWords* baseado na demanda de consumo. Igualmente, o usuário pode gerar o montante exato de *PayWords* para um pagamento.
- *Uso do último pagamento para autenticação* – O vendedor e o *broker* somente necessitam armazenar o compromisso válido e o par correspondente do último *Payword* $P_U = (w_u, u)$ usado para autenticar os pagamentos.
- *Quebra do anonimato do usuário* – Apesar de não haver a transmissão pela rede de informação que identifique o usuário, o *broker* estabelece no certificado o seu endereço de entrega A_U .

Algumas características quanto à eficiência computacional e necessidade de armazenamento.

- O *broker* assina para cada usuário seu certificado, verifica a assinatura do usuário de cada mensagem de compromisso de pagamento e aplica uma função *hash* por pagamento apresentado para ressarcimento pelo vendedor. Todos estes processamentos podem ser *off-line*. O *broker* armazena cópia dos certificados dos usuários e administra as contas dos usuários e dos vendedores.
- O usuário verifica seu próprio certificado, assina sua mensagem de compromisso de pagamento e aplica uma função *hash* por pagamento compromissado. Apenas a assinatura da mensagem de compromisso de pagamento é realizada *on-line*. O usuário necessita armazenar sua chave secreta de forma segura, manter os registros de todas as mensagens de compromisso e a correspondente série encadeada de *PayWords*, com o índice do próximo *PayWord* ainda não gasto.
- O vendedor verifica todos os certificados e mensagens de compromisso de pagamento. Aplica uma função *hash* por pagamento recebido ou salta os *PayWords* (i.e., aplica i vezes a função *hash*, quando $i > 1$ na mensagem P_U). Todos estes processamentos são realizados *on-line*. O vendedor armazena todas as mensagens de compromisso de pagamento e o último pagamento recebido por compromisso por dia ou por período.

Capítulo 4

MicroMint

O *MicroMint* [1] é um esquema de Micropagamento que foi proposto junto com o *PayWord* por *Ronald Rivest*, do *MIT LCS* e *Adi Shamir*, do *Weizmann Institute of Science*. É baseado no esquema de débito para o consumidor e o *broker* – agente financeiro e no esquema de crédito para o vendedor e o *broker*.

Como ocorre no *PayWord*, o esquema compõe-se de três partes: o *broker*, que produz as moedas eletrônicas e as vende ao usuário/consumidor, que faz compras e dá como pagamento moedas eletrônicas ao vendedor; este vende produtos/informações e coleta como pagamento essas moedas e compensa junto ao *broker*.

Figura 4.1 – Relacionamento entre *Broker*, Usuário e Vendedor

4.1 Visão Geral

Como outros esquemas de Micropagamento, o *MicroMint* é baseado em um *token* que representa a moeda eletrônica.

O *MicroMint* tem a propriedade de gerar muitas moedas eletrônicas com baixo custo. É necessário um grande investimento inicial para produzir a primeira moeda; depois, a geração de moedas adicionais pode ser feita de forma barata. Este processo é similar ao de cunhar moedas, cujo investimento inicial é muito alto em equipamentos e necessita de profissionais peritos para produzi-las de uma maneira econômica.

No cenário típico, um *broker* produz moedas eletrônicas que são válidas por um certo período, como por exemplo, por um mês. Para tal produção, o *broker* precisa de investimento substancial em *hardware*, que dá a ele uma vantagem computacional sobre um possível falsário. Por um mês, o *broker* roda continuamente neste equipamento o processamento para calcular ou produzir as moedas eletrônicas válidas para uso no mês seguinte. No início de cada mês, novas moedas são vendidas aos usuários. Com elas, o usuário efetua compra de produtos/informações, dando-as como pagamento ao vendedor. O vendedor retorna essas moedas ao *broker*, no final do dia ou no período de sua conveniência, para compensar as moedas eletrônicas apresentadas pelo vendedor.

A segurança do *MicroMint* está no processo de cunhar moedas eletrônicas. Esse processo seguro consiste na dificuldade de encontrar colisões em uma função de espalhamento ou *hash*, denotada por h .

Uma moeda eletrônica *MicroMint*, consiste de uma função *hash* de colisões de k -modos, representado pelo conjunto de valores $\{x_1, x_2, \dots, x_k\}$, onde $h(x_1) = h(x_2) = \dots = h(x_k) = y$. Uma função *hash* de k -modos significa que existem k valores diferentes de entrada que mapeiam um mesmo valor produzido pela função. Essa moeda eletrônica possui duas propriedades importantes: difícil de serem produzidas e fáceis de serem validadas.

É possível visualizar uma função *hash* como sendo um processo de lançar aleatoriamente bolas numeradas dentro de uma série de urnas também numeradas. Ocorre uma colisão quando mais que uma bola cai dentro de uma mesma urna. Uma moeda eletrônica *MicroMint* é uma urna com k bolas dentro dela.

4.2 Detalhes do *MicroMint*

4.2.1 Produção de Moedas Eletrônicas

Para melhor compreensão, seguem três estruturas propostas de moeda eletrônica. A primeira versão é uma forma genérica de moeda eletrônica, baseada no protocolo de dinheiro eletrônico, que mantém anônimo o usuário, pois não carrega qualquer informação a seu respeito. A segunda versão é uma forma de moeda eletrônica específica para um usuário com suas informações e somente utilizável por ele. A terceira versão combina a segunda com informações específicas de um vendedor, o que implica uma forma de moeda eletrônica que possui informações específicas de um usuário e de um vendedor.

4.2.1.1 Primeira versão – Moeda Eletrônica Genérica

Uma moeda eletrônica *MicroMint* é um elemento do conjunto de valores $\{x_1, x_2, \dots, x_k\}$, onde cada valor x_i , com m bits e $i=1, \dots, k$, tem um mesmo valor y , com n bits, mapeado por uma função *hash*, denotada por h . Essa característica é chamada de colisões de k -modos.

$$\text{Moeda Eletrônica} = \{x_1, x_2, \dots, x_k\} \quad \text{onde } h(x_i) = y \quad \text{para } i=1, \dots, k.$$

A função h é uma função criptográfica de mão única – *one-way function*, ou seja, é rápida de se calcular o resultado da saída dada uma entrada; contudo, é necessária uma imensa quantidade de pesquisa computacional para encontrar a entrada que produziu uma dada saída ou encontrar duas entradas para a mesma saída.

O tamanho em bits de um valor x_i , denotado por m , deve ser suficientemente grande para armazenar todos os possíveis valores x_i gerados. Esta é uma característica importante para evitar a falsificação dessas moedas.

Também para prevenir falsificação, o *broker*, na produção das moedas eletrônicas, atribui um critério para cada moeda em circulação para um determinado período de tempo; por exemplo, um critério mensal. Este critério é estabelecido como sendo um conjunto de bits $\{b_n, \dots, b_{n-t+1}\}$ fixados nos primeiros t bits do resultado da função *hash*. Desse modo, uma moeda válida para um determinado mês tem o seguinte formato:

$$h(x_i) = y = \{b_n, \dots, b_{n-t+1}, b_{n-t}, \dots, b_1\} \quad \begin{array}{l} \text{para } i=1, \dots, k, \\ \text{onde } \{b_n, \dots, b_{n-t+1}\} = \text{critério do mês.} \end{array}$$

4.2.1.2 Segunda versão – Moeda Eletrônica Específica por Usuário

Nessa versão, as moedas eletrônicas contêm informações específicas do usuário, de forma que tais moedas são apenas úteis para um mesmo usuário, que são facilmente validadas pelo vendedor. Caso sejam roubadas, não terão valor para outros, reduzindo enormemente a motivação de um oponente de roubar moedas.

Assim sendo, uma moeda eletrônica é um elemento do conjunto dos k valores, considerando as seguintes regras:

$$\begin{aligned} \text{Moeda Eletrônica} &= \{x_1, x_2, \dots, x_k\} \\ \text{onde } h(x_i) &= y_i \quad \text{para } i=1, \dots, k, \\ y_{j+1} - y_j &= d_j \pmod{2^u} \quad \text{para } j=1, \dots, k-1 \text{ e } u = n - t, \text{ e} \\ h(U) &= (d_1, d_2, \dots, d_{k-1}) \end{aligned}$$

U é a informação única que identifica um usuário. Caso o critério mensal seja estabelecido, todos os valores y_i variam apenas nos últimos u bits.

4.2.1.3 Terceira versão – Moeda Eletrônica Específica por Usuário e Vendedor

As moedas eletrônicas podem ser produzidas especificamente para um usuário e vendedor em particular. Com tal medida, um usuário ou um oponente é desencorajado a gastar a mesma moeda eletrônica com outros vendedores.

De igual maneira nesta versão, o resultado y produzido pela função h é de tamanho de n bits. Os primeiros t bits de y estabelecem o critério mensal fixado pelo *broker*. O restante dos bits, $u = n - t$, é dividido em duas partes: (1) y' os bits de mais alta ordem $(u - v)$ bits e (2) y'' os bits v de mais baixa ordem.

Figura 4.2 – Estrutura de uma moeda eletrônica

A Figura 4.2 mostra graficamente a estrutura de uma moeda eletrônica específica para um usuário e vendedor.

Tal moeda é um elemento do conjunto dos k valores, considerando as seguintes regras:

$$\begin{aligned}
 & \text{Moeda Eletrônica} = \{x_1, x_2, \dots, x_k\} \\
 & \text{onde} \\
 & h(x_i) = y_i = \{b_n, \dots, b_{n-t+1}, b_{n-t}, \dots, b_1\}_i \quad \begin{array}{l} \text{para } i=1, \dots, k, \\ \text{e } \{b_n, \dots, b_{n-t+1}\}_i = \text{critério do mês} \end{array} \\
 & e \\
 & y'_{j+1} - y'_j = d'_j \pmod{2^{u-v}} \quad \text{para } j=1, \dots, k-1 \text{ e } u = n - t \\
 & \text{onde} \\
 & h'(U) = (d'_1, d'_2, \dots, d'_{k-1}) \\
 & e \\
 & y''_{j+1} - y''_j = d''_j \pmod{2^v} \quad \text{para } j=1, \dots, k-1 \\
 & \text{onde} \\
 & h''(V) = (d''_1, d''_2, \dots, d''_{k-1}) \\
 & e \\
 & y''_1 \text{ valor único.}
 \end{aligned}$$

U e V são respectivamente as informações únicas que identificam um usuário e um vendedor. y''_1 é um valor único que identifica uma moeda eletrônica.

Observa-se que o vendedor somente poderá compensar uma moeda eletrônica se satisfizer o critério mensal estabelecido pelo *broker*, as identificações do usuário e do vendedor e também a identificação única da moeda eletrônica em y''_1 .

4.2.2 Compra de Moedas Eletrônicas Específicas por Usuário

Um *broker* pode organizar todos os valores x_i em grupos. Em cada grupo, os valores x_i têm os mesmos valores nos bits $t + (u - v)$. Quando um usuário compra moedas eletrônicas de um *broker*, este vende os grupos de moedas que satisfazem aquele usuário, de forma que a seguinte regra é válida:

$$\begin{aligned}
 & y'_{j+1} - y'_j = d'_j \pmod{2^{u-v}} \quad \text{para } j=1, \dots, k-1 \text{ e } u = n - t \\
 & \text{onde} \\
 & h'(U) = (d'_1, d'_2, \dots, d'_{k-1}).
 \end{aligned}$$

4.2.3 Compras

Quando um usuário efetua uma compra de produtos/informações de um vendedor, este toma um valor x de um grupo de valores x_i que foi comprado de um *broker* e monta a moeda eletrônica, considerando a seguinte regra:

$$y''_{j+1} - y''_j = d''_j \pmod{2^n} \quad \text{para } j=1, \dots, k-1$$

onde

$$h''(V) = (d''_1, d''_2, \dots, d''_{k-1})$$

e

y''_1 valor único.

O vendedor pode validar a moeda eletrônica aplicando-se a função *hash* e prevenindo-se da reapresentação de uma moeda, verificando o valor único y''_1 mantido, por exemplo, em uma base de dados.

Figura 4.3 – Compras

4.2.4 Compensação dos Pagamentos pelo *Broker* ao Vendedor

O vendedor retorna as moedas eletrônicas que foram coletadas nos pagamentos efetuados pelos usuários, as quais são compensadas junto ao *broker* no final de cada dia. Se a moeda eletrônica não foi previamente apresentada, o *broker* paga um centavo ao vendedor, menos uma taxa pela operação. Como o *broker* mantém um registro das moedas produzidas, vendidas e retornadas, no caso de uma moeda eletrônica ser apresentada mais que uma vez,

o *broker* não compensa o valor correspondente ao vendedor na reapresentação da mesma moeda.

Figura 4.4 – Compensação dos pagamentos

4.3 Avaliação de Segurança

A segurança do esquema *MicroMint* contra o processo de falsificação deriva do custo da grande base computacional, necessária para a produção de moedas eletrônicas. Dessa forma, tal mecanismo de segurança foi desenhado para desencorajar um ataque em grande escala, bem como uma falsificação em massa de moedas eletrônicas, ou uma persistente reapresentação de tais moedas.

4.3.1 Prevenção de Falsificação

A falsificação em grande escala de moedas eletrônicas é muito cara para despertar o interesse de um oponente. É de se supor que o esquema *MicroMint* seja parametrizado, com $k = 4$, $n = 54$ e $u = 31$. A geração das primeiras moedas eletrônicas falsas requer um processamento de 2^{54} operações de *hash*. Considerando-se que uma estação de trabalho padrão possa processar 2^{14} operações de *hash* por segundo, isso significa aproximadamente 80 anos para a produção de uma única moeda eletrônica no equipamento.

4.3.2 Detecção de Gasto Duplicado

Como o esquema de moeda eletrônica *MicroMint* não é anônima, o *broker* pode manter trilhas das moedas eletrônicas que foram vendidas aos usuários. Pode saber também qual dessas moedas foram utilizadas em pagamentos de produtos/informações pelo usuário junto aos vendedores, quando o vendedor apresentá-las na compensação (ver passo 2 nas Figuras 4.3 e 4.4). O valor único y''_i mantido, por exemplo, em uma base de dados pelo *broker* e pelo vendedor, previne a reapresentação de uma mesma moeda.

4.3.3 Outros Comentários

Algumas características do protocolo *MicroMint*.

- *Nenhuma chave secreta é utilizada, como também não é necessário o esquema de criptografia de chave pública* – Todas as informações nas moedas eletrônicas são públicas. Desse modo, não existe o problema do compartilhamento de chaves secretas entre as partes envolvidas, ou usar esquema de criptografia de chave pública, para garantir confidencialidade, autenticidade ou integridade das informações no protocolo *MicroMint*.
- *O usuário pode verificar a validade de suas moedas eletrônicas* – A verificação é feita aplicando-se sobre os valores x_i a função *hash* e obtendo um mesmo valor y .

Capítulo 5

MilliCent

O *MilliCent* [4] é um protocolo de segurança desenvolvido pela Compaq para o comércio eletrônico através da Internet. Usa uma conta eletrônica que contém dinheiro eletrônico que é chamado *scrip*. O *MilliCent* é um esquema de Micropagamento para transações de montante de menos de um centavo até cerca de R\$ 10,00.

5.1 Visão Geral

Uma transação *MilliCent* envolve as seguintes partes: *broker*, usuário/consumidor e vendedores. O ponto importante do *MilliCent* é o *scrip*, que representa uma conta eletrônica, onde é mantido o dinheiro eletrônico, produzido e atualizado por um *broker* ou vendedor, chamado aqui, respectivamente, por *scripBroker* e *scripVendor*.

Tanto o usuário, quanto o vendedor estabelecem suas respectivas contas em um *broker*, onde mantêm um relacionamento de longo prazo, como por exemplo, em uma empresa de cartão de crédito ou banco. Isso é necessário para que o *broker* possa adquirir *scripVendor* dos vendedores e possa revendê-los, como *scripBroker*, para os usuários. O ressarcimento do valor monetário do *scripVendor* pelo *broker* junto ao vendedor, ou do *scripBroker* pelo usuário junto ao *broker*, é feito por um método de pagamento tradicional como, por exemplo, pelo cartão de crédito, fora do esquema do *MilliCent*.

Em uma aplicação típica no esquema de Micropagamento, um vendedor oferece aos usuários produtos/informações em uma loja virtual na Internet. O vendedor produz seu *scripVendor* e os vende para seu *broker* que torna a revendê-los aos usuários na forma de *scripBroker*. Quando um usuário está interessado na compra de algum produto ou informações, ele deve ter um *scripVendor* daquele vendedor, pois este aceita como dinheiro eletrônico apenas seu próprio *scrip*. O usuário, que já adquiriu da forma tradicional *scripBroker* do seu *broker*, pode comprar *scripVendor*, usando *scripBroker* como dinheiro eletrônico. Um usuário coleciona *scripVendor* de diferentes vendedores para facilitar os pagamentos.

Os *scrips*, sejam eles *scripBroker* ou *scripVendor*, adquiridos por um usuário são armazenados em uma carteira eletrônica, responsável por enviar o respectivo *scrip*, como pagamento para um vendedor ou para *broker* na ocasião da compra de *scripVendor*.

