Relatório de Estudo:
Criptografia – Micro-Pagamento Eletrônico

Aluno: **João Carlos Néto**

Orientador: **Prof. Routo Terada**

MAC5701 – TÓPICOS EM CIÊNCIA DA COMPUTAÇÃO

Profa. Yoshiko Wakabayashi

USP/IME

1o. Semestre de 2.001

Resumo

Neste relatório são analisados os esquemas de Micro-Pagamento: *PayWord*, *MicroMint* e *MilliCent*. Inicialmente, o relatório descreve uma visão geral sobre Sistemas de Pagamento Eletrônico, suas propriedades desejáveis, as diferenças entre Macro e Micro-Pagamento, os protocolos de segurança e pagamento. É analisado cada um dos referidos esquemas de Micro-Pagamento, apresentando uma visão geral do protocolo, a descrição detalhada do protocolo 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. Finalmente, o relatório apresenta uma comparação dos esquemas de Micro-Pagamento analisados, sumarizando suas similaridades, diferenças e propriedades.

Conteúdo

1. Introdução 1

2. Sistemas de Pagamento Eletrônico 2

2.1 Visão Geral 2

2.2 Requisitos para Sistemas de Pagamento Eletrônico 3

2.2.1 Atomicidade 3

2.2.2 Segurança 3

2.2.3 Privacidade 3

2.2.4 Escalabilidade 4

2.2.5 Interoperabilidade 4

2.3 Classificação dos Sistemas de Pagamento Eletrônico 4

2.3.1 Operação *On-line* versus *Off-line* 4

2.3.2 Macro-Pagamento versus Micro-Pagamento 4

2.3.3 Solução por Hardware versus Software 5

2.4 Protocolos de Segurança e Pagamento 5

2.4.1 Ataques à Segurança de Computadores 5

2.4.2 Requisitos para um Sistema de Pagamento Eletrônico Seguro 6

2.5 Visão Geral dos Sistemas de Pagamento Eletrônico 7

2.5.1 Sistemas Tradicionais de Pagamento Baseados na Internet 7

2.5.2 Dinheiro Eletrônico 10

3. *PayWord* 11

3.1 Visão Geral 12

3.2 Detalhes do *PayWord* 13

3.2.1 Certificação do Usuário pelo *Broker* 13

3.2.2 Compras 13

3.1.3 Compensação dos Pagamentos pelo *Broker* ao Vendedor 16
3.2 Avaliação de Segurança 16
3.2.1 Prevenção de Falsificação 16
3.2.2 Detecção de Gasto Duplicado 17
3.2.3 Outros Comentários 17

4. *MicroMint* 19
4.1 Visão Geral 20
4.2 Detalhes do *MicroMint* 21
4.2.1 Produção de Moedas Eletrônicas 21
4.2.2 Compra de Moedas Eletrônicas Específicas por Usuário 23
4.2.3 Compras 24
4.2.4 Compensação dos Pagamentos pelo *Broker* ao Vendedor 24
4.3 Avaliação 25
4.3.1 Prevenção de Falsificação 25
4.3.2 Detecção de Gasto Duplicado 26
4.3.3 Outros Comentários 26

5. *MilliCent* 27
5.1 Visão Geral 27
5.2 Detalhes do *MilliCent* 29
5.2.1 Estrutura da Moeda Eletrônica *Scrip* 29
5.2.2 *Master Scrip Secret* e Certificação do *Scrip* 30
5.2.3 Validação do *Scrip* 31
5.2.4 *Customer Secret* e *Master Customer Secret* do *Scrip* 32
5.2.5 Protocolos do *MilliCent* 33
5.3 Interações entre *Broker*, Vendedor e Consumidor 35
5.3.1 Compra de *scripVendor* pelo *Broker* 35
5.3.2 Compra de *scripBroker* pelo Usuário 37
5.3.3 Compra de *scripVendor* pelo Usuário 37
5.3.4 Pagamento com *scripVendor* pelo Usuário ao Vendedor 37
5.4 Avaliação 38

5.4.1 Prevenção de Falsificação 38
5.4.2 Detecção de Gasto Duplicado 38
5.4.3 Outros Comentários 38

6. Comparação 39
6.1 *PayWord* versus *MicroMint* 39
6.2 *MilliCent* versus *MicroMint* 40
6.3 Comparação de Propriedades 42

7. Conclusões Finais 43

Referências Bibliográficas 44

Capítulo 1

Introdução

Neste relatório são analisados os seguintes esquemas de Micro-Pagamento: *PayWord* [1], *MicroMint* [1] e *MilliCent* [4].

Inicialmente, o relatório descreve uma visão geral sobre Sistemas de Pagamento Eletrônico, suas propriedades desejáveis, as diferenças entre Macro e Micro-Pagamento, os protocolos de segurança e pagamento.

Em seguida, é analisado cada um dos referidos esquemas de Micro-Pagamento, apresentando uma visão geral do protocolo, a descrição detalhada do protocolo 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.

Finalmente, o relatório apresenta uma comparação dos esquemas de Micro-Pagamento analisados, sumarizando suas similaridades, diferenças e propriedades.

Capítulo 2

Sistemas de Pagamento Eletrônico

2.1 Visão Geral

O comércio eletrônico começou a mais de vinte anos atrás, com o estabelecimento do padrão de troca eletrônica de documentos (EDI – *Electronic Data Interchange*) entre empresas, como ordens de compra, faturas, entregas e outros. EDI é baseado em um conjunto 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. Estes 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 (WWW – *World Wide Web*).

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

O primeiro passo para ganhar dinheiro neste 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 da WWW, as empresas estabeleceram lojas virtuais na *Web* vendendo produtos diretamente aos consumidores. Desta forma, os consumidores puderam encontrar os menores preços de produtos e serviços. Contudo, nesta época não havia tecnologia para os pagamentos pela Internet.

A primeira abordagem para este problema foi atendida com o envio do número do cartão de crédito do consumidor para o vendedor, através de formulários eletrônicos pela Internet, sem

uso de técnicas de criptografia. Para transações de valores altos isto é muito inseguro. 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 compensaria as taxas envolvidas com cartão de crédito.

Assim, foi 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 seguintes propriedades:

2.2.1 Atomicidade

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

2.2.2 Segurança

As transações devem ser seguras e não permitir abuso do 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 moedas é difícil, visto que são necessários equipamentos especiais e expertise. Por outro lado, dinheiro de circulação eletrônico é apenas dados binários que podem ser copiados facilmente. A cópia ou a reapresentação do dinheiro em circulação eletrônica deve ser prevenida e deve existir mecanismo de detecção. O ideal é que o dinheiro eletrônico seja incondicionalmente ou computacionalmente impossível de ser criado ilegalmente, copiado ou reutilizado. As transações de pagamento eletrônicas devem ser seguras contra interceptação, modificação ou fabricação.

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 deve ser revelada. Os usuários desejam que os Sistemas de Pagamento Eletrônico tenham o mesmo grau de privacidade que é provido quando se utiliza o dinheiro em espécie. Um compra feita

[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.

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 esta no seu caixa. Nem o banco sabe o que o cliente compra com o dinheiro que ele sacou. Alguns esquemas de pagamento ignoram a questão da privacidade e permite rastrear os pagamentos dos usuários. Outros protolocos são incondicionalmente não rastreáveis, onde os gastos de um indivíduo não podem ser determinados mesmos se as partes estejam em conluio. Para algumas transações, pode existir uma forma mais fraca de privacidade, de forma que o rastreamento seja dificultado o suficiente de maneira que o custo de obter informação seja maior que o benefício alcançado.

2.2.4 Escalabilidade

Os Sistemas de Pagamento Eletrônico devem possuir mecanismos para admitir o crescimento do número de usuários e sobre carga de acessos em certos períodos de pico sem impacto na performance 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 processar as transações dos usuários. Contudo, nem sempre os usuários são clientes do mesmo ambiente de servidores onde está hospedado o Sistema de Pagamento Eletrônico. Decorre disto, que o dinheiro de circulação eletrônica, que foi produzido para um certo sistema, deverá ser aceito e conferido mundialmente. Sem a aceitação mútua, este 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*, quanto participa da operação um servidor para autenticação e autorização em cada pagamento, ou *off-line*, quanto não há participação de uma terceira parte durante o pagamento entre o comprador e o vendedor.

2.3.2 Macro-Pagamento versus Micro-Pagamento

Muitas das soluções atuais de Comércio Eletrônico são focadas em Sistemas de Macro-Pagamento, como por exemplo, *Digicash*, *CyberCoin*, *Mondex* e *NetCash*. 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 US$ 2.00 ou superior.

Os Sistemas de Macro-Pagamento tipicamente trabalham de duas maneiras: o cliente paga a compra com cartão de crédito, ou o cliente estabelece uma conta junto ao vendedor. Dado o montante significativo do pagamento, as transações deste 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, e são caras, em função das taxas envolvidas de risco e administração. Obviamente esta abordagem não é praticável para sistemas de pagamento onde tipicamente o montante da compra é da ordem de centavos ou poucos dólares.