Quando a carteira eletrônica de um usuário recebe um pedido de compra, ela verifica se possui o montante apropriado no *scrip* daquele vendedor para pagar o requisitado item. Se não possuir, o *broker* do usuário é contatado e o *scripVendor* do vendedor em questão é comprado.

Em seguida, o *scripVendor* é enviado pela carteira eletrônica ao vendedor para pagar a compra como dinheiro eletrônico. Quando o vendedor recebe o *scripVendor*, certifica-se se não é falsificado ou forjado e se não é uma rerepresentação de um anterior. Validado o *scrip* recebido, é deduzido dele o valor da compra, e o novo *scrip* atualizado com o saldo restante é enviado de volta para a carteira eletrônica do usuário.

A carteira eletrônica mantém um histórico das transações realizadas, retratando as operações de pagamentos e respectivos saldos dos *scrips*.

Figura 5.1 – Relacionamento entre Broker, Usuário e Vendedor

A Figura 5.1 mostra o relacionamento entre as partes envolvidas no MilliCent, com o seguinte fluxo:

- 1) um *broker* compra uma quantidade grande de *scripVendor* de diferentes vendedores e produz *scripBroker*;

- 2) o usuário comprando *scripBroker* do *broker*, que possui um longo relacionamento, por um método de pagamento, como por exemplo, cartão de crédito. O *scripBroker* serve como uma moeda eletrônica para pagar *scripVendor* de um *broker*. Quando o usuário necessita de *scripVendor* de um vendedor específico, ele usa *scripBroker* para pagar;
- 3) quando um usuário realiza uma compra, o custo da compra é deduzido do valor do *scrip*; um novo *scrip*, com o novo valor de saldo, é retornado atualizado. Um usuário, ao completar uma série de transações, pode resgatar em dinheiro o restante de valor do *scrip*, o que significa fechar a conta.

5.2 Detalhes do *MilliCent*

5.2.1 Estrutura da Moeda Eletrônica *Scrip*

Por segurança, os pagamentos *MilliCent* possuem três segredos, que envolvem a produção, o gasto e a validação do *scrip*. O usuário envia um segredo, chamado *Customer Secret*, para provar que é proprietário de um *scrip*. O vendedor usa um outro segredo, chamado *Master Customer Secret*, para derivar o *Customer Secret* de informações do usuário no *scrip*. O terceiro segredo, chamado *Master Scrip Secret*, é usado pelo vendedor para prevenção de falsificação. O esquema *MilliCent* usa esses segredos de uma forma a mostrar que conhece um segredo sem que seja necessário revelá-lo.

Um *scrip* representa uma conta eletrônica que um usuário tem que estabelecer com um vendedor ou *broker*. O *scrip* inclui a identificação do vendedor, o valor do *scrip*, um certificado que prova sua validade e outras informações usadas para garantir a segurança na transmissão do *scrip*. Um *scrip* é somente válido para um apropriado vendedor ou *broker*. Ele pode ser gasto apenas uma vez e é resistente a falsificação. Só pode também ser gasto pelo proprietário e pode ser eficientemente produzido e validado. Um *scrip* consiste de um corpo e um certificado.

Figura 5.2 – Estrutura da Moeda Eletrônica *Scrip*

A Figura 5.2 mostra a estrutura do *scrip*, que inclui os seguintes campos:

Vendedor – identifica o vendedor para o *scrip*;

Valor – define o valor monetário do *scrip*;

ID# – identificador único do *scrip*, que é formado pelos campos:

ID_Serial# – usado para mapear um *Master Scrip Secret* na geração do certificado,

ID_Seqüencial# – um número seqüencial;

ID_Cliente# – identifica o usuário/consumidor, que é formado pelos campos:

ID_Cliente_Serial# – usado para selecionar o *Master Customer Secret*, usado para produzir o *customer secret*,

ID_Cliente_Seqüencial# – um número seqüencial;

Vencimento – define a data de vencimento do *scrip*;

Propriedades – informações adicionais do usuário para vendedor, tal como idade, estado da residência do usuário;

Certificado - consiste da assinatura do *scrip* usado para verificar se ele é válido e não foi falsificado, conforme seção a seguir.

5.2.2 Master Scrip Secret e Certificação do Scrip

O *Master Scrip Secret* é usado para gerar o certificado do *scrip* para assegurar sua integridade e prevenir falsificação.

Figura 5.3 – Master Scrip Secret e Certificação do Scrip

O vendedor e o *broker* geram e armazenam uma série única de *Master Scrip Secret* para uso futuro. A estrutura de um *Master Scrip Secret* não é descrita em [4]. Pode-se, todavia, supor que seu tamanho seja de 128 bits no mínimo.

Quando um novo *scrip* é gerado por um vendedor ou por um *broker*, é igualmente produzido um certificado para aquele *scrip*, concatenando-se o respectivo *Master Scrip Secret* ao corpo do *scrip* e, em seguida, aplicando-se uma função de espalhamento, ou função *hash* de mão única (*one-way*) e criptograficamente forte, tal como MD5 [2] ou SHA [3]. O *Master Scrip Secret* é selecionado com base no *ID_Serial#* parte do *ID#* do *scrip*. O *ID_Serial#* permite que seja estabelecido uma faixa de *ID#s* de *scrips* para um determinado vendedor, de forma que o *broker* e aquele vendedor possam calcular o certificado baseado no mesmo *Master Scrip Secret* daquele serial.

5.2.3 Validação do Scrip

A validação de um *scrip* ocorre quando um usuário paga um item junto ao vendedor com um *scripVendor*. O vendedor valida o *scrip* com os seguintes passos:

- 1) um novo certificado é calculado pelo vendedor e comparado com o certificado que foi enviado com o *scrip*, conforme mostra a Figura 5.4. O *ID_Serial#* parte do *ID#* do *scrip* é que determina o *Master Scrip Secret*. O corpo do *scrip* é concatenado com este segredo e é aplicada a função *hash* para produzir o certificado recalculado. Se o *scrip* não foi falsificado, o certificado original e o recalculado serão idênticos;
- 2) a cada *scrip* recebido pelo vendedor, ele verifica sua data de vencimento e consulta uma base de dados, usando o *ID#* do *scrip*, para assegurar que não está havendo reapresentação deste *scrip*.

Figura 5.4 – Validação do certificado do *scrip*

5.2.4 Customer Secret e Master Customer Secret do Scrip

O *Customer Secret* prova quem é o proprietário de um *scrip* e pode ser usado para sua transmissão segura. O vendedor e *broker* geram e armazenam uma série única de *Master Customer Secrets* para uso futuro. A estrutura de uma *Master Customer Secret* também não é descrita em [4], contudo, pode-se supor que seu tamanho seja de 128 bits no mínimo.

Quando um usuário compra um novo *scripVendor* de um *broker*, este gera um identificador de usuário (*ID_Cliente#*) e atribui ao *scripVendor*. Dessa forma, é impossível que um vendedor venha trilhar as compras de um usuário, pois cada novo *scripVendor* gerado, é produzido novo identificador de usuário. O *broker* envia o novo *scrip* e o respectivo *Customer Secret* para o usuário.

O *Customer Secret* é produzido pela concatenação do respectivo *Master Customer Secret* ao identificador de usuário (*ID_Cliente#*) do *scrip*, aplicando-se em seguida uma função de espalhamento ou função *hash* de mão única (*one-way*) e criptograficamente forte, tal como MD5 [2] ou SHA [3]. O *Master Customer Secret* é selecionado com base no *ID_Cliente_Serial#*, parte do *ID_Cliente#*. O *ID_Cliente_Serial#* permite que seja estabelecida uma faixa de *ID#s* de *scrips* para um determinado vendedor, de modo que o *broker* e aquele vendedor possam calcular o *Customer Secret* baseado no mesmo *Master Customer Secret* daquele serial.

Figura 5.5 – Customer Secret e Master Customer Secret

5.2.5 Protocolos do *MilliCent*

O *MilliCent* se baseia em três protocolos para abranger diferentes níveis de segurança. O primeiro protocolo, chamado *scrip in the clear*, é o mais simples e eficiente e serve de base para os outros dois. Contudo, não é útil na prática, porque é muito inseguro como se pode avaliar abaixo. O segundo, de nome *private and secure*, é seguro e oferece uma boa privacidade, mas é bastante caro pelo fato de consumir muito tempo computacional se comparado com os outros dois protocolos. O terceiro protocolo, chamado *secure without encryption*, também é seguro, eficiente e não usa criptografia no canal de comunicação.

5.2.5.1 *Scrip in the clear*

Neste protocolo mais simples, o usuário/consumidor envia pelo meio de transmissão o *scripVendor* em claro (*i.e.*, não criptografado) para o vendedor como forma de pagamento. O vendedor retorna o *scripVendor* atualizado, deduzindo o valor do pagamento da compra do valor do *scrip*. Restando valor no *scrip*, o usuário pode apresentar o *scripVendor* atualizado para pagamento da próxima compra.

Este esquema quase não oferece segurança porque tem condições de um oponente interceptar um *scripVendor* atualizado, quando da sua transmissão do vendedor para o usuário. Com efeito, o oponente pode utilizar de imediato este *scripVendor* interceptado no pagamento de sua compra. Mais tarde, portanto, quando o usuário for tentar apresentar o *scripVendor* interceptado no pagamento de sua compra, o vendedor irá recusar o pagamento, uma vez que o *scripVendor* já tinha sido gasto pelo oponente.

5.2.5.2 *Private and secure*

Este protocolo implementa segurança e privacidade na transmissão das transações, utilizando um segredo compartilhado entre o usuário e o vendedor. O segredo é usado para estabelecer um canal de comunicação seguro, com um método eficiente de criptografia simétrica, tal como, IDEA [5] ou AES [6]. O próprio *scripVendor* pode ser usado para estabelecer a chave secreta compartilhada entre as duas partes. Quando um usuário efetua a compra de um *scripVendor* do *broker*, é gerado um segredo (*Customer Secure*) baseado na identificação do usuário estabelecida pelo *broker* na geração de cada novo *scripVendor* (ver seção 5.2.4). O usuário armazena o *scripVendor* e o correspondente *Customer Secure* do *scrip*.

A compra do *scripVendor* pelo usuário junto ao *broker* deve ser efetuada através de um protocolo seguro fora do *MilliCent*, ou o *scripVendor* deve ser comprado usando uma transação segura do *MilliCent*.

Em uma transação de pagamento de um produto/informação por um usuário junto a um vendedor, esse usuário transmite ao vendedor seu *scripVendor* criptografado, por um método simétrico de criptografia usando como chave secreta o *Customer Secret* daquele *scripVendor*;

da mesma forma é transmitido, porém em claro, o campo de identificação do vendedor (*Vendedor*) e do usuário (*ID_Cliente#*).

Quando o vendedor recebe o pedido de pagamento, este deriva o *Customer Secret* com base na identificação do usuário usando o *ID_Cliente_Serial#* para selecionar o *Master Customer Secret* (ver seção 5.2.4). Com o *Customer Secret* recalculado, passa a ser usado como chave secreta para decriptografar o *scripVendor* recebido do usuário.

5.2.5.3 Secure without Encryption

No protocolo *private and secure* é descrito como compartilhar o segredo entre o usuário e o vendedor, de forma a implementar um canal seguro na transmissão das transações de pagamento do *MilliCent*. Contudo, o custo computacional de manter um canal criptografado pode inviabilizar algumas aplicações de Micropagamento. Para tanto, este protocolo *secure without encryption* descreve como dar privacidade em um pedido de pagamento e em sua resposta sem que o canal seja criptografado.

Neste protocolo, como no anterior, o usuário compra um *scripVendor* do *broker* e recebe o *Customer Secret*. Para o usuário fazer uma compra, ele envia ao vendedor o pedido, o *scripVendor* e a assinatura do pedido. A assinatura é produzida da mesma forma que o certificado do *scrip* (ver seção 5.2.2). O *scrip* e o pedido são concatenados com o *Customer Secret* daquele *scripVendor*. Em seguida o usuário aplica uma função de espalhamento ou função *hash* de mão única (*one-way*) e criptograficamente forte, tal como MD5 [2] ou SHA [3].

Quando o vendedor recebe o pedido, ele deriva o *Customer Secret* do *scrip* e recalcula a assinatura do pedido. Se a assinatura recebida for idêntica à assinatura recalculada, o pedido é válido; ou seja, o pedido ou *scrip* não foi falsificado ou forjado.

Figura 5.6 – Secure without encryption

Em seguida, o vendedor deduz o valor da compra do valor do *scrip* recebido e devolve ao usuário um novo *scrip* atualizado. O *scrip* devolvido possui o mesmo identificador de usuário

do *scrip* originalmente recebido; assim, o usuário pode usar o mesmo *Customer Secret* para calcular a próxima assinatura de pedido da compra seguinte.

Neste protocolo não é necessário que qualquer resposta seja criptografada. Um oponente não pode roubar o *scrip*, uma vez que a assinatura do pedido não pode ser calculada sem o conhecimento do *Customer Secret*. O vendedor precisa assinar também as respostas com o *Customer Secret* para prover autenticidade para o usuário.

5.3 Interações entre *Broker*, Vendedor e Consumidor

5.3.1 Compra de *scripVendor* pelo *Broker*

Os vendedores e os *brokers* possuem, por hipótese, um longo relacionamento nos negócios. Existem diversos modelos de negócio pelos quais o *broker* pode comprar *scrips* dos vendedores, tais como pagamento antecipado, produção licenciada ou venda em consignação. Em todos estes exemplos, o *broker* pode obter receita na venda dos *scrips*, pelo fato de vender aos usuários/consumidores partes do valor dos *scrips* por preço maior do que pagou ao vendedor na compra em grande volume.

5.3.1.1 *Scrip Warehouse*

Neste modelo de interação do *broker* com vendedor, o vendedor produz *scripVendors* antecipadamente ou sob pedido. Para validação futura do *scripVendor*, o vendedor armazena os *ID#s* únicos de todos os *scripVendors*. O *broker* funciona como um armazém de *scrips* (*Scrip Warehouse*), que compra *scripVendor* em grande volume de um vendedor, armazena os *scripVendors* e vende em pequenas quantidades aos usuários por um preço maior.

Figura 5.7 – Armazém de *scrips*

5.3.1.2 *Scrip Licenciado*

Se um usuário de um *broker* comprar muitos *scrips* de um vendedor específico, é desejável que aquele vendedor venha a licenciar a produção de seus *scrips* para o *broker*. Isso quer dizer que o *broker* produz *scrips* os quais o vendedor venha a aceitar e possa validar. O vendedor vende o direito da geração de *scrips* para o *broker*. Com essa finalidade o *broker* obtém do vendedor um *Master Scrip Secret*, uma série de *ID#s* de *scrips*, um *Master Customer Secret* e uma série de identificadores de usuários. São informações necessárias para a produção de *scrips*. O vendedor pode validar o *scrip* licenciado, uma vez que o *Master Scrip Secret* pode ser derivado da série de *ID#s* de *scrips* e o *Master Customer Secret* ser derivado da série dos identificadores dos usuários (ver Seções 5.2.2 e 5.2.4).

Figura 5.8 – Produção de *Scrips* Licenciados

5.3.1.3 *Múltiplos Brokers*

Nesse ambiente com múltiplos *brokers*, é possível que o usuário de um *broker* queira fazer uma compra de um vendedor associado a um outro *broker*. Se cada vendedor tem conta com um único *broker*, o usuário precisa possuir *scripBroker* de diferentes *brokers*. Para simplificar a situação do usuário, o seu *broker* pode comprar *scripBroker* do *broker* do vendedor do qual deseja fazer compras e revendê-lo ao seu usuário. Em seguida, o usuário pode comprar *scripVendor* do *broker* do vendedor interessado, para tanto pagando com *scripBroker* comprado do seu *broker*.

Figura 5.9 – Múltiplos *Brokers*

5.3.2 Compra de *scripBroker* pelo Usuário

Inicialmente o usuário não possui nenhum *scrip* de um *broker* ou de um vendedor. Ele estabelece, então, uma comunicação com um *broker* com quem possui relacionamento e compra *scripBroker* usando um método de pagamento fora do *MilliCent* como, por exemplo, cartão de crédito. O *broker*, recebendo seu pagamento, envia o *scripBroker* cujo valor monetário é o valor contido no próprio *scrip*. Em seguida, o usuário pode comprar *scripVendor* dos vendedores de seu interesse associados ao seu *broker*, usando como forma de pagamento o *scripBroker* do seu *broker* (ver Figuras 5.1 e 5.8).

5.3.3 Compra de *scripVendor* pelo Usuário

Considerando-se que o usuário possui *scripBroker* mas não possui *scripVendor*, e deseja fazer um pagamento de algum produto/informação de um vendedor, ele necessita de *scripVendor* daquele vendedor. Nesse caso, o usuário contata o *broker* do vendedor, compra *scripVendor* e paga com seu *scripBroker* adquirido anteriormente (ver Figuras 5.1 e 5.8).

5.3.4 Pagamento com *scripVendor* pelo Usuário ao Vendedor

Se um usuário deseja comprar algo de um vendedor e possui *scripVendor* desse vendedor, ele pode enviar o *scripVendor* como forma de pagamento ao vendedor. Este, por sua vez,

verifica e valida o *scrip* (ver seção 5.2.3). Em seguida, o vendedor deduz o valor da compra do valor do *scrip* e retorna ao usuário novo *scripVendor* atualizado (ver Figuras 5.1 e 5.8).

5.4 Avaliação de Segurança

A segurança do esquema *MilliCent* decorre da proteção de todas as transações que abrangem as partes (*broker*, vendedor e usuário). Para tanto, usam-se os três segredos que envolvem a produção, o gasto e a validação do *scrip* (ver Seções 5.2.2, 5.2.3 e 5.2.4). O usuário envia um segredo, o *Customer Secret*, para provar que é proprietário de um *scrip*. O vendedor usa um outro segredo, o *Master Customer Secret*, para derivar o *Customer Secret* de informações do usuário no *scrip*. O terceiro segredo, o *Master Scrip Secret*, é usado pelo vendedor para prevenção de falsificação.

5.4.1 Prevenção de Falsificação

Um *scrip*, conforme descrito na seção 5.2, é composto do corpo do *scrip* e seu certificado. O certificado é produzido pela concatenação do respectivo *Master Scrip Secret* com o corpo do *scrip* e, em seguida, aplica-se uma função *hash*, que assegura a integridade e previne a falsificação do *scrip*.

5.4.2 Detecção de Gasto Duplicado

Conforme descrito na seção 5.2.3, quando um usuário apresenta um *scrip* como forma de pagamento ao vendedor, este valida o *scrip* pesquisando pelo seu *ID#* em uma base de dados, para assegurar que não está havendo reapresentação deste *scrip*.

5.4.3 Outros Comentários

Algumas características do protocolo *MilliCent*:

- *Não existe compensação dos pagamentos* – O protocolo não exige que os pagamentos realizados pelos usuários junto aos vendedores sejam ressarcidos monetariamente pelo *broker*. Um usuário sempre realiza o pagamento de produtos/informações com *scripVendor* do próprio vendedor; portanto, o vendedor já obteve seu benefício ou lucro quando vendeu os *scrips* ao *broker* (ver seção 5.3.1).
- *Não é necessário o esquema de criptografia de chave pública* – O protocolo não aplica o esquema de criptografia de chave pública para garantir a autenticação das partes envolvidas, confidencialidade e integridade nas mensagens transmitidas, anonimato dos usuários e suas respectivas compras. Para tanto, a segurança do esquema *MilliCent* faz uso de criptografia relativamente barata, que é a aplicação de uma função *hash* e do uso de três segredos, no processo de produção, gasto e validação do *scrip* (ver seção 5.2). Contudo, o uso de três segredos pode ser caro na manutenção do *Master Scrip Secret* e do *Master Customer Secret* entre o *broker* e o vendedor (ver Seções 5.2.2 e 5.2.4).

Capítulo 6

Comparação dos Protocolos

Como registrado no início deste estudo, o presente capítulo mostra uma análise de comparação entre os pontos similares e os pontos divergentes do *PayWord* com *MicroMint* e do *MilliCent* com *MicroMint*. Em seguida, é feita ainda uma comparação das propriedades desejáveis de um Sistema de Pagamento Eletrônico, bem como, considerações específicas a respeito da escolha do protocolo *PayWord* para a Implementação do Sistema Protótipo de Micropagamento.

6.1 *PayWord* versus *MicroMint*

Alguns pontos similares do *PayWord* com *MicroMint*:

- *Operação Off-line.* O usuário necessita contatar seu *broker* no início de cada mês ou período para obter novo certificado digital *PayWord*. Esse certificado possui uma data de vencimento e deve ser renovado pelo *broker* que verifica a validade da conta do usuário. Com esse certificado, o usuário está autorizado a produzir uma série encadeada de *PayWords*, que representa dinheiro de circulação eletrônica. No *MicroMint*, um *broker* produz moedas eletrônicas válidas por um certo período, por exemplo, um mês. O *broker* roda continuamente no seu equipamento o processamento para calcular ou produzir as moedas eletrônicas válidas para uso no próximo mês. No início de cada mês, novas moedas são vendidas aos usuários.
- *As moedas eletrônicas são específicas por usuário e por vendedor.* Uma série encadeada de *PayWord* é autenticada pelo compromisso de pagamento, onde contém w_0 , que é o certificado que identifica o usuário junto ao vendedor (ver seção 3.2). Do mesmo modo, a moeda eletrônica *MicroMint* é produzida especificamente para um usuário e para um vendedor em particular (ver Figura 4.2).

Alguns pontos divergentes entre o *PayWord* e *MicroMint*:

- *PayWord* utiliza o esquema de criptografia de chave pública. Apesar desse esquema ser considerado caro, isso não constitui obstáculo uma vez que um vendedor necessita verificar a assinatura de cada usuário apenas uma vez ao dia, para que o mesmo possa realizar os pagamentos de suas compras (ver seção 3.2.2.1). Dessa forma, o seu custo é amortizado quando da realização de várias compras. Contudo, se um usuário tem o padrão de gasto de uma ou duas compras em um mesmo vendedor por dia, o custo computacional pode ser alto por moeda eletrônica utilizada.
- Os usuários *PayWord* geram sua própria moeda eletrônica. No caso do *MicroMint*, o broker passa o mês produzindo as moedas eletrônicas que serão vendidas aos usuários no mês seguinte.
- *PayWord* é baseado no esquema de crédito. O broker do *MicroMint*, envia mensalmente ao usuário um número específico de moedas eletrônicas e debita da sua conta o valor monetário referente às moedas compradas. No caso do *PayWord*, é debitado somente depois que o vendedor obtiver o ressarcimento monetário da série encadeada de *PayWords* que consumiu.
- Espaço de armazenamento exigido no broker. O broker *PayWord* necessita de espaço de armazenamento para manter os certificados dos usuários, a série encadeada de *PayWords* e demais informações da conta do usuário. O broker *MicroMint*, por seu lado, necessita de um enorme espaço de armazenamento para a produção mensal da moeda eletrônica *MicroMint*.
- Espaço de armazenamento exigido no vendedor. O vendedor *PayWord* precisa de espaço de armazenamento para manter o último *PayWord* gasto por usuário e o compromisso estabelecido no início do dia. O vendedor *MicroMint* necessita de espaço de armazenamento para manter cada moeda eletrônica *MicroMint* recebida como pagamento dos usuários.

6.2 *MilliCent* versus *MicroMint*

Alguns pontos similares do *MilliCent* com *MicroMint*:

- Não utilizam o esquema de criptografia de chave pública. O *scrip*, moeda eletrônica do *MilliCent*, é autenticado usando-se uma função *hash* com uma chave secreta; as mensagens são autenticadas aplicando-se uma função *hash* com uma chave secreta compartilhada entre as partes. A moeda eletrônica *MicroMint* consiste de uma função *hash* de colisões de k -modos, representado pelo conjunto de valores $\{x_1, x_2, \dots, x_k\}$, onde $h(x_1) = h(x_2) = \dots = h(x_k) = y$.

- *A moeda eletrônica é específica por usuário e vendedor.* Quando um usuário compra um novo *scripVendor* de um *broker*, ele gera um identificador de usuário (*ID_Cliente#*) e atribui ao *scripVendor*. Da mesma forma, a moeda eletrônica *MicroMint* é produzida especificamente para um usuário e vendedor em particular (ver Figura 4.2).

Alguns pontos em que o *MilliCent* difere do *MicroMint*:

- *Não é o broker o proprietário do scrip e, sim, o vendedor.* A descentralização da geração ou produção do *scrip* no *MilliCent* resulta em uma redução de tempo quando comparado com o processo equivalente do *MicroMint*.
- *É necessário que o broker seja envolvido a cada pagamento junto a um novo vendedor.* Caso um usuário não possua o apropriado *scripVendor* do vendedor, precisa haver a interação prevista na seção 5.3.1.3, para atender a múltiplos *brokers*. No *MicroMint*, somente uma transação é necessária entre o usuário e o *broker* durante a produção da moeda eletrônica, independentemente da quantidade de vendedores em que o usuário venha a realizar compras/pagamentos.
- *O usuário deve estabelecer uma chave secreta compartilhada com cada vendedor.* No *MicroMint* não é necessário estabelecer um segredo compartilhado entre as partes, e o usuário necessita somente da identificação do vendedor, que é uma informação pública, para produzir a moeda eletrônica específica para aquele vendedor.
- *Os usuários não podem verificar a validade de seu dinheiro eletrônico.* Como no *MilliCent*, a moeda eletrônica *scrip* é autenticada com uma chave secreta conhecida apenas pelo vendedor ou pelo *broker*; então o usuário não possui uma maneira de verificar se o *scrip* em seu poder é válido. No *MicroMint*, a função *hash* usada para validar a moeda eletrônica é pública; assim, os usuários podem facilmente verificar se suas moedas eletrônicas que foram compradas de um *broker* são válidas.

6.3 Comparação de Propriedades

Nas seções 2.2, 2.3 e 2.4 foram identificadas diversas propriedades que uma solução de Pagamento Eletrônico deve possuir. Na Tabela 6.1 são apresentados os resultados da comparação entre os Sistemas de Pagamento Eletrônico *SET*, *PayWord*, *MicroMint* e *MilliCent*.

Propriedades	SET	PayWord	MicroMint	MilliCent
Macropagamento	√			
Micropagamento		√	√	√
Atomicidade da Transação	√	√	√	
Transação Segura	√	√	√	√
Privacidade da Identidade do Usuário	√	√	√	√
Privacidade do Usuário vs. Compra	√			
Prevenção de Falsificação	√	√	√	√
Detecção de Gasto Duplicado	√	√	√	√
Escalabilidade	√	√	√	√
Operação <i>On-line</i>	√			√
Operação <i>Off-line</i>		√	√	
Solução por <i>Software</i>	√	√	√	√

Tabela 6.1 – Análise comparativa das propriedades dos Sistemas de Pagamento Eletrônico

Verifica-se através dessa tabela que as três propostas estão aderentes aos principais requisitos apresentados nas seções 2.2, 2.3 e 2.4.

6.4 Escolha do *PayWord*

A escolha do protocolo *PayWord* na Implementação do Sistema Protótipo de Micropagamento levou em consideração as seguintes características importantes, já apresentadas e analisadas nos capítulos 3, 4 e 5 e nas comparações deste capítulo:

- pelo esquema *PayWord*, as operações de compra/pagamento entre o Usuário e o Vendedor são *Off-line* com relação ao Broker;
- o *PayWord* é um esquema de crédito. O Usuário não necessita pagar antecipadamente pela Moeda Eletrônica utilizada na operação de pagamento de uma compra. Sua conta é debitada somente quando o Vendedor for ressarcido do pagamento pelo Broker;
- o Usuário gera sua própria Moeda Eletrônica (*PayWords*);

- *o Usuário não precisa estabelecer uma chave secreta compartilhada com cada Vendedor;*
- *os Usuários podem verificar a validade de sua Moeda Eletrônica;*
- *o PayWord utiliza o esquema de criptografia de chave pública e função Hash;*
- *o PayWord usa espaço reduzido de armazenamento no Broker e no Vendedor;*
- *o Broker não necessita ser envolvido a cada pagamento junto a um novo vendedor;*
- *o PayWord não exige grande investimento em sua base computacional para a produção de Moedas Eletrônicas (PayWords).*

Cabe observar neste momento que a Implementação do Sistema Protótipo de Micropagamento com o protocolo *PayWord* teve por objetivo consolidar os estudos que esta dissertação pretendeu desenvolver em relação tanto ao tema Micropagamento como também às técnicas de criptografia empregadas.

A Arquitetura da implementação desse sistema é assunto que será desenvolvido no capítulo seguinte.

Capítulo 7

Arquitetura

Este capítulo descreve as principais características da Arquitetura para implementar o Sistema Protótipo de Micropagamento que foi desenvolvido com a utilização do protocolo *PayWord* apresentado no capítulo 3. A seção 7.1 apresenta uma visão geral dos Componentes do Sistema Protótipo. Na seção 7.2 são descritas as Interações entre os diversos módulos que compõem o Sistema Protótipo. A seção 7.3 procura mostrar a Infra-Estrutura utilizada na construção do Sistema Protótipo, quanto aos Sistemas Operacionais, Pacotes de Software, as Linguagens de Desenvolvimento e a Infra-Estrutura *Web* necessária para operacionalizar o Protótipo. A seção 7.4 mostra os Bancos de Dados que compõem o Sistema e as respectivas tabelas com sua composição e descrição de campos. A seção 7.5 fala da Infra-Estrutura de Comunicação necessária ao Sistema Protótipo. As seções 7.6 e 7.7 descrevem a Infra-Estrutura de Segurança, descrevendo as rotinas de Criptografia e o uso de Certificados Digitais.

7.1 Componentes

O Sistema Protótipo de Micropagamento foi implementado em três componentes básicos, associados às três entidades do modelo do protocolo *PayWord*, para atender às suas funções de forma distribuída em uma rede como a Internet. Isso significa que os componentes podem rodar em diferentes computadores na rede. No equipamento do *broker* roda o primeiro componente do Sistema Protótipo chamado Servidor de Certificação/Compensação. No equipamento do Usuário, associado a um navegador (*browser*), roda o segundo componente chamando Carteira Eletrônica. O Vendedor é constituído de uma Loja Virtual na *Web* onde roda um Servidor de Páginas *Web* associado ao terceiro componente do Sistema Protótipo chamado Servidor de Pagamentos.

7.2 Interações

A Figura 7.1 expõe uma visão geral dos principais componentes e respectivos módulos do Sistema Protótipo de Micropagamento, com as suas interações e dependências.

Figura 7.1 – Arquitetura do Sistema Protótipo

Descrição sucinta das interações entre os módulos do Sistema Protótipo:

- (a) O administrador do Sistema Protótipo no *broker*, através de uma interface administrativa no banco de dados do Servidor de Certificação/Compensação, registra a conta do novo Usuário e fornece ao mesmo, por um meio seguro, o código de acesso a este Servidor. As formas seguras de distribuição do código de acesso podem ser por carta (através de serviços especiais dos Correios), por correio eletrônico (*e-mail*) ou pela combinação do correio eletrônico e páginas *Web* configuradas com protocolo seguro como, por exemplo, SSL – *Secure Socket Layer* [11].
- (b) O Usuário, de posse do código de acesso ao Servidor de Certificação/Compensação e com a Carteira Eletrônica instalada em seu equipamento, registra no banco de dados local as informações de sua conta, através da Interface do Usuário, e inicia a Certificação junto ao Servidor de Certificação/Compensação.
- (c) Na ocasião do processo de cadastramento de nova conta do Usuário, a Carteira Eletrônica solicita ao Servidor de Certificação/Compensação a emissão e a assinatura do certificado digital para esse novo usuário, com a devida verificação do código de acesso ao Servidor provido pelo Usuário.
- (d) O Servidor de Certificação/Compensação, reconhecendo a validade do código de acesso e pedido de emissão de novo certificado digital para o usuário em questão, retorna para a Carteira Eletrônica o referido certificado digital assinado.
- (e) O Usuário acessa a rede Internet, através de um navegador (*browser*) e com sua Carteira Eletrônica e conta cadastrada e seu certificado digital emitido/assinado, poderá iniciar a consulta, através da Interface do Usuário, ao Servidor de Páginas *Web* onde reside a Loja Virtual do Vendedor.
- (f) O Usuário solicita a visualização das Páginas *Web* da Loja Virtual do Vendedor.
- (g) O Servidor de Páginas *Web* encaminha a página solicitada ao *browser* do usuário, para analisar uma possível compra de produto ou informação ofertada.
- (h) O Usuário faz a opção pela compra de um item ofertado na Página *Web* da Loja Virtual, a qual aciona o processo de aquisição, o que provoca o “instanciamento” da Carteira Eletrônica para identificar e autenticar o Usuário.
- (i) A Carteira Eletrônica conecta-se ao Servidor de Pagamentos para estabelecer o Compromisso de Pagamento desse Usuário, enviando seu certificado digital e os *PayWords* necessários para efetuar o pagamento do item adquirido.

- (j) O Servidor de Pagamentos valida o Compromisso de Pagamento e os *PayWords* enviados pela Carteira Eletrônica. Se forem aceitos, o Servidor envia uma resposta com uma chave aleatória para acesso ao item adquirido pelo Usuário.
- (k) A Carteira Eletrônica, recebendo a chave aleatória para acesso ao item adquirido pelo Usuário, monta o acesso via endereço URL (*Universal Resource Locator*) [25] para solicitar o *download* ao Servidor de Páginas Web.
- (l) O Servidor de Páginas Web obtém o arquivo solicitado pela chave aleatória para *download*.
- (m) O Servidor de Páginas Web aciona o processo de *download* para o equipamento do Usuário que efetuou a compra.
- (n) O administrador do Servidor de Pagamentos, no final do dia ou do período, aciona o processo de compensação de todas as vendas realizadas.
- (o) O Servidor de Pagamentos conecta-se ao Servidor de Certificação/Compensação para encaminhar as mensagens de compromisso de pagamento e o último *PayWord* utilizado como pagamento nas compras pelos Usuários.
- (p) O Servidor de Certificação/Compensação verifica cada mensagem de compromisso assinado pelos Usuários e o último *PayWord* utilizado como pagamento, enviada pelo Servidor de Pagamentos.
- (q) O administrador do Servidor de Certificação/Compensação, através da Interface Administrativa, verifica a situação do cadastramento dos Usuários do Sistema Protótipo, os certificados digitais emitidos e as vendas que já foram compensadas.