Os Sistemas de Micro-Pagamento são soluções para compras de produtos e serviços de valor da ordem de centavos ou poucos dólares e possuem flexibilidade. Tais sistemas também provêm velocidade o suficiente na transação de pagamento, de forma que o usuário não tenha espera na entrega do produto ou serviço comprado. Por exemplo, se um usuário está pagando US$ 0.02 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*, que provêm um armazenamento resistente a ataques às informações sensíveis 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 conta 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 pela rede de comunicação insegura como a Internet, não podem ser lidas por uma atacante que pode usar tais informações de forma indevida. Isto deve 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 ganha o controle do sistema e se passar como um usuário autorizado do mesmo.

2.4.1 Ataques à Segurança de Computadores

As seguintes categorias de ataques à Segurança de Computadores devem ser 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 a 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, não para obter lucro financeiro e sim para adulterar os dados do usuário ou entidade.

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 os seguintes requisitos:

2.4.2.1 Prevenção de Espionagem

Isto deve ser assegurado 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 disfarce-se como sendo a outra parte no sistema. Para isto, a assinatura digital é utilizada para autenticar as mensagens. A assinatura digital é baseada em algoritmos criptográficos e atestam a integridade como também a autoria da mensagem. Se uma mensagem for alterada isto pode ser detectado. 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. Aplicando-se sumarizações na mensagem (*message digests*) ou assinatura digital, isto assegura 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 disto seria a realização de múltiplos saques não autorizados em uma conta bancária, por alguém, que copia uma mensagem legítima de saque de um sistema de pagamento e substitui várias mensagens de saques com mensagens não legítimas. Para evitar este problema é incluído um número seqüencial 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 fazendo uso da infra-estrutura existente dos bancos ou empresas de cartão de crédito e criaram um sistema de comunicação eletrônica entre o vendedor, consumidor e a instituição financeira.

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

2.5.1 Sistemas Tradicionais de Pagamento Baseados na Internet

2.5.1.1 Sistemas Baseados em Cartão de Crédito

Estes sistemas possuem um mecanismo de pagamento através da infra-estrutura existente de pagamento por cartão de crédito. Nas transações com cartão de crédito, informações sensíveis 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 que 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 da empresa de cartão de crédito.

Muitos destes sistemas utilizam criptografia para proteger as informações sensíveis 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* ou TLS – *Transport Layer Security.*

Os sistemas de pagamento pela Internet baseados em cartão de crédito não são desenhados para transações de pequeno valor monetário. Na verdade estes sistemas visam compras de montantes acima de dois dólares, pois abaixo deste limite o custo para processar a transação poderia ficar acima do preço do produto ou serviço comprado.

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

Resumidamente, um pagamento eletrônico usando-se o SET, é iniciado pelo consumidor possuidor de um cartão de crédito, para pagar um produto ou serviço contratado de uma loja virtual na *Web*. O sistema desta 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 serviço é entregue ao consumidor. Depois o vendedor solicita o ressarcimento financeiro junto à instituição financeira do consumidor, com o auxílio do *gateway* de pagamento.

O protocolo SET usa uma técnica de criptografia chamada de 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 consiste de duas partes, uma para o vendedor e outra para a instituição financeira do consumidor. 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. Esta assinatura é construída gerando uma *message digests* da concatenação dessas duas *message digests* e assinando-a (assinatura dual). Esta assinatura dual permite que cada parte possa ler e validar a assinatura da outra parte do pedido de compra, sem decriptografar informações da terceira parte do processo.

Com isto, o protocolo SET provê tanto confidencialidade das informações de pagamento, como das informações da compra realizada que são transmitidas junto com as informações de pagamento. Também, 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, cuja validade e aceitação, exige a assinatura no mesmo 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 cheque eletrônicos funcionam de forma equivalente, possuem assinatura digital e certificados para autorizar o pagamento como um cheque em papel. O correntista usa um talão de cheques eletrônico que é baseado em alguma forma segura de *hardware* ou *software*. Este dispositivo ou programa garante o armazenamento seguro da chave privada e os 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, o qual encaminha eletronicamente para seu banco. O banco efetua a compensação do cheque e transfere o dinheiro do banco do comprador para conta do destinatário.

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

Um pagamento começa quanto um correntista emite um documento eletrônico que inclui seu nome e número de sua conta corrente, o nome de sua instituição financeira, nome do destinatário e montante do cheque. Usando o sistema de segurança *Kerberos*, uma assinatura eletrônica é incluída ao cheque eletrônico que autentica o cheque. Como um cheque em papel isto 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 pode ser 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 chamado *chip*, que garante o armazenamento e processamento de informações de forma segura. Existe uma variedade de *chips* com características para armazenar informações financeiras para uso em máquinas automáticas de venda, outros para armazenar informações e aplicações que exigem segurança para operar, com 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 esta 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 que são administradas por uma instituição financeira. O dinheiro eletrônico é armazenado 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 quando da utilização do dinheiro em espécie. Como o dinheiro de circulação eletrônica pode ser facilmente copiado, este deve possuir mecanismos para prevenir sua reapresentação em duplicidade. O dinheiro eletrônico é muito flexível e é utilizado nos protocolos de Macro-pagamento e Micro-pagamento.

2.5.2.1 PayWord

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 onde participam o consumidor, o vendedor e um agente financeiro (*broker*) no Sistema de Micro-Pagamento 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* e é baseado no esquema de débito para o consumidor e um corretor (*broker*), enquanto que para o vendedor o esquema é de crédito. O *MicroMint* usa a 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. Este usa uma forma de dinheiro de circulação eletrônica chamado de *scrip*. O *MilliCent* foi planejado para transações de montante de menos de um centavo de dólar até no máximo cerca de US$ 5.00. No Capítulo 5 é dada uma descrição detalhada do *MilliCent*.

Capítulo 3

PayWord

O *PayWord* [1] é um esquema de Micro-Pagamento proposto por *Ronald Rivest* do *MIT LCS* e *Adi Shamir* do *Weizmann Institute of Science*, baseado no esquema de crédito[3] para transações de pequeno valor.

Participam deste esquema 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.

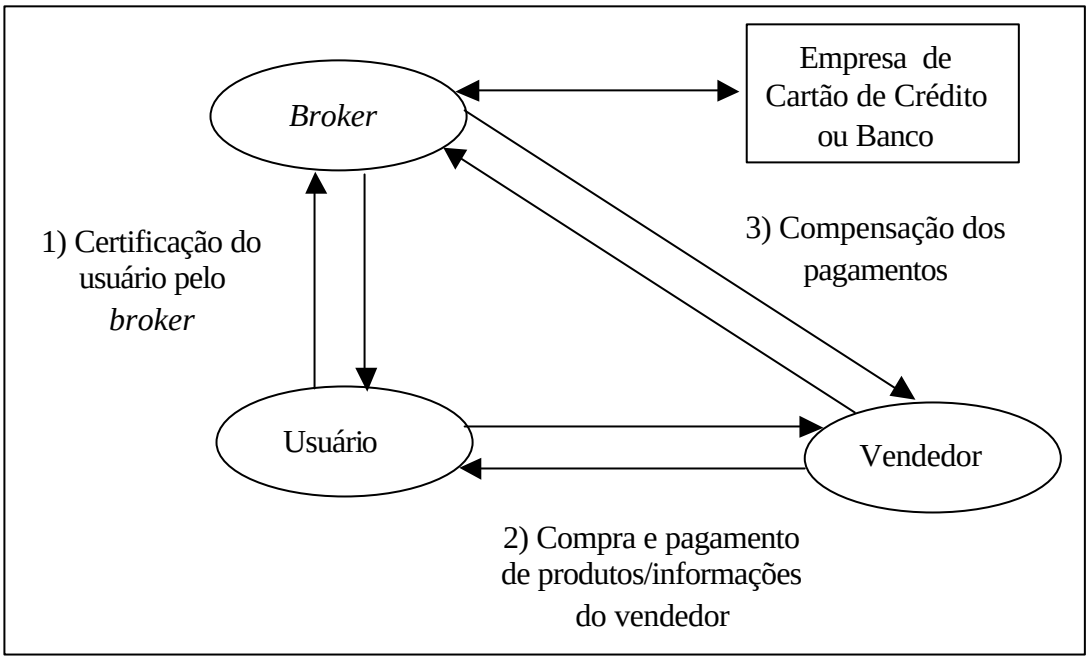

Figura 3.1 Relacionamento entre *Broker*, Usuário e Vendedor

[3] O usuário/consumidor paga uma taxa mensalmente pelas operações realizadas.

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 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 $\{M\}_{SK}$. Esta mensagem pode ser verificada usando a correspondente chave pública *PK*.