7.3 Infra-Estrutura

A Infra-Estrutura empregada para operacionalizar a implementação do Sistema Protótipo de Micropagamento desta dissertação foi a do Sistema Operacional *Windows 2000*, utilizando-se o modelo de aplicação de rede *Web Client/Server*. As partes *Server* do Protótipo que são o Servidor de Certificação/Compensação, o Servidor de Páginas Web e o Servidor de Pagamentos rodam no ambiente *Windows 2000 Server*. Já a parte *Client*, composta pelo navegador (*browser* – *MS-Internet Explorer 6.0*) e pela Carteira Eletrônica, roda nos sistemas operacionais *Windows 2000 Professional*, *Windows 95*, *98* ou *Millennium*. O Servidor de Páginas Web é o *IIS (MS-Internet Information Server)* [27] que gerencia tanto páginas *HTML (HypeText Markup Language)* [15] como páginas *ASP (Active Server Pages)* [23].

O ambiente de desenvolvimento dos componentes do Sistema Protótipo foi dividido nas seguintes linguagens:

- o Servidor de Certificação/Compensação, a Carteira Eletrônica e o Servidor de Pagamentos foram desenvolvidos em *MS-Visual Basic 6.0 Service Pack 5.0*;
- as páginas da Loja Virtual do Servidor de Páginas *Web* foram desenvolvidas em ASP (*Active Server Pages*) [23] pela ferramenta *MS-Visual InterDev 6.0*;
- as rotinas de Interface com as rotinas de Criptografia foram desenvolvidas como DLL (*Dynamic Link Library*) em *MS-Visual C++ 6.0* e são acionadas pelos componentes do Servidor de Certificação/Compensação, Carteira Eletrônica e Servidor de Pagamentos;

Vale observar que a Carteira Eletrônica foi desenvolvida para ser um componente acionado pelo *browser* e também executado como um programa normal. Para tanto, sua estrutura de desenvolvimento é a de um *ActiveX Control* [19] que pode ser instanciado por uma página ASP pelo Servidor de Páginas *Web*.

A Infra-Estrutura de rotinas de criptografia necessária para a implementação do protocolo *PayWord* nos componentes do Sistema Protótipo, foi provida pelo Pacote de Desenvolvimento de Infra-Estrutura de Chaves Pública/Privada (*PKI-Plus*) da Empresa *Baltimore Technologies* [22]. Utilizaram-se algoritmos, rotinas e padrões, como a função *Hash* pelo algoritmo SHA [3], o algoritmo de criptografia assimétrica *RSA* [26], uma Infra-Estrutura de Chaves Pública/Privada, com Certificados Digitais X.509 [20], o armazenamento seguro de certificados digitais para proteção da chave privado no padrão *PKCS#12 – Personal Information Exchange Syntax Standard* [21] e geração de números aleatórios pelo algoritmo *Blum-Blum-Shub (BBS)* [24].

7.4 Bancos de Dados

O Sistema Protótipo de Micropagamento foi desenvolvido usando a estrutura de Bancos de Dados e acesso via ODBC (*Open Database Connectivity*) tendo em vista facilitar a integração dos seus componentes, como por exemplo, o Servidor de Páginas *Web* com o Servidor de Pagamentos. Com isto, houve facilidade na consulta e na navegação, tanto pela parte *client*, como pela parte *Server* do Sistema. Nos Bancos de Dados, as informações são organizadas em tabelas, as quais possuem linhas e colunas e relacionamento cruzado entre linhas de duas tabelas; isso permite estabelecer integridade relacional das informações contidas em um Banco de Dados. A construção dos Bancos de Dados do Sistema Protótipo foi através da ferramenta *MS-ACCESS*, que possui compatibilidade e facilidade na migração futura para um Software de Gerenciamento de Bancos de Dados (*SGBD*), como no caso do *MS-SQL Server*, *Oracle*, *DB2* e outros. O Sistema Protótipo de Micropagamento possui três Bancos de Dados, sendo que o primeiro é consultado pelo Servidor Certificação/ Compensação, o segundo pela Carteira Eletrônica e o terceiro é consultado simultaneamente pelo Servidor de Páginas *Web* e pelo Servidor de Pagamentos.

7.4.1 Banco de Dados do Servidor de Certificação/Compensação

A Figura 7.2 mostra o relacionamento das três tabelas do Banco de Dados que é consultado pelo Servidor de Certificação/Compensação.

Figura 7.2 – Banco de Dados do Servidor de Certificação/Compensação

A tabela de *Clientes* armazena dados cadastrais e controles de acesso dos Usuários do Sistema Protótipo. A tabela de *Certificados*, para cada Cliente registrado na tabela de *Clientes*, contém tantas linhas quantos forem os certificados assinados pelo *Broker*. Cada linha da tabela de *Compromissos* está associada a um cliente e a um certificado digital, com os dados do compromisso de pagamento das compras realizadas pelo Usuário junto a um Vendedor, e a indicação, se o processo de compensação entre o *Broker* e o Vendedor foi efetivado.

Em seguida, a descrição de cada coluna das tabelas do Banco de Dados do Servidor de Certificação/Compensação.

7.4.1.1 Tabela de Clientes

IDCliente – identificador único do Cliente atribuído automaticamente pelo SGBD.

ContaCliente – número da conta do cliente.

NomeCliente – nome do cliente.

EndCliente – endereço eletrônico do cliente (*e-mail*).

CodigoAcesso – código de acesso ao Servidor de Certificação/Compensação.

CodigoAleatorio – código aleatório utilizado para validar o código de acesso.

7.4.1.2 Tabela de Certificados

IDCertificado – identificador único do certificado atribuído automaticamente pelo SGBD.

ContaCliente – número da conta do cliente desse certificado.

DataCertificado – data de assinatura do certificado.

VencimentoCertificado – define a data de vencimento do certificado.

ChavePublica – chave pública do certificado.

7.4.1.3 Tabela de Compromissos

IDCompromisso – identificador único de compromisso de pagamento atribuído automaticamente pelo SGBD.

IDCertificado – identificador certificado associado a esse compromisso de pagamento.

Compromisso – consiste do compromisso de pagamento de um Usuário.

Assinatura – assinatura do Usuário do compromisso de pagamento.

LastPayWord – define o último *PayWord* utilizado pelo Usuário como moeda eletrônica junto ao Vendedor.

CountPayWords – estabelece a quantidade de *PayWords* utilizado como moeda eletrônica pelo Usuário.

DataCompensacao – data e hora da realização da transação de compensação entre o Servidor de Certificação/Compensação e o Servidor de Pagamentos.

7.4.2 Banco de Dados da Carteira Eletrônica

A Figura 7.3 mostra o relacionamento das duas tabelas do Banco de Dados que é consultado pela Carteira Eletrônica.

Figura 7.3 – Banco de Dados da Carteira Eletrônica

A tabela de *Contas* mantém os dados cadastrais, a senha de acesso e as informações sobre o certificado digital do Usuário da Carteira Eletrônica. A tabela de *Histórico* registra para cada Usuário da Carteira Eletrônica toda transação realizada no Sistema Protótipo de Micropagamento.

Segue a descrição de cada coluna das tabelas do Banco de Dados da Carteira Eletrônica.

7.4.2.1 Tabela de Contas

ID – identificador único da Conta do Usuário atribuído automaticamente pelo SGBD.

Nome – nome do Usuário.

Conta – número da conta do Usuário.

Senha – senha criptográfica de acesso à Carteira Eletrônica.

VenctoCertificado – define a data de vencimento do certificado.

Certificado – consiste do certificado provido pelo *Broker*.

AssinaturaCertificado – assinatura do *Broker* do certificado.

CodigoAleatorio – código aleatório utilizado para validar a senha criptográfica de acesso.

7.4.2.2 Tabela de Histórico

IDConta – identificador único da Conta do Usuário.

Loja – nome da loja virtual do Vendedor.

Status – estado da transação.

Compra – código do item adquirido.

Item – nome do item adquirido.

Custo – valor monetário do item adquirido.

Data – data e hora da realização da transação de compra do item.

Chave – chave aleatória de acesso para viabilizar o *download* do item adquirido.

7.4.3 Banco de Dados do Servidor de Páginas Web e do Servidor de Pagamentos

A Figura 7.4 mostra o relacionamento das três tabelas do Banco de Dados que é consultado simultaneamente pelo Servidor de Páginas Web e pelo Servidor de Pagamentos.

Figura 7.4 – Banco de Dados do Servidor de Páginas Web e do Servidor de Pagamentos

A tabela de *Produtos* armazena os itens de venda do Vendedor, que é publicado pelo Servidor de Páginas Web e consultado pelo Usuário através do navegador (*browser*). Esta tabela também é consultada pelo Servidor de Pagamentos para verificar o item cuja compra foi solicitada pelo Usuário. Na tabela de *Compromissos*, o Servidor de Pagamentos registra cada compromisso de pagamento que um Usuário estabelece junto ao Vendedor quando do processo de compra. A cada item adquirido pelo Usuário, o Servidor de Pagamentos registra a transação na tabela de *Compras*.

Segue a descrição de cada coluna das tabelas do Banco de Dados do Servidor de Páginas Web e do Servidor de Pagamentos.

7.4.3.1 Tabela de Produtos

ID – identificador único do produto atribuído automaticamente pelo SGBD.

CodProduto – código do produto.

Produto – nome do produto.

Custo – valor monetário do produto.

Arquivo – nome do arquivo em venda (necessário no *download*).

7.4.3.2 Tabela de Compromissos

IDCompromisso – identificador único do compromisso atribuído automaticamente pelo SGBD.

IDBroker – identificador do *Broker*.

IDCertificado – identificador do certificado do Usuário.

FirstPayWord – define o primeiro *PayWord* utilizado por este compromisso.

LastPayWord – define o último *PayWord* utilizado pelo Usuário como moeda eletrônica.

CountPayWords – estabelece a quantidade de *PayWords* utilizado como moeda eletrônica pelo Usuário.

VencimentoCompromisso – define a data de vencimento do compromisso.

Certificado – consiste do certificado provido pelo *Broker*.

Assinatura – assinatura do *Broker* do certificado.

Compensado – indica se já foi realizada a transação de compensação das compras pelo Servidor de Pagamentos junto ao Servidor de Certificação/Compensação.

7.4.3.3 Tabela de Compras

IDCompromisso – identificador do compromisso de pagamento.

IDCertificado – identificador do certificado associado ao compromisso de pagamento.

Data – define a data e hora da realização da compra do produto.

IDProduto – identificador do produto.

Chave – chave aleatória de acesso para viabilizar o *download* do item adquirido.

7.5 Comunicação

O Sistema Protótipo de Micropagamento foi desenvolvido baseado na infra-estrutura de comunicação provida pela rede Internet, que permite o acesso global e em tempo real aos sistemas de informação hipermídia que, por sua vez, constitui a arquitetura *World Wide Web*, ou simplesmente *Web*.

A arquitetura da *Web* é tipicamente uma arquitetura de aplicação *client/server*. De um lado ficam os *clients*, compostos pelos navegadores (*browsers*), capazes de solicitar e exibir um documento hipermídia da rede. Um documento hipermídia pode ser definido como a reunião de hipertexto e multimídia numa única página *Web*. Hipertexto é combinação de textos não lineares, de forma a poder saltar de um texto para outro dinamicamente por meio de vínculos (*links*). Multimídia é a combinação de textos e outras mídias, como imagens estáticas ou em movimento, sons, vídeos, gráficos e outros. Do outro lado da arquitetura *Web* ficam os *servers*, compostos pelos Servidores de Páginas *Web*, cuja principal função é atender os pedidos dos *browsers* com o envio de páginas *Web* armazenadas no sistema de arquivos da plataforma onde se encontra instalado. Dependendo do pedido do *browser*, o Servidor de Páginas *Web* pode disparar a comunicação com uma aplicação externa, como por exemplo, a execução de um componente de uma aplicação que consulta um banco de dados.

Uma característica importante é que a *Web* foi projetada para contemplar vários protocolos de comunicação da Internet, incluindo FTP (*File Transfer Protocol*), Telnet, e outros.

O Sistema Protótipo de Micropagamento baseou-se na arquitetura *Web* e implementou o uso do *browser* para o Usuário consultar o Servidor de Páginas *Web*, considerando a Loja Virtual do Vendedor, a qual oferece produtos à venda. A seguir é descrita a comunicação entre o *browser* e o Servidor de Páginas *Web*, como também, a forma de acionamento e comunicação do *browser* com a Carteira Eletrônica. Da mesma maneira é descrita a forma de comunicação que é estabelecida entre os três componentes: da Carteira Eletrônica com o Servidor de Pagamentos, bem como com o Servidor de Certificação/Compensação. Essa mesma forma de comunicação é também estabelecida entre o Servidor de Pagamentos com o Servidor de Certificação/Compensação.

7.5.1 Browser versus Servidor de Páginas Web e Carteira Eletrônica

Basicamente, o *browser* comunica-se com o Servidor de Páginas *Web* através do protocolo HTTP (*HyperText Transport Protocol*) [7], que é um protocolo de aplicação do padrão ISO adequado para reconhecer e transferir dados em sistemas hipermídia distribuídos.

Pelo *browser*, o Usuário visualiza e navega pelas páginas *Web*, as quais são construídas utilizando a linguagem HTML (*HyperText Markup Language*) [15], que é um padrão para apresentar e formatar documentos *Web*. Essa linguagem é baseada no padrão SGML (*Standard Generalized Markup Language - ISO 8879:1986*) [16] e, como tal, independe da

plataforma em que é exibida, cabendo ao *browser* interpretá-la de acordo com sua configuração.

Uma importante característica da linguagem HTML é a possibilidade de a página *Web* conter um *script* que aciona a execução de um programa complementar ao *browser*, que é chamado de extensão do *browser*.

No Sistema Protótipo utilizou tal função para estabelecer a comunicação do *browser* com a Carteira Eletrônica, a qual foi construída com a tecnologia *MS-ActiveX* [19] e pode ser instanciada por uma página *Web*.

ActiveX Controls, formalmente conhecido como *OLE Controls* [19], são objetos inseridos nas páginas *Web*, usando o padrão da linguagem HTML *tag* <OBJECT>. Uma *tag* <OBJECT> estabelece um conjunto de parâmetros que são passados da página *Web* para o objeto e permite o acionamento de métodos do objeto.

No Sistema Protótipo de Micropagamento, quando um Usuário opta pela compra de um item que é apresentado na página *Web* da Loja Virtual do Vendedor, através de um *click* no botão “comprar”, o *browser* aciona um método da Carteira Eletrônica, que recebe o código do produto do qual foi solicitada a compra. Assim, a Carteira Eletrônica poderá iniciar o processo de autenticação do Usuário e estabelecer a comunicação com o Servidor de Pagamentos, dando continuidade ao processo de pagamento da compra.

7.5.2 Carteira Eletrônica versus Servidores de Pagamentos e Certificação / Compensação

A comunicação da Carteira Eletrônica com o Servidor de Pagamentos e com o Servidor de Certificação/Compensação, assim como, entre estes dois últimos, foi viabilizada utilizando-se a infra-estrutura de comunicação *Windows Sockets (WinSock)* [17] que é uma API (*Application Programming Interface*) provida pelos Sistemas Operacionais onde rodam os componentes do Sistema Protótipo.

O desenvolvimento da comunicação entre os componentes do Sistema Protótipo foi facilitado usando-se os controles *WinSock*, da linhagem de *MS-Visual Basic*, que permitem fácil acesso aos serviços de rede com protocolo TCP (*Transmission Control Protocol*) [18].

O protocolo TCP permite criar e manter uma conexão entre dois computadores remotos, de forma que ambos possam trocar dados entre si.

Para exemplificar o uso da facilidade da comunicação *Winsock*, a seguir é descrita a comunicação entre a Carteira Eletrônica, considerada uma aplicação *client*, com o Servidor de Pagamentos, considerado uma aplicação *server*.

Para iniciar a conexão entre as partes, procede-se da seguinte maneira: de uma lado, a Carteira Eletrônica que necessita prover ao controle *WinSock*, o nome do servidor remoto ou o seu endereço IP (*Internet Protocol*) através da propriedade *RemoteHost*, e também o número da porta ao qual o servidor remoto estará “escutando” a comunicação. O número da porta do servidor remoto é passado através da propriedade *RemotePort*. Em seguida, a Carteira Eletrônica aciona o método *Connect* para estabelecer a conexão com o servidor remoto.

Do outro lado está o *server*, o Servidor de Pagamentos que necessita prover ao controle *WinSock*, através da propriedade *LocalPort*, o número da porta ao qual ficará “escutando” a comunicação e, em seguida, deve acionar o método *Listen* para ficar aguardando o estabelecimento da conexão da Carteira Eletrônica.

Quando a Carteira Eletrônica requisitar uma conexão ao Servidor de Pagamentos, o evento *ConnectionRequest* ocorrerá no Servidor de Pagamentos.

Para completar a conexão, dentro do evento *ConnectionRequest* no Servidor de Pagamentos, é necessário acionar o método *Accept* para aceitar o pedido de conexão remota. Uma vez a conexão tenha sido aceita, a Carteira Eletrônica pode enviar dados ao Servidor de Pagamentos e dele receber.

Para a Carteira Eletrônica enviar uma mensagem ao Servidor de Pagamentos é preciso que ela acione o método *SendData* provendo, assim, os dados a serem transmitidos ao servidor remoto.

Quando os dados são recebidos pelo Servidor de Pagamentos, ocorre o evento *DataArrival*, e dentro desse evento o método *GetData* pode ser acionado para extrair os dados enviados pela Carteira Eletrônica.

7.5.3 Servidor de Pagamentos versus Servidor de Certificação/Compensação

O Servidor de Pagamentos se comunica com o Servidor de Certificação/Compensação usando a mesma infra-estrutura descrita na seção 7.5.2. O Servidor de Pagamentos é considerado uma aplicação *client*, e o Servidor de Certificação/Compensação é considerado uma aplicação *server*.

Vale observar que o Servidor de Pagamentos tanto se comporta como uma aplicação *server*, quando está aguardando uma conexão da Carteira Eletrônica, como uma aplicação *client*, quando estabelece a conexão com o Servidor de Certificação/Compensação.

Por outro lado, o Servidor de Certificação/Compensação se comporta apenas como aplicação *server*; contudo, admite duas conexões simultâneas, sendo uma com a aplicação *client* da Carteira Eletrônica, e a outra com aplicação *client* do Servidor de Pagamentos.

7.6 Criptografia e Certificado Digital

Na implementação do protocolo *PayWord* no Sistema Protótipo de Micropagamento foi necessário utilizar um conjunto de algoritmos e rotinas de criptografia, de domínio público, reconhecidos e testados no âmbito acadêmico e comercial. Muitos algoritmos de criptografia possuem seus códigos fontes publicados em livros, como é o caso de *Menezes*, *Van Oorschot* e *Vanstone* [8], de *Schneier* [9], de *Routo Terada* [10] e de fácil acesso pela Internet [14].

Com a pesquisa na Internet buscou-se localizar uma biblioteca de algoritmos e rotinas de criptografia que melhor atendesse a necessidade de implementar a função *Hash* pelo algoritmo SHA-1 [3] para gerar *PayWords* (dinheiro eletrônico) do protocolo, como também, rotinas e algoritmos para montar uma Infra-Estrutura de Chaves Pública/Privada e Certificados Digitais [20].

Finalizou-se com a escolha do Pacote de Desenvolvimento de Infra-Estrutura de Chaves Pública/Privada (*PKI-Plus*) da Empresa *Baltimore Technologies* [22] que não só forneceu uma biblioteca completa de algoritmos e rotinas de criptografia, como também atendeu às necessidades da implementação do protocolo *PayWord* no Sistema Protótipo de Micropagamento.

Houve inicialmente dificuldades na utilização da biblioteca de algoritmos e rotinas de criptografia *PKI-Plus* pelos componentes do Sistema Protótipo de Micropagamento. Em decorrência da necessidade de uma interface amigável com os Usuários da Carteira Eletrônica, do Servidor de Certificação/Compensação e do Servidor de Pagamentos, estes foram escritos na linguagem *MS-Visual Basic 6.0* e precisavam acionar rotinas de criptografia para implementar o protocolo *PayWord*. Entretanto, a biblioteca de algoritmos e rotinas de criptografia *PKI-Plus* foi escrita na linguagem C++. Isto gerou incompatibilidade no acionamento direto das rotinas de criptografia *PKI-Plus* pelos componentes do Sistema Protótipo. A solução usada foi implementar rotinas de Interfaces em DLL, escritas na linguagem *MS-Visual C++ 6.0*, as quais possuem compatibilidade para serem acionadas por componentes escritos na linguagem *MS-Visual Basic 6.0*.

Foram desenvolvidas cinco rotinas de Interfaces em DLL, incorporando a biblioteca de algoritmos e rotinas de criptografia *PKI-Plus*. Tais rotinas contemplam: (1) geração de certificados digitais, (2) assinatura digital, (3) verificação de assinatura digital, (4) rotina para criptografar e decriptografar mensagens usando o algoritmo RSA e (5) função *Hash* que implementa o algoritmo SHA-1.

Nos itens 7.6.1 a 7.6.5 são descritas as cinco rotinas de Interfaces em DLL, mostrando em quais componentes são usadas, suas interfaces de acionamento em *MS-Visual Basic 6.0* e a descrição dos campos que devem ser passados e recebidos.

7.6.1 Geração de Certificados Digitais

A rotina *Certificacao.dll* foi desenvolvida para gerar certificados digitais no padrão *X.509* [20] e armazená-los com suas respectivas chaves privadas de seus Usuários em um arquivo em disco no padrão *PKCS#12 – Personal Information Exchange Syntax Standard* [21].

Os parâmetros utilizados nas rotinas de criptografia *PKI-Plus* para gerar certificados digitais foram os seguintes:

- chaves pública e privada com 512 bits;
- possibilidade de assinar e verificar mensagens com as chaves pública e privada;
- possibilidade de criptografar e decriptografar mensagens, utilizando o algoritmo RSA, com base nas chaves pública e privada do Usuário;
- possibilidade de exportação do certificado digital armazenado em disco no padrão *PKCS#12* para o padrão de certificados digitais *X.509* aceitos pelos *browsers*, softwares de correio eletrônico e outros.

A Figura 7.5 mostra a interface da rotina *Certificacao.dll* usada pela Carteira Eletrônica para certificação do Usuário pelo *Broker* (ver seção 3.2.1).

```
Private Declare Function fnCertificacao Lib "Certificacao.dll" _  
    (ByVal certNome As String, _  
    ByVal certNomeTam As Integer, _  
    ByVal certSenha As String, _  
    ByVal certSenhaTam As Integer, _  
    ByVal certSementeRandom As String, _  
    ByVal certSementeRandomTam As Integer, _  
    ByVal certChavePublica As String, _  
    ByVal certChavePublicaDump As String) _  
    As Integer
```

Figura 7.5 – Interface da rotina de Certificação

Campos que devem ser fornecidos quando é chamada a rotina *Certificacao.dll*:

Campo	Descrição
<i>certNome</i>	Nome do usuário do certificado digital.
<i>certNomeTam</i>	Tamanho do nome do usuário.
<i>cerSenha</i>	Senha para uso futuro no acesso ao arquivo em disco padrão PKCS#12 que conterá a chave privada e o certificado digital do usuário.
<i>certSenhaTam</i>	Tamanho da senha de acesso.
<i>certSementeRandom</i>	<i>String</i> binária de dados randômicos como semente aleatória para gera a chave pública e privada do usuário.
<i>certSementeRandomTam</i>	Tamanho da semente randômica.

Campos que retornarão após a execução da rotina *Certificacao.dll*:

Campo	Descrição
<i>certRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica erro na execução.
<i>certChavePublica</i>	Chave pública do usuário do certificado digital.
<i>certChavePublicaTam</i>	Tamanho da chave pública.

A Figura 7.6 expõe o diretório em disco onde a rotina *Certificacao.dll* gerou o certificado digital do usuário *neto*. O arquivo *neto.p12* está no padrão *PKCS#12* e foi exportado para o arquivo *neto.cer*, que está no padrão *X.509* aceito pelos *browsers* e softwares de correio eletrônico e outros.

Figura 7.6 – Diretório de Certificados Digitais

A Figura 7.7 reproduz detalhes dos principais campos do certificado digital do Usuário *neto*, gerado pela rotina *Certificacao.dll* no processo de cadastramento desse Usuário pela Carteira Eletrônica e, em seguida, exportado para uso no *browser* – *MS-Internet Explorer 6.0*. para visualização como segue.

Figura 7.7 – Certificado Digital do Usuário *neto*

7.6.2 Assinatura Digital

A rotina *Assinatura.dll* foi desenvolvida para gerar assinaturas digitais pelo algoritmo RSA usando a chave privada do Usuário que consta do seu certificado digital padrão *X.509*, gerado e armazenado em arquivo em disco no padrão *PKCS#12*.

A Figura 7.8 mostra a interface da rotina *Assinatura.dll* usada pela Carteira Eletrônica para assinar a mensagem de compromisso de pagamento (ver Figura 3.3).

```
Private Declare Function fnAssinatura Lib "Assinatura.dll" _
    (ByVal assNome As String, _
    ByVal assNomeTam As Integer, _
    ByVal assSenha As String, _
    ByVal assSenhaTam As Integer, _
    ByVal Mensagem As String, _
    ByVal MensagemTam As Integer, _
    ByVal Assinatura As String, _
    ByVal AssinaturaDump As String) _
    As Integer
```

Figura 7.8 – Interface da rotina de Assinatura Digital

Descrição dos campos que devem ser fornecidos quando é chamada a rotina *Assinatura.dll*:

Campo	Descrição
<i>assNome</i>	Nome do usuário do certificado digital.
<i>assNomeTam</i>	Tamanho do nome do usuário.
<i>assSenha</i>	Senha de acesso ao arquivo em disco padrão <i>PKCS#12</i> que contém a chave privada e o certificado digital do usuário.
<i>assSenhaTam</i>	Tamanho da senha de acesso.
<i>Mensagem</i>	Mensagem que será assinada.
<i>MensagemTam</i>	Tamanho da mensagem que será assinada.
<i>assWorkArea</i>	Área de trabalho exigida pela rotina.

Campos que retornarão após a execução da rotina *Assinatura.dll*:

Campo	Descrição
<i>assRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica erro na execução.
<i>Assinatura</i>	Assinatura digital da mensagem.
<i>AssinaturaDump</i>	Assinatura digital da mensagem no formato <i>Dump</i> , ou seja, formato hexadecimal.

7.6.3 Verificação de Assinatura Digital

A rotina *Verificacao.dll* foi desenvolvida para verificar a assinatura digital de uma mensagem assinada pelo algoritmo RSA, usando a chave pública do Usuário que assinou tal mensagem.

Essa rotina possui duas interfaces, conforme são mostradas nas Figuras 7.9 e 7.10. São usadas pelo Servidor de Pagamentos para verificar a assinatura digital do Usuário e do *Broker* (ver Figura 3.3) e pelo Servidor de Certificação/Compensação para verificar sua própria assinatura digital e do Usuário (ver Figura 3.5).

A Figura 7.9 traz a interface da rotina *Verificacao.dll* que obtém a chave pública do usuário de seu certificado digital, exportado e armazenado em disco no padrão de certificados digitais X.509 aceitos pelos *browsers*, softwares de correio eletrônico e outros.

```
Private Declare Function fnVerAssinatura1 Lib "Verificacao.dll" _
    (ByVal verNome As String, _
    ByVal verNomeTam As Integer, _
    ByVal verMensagem As String, _
    ByVal verMensagemTam As Integer, _
    ByVal verAssinatura As String, _
    ByVal verAssinaturaTam As Integer, _
    ByVal verWorkArea As String) _
    As Integer
```

**Figura 7.9 – Interface da rotina de Verificação da Assinatura Digital
Chave Pública obtida do Certificado Digital**

A Figura 7.10 mostra a interface da rotina *Verificacao.dll* que recebe a chave pública do Usuário como parâmetro em sua chamada.

```
Private Declare Function fnVerAssinatura2 Lib "Verificacao.dll" _
    (ByVal verChavePublica As String, _
    ByVal verChavePublicaTam As Integer, _
    ByVal verMensagem As String, _
    ByVal verMensagemTam As Integer, _
    ByVal verAssinatura As String, _
    ByVal verAssinaturaTam As Integer, _
    ByVal verWorkArea As String) _
    As Integer
```

**Figura 7.10 – Interface da rotina de Verificação da Assinatura Digital
Chave Pública fornecida na chamada da rotina**

Descrição dos campos que devem ser fornecidos quando é chamada a rotina *Verificacao.dll*. Pode-se observar que os campos *verNome/verNomeTam* e os campos *verChavePublica/verChavePublicaTam* se excluem mutuamente.

Campo	Descrição
<i>verNome</i>	Nome do usuário do certificado digital.
<i>verNomeTam</i>	Tamanho do nome do usuário.
<i>verChavePublica</i>	Chave pública do usuário do certificado digital.
<i>verChavePublicaTam</i>	Tamanho da chave pública.
<i>verMensagem</i>	Mensagem que foi assinada.
<i>verMensagemTam</i>	Tamanho da mensagem que foi assinada.
<i>verAssinatura</i>	Assinatura digital da mensagem.
<i>verAssinaturaTam</i>	Tamanho da assinatura digital.
<i>verWorkArea</i>	Área de trabalho exigida pela rotina.

Descrição do campo que retornará após a execução da rotina *Verificacao.dll*:

Campo	Descrição
<i>verRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica que a assinatura digital não corresponde à mensagem, ou a chave pública do usuário não corresponde à chave privada usada para assinar a mensagem.

7.6.4 Criptografia RSA

A rotina *CriptoRSA.dll* foi desenvolvida para: (1) criptografar uma mensagem pelo algoritmo RSA usando a chave pública do Usuário a quem se deseja enviar a mensagem criptografada; (2) decriptografar uma mensagem pelo algoritmo RSA usando a chave privada do Usuário que recebeu a mensagem criptografada.

Essa rotina possui três interfaces, conforme mostradas nas Figuras 7.11 a 7.13. São usadas pela Carteira Eletrônica e pelo Servidor Certificação/Compensação, para estabelecer um canal seguro no envio de informações confidenciais do Usuário no processo de certificação pelo *broker* (ver Figura 3.2).

A Figura 7.11 apresenta a interface da rotina *CriptoRSA.dll* que efetua a criptografia de uma mensagem usando a chave pública do Usuário, a quem se deseja enviar uma mensagem criptografada. A chave pública desse Usuário será obtida do certificado digital que foi exportado e armazenado em disco no padrão de certificados digitais X.509 aceitos pelos *browsers*, softwares de correio eletrônico e outros.

```
Private Declare Function fnEnCriptraRSA1 Lib "CriptoRSA.dll" _
    (ByVal enCriptraNome As String, _
    ByVal enCriptraNomeTam As Integer, _
    ByVal enCriptraMensagem As String, _
    ByVal enCriptraMensagemTam As Integer, _
    ByVal enCriptraMsgCriptra As String, _
    ByVal enCriptraMsgCriptraDump As String) _
    As Integer
```

**Figura 7.11 – Interface da rotina de Encriptação RSA
Chave Pública obtida do Certificado Digital**

A Figura 7.12 expõe a interface da rotina *CriptoRSA.dll* que efetua a criptografia de uma mensagem usando a chave pública do Usuário recebida como parâmetro na sua chamada.

```
Private Declare Function fnEnCriptraRSA2 Lib "CriptoRSA.dll" _
    (ByVal enChavePublica As String, _
    ByVal enChavePublicaTam As Integer, _
    ByVal enCriptraMensagem As String, _
    ByVal enCriptraMensagemTam As Integer, _
    ByVal enCriptraMsgCriptra As String, _
    ByVal enCriptraMsgCriptraDump As String, _
    ByVal enWorkArea As String) _
    As Integer
```

**Figura 7.12 – Interface da rotina de Encriptação RSA
Chave Pública fornecida na chamada da rotina**

Descrição dos campos que devem ser fornecidos quando é chamada a rotina *CriptoRSA.dll* para efetuar a criptografia de uma mensagem. Pode-se observar que os campos *enCriptaNome/ enCriptaNomeTam* e os campos *enChavePublica/enChavePublicaTam* se excluem mutuamente.

Campo	Descrição
<i>enCriptaNome</i>	Nome do usuário do certificado digital.
<i>enCriptaNomeTam</i>	Tamanho do nome do usuário.
<i>enChavePublica</i>	Chave pública do usuário do certificado digital.
<i>enChavePublicaTam</i>	Tamanho da chave pública.
<i>enCriptaMensagem</i>	Mensagem a ser criptografada.
<i>enCriptaMensagemTam</i>	Tamanho da mensagem a ser criptografada.
<i>enWorkArea</i>	Área de trabalho exigida pela rotina.

Seguem os campos que retornarão após a execução da rotina *CriptoRSA.dll*:

Campo	Descrição
<i>enCriptaRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica erro na execução.
<i>enCriptaMsgCripta</i>	Mensagem criptografada.
<i>enCriptaMsgCriptaDump</i>	Mensagem criptografada no formato <i>Dump</i> , ou seja, formato hexadecimal.

A Figura 7.13 mostra a interface da rotina *CriptoRSA.dll* que efetua a decriptação de uma mensagem criptografada, usando a chave privada do Usuário que recebeu tal mensagem. A chave privada desse Usuário será obtida do arquivo onde consta junto ao seu certificado digital padrão *X.509*, gerado e armazenado em disco no padrão *PKCS#12*.

```
Private Declare Function fnDeCriptraRSA Lib "CriptraRSA.dll" _
    (ByVal deCriptraNome As String, _
    ByVal deCriptraNomeTam As Integer, _
    ByVal deCriptraSenha As String, _
    ByVal deCriptraSenhaTam As Integer, _
    ByVal deCriptraMensagem As String, _
    ByVal deCriptraMensagemTam As Integer, _
    ByVal deCriptraMsgDeCriptra As String, _
    ByVal deCriptraMsgDeCriptraDump As String) _
    As Integer
```

Figura 7.13 – Interface da rotina de Decriptação RSA

Descrição dos campos que devem ser fornecidos quando é chamada a rotina *CriptoRSA.dll*:

Campo	Descrição
<i>deCriptraNome</i>	Nome do usuário do certificado digital.
<i>deCriptraNomeTam</i>	Tamanho do nome do usuário.
<i>deCriptraSenha</i>	Senha de acesso ao arquivo em disco padrão PKCS#12 que contém a chave privada e o certificado digital do usuário.
<i>deCriptraSenhaTam</i>	Tamanho da senha de acesso.
<i>deCriptraMensagem</i>	Mensagem criptografada.
<i>deCriptraMensagemTam</i>	Tamanho da mensagem criptografada.
<i>deWorkArea</i>	Área de trabalho exigida pela rotina.