Também é utilizada uma função de espalhamento ou função *hash*, denotada por *h* e criptograficamente forte, tal como MD5 [2] ou SHA [3]. A propriedade importante desta função *h* é o fato de ser uma função 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.*

Supomos 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* $w_1 \ldots w_n$. O usuário produz uma série encadeada de *PayWords* na ordem reversa, pegando um número aleatório para produzir o *PayWord* w_n, que é a semente para produzir os demais *PayWords* como segue:

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

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, 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 *us* 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:

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

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

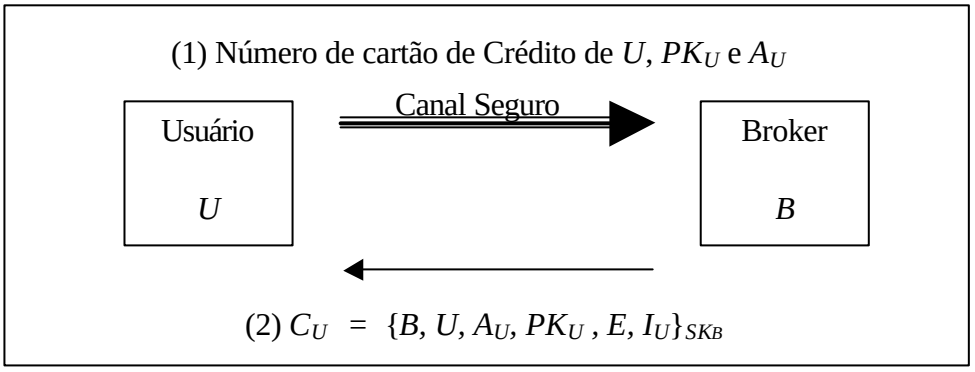

Figura 3.2 Certificação do usuário pelo *broker*

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. Isto significa que um usuário *U* pode visitar uma página da loja virtual na *Web* do vendedor *V* e comprar produtos/informações deste *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* $w_1, ..., w_n$, com raiz w_0. Esta série encadeada de *PayWords* pode ser de qualquer tamanho.

O usuário *U* então constrói seu compromisso para esta série encadeada de *PayWords*, com as seguintes informações:

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

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*. Esta mensagem M_U é assinada pelo usuário *U* e enviada ao vendedor V. Este compromisso autoriza o *broker B* a pagar ao vendedor *V*, por qualquer *Payword* $w_1 \ldots w_n$, 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.

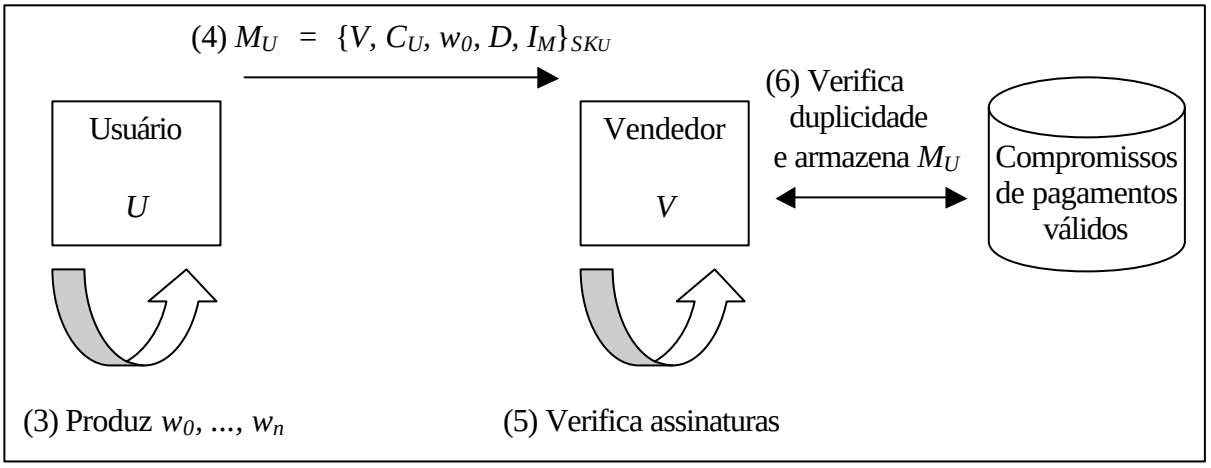

Figura 3.3 Estabelecendo compromisso de pagamento

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. Isto 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. Em nossa aplicação típica de loja virtual na *Web*, com esquema de Micro-Pagamento, 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 correspondente índice, como segue:

$$P_U = (w_i, i) \quad para\ i=1, ..., n.$$

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 estes valores, isto assegura a autenticidade do pagamento P_U.

Com este processo, o vendedor V necessita apenas armazenar por usuário U último par (w_i, i), para validar o próximo pagamento este usuário U ou, no final do dia, para o vendedor V 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, supondo 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. Neste 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 $P_U = (w_{i+5}, 5)$.

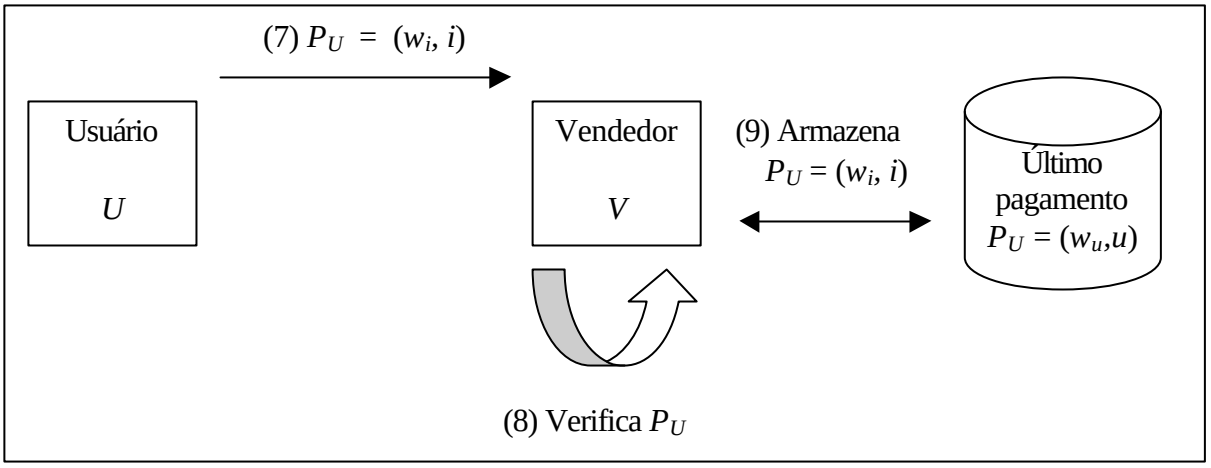

Figura 3.4 Fazendo pagamento

3.1.3 Compensação dos Pagamentos pelo *Broker* ao Vendedor

No final do dia ou 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. Também 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. Supomos que o *broker B* honra cada pedido de ressarcimento autêntico.

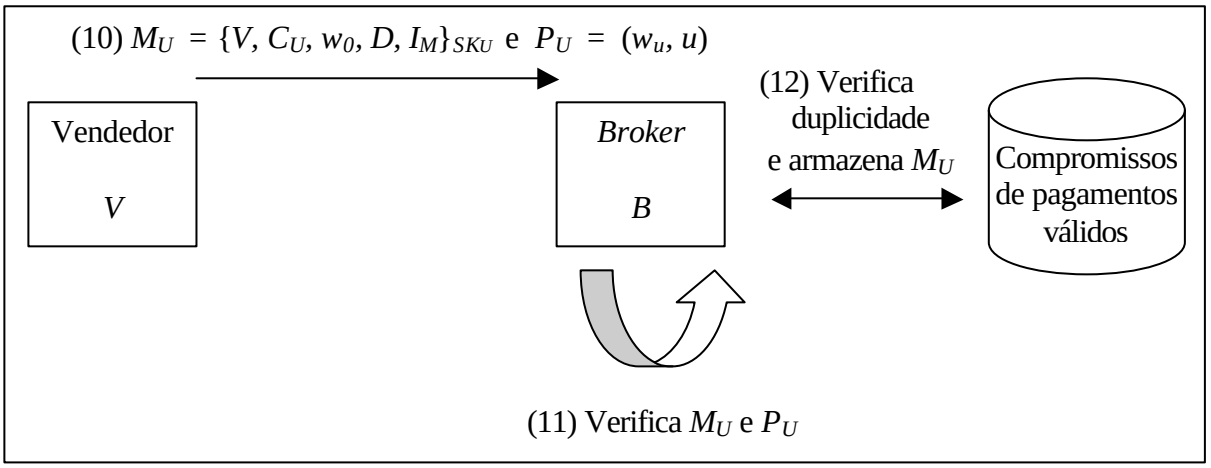

Figura 3.5 Compensação dos pagamentos

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

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

3.2 Avaliação de Segurança

3.2.1 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*.

3.2.2 Detecção de Gasto Duplicado

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

Um vendedor ou um *broker* podem 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_o.

A reapresentação 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). Desta forma, tanto o vendedor como o *broker*, até seu vencimento dos compromissos armazenados, previne a reapresentação de um compromisso válido por um usuário ao vendedor ou pelo vendedor ao *broker*.

3.2.3 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. Também, o usuário pode gerar o montante exato de *PayWords* para um pagamento.

- *Uso do último pagamento para autenticação* – O vendedor e o broker somente necessita armazenar o compromisso válido e o correspondente par 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 informações que identifica 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 mensagem 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 mensagem 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 período.

Capítulo 4

MicroMint

O *MicroMint* [1] é um esquema de Micro-Pagamento 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, enquanto que para o vendedor e o *broker* o esquema é de crédito.

Participam deste esquema 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, o qual vende produtos/informações e coleta como pagamento essas moedas e compensa junto ao *broker*.

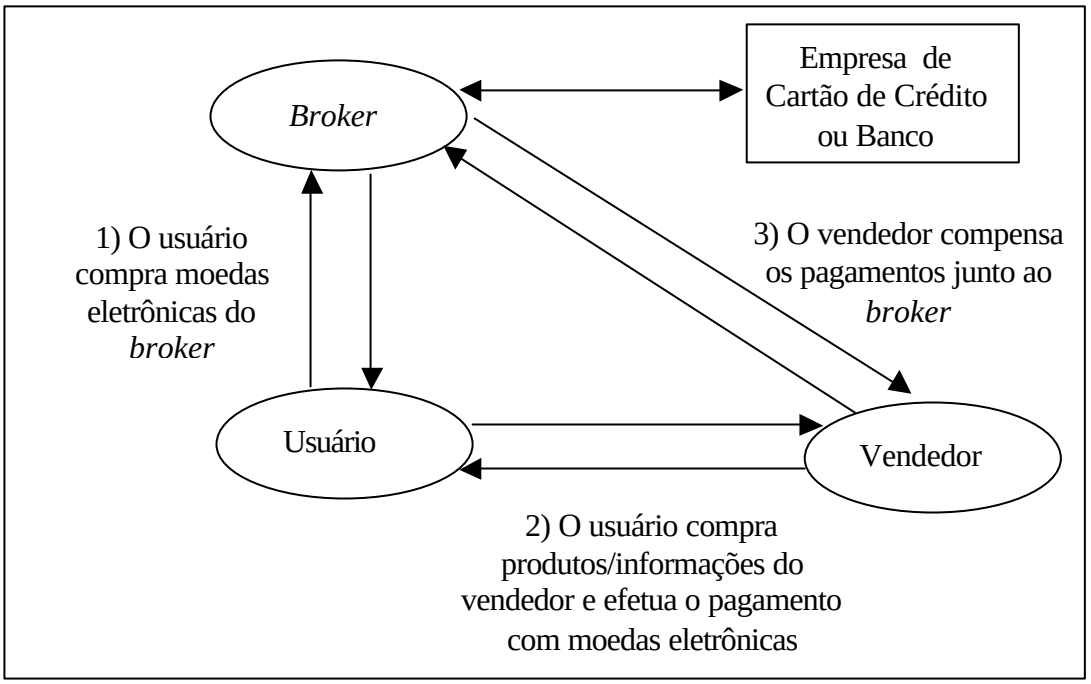

Figura 4.1 Relacionamento entre *Broker*, Usuário e Vendedor

4.1 Visão Geral

Como outros esquemas de Micro-Pagamento, 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 gerar a primeira moeda, mas depois para gerar as moedas adicionais, isto pode ser feito de forma barata. Este processo é similar ao de cunhar moedas, cujo investimento inicial é muito alto em equipamentos e expertise para produzir moedas economicamente baratas.

No cenário típico, um *broker* produz moedas eletrônicas que são válidas por um certo período, por exemplo, um mês. O *broker* necessita de investimento substancial em *hardware*, que dá a ele uma vantagem computacional sobre um possível falsário. O *broker* roda continuamente neste equipamento por um mês 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. Com estas moedas, o usuário efetua compra de produtos/informações e dá como pagamento ao vendedor tais moedas. O vendedor retorna essas moedas coletadas 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, que 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, que é representado pelo conjunto de valores $\{x_1, x_2, ..., x_k\}$, onde $h(x_1) = h(x_2) = ... = h(x_k) = y$. 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: (1) difícil de serem produzidas e (2) fáceis de serem validadas.

Podemos visualizar uma função *hash* como um processo de lançar aleatoriamente bolas numeradas dentro de uma série de urnas numeradas. Uma colisão ocorre 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, segue 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 do mesmo. A segunda versão é forma de moeda eletrônica específica para um usuário com suas informações e somente utilizável pelo próprio usuário. A terceira versão combina a segunda versão com informações específicas de um vendedor, o que implica em 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 $\{x_1, x_2, ..., x_k\}$, onde cada valor x_i, com m bits e $i=1, ..., k$, tem um mesmo valor y, com n bits, mapeado por uma função *hash*, denotada por h. Esta característica é chamada de colisões de k-modos.

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

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 suficiente grande para armazenar todos os possíveis valores x_i gerados. Esta é uma importante característica para evitar a falsificação dessas moedas.

Também, para prevenir falsificação, o *broker* atribui na produção das moedas eletrônicas, 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 $\{b_n, ..., b_{n-t+1}\}$ fixados nos primeiros t bits do resultado da função *hash*. Desta forma, uma moeda válida para um determinado mês tem o seguinte formato:

$$h(x_i) = y = \{b_n, ..., b_{n-t+1}, b_{n-t}, ..., b_1\} \qquad \textit{para } i=1, ..., k,$$
$$\textit{onde } \{b_n, ..., b_{n-t+1}\} = \textit{critério do mês.}$$

4.2.1.2 Segunda versão – Moeda Eletrônica Específica por Usuário

Nesta 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 tais moedas sejam roubadas, não terá valor para outros, reduzindo enormemente a motivação de um oponente de roubar moedas.

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

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

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 isto, um usuário ou um oponente é desencorajado a gastar a mesma moeda eletrônica com outros vendedores.

Também, nesta versão o resultado y produzido pela função h é de tamanho de n bits. Os primeiros t bits de y estabelece 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.

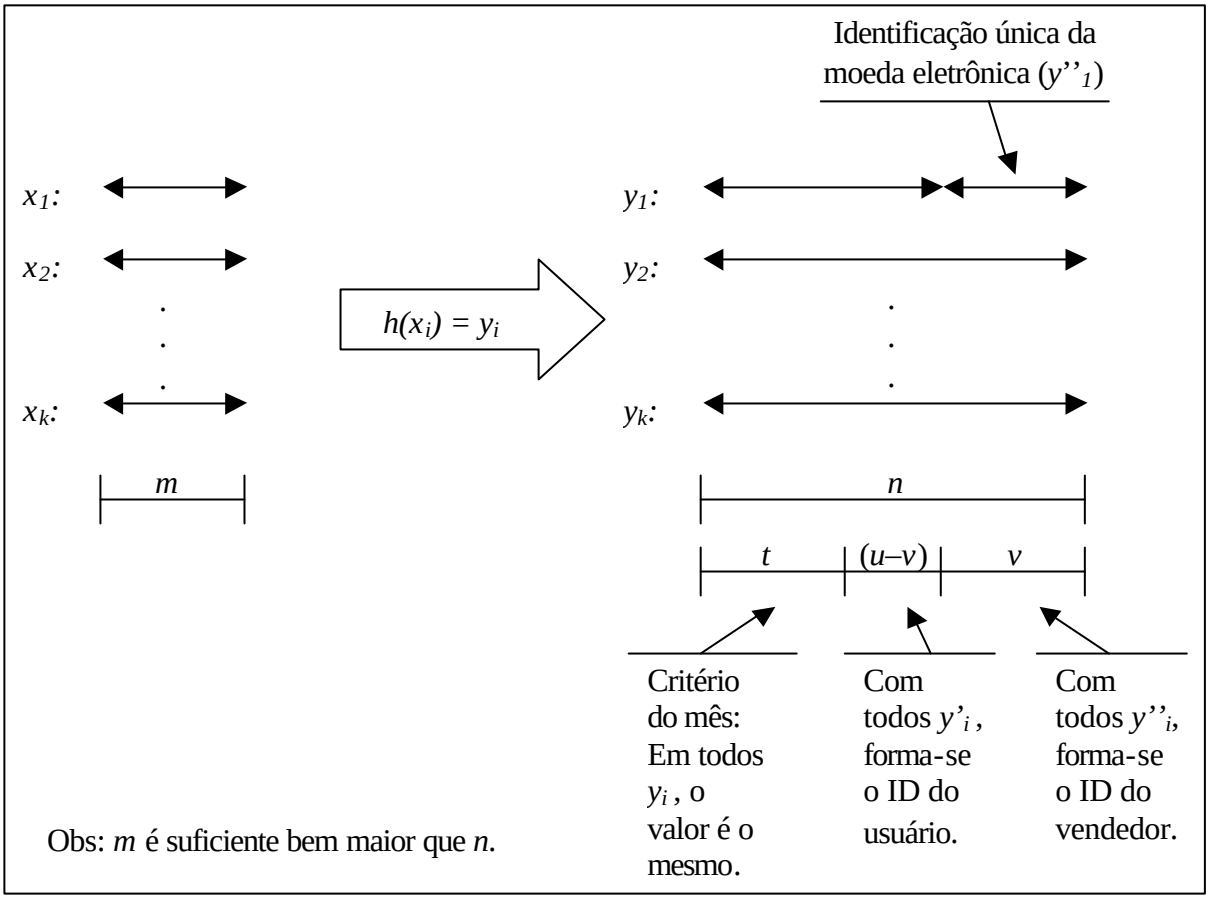

Figura 4.2 Estrutura de uma moeda eletrônica

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:

$$\textit{Moeda Eletrônica} = \{x_1, x_2, ..., x_k\}$$
onde
$$h(x_i) = y_i = \{b_n, ..., b_{n-t+1}, b_{n-t}, ..., b_1\}_i \qquad \textit{para } i=1, ..., k,$$
$$\textit{e } \{b_n, ..., b_{n-t+1}\}_i = \textit{critério do mês}$$

e

$$y'_{j+1} - y'_j = d'_j \ (mod\ 2^{u-v}) \qquad \textit{para } j=1, ..., k\text{-}1 \ \textit{ e } \ u = n - t$$
onde
$$h'(U) = (d'_1, d'_2, ..., d'_{k-1})$$

e

$$y''_{j+1} - y''_j = d''_j \ (mod\ 2^{v}) \qquad \textit{para } j=1, ..., k\text{-}1$$
onde
$$h''(V) = (d''_1, d''_2, ..., d''_{k-1})$$

e

$$y''_1 \ \textit{ valor único.}$$

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:

$$y'_{j+1} - y'_j = d'_j \ (mod\ 2^{u-v}) \qquad \textit{para } j=1, ..., k\text{-}1 \ \textit{ e } \ u = n - t$$
onde
$$h'(U) = (d'_1, d'_2, ..., d'_{k-1}).$$