Descrição dos campos que retornarão após a execução da rotina *Verificacao.dll*:

Campo	Descrição
<i>deCriptraRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica erro na execução.
<i>deCriptraMsgDeCriptra</i>	Mensagem decriptografada.
<i>deCriptraMsgDeCriptraDump</i>	Mensagem decriptografada no formato <i>Dump</i> , ou seja, formato hexadecimal.

7.6.5 Função Hash

A rotina *Hash.dll* foi desenvolvida para implementar a função *hash* pelo algoritmo SHA-1 para gerar os *PayWords* (dinheiro eletrônico) na implementação do protocolo *PayWord*.

Ela possui duas interfaces, conforme são mostradas nas Figuras 7.14 e 7.15. São usadas pela Carteira Eletrônica na geração dos *PayWords*, pelo Servidor de Pagamentos para verificar os *PayWords* na compras (ver seção 3.2.2) e pelo Servidor de Certificação/Compensação para efetuar a verificação e compensação das vendas (ver seção 3.2.3).

A Figura 7.14 apresenta a interface da rotina *Hash.dll* que produz um *PayWord* a partir de uma semente que deve ser um *string* binário de dados aleatório.

```
Private Declare Function fnHash Lib "Hash.dll" _
    (ByVal hashSemente As String, _
    ByVal hashSementeTam As Integer, _
    ByVal hashPayWord As String, _
    ByVal hashPayWordDump As String, _
    ByVal hashWorkArea As String) _
    As Integer
```

Figura 7.14 – Interface da rotina *Hash.dll*
Produz o primeiro *Payword* de uma série encadeada

Descrição dos campos que devem ser fornecidos quando é chamada a rotina *Hash.dll*.

Campo	Descrição
<i>hashSemente</i>	<i>String</i> binária de dados randômicos como semente aleatória para gerar o primeiro <i>PayWord</i> .
<i>hashSementeTam</i>	Tamanho da semente aleatória.
<i>hashWorkArea</i>	Área de trabalho exigida pela rotina.

Descrição dos campos que retornarão após a execução da rotina *Hash.dll*:

Campo	Descrição
<i>hashRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica erro na execução.
<i>hashPayWord</i>	Primeiro <i>PayWord</i> de uma série encadeada.
<i>hashPayWordDump</i>	Primeiro <i>PayWord</i> de uma série encadeada no formato <i>Dump</i> , ou seja, formato hexadecimal.

A Figura 7.15 mostra a interface da rotina *Hash.dll* que produz um próximo *PayWord* a partir de um *PayWord* produzido anteriormente.

```
Private Declare Function fnHashDump Lib "Hash.dll" _
    (ByVal hashPayWordDump As String, _
    ByVal hashPayWordDumpTam As Integer, _
    ByVal hashNextPayWord As String, _
    ByVal hashNextPayWordDump As String, _
    ByVal hashWorkArea As String) _
    As Integer
```

**Figura 7.15 – Interface da rotina *Hash.dll*
Produz o próximo *PayWord* de uma série encadeada**

Descrição dos campos que devem ser fornecidos quando é chamada a rotina *Hash.dll*.

Campo	Descrição
<i>hashPayWordDump</i>	Primeiro <i>PayWord</i> de uma série encadeada no formato <i>Dump</i> , ou seja, formato hexadecimal.
<i>hashPayWordDumpTam</i>	Tamanho do primeiro <i>PayWord</i> .
<i>hashWorkArea</i>	Área de trabalho exigida pela rotina.

Descrição dos campos que retornarão após a execução da rotina *Hash.dll*:

Campo	Descrição
<i>hashRetorno</i>	Código de retorno da rotina – retornando valor diferente de zeros implica erro na execução.
<i>hashNextPayWord</i>	Próximo <i>PayWord</i> de uma série encadeada.
<i>hashNextPayWordDump</i>	Próximo <i>PayWord</i> de uma série encadeada no formato <i>Dump</i> , ou seja, formato hexadecimal.

Capítulo 8

Implementação

Este capítulo descreve a Implementação dos componentes do Sistema Protótipo de Micropagamento que foi desenvolvido através do emprego da Arquitetura apresentada capítulo 7. A implementação deste Sistema Protótipo consolidou os estudos pretendidos por esta dissertação. As seções 8.1, 8.2 e 8.3 apresentam os principais componentes que constituem o Sistema Protótipo, como o Servidor de Certificação/Compensação, a Carteira Eletrônica e o Servidor de Pagamentos. A seção 8.4 procura detalhar as páginas *Web* que compõem a Loja Virtual.

8.1 Servidor de Certificação/Compensação

O Servidor de Certificação/Compensação desenvolve o importante processo no protocolo *PayWord*. Sua função é mediar as transações de pagamento entre o Usuário/Cliente e o Vendedor no Sistema Protótipo de Micropagamento. Isto inclui o Cadastramento de Novo Usuário/Cliente no Banco de Dados, a Certificação do Usuário/Cliente com a assinatura do Certificado Digital *PayWord* e a Compensação das Transações realizadas pelo Usuário/Cliente junto ao Vendedor.

8.1.1 Componentes e Estrutura

Diagrama da visão geral do Servidor de Certificação/Compensação no *Broker*:

Figura 8.1 – Componentes e Estrutura do Servidor de Certificação/Compensação

Breve descrição dos principais componentes do Servidor de Certificação/ Compensação:

- **SrvCertComp.exe** – Executável do Servidor de Certificação/Compensação, componente *server* desenvolvido na linguagem *MS-Visual Basic 6.0*, roda no ambiente do sistema operacional *Windows 2000 Server*, responsável por manter o Banco de Dados e interagir pelo protocolo *PayWord* com a Carteira Eletrônica do Usuário e o Servidor de Pagamentos no Vendedor;
- **Assinatura.dll, Verificacao.dll, CritoRSA.dll. e Hash.dll** – Rotinas de Interface em DLL que implementa as rotinas de Criptografia e Certificação Digital do protocolo *PayWord* (ver seção 7.6);
- **Banco de Dados** – Armazena os dados cadastrais, os Certificados Digitais assinados, as transações realizadas e compensadas de cada Usuário do Sistema Protótipo.

Atualmente está implementado em *MS-Access* e pode admitir qualquer SGDB que dá suporte à interface ODBC (ver seção 7.4).

8.1.2 Monitoração

O executável do Servidor de Certificação/Compensação deve ser mantido rodando no equipamento do *Broker* para mediar a comunicação prevista no protocolo *PayWord* entre um Usuário e Vendedor. Uma janela de monitoração da comunicação foi desenvolvida conforme mostra a Figura 8.2.

Figura 8.2 – Janela de Monitoração do Servidor de Certificação/Compensação

Na janela de Monitoração do Servidor de Certificação/Compensação foram desenvolvidos botões para acionar as seguintes funções:

- **Novo Cliente** – Cadastrar novo Usuário/Cliente no Banco de Dados;
- **Lista Clientes** – Listar os Usuários/Clientes cadastrados no Banco de Dados;
- **Lista Certificados** – Listar os Certificados Digitais dos Usuários/Clientes;
- **Lista Vendas** – Listar as transações realizadas e compensadas dos Usuários/Clientes.

8.1.3 Cadastramento de Novo Usuário/Cliente

Cada Usuário/Cliente do Sistema Protótipo de Micropagamento deve ser previamente cadastrado no Banco de Dados do Servidor de Certificação/Compensação. Para tanto, é necessário informar os dados da conta do Usuário junto ao *Broker* e estabelecer uma senha de acesso que será usada pelo Usuário no momento da abertura da conta na Carteira Eletrônica.

Pelo botão *Novo Cliente* na janela de Monitoração do Servidor de Certificação/Compensação, pode ser acionado o Cadastramento de Novo Cliente/Usuário, conforme pode ser visto na Figura 8.3.

A imagem mostra uma janela de software com o título "Servidor de Certificação/Compensação - Cliente". A janela contém os seguintes elementos:

- Uma seção "Broker" com o campo "Nome:" preenchido com "Broker-001".
- Uma seção "Novo Cliente" com o texto "Digite os dados para criar uma nova conta:".
- Dentro da seção "Novo Cliente", há quatro campos de entrada de texto:
 - "Nome:" com o valor "neto".
 - "Conta:" com o valor "1234".
 - "Endereço Eletrônico (email):" com o valor "neto@ime.usp.br".
 - Uma subseção "Senha de Acesso ao Servidor de Certificados" com dois campos: "Senha:" e "Confirmar senha:", ambos preenchidos com caracteres de máscara (sete "x" e sete "x" respectivamente).
- Na base da janela, há dois botões: "Cadastrar" e "Monitoração".

**Figura 8.3 – Janela de Cadastramento de Novo Cliente
Servidor de Certificação/Compensação**

8.1.4 Lista dos Usuários/Clientes

A Figura 8.4 mostra a janela dos Usuários/Clientes que estão cadastrados no Banco de Dados do Sistema Protótipo; tal janela pode ser acionada pelo botão *Lista Clientes* na janela de Monitoração do Servidor de Certificação/Compensação.

Figura 8.4 – Janela da Lista dos Clientes/Usuários Cadastrados no Servidor de Certificação/Compensação

8.1.5 Lista dos Certificados Digitais

A Figura 8.5 apresenta a janela dos Certificados Digitais cadastrados dos Usuários/Clientes que efetuaram a abertura da conta na Carteira Eletrônica. Esta janela pode ser acionada pelo botão *Lista Certificados* na janela de Monitoração do Servidor de Certificação/Compensação.

Figura 8.5 – Janela da Lista dos Certificados Digitais cadastrados no Servidor de Certificação/Compensação

8.1.6 Lista das Vendas

A Figura 8.6 mostra a janela das vendas realizadas pelos Vendedores aos Usuários/Clientes do Sistema Protótipo de Micropagamento cujos Vendedores já tenham acionado o processo de Compensação no Servidor de Pagamentos junto ao Servidor de Certificação/Compensação. Esta janela pode ser acionada pelo botão *Lista Vendas* na janela de Monitoração do Servidor de Certificação/Compensação.

Conta	Nome do Cliente	Certificado	Compromisso	Loja Virtual	Último PayWord	Consumo	Compensação
1234	neto	112	100	InfoX	332A8216ACFDD967174707EA1F7BE714	R\$ 13,91	30/12/2001 00:04

Figura 8.6 – Janela da Lista das Vendas compensadas pelos Vendedores no Servidor de Certificação/Compensação

8.2 Carteira Eletrônica

A Carteira Eletrônica desenvolve o processo de produzir e controlar o Dinheiro Eletrônico (*PayWords*) do Usuário; é empregada para os pagamentos de produtos/informação de baixo valor monetário junto ao Vendedor pelo esquema do protocolo *PayWord* no Sistema de Protótipo de Micropagamento.

8.2.1 Componentes e Estrutura

Diagrama da visão geral da Carteira Eletrônica do Usuário:

Figura 8.8 – Componentes e Estrutura da Carteira Eletrônica

Breve descrição dos principais componentes da Carteira Eletrônica:

- **CartEletronica.exe** – Executável da Carteira Eletrônica, componente *MS-ActiveX Control* que é acionado pelo *browser*, desenvolvido na linguagem *MS-Visual Basic 6.0*, roda nos ambientes dos sistemas operacionais *Windows 2000 Professional*, *Windows 95*, *98* ou *Millennium*, responsável por produzir e controlar o Dinheiro Eletrônico do Usuário, manter/registrar as operações de pagamento junto ao Vendedor no Banco de Dados e interagir pelo protocolo *PayWord* com o Servidor de Pagamentos e o Servidor de Certificação/Compensação;
- **Certificacao.dll, Assinatura.dll, CriptoRSA.dll. e Hash.dll** – Rotinas de Interface em DLL que implementa as rotinas de Criptografia e Certificação Digital do protocolo *PayWord* (ver seção 7.6);
- **Banco de Dados** – Armazena os Certificados Digitais assinados pelo *Broker*, mantém o histórico das operações de pagamento das compras realizadas por cada Usuário da Carteira Eletrônica. Atualmente está implementado em *MS-Access* e pode admitir qualquer SGDB que dá suporte à interface ODBC (ver seção 7.4).

O pacote de software que compõe a Carteira Eletrônica pode ser obtido do *site* da Loja Virtual do Vendedor via *download* e ser instalado no equipamento do Usuário através de seu *Setup*, montado pela ferramenta *MS-Package and Deployment Wizard* e permite incluir todos os componentes citados em um único arquivo executável e compactados.

8.2.2 Login do Usuário

O executável da Carteira Eletrônica pode ser chamado de duas formas:

- a primeira, pelo botão *Start* → *Programs* → *Carteira Eletrônica* no equipamento do Usuário com um dos sistemas operacionais *Windows* suportado. Desta forma, a Carteira Eletrônica estará rodando sem vínculo com o *browser* e seu Usuário poderá efetuar a abertura de sua Conta, a exclusão de uma Conta ou, ainda, verificar o histórico das transações de pagamento realizadas;
- a segunda é através do *browser*, quando o Usuário optar pelo pagamento de uma compra junto ao Vendedor.

De qualquer maneira, a primeira função que a Carteira Eletrônica desenvolve logo que seja chamada sua execução é a apresentação da janela de *Login* do Usuário, conforme mostra a Figura 8.9.

**Figura 8.9 – Janela de *Login* do Usuário
Carteira Eletrônica**

8.2.3 Abertura de Nova Conta

O Usuário que teve sua Conta Cadastrada no Servidor de Certificação/Compensação e possua a senha de acesso ao Servidor, poderá abrir sua conta na Carteira Eletrônica. Para isso, na janela de *Login* do Usuário (ver Figura 8.9), o usuário opta pelo botão *Nova Conta* e a Carteira Eletrônica apresentará a janela, conforme mostra a Figura 8.10.

**Figura 8.10 – Janela Abertura Nova Conta
Carteira Eletrônica**

8.2.4 Certificação da Nova Conta

O Usuário preenche os campos: *Nome*, *Conta*, *Senha* e *Confirmar senha* na Janela *Abertura de Nova Conta*, conforme a Figura 8.10; em seguida, opta pelo botão *Cadastrar*, a Carteira Eletrônica procederá ao cadastramento dos dados do Usuário no Banco de Dados da Carteira Eletrônica, a geração do Certificado Digital do Usuário, conforme seção 7.6.1 e apresentará a janela *Confirmar Certificação*, como mostra a Figura 8.11.

**Figura 8.11 – Janela Confirmar Certificação
Carteira Eletrônica**

Em seguida, o Usuário de posse da senha de acesso ao Servidor de Certificação/Compensação, preenchendo os campos *Senha* e *Confirmar senha* e optando pelo botão *Confirmar* na janela *Confirmar Certificação*, a Carteira Eletrônica estabelecerá a comunicação com o Servidor de Certificação/Compensação e procederá a Certificação do Usuário junto ao *Broker*, segundo o protocolo *PayWord* (ver seção 3.2.1).

8.2.5 Monitoração da Certificação

A Figura 8.12 mostra a janela de Monitoração do Servidor de Certificação/Compensação quando do processo de Certificação do Usuário, após a confirmação descrito na seção 8.2.4.

Figura 8.12 – Janela de Monitoração do Servidor de Certificação/Compensação
Certificação do Usuário *neto*

Breve descrição do processo de Monitoração da Certificação do Usuário *neto*:

- (1) **Mensagem de código CERT:** Enviada pela Carteira Eletrônica ao Servidor de Certificação/Compensação, com informações confidenciais do Usuário e sua Chave Pública, de acordo com o protocolo *PayWord* (ver Figura 3.2).
- (a) Corpo da mensagem criptografada com a Chave Privada do Usuário, como apresentado na seção 7.6.4, onde contém as informações confidenciais do Usuário, como o número da conta, o nome do usuário, a senha de acesso ao Servidor de Certificação/Compensação, a *String* binária de dados aleatórios e a assinatura digital

dessas informações confidenciais pelo Usuário com sua Chave Privada, conforme seção 7.6.2.

(b) Corpo da mensagem que contém a Chave Pública do Certificado Digital do Usuário.

(2) **Mensagem de código CERT:** Enviada pelo Servidor de Certificação/Compensação a Carteira Eletrônica, em resposta a Certificação do Usuário pelo *Broker*, com o Certificado Digital *PayWord* e sua Assinatura Digital, com previsto no protocolo *PayWord* (ver Figura 3.2).

(c) Corpo da mensagem com as informações do Certificado Digital *PayWord*, como código do *Broker*, data de vencimento, Chave Pública do Usuário e assinatura digital destas informações pelo *Broker* com sua Chave Privada, como apresentado na seção 7.6.2.