4.2.3 Compras

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

$$y''_{j+1} - y''_j = d''_j \ (mod\ 2^v) \qquad para\ j=1, ..., k-1$$

onde

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

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.

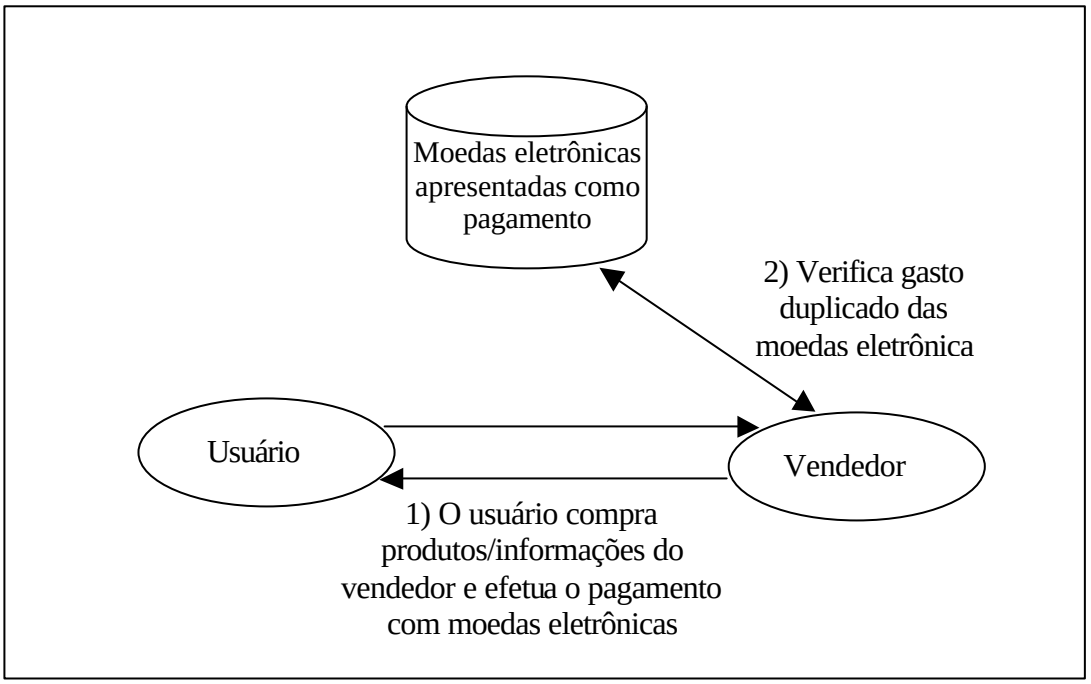

Figura 4.3 Compras

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, se uma moeda eletrônica é apresentada mais que uma vez o *broker* não compensa o valor correspondente ao vendedor na reapresentação da mesma moeda.

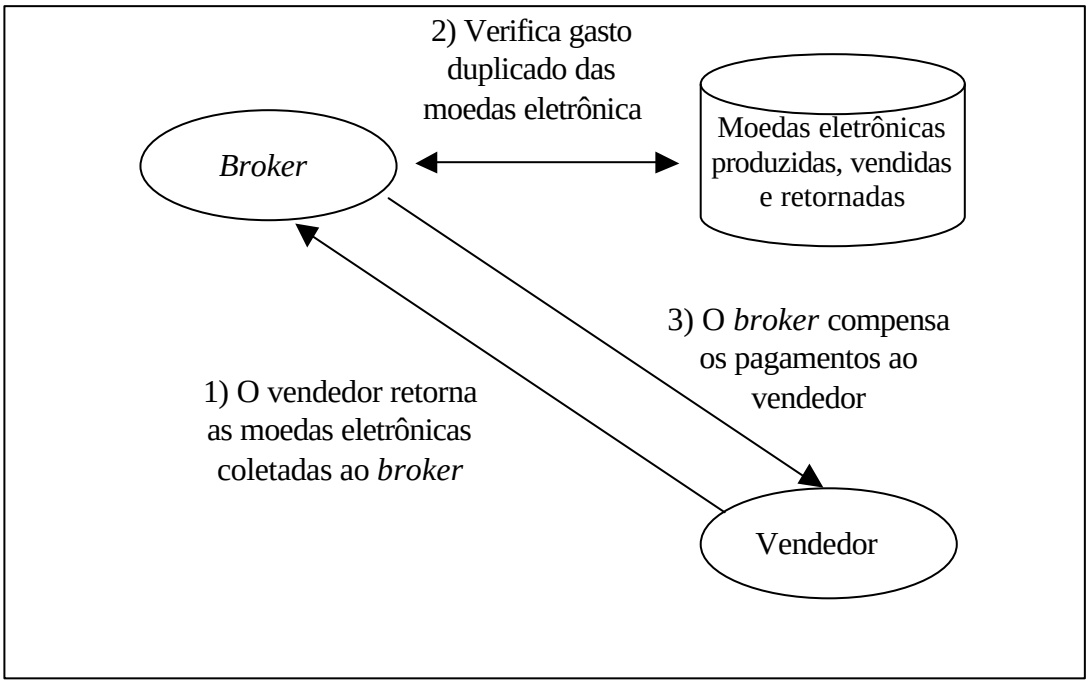

Figura 4.4 Compensação dos pagamentos

4.3 Avaliação

A segurança do esquema *MicroMint* contra o processo de falsificação, deriva do custo da grande base computacional necessária para a produção de moedas eletrônicas. Desta forma, tal mecanismo de segurança foi desenhado para desencorajar um ataque em grande escala, tal 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 caro para ser de interesse de um oponente. Suponha 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, que uma estação de trabalho padrão, pode processar 2^{14} operações de *hash* por segundo, isto significa aproximadamente 80 anos, para a produção de uma única moeda eletrônica neste 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 e quais destas moedas foram utilizadas em pagamentos de produtos/informações pelo usuário junto aos vendedores, quando da apresentação das mesmas na compensação pelo vendedor (ver passo 2 nas Figuras 4.3 e 4.4). O valor único y''_1 mantido, por exemplo, em uma base de dados pelo *broker* e pelo vendedor, previne a reapresentação de uma 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. Desta forma, 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* – O usuário pode verificar a validade de suas moedas eletrônicas, 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 que foi desenvolvido pela Compaq para o comércio eletrônico pela Internet. Este usa uma conta eletrônica que contém dinheiro eletrônico que é chamado *scrip*. O *MilliCent é* um esquema de Micro-Pagamento para transações de montante de menos de um centavo até cerca de US$ 5.00.

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*.

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

Em uma aplicação típica no esquema de Micro-Pagamento, 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 de um vendedor, este deve ter um *scripVendor* daquele vendedor, pois o mesmo apenas aceita como dinheiro eletrônico 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, a qual é responsável por enviar o respectivo *scrip*, como pagamento para um vendedor ou para *broker* quando da compra de *scripVendor*.

Quando a carteira de um usuário recebe um pedido de compra, esta verifica se a mesma possui o apropriado montante 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*, este é verificado se não é falsificado ou forjado e se não é uma reapresentação de um anterior. Validado o *scrip* recebido, deste é deduzido 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*.

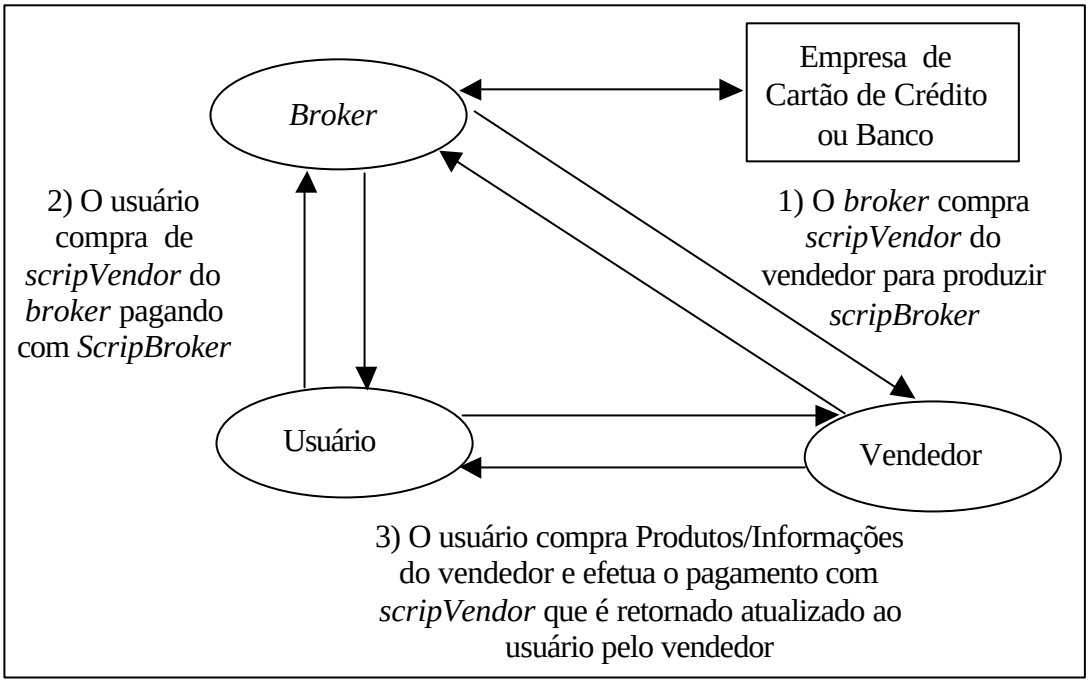

Figura 5.1 Relacionamento entre *Broker*, Usuário e Vendedor

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 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, o usuário usa *scripBroker* para pagar.
3) Quando um usuário realiza uma compra, o custo da compra é deduzido do valor do *scrip* e novo *scrip*, com o novo valor de saldo, é retornado atualizado. Quando um usuário completou uma série de transações, este 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, gasto e 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 estes segredos de uma forma a mostrar que conhece um segredo sem que seja necessário revelar o segredo.

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*. Este pode ser gasto apenas uma vez e é resistente a falsificação. Também, só pode ser gasto pelo proprietário e pode ser eficientemente produzido e validado. Um *scrip* consiste de um corpo e um certificado.

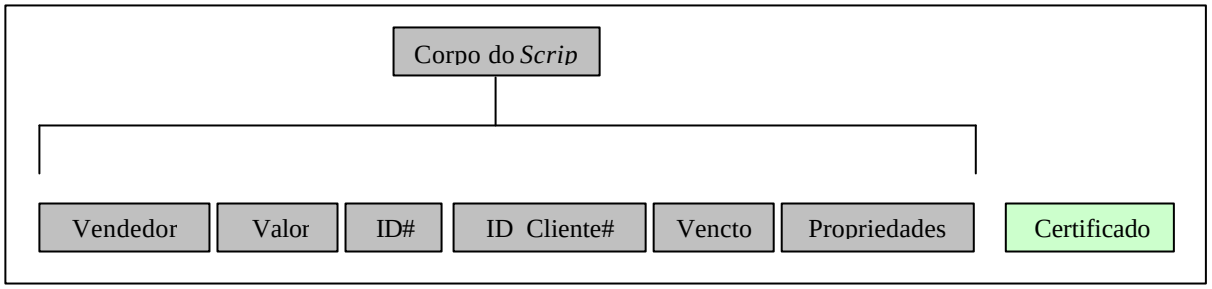

Figura 5.2 Estrutura da Moeda Eletrônica *Scrip*

A figura 5.2 mostra a estrutura do *scrip*, onde 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*, que foi 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 o *scrip* 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.

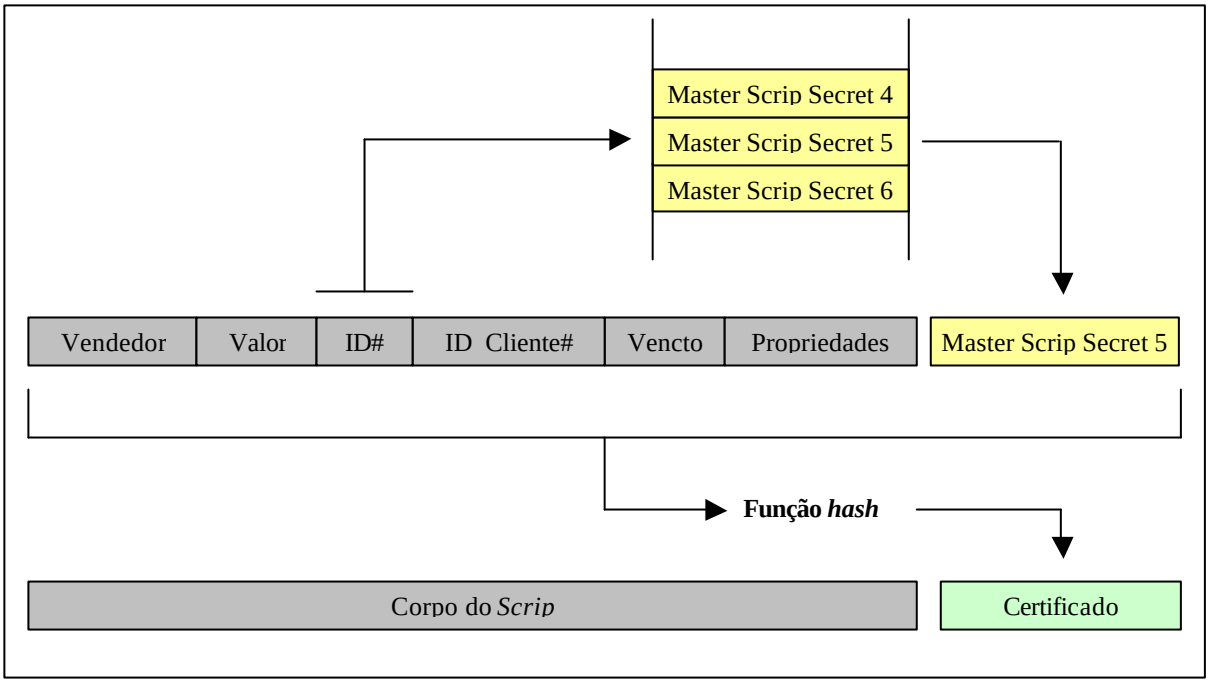

Figura 5.3 *Master Scrip Secret* e Certificação do *Scrip*

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

Quando um novo *scrip* é gerado por um vendedor ou *broker*, um certificado é produzido para aquele *scrip*, concatenando-se o respectivo *Master Scrip Secret* ao corpo do *scrip* e, em seguida, aplica-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 quanto 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 aplicada a função *hash* para produzir o certificado recalculado. Se o *scrip* não foi falsificado, então o certificado original e o recalculado serão idênticos.
2) A cada *scrip* recebido pelo vendedor, este 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*.

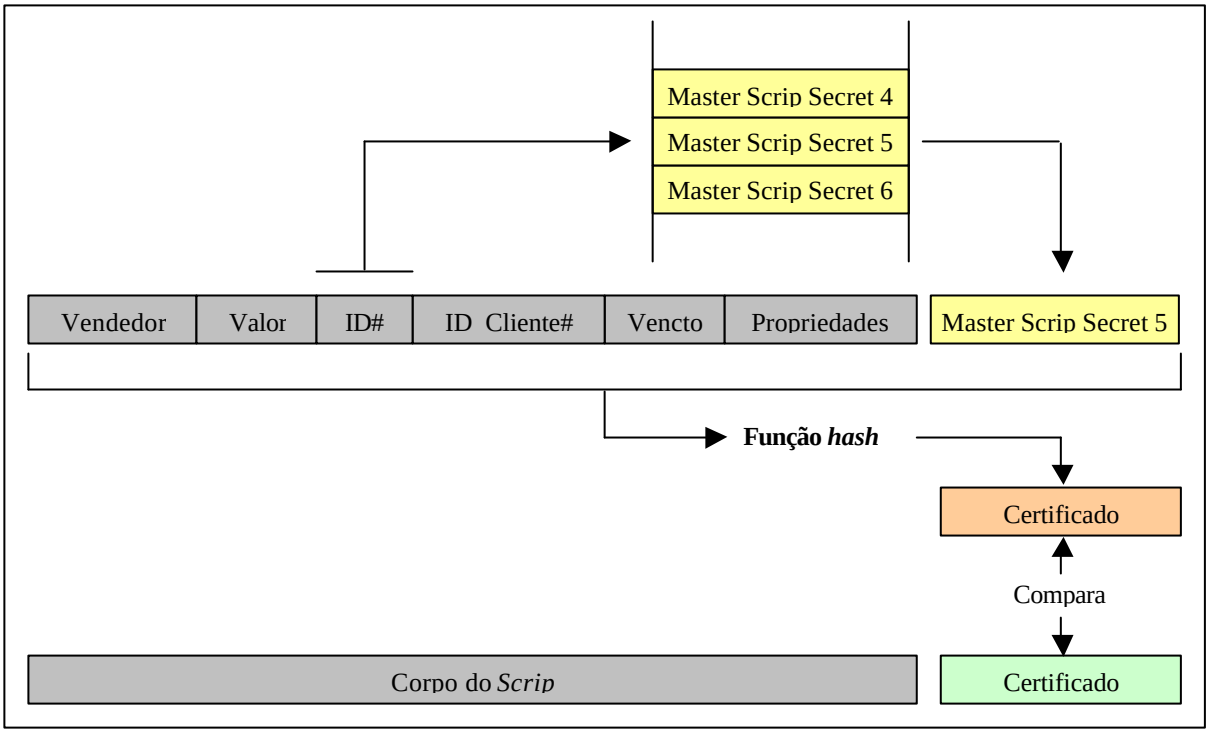

Figura 5.4 Validação do certificado do *scrip*

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 uma transmissão segura do *scrip*. O vendedor e *broker* geram e armazenam uma série de *Master Customer Secrets* única para uso futuro. Também a estrutura de uma *Master Customer Secret* não é descrita em [4], contudo, pode-se supor que seu tamanho é de no mínimo 128 bits.