(d) Corpo da mensagem criptografada com a Chave Privada do *Broker*, conforme seção 7.6.4, que contém as mesmas informações enviadas pela Carteira Eletrônica, visto em (a).

(3) **Mensagem de Monitoração código MsgDeCripa:** Serve para ilustrar e testar o uso das rotinas de Criptografia e Certificados Digitais do Sistema Protótipo de Micropagamento.

(e) Corpo (a) da mensagem (1) que possui as informações confidenciais do Usuário, onde mostra o conteúdo do resultado da rotina de decryptografia pelo Servidor de Certificação/Compensação, conforme seção 7.6.4.

8.3 Servidor de Pagamentos

O Servidor de Pagamentos desenvolve no protocolo *PayWord* o processo de receber o Dinheiro Eletrônico (*PayWords*) da Carteira Eletrônica do Usuário, como forma de pagamento de produtos/informação de baixo valor monetário adquiridos do Vendedor no Sistema Protótipo de Micropagamento. O processo inclui, validar o Dinheiro Eletrônico, liberar o acesso à Carteira Eletrônica para solicitar o *download* do produto/informação adquirido pelo Usuário e, no final do dia ou do período, enviar ao Servidor de Certificação/Compensação as transações realizadas pelo Usuário junto ao Vendedor para obter o ressarcimento do valor monetário, transações essas correspondentes às vendas.

8.3.1 Componentes e Estrutura

Diagrama da visão geral do Servidor de Pagamentos no Vendedor:

Figura 8.13 – Componentes e Estrutura da Carteira Eletrônica

Breve descrição dos principais componentes do Servidor de Pagamentos:

- **SrvPagamentos.exe** – Executável do Servidor de Pagamentos, é um componente *server* e *client* desenvolvido na linguagem *MS-Visual Basic 6.0*; ele roda no ambiente do sistema operacional *Windows 2000 Server*; é ainda responsável por validar o Dinheiro Eletrônica enviado pela Carteira Eletrônica como forma de pagamento dos produtos/informações adquiridos pelo Usuário; responsável também por manter/registrar as operações de pagamento em Banco de Dados para ressarcimento das vendas junto ao *Broker* e interagir pelo protocolo *PayWord* com a Carteira Eletrônica e o Servidor de Certificação/Compensação;
- **Verificacao.dll e Hash.dll** – Rotinas de Interface em DLL que implementa as rotinas de Criptografia e Certificação Digital do protocolo *PayWord* (ver seção 7.6);
- **Banco de Dados** – Armazena os produtos ofertados pela Loja Virtual do Vendedor, mantém os compromissos de pagamento estabelecido pelos Usuários, bem como o histórico das operações de pagamento das compras realizadas por cada Usuário para posterior ressarcimento junto ao *Broker*. Atualmente está implementado em *MS-Access* e pode admitir qualquer SGDB que dá suporte à interface ODBC (ver seção 7.4).

8.3.2 Monitoração

O executável do Servidor de Pagamentos também deve ser mantido rodando no equipamento do Vendedor para atender à comunicação prevista no protocolo *PayWord* entre um Usuário e *Broker*. Uma janela de monitoração da comunicação foi desenvolvida, como mostra a Figura 8.14.

Figura 8.14 – Janela de Monitoração do Servidor de Pagamentos

Na janela de Monitoração de Pagamentos foram desenvolvidos botões para acionar as seguintes funções:

- **Compensação** – Enviar ao Servidor de Certificação/Compensação as transações realizadas pelo Usuário junto ao Vendedor para obter o ressarcimento do valor monetário, transações essas correspondentes às vendas.
- **Lista Vendas** – Listar as transações realizadas e compensadas dos Usuários/Clientes.

8.3.3 Compromisso e Pagamento

A Figura 8.15 apresenta a janela de Monitoração do Servidor de Pagamentos, por ocasião do estabelecimento do compromisso de pagamento e os pagamentos realizados junto ao Servidor de Pagamentos pela Carteira Eletrônica do Usuário *neto*.

Figura 8.15 – Janela de Monitoração de um Compromisso e Pagamentos do Usuário *neto* junto ao Servidor de Pagamentos

Breve descrição deste processo de Monitoração do Compromisso e Pagamentos do Usuário *neto*:

- (1) **Mensagem de código CPAG:** Enviada pela Carteira Eletrônica ao Servidor de Pagamentos, para estabelecer o compromisso de pagamento entre o Usuário e o Vendedor, como previsto no protocolo *PayWord* (ver Figura 3.3).

- (a) Corpo da mensagem que contém o código da Loja Virtual.
 - (b) Corpo da mensagem com as informações do Certificado Digital *PayWord*, como código do *Broker*, data de vencimento, Chave Pública do Usuário e assinatura digital destas informações pelo *Broker* com sua Chave Privada, conforme seção 7.6.2.
 - (c) Corpo da mensagem contendo o *PayWord* w_0 , ou seja, a raiz da série encadeada de *PayWords*.
 - (d) Corpo da mensagem com a data de vencimento do compromisso de pagamento e a assinatura digital das informações dessa mensagem de compromisso de pagamento pelo Usuário com sua Chave Privada, como apresentado na seção 7.6.2..
- (2) **Mensagem de código CPAG Ok:** Enviada pelo Servidor de Pagamentos a Carteira Eletrônica do Usuário, como resposta de aceitação do compromisso de pagamento. Esta mensagem foi implementada no Sistema Protótipo de Micropagamento; contudo, não está prevista no protocolo *PayWord* (ver Figura 3.3).
- (e) Corpo da mensagem com o código do compromisso de pagamento estabelecido com o Vendedor. Será enviada pela Carteira Eletrônica por ocasião da mensagem de pagamento (3).
- (3) **Mensagem de código PAGA:** Enviada pela Carteira Eletrônica ao Servidor de Pagamentos, como forma de pagamento de um produto/informação adquirido pelo Usuário junto ao Vendedor, como previsto no protocolo *PayWord* (ver Figura 3.4).
- (f) Corpo da mensagem que contém o código do compromisso de pagamento estabelecido com o Vendedor, o próximo *PayWord* da série encadeada e a quantidade de *PayWords* que estão sendo utilizados como Dinheiro Eletrônico. No Sistema Protótipo de Micropagamento, cada *PayWord* tem o valor monetário de R\$ 0,01 (Um centavo de real).
- (4) **Mensagem de código PAGA Ok:** Enviada pelo Servidor de Pagamentos a Carteira Eletrônica do Usuário, como resposta de aceitação do pagamento efetuado pelo Usuário junto ao Vendedor. Esta mensagem foi implementada no Sistema Protótipo de Micropagamento; entretanto, não está prevista no protocolo *PayWord* (ver Figura 3.4).
- (g) Corpo da mensagem contendo um código aleatório de acesso ao produto/informações que o Usuário adquiriu do Vendedor. Utilizado para solicitar junto ao Servidor de Páginas Web o *download* do arquivo que contém a compra solicitada pela Carteira Eletrônica do Usuário.

8.3.4 Compensação

Para obter o ressarcimento do valor monetário correspondente a vendas, o Vendedor, no final do dia ou do período, necessita enviar ao Servidor de Certificação/ Compensação no *Broker*, as mensagens de compromisso de pagamento de cada Usuário do *Broker* que estabeleceram o compromisso de pagamento e o último *PayWord* utilizado como forma de pagamento dos produtos/informações adquiridos pelo Usuário.

Pelo botão *Compensação* na janela de Monitoração do Servidor de Pagamentos (ver Figura 8.14) pode ser acionado o processo de ressarcimento das vendas realizadas, conforme o protocolo *PayWord* (ver seção 3.2.3). Em seguida, o Servidor de Pagamentos solicita a confirmação de tal função, como mostra a Figura 8.16

Figura 8.16 – Janela de Confirmação da Compensação no Servidor de Pagamentos

Após a troca de mensagens entre o Servidor de Pagamentos e o Servidor de Certificação/ Compensação, a janela conforme mostra a Figura 8.17, informa que a Compensação das Vendas foi concluída com sucesso.

Figura 8.17 – Janela de Conclusão da Compensação no Servidor de Pagamentos

8.3.5 Lista das Vendas

A Figura 8.18 mostra a janela das vendas realizadas pelo Vendedor aos Usuários do Sistema Protótipo de Micropagamento. São sumarizados por compromisso de pagamento os valores das vendas, a data de vencimento do compromisso, o último *PayWord* utilizado como Dinheiro Eletrônico, o código do *Broker*, bem como se o compromisso de pagamento foi ou não compensado pelo Servidor de Certificação/Compensação junto ao *Broker*. Essa janela pode ser acionada pelo botão *Lista Vendas* na janela de Monitoração do Servidor de Pagamentos.

Figura 8.18 – Janela da Lista das Vendas realizadas pela Loja Virtual no Servidor de Pagamentos

8.4 Loja Virtual

O Servidor de Páginas *Web* desenvolve uma importante função na Arquitetura do Sistema Protótipo de Micropagamento. Com efeito, trata-se do processo de publicar na Internet a Loja Virtual do Vendedor, através de páginas *Web*, as quais ofertam produtos/informações aos Usuários/Clientes, de acionar a Carteira Eletrônica do Usuário na compra e pagamento pelo protocolo *PayWord* e, finalmente, de controlar o acesso e de prover o *download* do arquivo contendo o produto/informação adquirida para o equipamento do Usuário.

8.4.1 Componentes e Estrutura

Diagrama da visão geral da Loja Virtual:

Figura 8.19 – Componentes e Estrutura da Loja Virtual

Breve descrição dos principais componentes da Loja Virtual:

- **Servidor de Página Web (MS-Internet Information Server)** – Pacote de software composto por diversos componentes para prover suporte completo à publicação pela Internet, incluindo serviços de transporte, suporte às aplicações *clients*, ferramentas de administração, conectividade às aplicações e bancos de dados e também comunicação com criptografia [27];
- **Páginas Web** – Desenvolvidas em páginas ASP (*Active Server Pages*) [23], um ambiente de programação por *scripts* que rodam no Servidor de Páginas Web. Páginas ASP podem ser usadas para criar páginas dinâmicas, interativas de alta performance. O Servidor de Páginas Web transforma os *scripts* em HTML padrão, fazendo com que qualquer *browser* seja capaz de consultar uma página que possua código ASP. A implementação dos *scripts* pode ser feita usando *VBScript (MS-Visual Basic Script)* ou *JScript (MS-Java Script)*. Para construir as Páginas Web em ASP da Loja Virtual do Sistema Protótipo de Micropagamento foi utilizada a ferramenta *MS-Visual InterDev 6.0*;
- **Banco de Dados** – Armazena os produtos/informações ofertados pela Loja Virtual do Vendedor aos Usuários/Clientes, assim como, o histórico das operações das compras realizadas por cada Usuário. Os produtos são publicados através de páginas ASP que são consultadas pelo *browser* do Usuário. Atualmente está implementado em *MS-Access* e pode admitir qualquer SGDB que dá suporte à interface ODBC (ver seção 7.4);
- **Arquivos para Venda** – Cada arquivo contém um produto ofertado pela Loja Virtual do Vendedor aos Usuários/Clientes. O Usuário opta pela compra de um dos produtos/informações ofertado; uma vez realizado o pagamento pelo Sistema Protótipo de Micropagamento pela Carteira Eletrônica, esta solicita ao Servidor de Páginas Web o *download* do correspondente arquivo para o *browser*.

8.4.2 Home Page

A Figura 8.20 mostra a *Home Page* da Loja Virtual do Vendedor, implementada para o Sistema Protótipo de Micropagamento. No ambiente de simulação da Internet para o Sistema Protótipo, essa página pode ser consultada pelo *browser*; basta apenas que o Usuário informe o endereço da Loja Virtual.

Figura 8.20 – Página *Home Page* da Loja Virtual

Breve descrição dos itens do menu (*links*) da *Home Page* da Loja Virtual.

- **Lista de produtos** – *Link* para a página que contém a lista dos produtos ofertados pela Loja Virtual;
- **Produtos adquiridos** – *Link* para a página que mostra uma tabela com os produtos adquiridos pelo Usuário na última hora. Cada produto nessa página está associado a um *link* com o qual, durante o período de uma hora desde sua compra, o Usuário poderá consultar o arquivo que contém o produto/informações para um novo *download*, que, eventualmente, não tenha sido antes realizado com sucesso;
- **Download da Carteira Eletrônica** – *Link* para o arquivo que contém um arquivo compactado com o pacote de componentes da Carteira Eletrônica, que os Usuários podem obter por *download*.

8.4.3 Lista de Produtos

A Figura 8.21 apresenta a página da Loja Virtual do Vendedor que expõe a lista de produtos ofertados aos Usuários/Clientes.

Figura 8.21 – Página Lista de Produtos da Loja Virtual

Um Usuário ao optar por uma das categorias dos produtos ofertados, como por exemplo, *Artigos*, o Servidor de Páginas Web consulta a Tabela de *Produtos* do seu Banco de Dados e seleciona os produtos daquela categoria. Monta dinamicamente a página, conforme demonstra a Figura 8.22, com os detalhes de cada produto, como a descrição do produto, o preço e, também, a opção de “Comprar” que é um botão que, se acionado, ativa a Carteira Eletrônica para autenticar o Usuário e proceder ao pagamento pelo protocolo *PayWord*.

Figura 8.22 – Página Detalhe dos Produtos Artigos da Loja Virtual

8.4.4 Compra

Para adquirir produtos/informações da Loja Virtual do Vendedor, o Usuário do Sistema Protótipo de Micropagamento deve inicialmente abrir sua conta junto ao *Broker* e obter a senha de acesso ao Servidor de Certificação/Compensação (ver seção 8.1.3), instalar o pacote de componentes da Carteira Eletrônica em seu equipamento e abrir sua conta na Carteira Eletrônica (ver seção 8.2.3), procedendo à certificação da nova conta junto ao *Broker* (ver seção 8.2.4).

Usando o *browser* e navegando em uma das páginas de produtos publicados pelo Servidor Páginas Web da Loja Virtual, como por exemplo, na página mostrada na Figura 8.22, o Usuário pode optar pela compra de um item. Para tanto, basta que o Usuário dê um *click* no botão *Comprar* da linha do produto que deseja. Em seguida, o *browser* aciona a Carteira Eletrônica, que abre sobre a página dos produtos publicados a janela de *Login* do Usuário, como mostra a Figura 8.23.

Para guardar segurança ao acesso à Carteira Eletrônica, o *Login* do Usuário será solicitado na primeira opção de compra, bem como depois de um certo período de tempo, como por exemplo, três minutos sem atividade de compra pelo Usuário.

Figura 8.23 – Página com a janela *Login* do Usuário da Carteira Eletrônica na ocasião da Compra

8.4.5 Pagamento

A Carteira Eletrônica, estando “instanciada” pelo *browser* e devidamente autenticada com o *Nome* e *Senha* do Usuário, como descrita na seção 8.4.4, toda vez que o Usuário optar pela compra de um produto, ela abrirá, sobre a página dos produtos publicados, a janela de *Confirmar Pagamento*, conforme Figura 8.24; bastará, então, um *Click* no botão *Confirmar* pelo Usuário para que a Carteira Eletrônica efetue o pagamento da compra pelo protocolo *PayWord* ao Vendedor, de acordo com o que foi exposto na seção 3.2.2.

A Figura 8.24 mostra a janela de *Confirmar Pagamento* que identifica o Usuário da Carteira Eletrônica pelo *Nome* e *Conta*. Tal janela possui os botões de *Confirmar* ou *Cancelar* o pagamento da compra, bem como o botão *Outra Carteira* para efetuar eventualmente o *Login* de novo Usuário na Carteira Eletrônica.

Figura 8.24 – Página com a janela *Confirmar Pagamento*

8.4.6 Download da Compra

Após a confirmação do pagamento pelo Usuário junto ao Vendedor, descrito no item anterior, o Servidor de Pagamentos envia para a Carteira Eletrônica do Usuário uma chave aleatória para solicitar acesso ao arquivo que contém o produto/informações adquirido (ver seção 8.3.3).

Em seguida, a Carteira Eletrônica solicita o *download* ao Servidor de Páginas Web, acionando a página ASP da Loja Virtual chamada *arquivo.asp*, que associa em sua URL, o parâmetro da chave aleatório de acesso de tal arquivo, como por exemplo, <http://localhost/LojaVirtual/arquivo.asp?Carteira=1&Chave=E2008A6136EB30991333F877BF0D7888>.

Em tal página roda um *script* no Servidor de Páginas Web; quando acionado, efetua uma seleção pela chave aleatória de acesso na Tabela de *Compras* de seu Banco de Dados, buscando a linha que possui o correspondente nome do arquivo com o produto/informações adquirido. Em seguida, o arquivo é lido pelo Servidor de Páginas Web e enviado ao equipamento do Usuário efetuando o *download*, como se pode ver no exemplo da Figura 8.25.

Figura 8.25 – Página com a janela de *Download* do *arquivo.zip* contendo a Compra

8.4.7 Produtos Adquiridos

A página de *Produtos adquiridos* da Loja Virtual do Vendedor mostra todas as compras realizadas nos últimos sessenta minutos feitas pelo Usuário. Por essa página, o Usuário pode solicitar o *download* dos arquivos que contêm os produtos/informações adquiridos dentro desse período. O objetivo é auxiliar o Usuário a solicitar novo *download* de um arquivo que não tenha sido eventualmente concluído com sucesso pelos procedimentos descritos na seção 8.4.6.