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

O *Customer Secret* é produzido, concatenando-se o respectivo *Master Customer Secret* ao identificador de usuário (*ID_Cliente#*) do *scrip* e, em seguida, aplica-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 Customer Secret* é selecionado com base no *ID_Cliente_Serial#*, parte do *ID_Cliente#*. O *ID_Cliente_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 *Customer Secret* baseado no mesmo *Master Customer Secret* daquele serial.

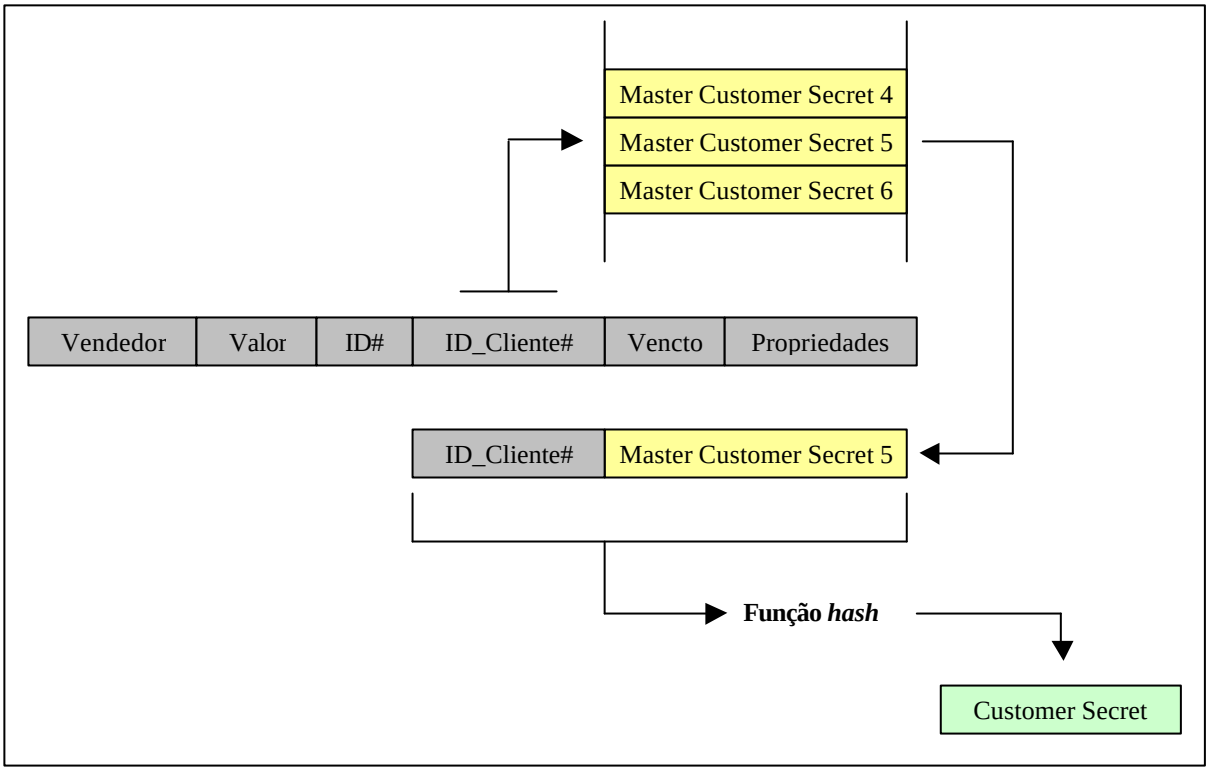

Figura 5.4 *Customer Secret* e *Master Customer Secret*

5.2.5 Protocolos do *MilliCent*

O *MilliCent* é baseado em três protocolos para abranger diferentes níveis de segurança. O primeiro protocolo, chamado "*scrip in the clear*", é o mais simples, eficiente e base para os outros dois protocolos. Contudo, este protocolo não é útil na prática, pois é muito inseguro como veremos a seguir. O segundo protocolo, chamado "*private and secure*", é seguro e oferece uma boa privacidade, mas é muito caro, pois consome muito tempo computacional 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

Este protocolo é o mais simples do *MilliCent*. 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, pois, um oponente pode interceptar um *scripVendor* atualizado, quanto da sua transmissão do vendedor para o usuário. Imediatamente, o oponente pode utilizar este *scripVendor* interceptado no pagamento de sua compra. Mais tarde, quando o usuário for tentar apresentar o *scripVendor* interceptado no pagamento de sua compra, o vendedor irá recusar o pagamento, pois este *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*, um segredo (*Customer Secure*) é gerado 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, o usuário transmite ao vendedor seu *scripVendor* criptografado, por um método simétrico de criptografia usando como chave secreta o *Customer Secret* daquele *scripVendor*

e, também 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, este é 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 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 Micro-Pagamento. Para tanto, este protocolo *Secure without Encryption*, descreve como dar privacidade em um pedido de pagamento e 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, este envia ao vendedor, um pedido, o *scripVendor* e uma 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, esse deriva o *Customer Secret* do *scrip* e recalcula a assinatura do pedido. Se a assinatura do pedido recebida for idêntica à assinatura recalculada, então o pedido é válido, ou seja, o pedido ou *scrip* não foi falsificado ou forjado.

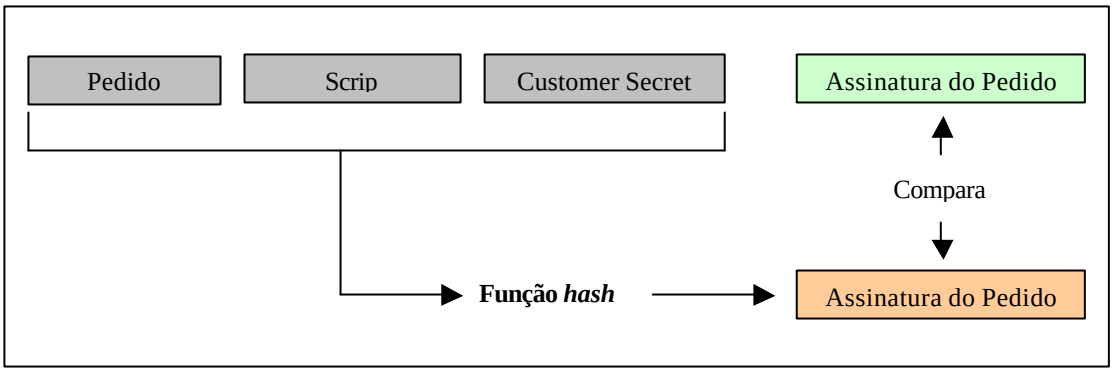

Figura 5.5 *Secure without encryption*

Em seguida, o vendedor deduz o valor da compra do valor do *scrip* recebido e retorna um novo *scrip* atualizado ao usuário. O *scrip* devolvido pelo vendedor 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 próxima compra.

Neste protocolo não é necessário que qualquer resposta seja criptografada. Um oponente não pode roubar o *scrip*, pois a assinatura do pedido não pode ser calculada sem o conhecimento do *Customer Secret*. O vendedor deve também assinar 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. Diversos modelos de negócio existem para que o *broker* possa comprar *scrips* dos vendedores, tal como, pagamento antecipado, produção licenciada ou venda em consignação. Em todos este modelos o *broker* pode obter receita na venda dos *scrips*, pois este vende aos usuários/consumidores partes do valor dos *scrips* por preço maior 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.