A Figura 8.26 apresenta um exemplo de página de *Download* com os produtos que foram adquiridos pelo Usuário *neto*. Tal página foi consultada pelo item *Produtos adquiridos* do menu (*links*) da *Home Page* da Loja Virtual do Vendedor. Nesse exemplo, para efetuar *download* de um dos arquivos que contém o produto adquirido, é suficiente que o Usuário dê um *Click* sobre a palavra *Download* na coluna à direita da descrição do produto desejado e os procedimentos de *download* seriam realizados.

Figura 8.26 – Página com os Produtos Adquiridos para *Download* no período de 1 hora após a Compra

8.4.8 Registro de Operações

A qualquer momento, a Carteira Eletrônica do Usuário pode mostrar o registro das operações das compras realizadas, conforme mostra a Figura 8.27.

Para essa operação, o Usuário deve “instanciar” sua Carteira Eletrônica em seu equipamento, de acordo com o descrito na seção 8.2.2; ou seja, pelo botão *Start* → *Programs* → *Carteira Eletrônica* no *Windows*; depois, efetuar seu *Login*, informando *Nome* e *Senha* na janela, (ver Figura 8.9) e, em seguida, a Carteira Eletrônica mostra a janela a seguir.

Pelo botão *Abrir Downloads* na janela de Registros de Operações da Carteira Eletrônica, o Usuário pode acionar o *browser* e também abrir a página de *Produtos Adquiridos*, como mostra a Figura 8.26.

Figura 8.27 – Janela de *Registro de Operações* da Carteira Eletrônica do Usuário *neto*

8.4.9 Download da Carteira Eletrônica

Já mencionado na seção 8.4.2, este item *Download da Carteira Eletrônica* do menu (*links*) da *Home Page* da Loja Virtual do Vendedor pode permitir ao Usuário/Cliente do Sistema Protótipo de Micropagamento efetuar o *download* do pacote de componentes da Carteira Eletrônica do Servidor de Páginas *Web* para o equipamento do Usuário e, em seguida, permite proceder a sua instalação e uso, como descrito na seção 8.2.3.

A Figura 8.28 traz um exemplo da janela de interação com um Usuário, cujo *download* foi solicitado e que iniciará a cópia do arquivo compactado *CarteiraEletronica.zip*, que contém o pacote de componentes da Carteira Eletrônica. Após a conclusão, com sucesso, do *download* do arquivo para o equipamento do Usuário, este deve descompactá-lo e executar o procedimento de Instalação da Carteira Eletrônica em seu equipamento. Para isso, basta que o Usuário execute o programa *Setup.exe* que consta no diretório onde foram descompactados os componentes da Carteira Eletrônica e tal programa guiará o Usuário nos procedimentos necessários.

**Figura 8.28 – Janela de Download da Carteira Eletrônica
Acionado pelo menu da Home Page da Loja Virtual do Vendedor**

Capítulo 9

Conclusões e Perspectivas

Este capítulo é destinado às considerações finais sobre o que foi realizado nesta dissertação e os desafios que foram surgindo no decorrer no seu desenvolvimento. Destina-se, da mesma forma, a comentar as conclusões obtidas com os estudos e análises e a traçar perspectivas para trabalhos futuros.

O principal objetivo desta dissertação, de acordo com o exposto na seção 1.1, foi apresentar uma análise de esquemas de Micropagamento Eletrônico e o emprego de técnicas de criptografia, tendo como meta implementar uma solução que atendesse aos requisitos e às características aplicáveis ao comércio eletrônico nas transações de pequeno valor monetário, que contemplasse a venda, o pagamento e a entrega de produtos/informações realizadas via Internet.

Com esse ponto de partida, levou-se a efeito uma abrangente pesquisa bibliográfica do assunto Micropagamento Eletrônico. Dela resultou nas referências bibliográficas que fazem parte integrante desta dissertação. Resultou igualmente na apresentação de uma visão geral dos Sistemas de Pagamento Eletrônico, conforme mostrou o capítulo 2, com os seus requisitos e propriedades desejáveis, seus critérios de classificação dos Sistemas de Pagamento Eletrônico, as diferenças entre Macro e Micropagamento e alguns protocolos de segurança e pagamento.

Os esquemas de Micropagamento *PayWord* [1], *MicroMint* [1] e *MilliCent* [4], consultados na literatura pesquisada, foram respectivamente analisados em detalhes nos capítulos 3, 4, e 5. Cada capítulo apresentou uma visão geral desses três diferentes protocolos, seguida de uma descrição pormenorizada das partes envolvidas no esquema. Da mesma forma, apresentou todas as fases que constituem um processo de Micropagamento de cada um desses três esquemas, quais sejam: a estrutura do dinheiro eletrônico, as técnicas de criptografia empregadas, a geração ou produção de moeda eletrônica, as compras e seus pagamentos, o processo de ressarcimento do dinheiro eletrônico e uma avaliação de segurança envolvendo prevenção de falsificação, detecção de gasto duplicado e outros comentários sobre cada esquema. Com a análise desses esquemas de Micropagamento pode-se concluir que é possível e exequível chegar ao objetivo proposto que é construir uma solução para

atender ao comércio eletrônico de transações de pequeno valor monetário, que contemple a venda, o pagamento e a entrega de produtos/informações, usando uma rede de comunicação de dados como a Internet.

O capítulo 6, por seu lado, expôs uma análise comparativa dos três esquemas de Micropagamento estudados, resumindo suas similaridades, diferenças e propriedades. Os estudos realizados nesse capítulo, em conjunto com os que foram feitos nos capítulos anteriores, contribuíram no apoio à decisão de utilizar o protocolo *PayWord* na implementação do Sistema Protótipo de Micropagamento. Tal implementação visou consolidar os pontos de vista contidos nesta dissertação, no que diz respeito tanto ao assunto Micropagamento quanto às técnicas de criptografia empregadas.

O grande desafio deste estudo foi definir uma arquitetura e a posterior implementação de uma solução para um Sistema Protótipo de Micropagamento que demonstrasse, através de uma aplicação de Loja Virtual na Internet, um esquema a um só tempo prático e exeqüível de uso dos mecanismos de Micropagamento Eletrônico.

Em decorrência disso, o capítulo 7 foi desenvolvido para apresentar as principais características da Arquitetura para implementar um Sistema Protótipo de Micropagamento que foi desenvolvido utilizando-se o protocolo *PayWord*. O capítulo 8, enfim, detalhou a Implementação dos componentes do Sistema Protótipo de Micropagamento utilizando-se a Arquitetura apresentada no capítulo 7.

Com isto, pode-se concluir que é possível implementar uma solução completa para atender aos requisitos e às características de uma aplicação de comércio eletrônico para transações de pequeno valor monetário, que contemplasse a venda, o pagamento e a entrega de produtos/informações via Internet.

Muitas dificuldades, contudo, foram vencidas na definição da Arquitetura e Implementação do Sistema Protótipo de Micropagamento. Para vencê-las, cabe destacar a utilização de algoritmos, de rotinas e padrões, como a função *Hash* pelo algoritmo SHA [3], o algoritmo de criptografia assimétrica *RSA* [26], a montagem de uma Infra-Estrutura de Chave Pública/Privada para a geração de Certificados Digitais X.509 [20], o armazenamento seguro de certificados digitais para proteção da chave privada no padrão *PKCS#12 – Personal Information Exchange Syntax Standard* [21], a geração de números aleatórios pelo algoritmo *Blum-Blum-Shub (BBS)* [24], o desenho e a implementação de uma Carteira Eletrônica como extensão de um navegador (*browser*), a definição e implementação de um canal seguro para a Certificação do Usuário junto ao *Broker*, como também, a definição e implementação das mensagens de respostas complementares no processo de compra, cujas definições não estavam previstas no protocolo *PayWord*.

Quanto às perspectivas em relação ao futuro, existe ainda muito por fazer para construir e implementar sistemas de Micropagamento Eletrônico que atendam aos requisitos do ponto de vista tecnológico, à facilidade de uso e à viabilidade econômica frente a um cenário de comércio eletrônico para transações de pequeno valor monetário.

Do ponto de vista tecnológico, é importante destacar que uma ampla utilização e grande interesse dos usuários implicará na dependência de uma rede de comunicação com disponibilidade e servidores de alto desempenho para admitir a carga das operações. Do mesmo modo, a implementação de sistemas para uma solução de Micropagamento Eletrônico deverá ser otimizada, de forma a possuir performance e eficácia no atendimento às transações de pagamento.

O estudo apresentado permite definir a facilidade de uso como uma característica prioritária na construção de tais sistemas, propiciando uma perfeita familiaridade do Usuário com sua interface. Não só facilidade, como também simplificação e portabilidade para diferentes equipamentos dos mecanismos de identificação e autenticação do Usuário. Para tanto, a construção de uma solução de Carteira Eletrônica deve ser aprimorada buscando um equilíbrio para tornar de fácil uso estes mecanismos, assim como garantir a segurança na produção e controle da Moeda Eletrônica.

Uma aplicação de comércio eletrônico para transações de pequeno valor monetário terá sem dúvida sua viabilidade econômica, quando houver um equilíbrio entre os investimentos e o retorno financeiro das operações realizadas. Para que isso aconteça, há de existir de um lado uma grande base de comerciantes nesse mercado, oferecendo atrativos, produtos/informações, e, do outro lado, uma base de usuários que estejam habilitados a fazer compras de tais produtos/informações e dispostos a efetuar pagamentos por um sistema de Micropagamento Eletrônico. Da mesma forma, uma terceira parte neste cenário, que são os *Brokers* ou Instituições Financeiras, deverá facilitar a relação de confiança entre os Usuários e os Comerciantes.

Os trabalhos aqui expostos levam também em consideração que as Instituições Financeiras, como Bancos e Empresas de Cartão de Crédito, poderão ser os grandes impulsionadores e facilitadores de um processo de massificação de uso de soluções de Micropagamento Eletrônico, visto possuírem uma significativa base de clientes que são usuários de soluções de *Internet Banking*, bem como um conjunto de clientes pessoas jurídicas, que são os comerciantes.

Por fim, entende-se ainda que os mecanismos apresentados nesta dissertação sobre Micropagamento Eletrônico e sobre o emprego de técnicas de criptografia para aplicações de comércio eletrônico com transações de pequeno valor monetário não se restringem aos negócios via Internet; eles poderão ser adaptados e serem empregados em outras soluções tecnológicas e com diversos outros equipamentos eletrônicos, como por exemplo, o telefone celular, o PAD (*Personal Digital Assistant*) e também em implementações usando *smart cards*, como *tickets* de pagamento.

Referências Bibliográficas

- [1] *Ronald L. Rivest e Adi Shamir, PayWord and MicroMint: Two simple micropayment schemes. May 7, 1996.*
<http://theory.lcs.mit.edu/~rivest/RivestShamir-mpay.ps>
- [2] *Ronald L. Rivest, The MD5 message-digest algorithm. Internet Request for Comments, April 1992. RFC 1321.*
- [3] *National Institute of Standards and Technology (NIST) FIPS Publication 180, Secure Hash Standard (SHS), May 11, 1993.*
- [4] *S. Glassman, M. Manasse, M. Abadi, P. Gauthier e P. Sobalvarro, The MilliCent Protocol for Inexpensive Electronic Commerce, 1995.*
<http://www.MilliCent.digital.com/works/details/papers/MilliCent-w3c4/MilliCent.html>
- [5] *X. Lai, On the Design and Security of Block Ciphers, Institute for Signal and Information Processing, ETH-Zentrum, Zurich, Switzerland, 1992.*
- [6] *J. Daemen e V. Rijmen, AES Proposal: Rijndael, First Advanced Encryption Standard (AES) Conference, Ventura, CA, 1998.*
- [7] *T. Berners-Lee, R. Fielding e H. Frystyknielsen, Hypertext Transfer Protocol - HTTP/1.0 - RFC 1945, May 1996.*
<http://www.ietf.org/rfc/rfc1945.txt>
- [8] *A. J. Menezes, P. C. Van Oorschot e S. A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1997.*
- [9] *B. Schneier, Applied Cryptography – Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Second Edition, 1996.*
- [10] *Routo Terada, Segurança de Dados – Criptografia em Redes de Computadores. Editora Edgard Blücher Ltda, 1ª edição – 2000.*
- [11] *Netscape Communications Corporation, Introduction to SSL, 1998.*
<http://devedge.netscape.com/docs/manual/security/ssl/index.htm>

- [12] *T. Dierks e C. Allen, The TLS protocol version 1.0, RFC 2246, January 1999.*
<http://www.ietf.org/rfc/rfc2246.txt>
- [13] **SET Secure Electronic Transaction LLC. SET specification books.**
http://www.setco.org/set_specifications.html
- [14] **Índice de los Códigos Fuente (V 2.0).**
http://tirnanog.ls.fi.upm.es/Servicios/Software/ap_crypt/indice.html
- [15] *T. Berners-Lee, HyperText Markup Language Specification - 2.0, RFC1866, November 1995.*
<http://www.w3.org/MarkUp/html-spec/>
- [16] *C. M. Sperberg-McQueen e L. Burnard, A Gentle Introduction to SGML.*
<http://www-tei.uic.edu/orgs/tei/sgml/teip3sg/index.html>
- [17] *B. Quinn e D. Shute, Windows Sockets Network Programming, Addison Wesley, November 1995.* <http://www.sockets.com>
- [18] *J. Postel, Transmission Control Protocol, RFC793, September 1981.*
<http://www.rfc-editor.org/rfc/rfc793.txt>
- [19] *D. Chappell, Understanding ActiveX and OLE. Microsoft Press, 1996.*
- [20] **Public-Key Infrastructure (X.509) (pkix).**
<http://www.ietf.org/html.charters/pkix-charter.html>
- [21] *RSA Laboratories, PKCS #12 - Personal Information Exchange Syntax Standard - Version 1.0. June, 1999.*
<http://www.rsasecurity.com/rsalabs/pkcs/pkcs-12/index.html>
- [22] *Baltimore Technologies – Web Site.*
<http://www.baltimore.com/index.html>
- [23] *Microsoft, The ABCs of Active Page Server.*
<http://msdn.microsoft.com/workshop/server/asp/ASPover.asp>
- [24] *L. Blum, M. Blum e M. Shud, A Simple Unpredictable Pseudo-Random Number Generator. SIAM Journal on Computing, volume 15, pages 364-383, May 1986.*
- [25] *R. T. Fielding, Relative Uniform Resource Locators, RFC1808, June 1995.*
<http://www1.ics.uci.edu/pub/ietf/uri/rfc1808.txt>

- [26] *RSA Laboratories, Frequently Asked Questions about Today's Cryptography. Version 4.1, May, 2000.*
<http://www.rsasecurity.com/rsalabs/faq/>
-
- [27] *K. Bergmann, A High-Level Look at Microsoft Internet Information Server. Microsoft Developer Network Technology Group, November 1995.*
- [28] *Internet Software Consortium, Internet domain survey, January 2000.*
<http://www.isc.org/www-20001/report.html>
- [29] *Amazon.com, Inc – Amazon Web Site.*
<http://www.amazon.com>
- [30] *J. Burstein, An Implementation of MicroMint. MIT EECS, Master's Thesis, May 1998.*
- [31] *W3C, Common Markup for Micropayment per-fee-links. W3C Working Draft, August 1999.*
<http://www.w3.org/TR/MicroPayment-Markup/>
- [32] *IBM, IBM Micro Payments – Charge per click on the Web. October 1999.*
<http://www.hrl.il.ibm.com/mpay/>
- [33] *W3C, Micropayments Overview. August 1999.*
<http://www.w3.org/ECommerce/Micropayments/Overview.html>
- [34] *A. Herzberg, Safeguarding Digital Library Contents – Charging for Online Content. D-Lib Magazine, January 1998.*
<http://www.dlib.org/dlib/january98/ibm/01herzberg.html>
- [35] *University of Southern California, Information Sciences Institute, The NetCheque (SM) Network Payment System.*
<http://www.netcheque.org>
- [36] *W. Stallings, Cryptography and Network Security: Principle and Practice. Second Edition, Prentice-Hall, New Jersey, 1999.*
- [37] *USC/ISI's GOST Group, The Kerberos network authentication service.*
<http://www.isi.edu/gost/info/kerberos>
- [38] *UN/EDIFACT Working Group, United Nation Directories for EDI, 1995.*
<http://www.unece.org/trade/untdid>

- [39] *D. Eastlake, B. Boesch, S. Crocker e M. Yesil, CyberCash credit card protocol version 0.8. Internet Draft, July 1995.*
<ftp://ds.internic.net/internet-drafts/draft-eastlake-cybercash-v08-00.txt>
-
- [40] *CyberCash, Inc – CyberCash Web Site.*
<http://www.cybercash.com>
- [41] *T. Jones, Mondex on ‘the future of money’. Submission to the US House of Representatives, 1996.*
<http://www.mondex.com/hserep.htm>
- [42] *G. Medvinsky e B. C. Neuman, Netcash: A design for practical electronic currency on the internet. In Proceedings of 1st the ACM Conference on Computer and Communication Security, November 1993.*