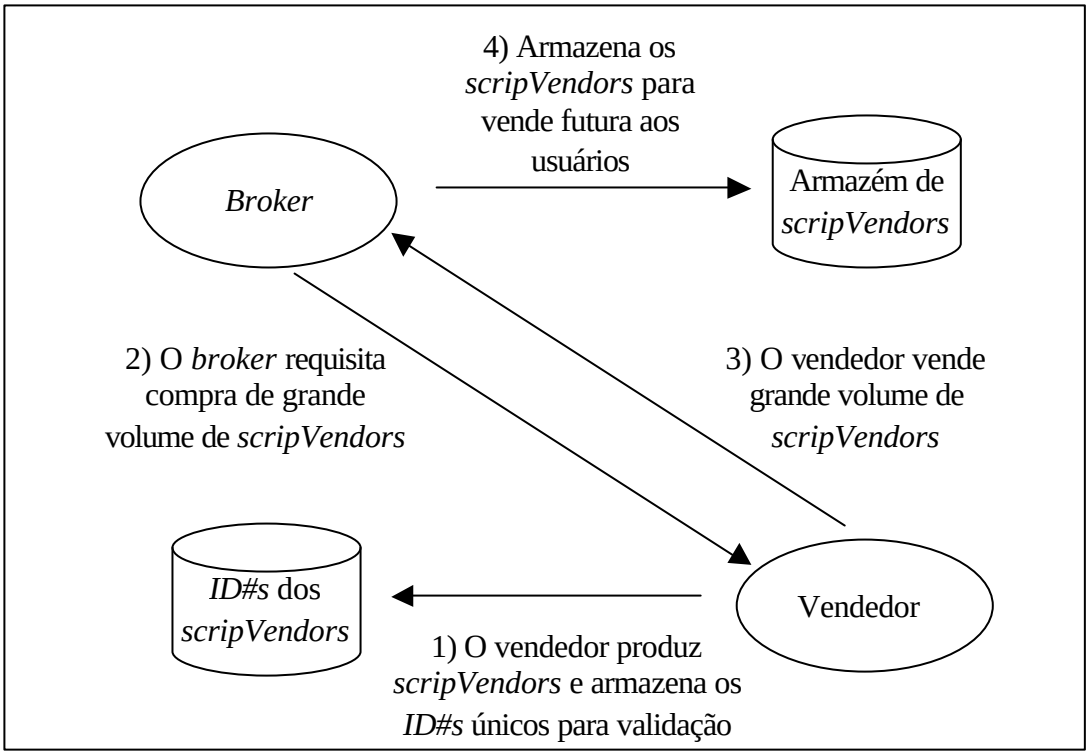

Figura 5.6 Armazém de *scrips*

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*. Isto significa que o *broker* produz *scrips* que o vendedor venha a aceitar e possa validar. O vendedor vende o direito da geração de *scrips* para o broker. Para esta 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. Estas informações são necessárias para a produção de *scrips*. O vendedor pode validar o *scrip* licenciado, pois o *Master Scrip Secret* pode ser derivado da série de *ID#s* de *scrips* e o *Master Customer Secret* pode ser derivado da série dos identificadores dos usuários (ver Seções 5.2.2 e 5.2.4).

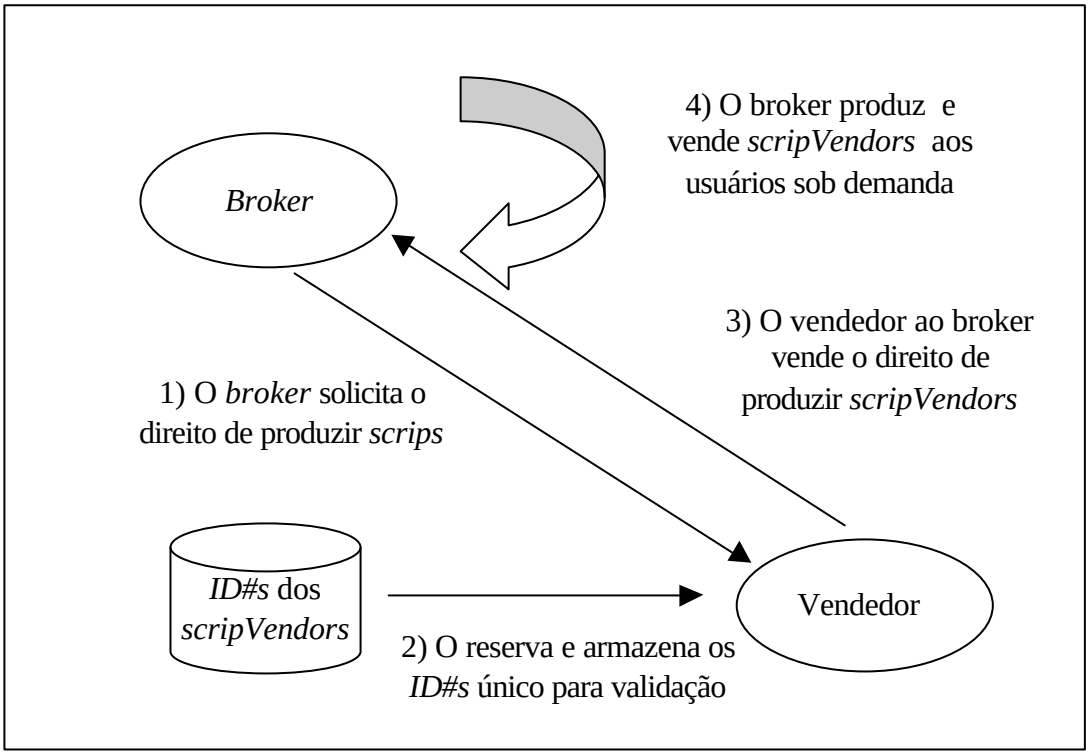

Figura 5.7 Produção de *Scrips* Licenciados

5.3.1.3 Múltiplos Brokers

Neste ambiente com múltiplos *brokers*, o usuário de um *broker* pode querer fazer uma compra de um vendedor associado a um outro *broker*. Se cada vendedor tem conta com um único *broker*, o usuário deve possuir *scripBroker* de diferentes *brokers*. Para simplificar a situação deste usuário, o seu *broker* pode comprar *scripBroker* do *broker* do vendedor do qual este deseja fazer compras e revendê-lo ao seu usuário. Em seguida, este usuário pode comprar *scripVendor* do *broker* do vendedor interessado, pagando com *scripBroker* comprado do seu *broker*.

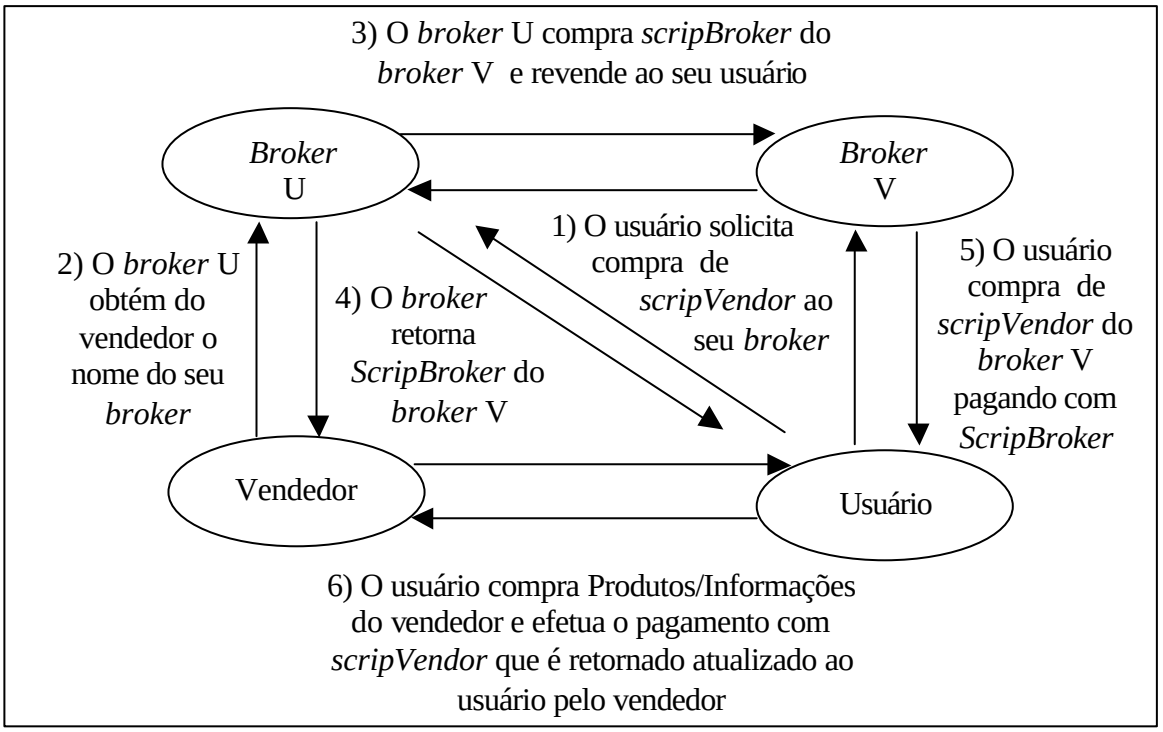

Figura 5.8 Múltiplos *Brokers*

5.3.2 Compra de *scripBroker* pelo Usuário

Inicialmente o usuário não possui nenhum *scrip* de um *broker* ou de um vendedor. O usuário estabelece uma comunicação com um *broker* que possui relacionamento e compra *scripBroker* usando um método de pagamento fora do *MilliCent* como, por exemplo, cartão de crédito. O *broker* recebendo seu pagamento, este 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 que o usuário possui *scripBroker* mas não possui *scripVendor* e deseja fazer um pagamento de algum produto/informação de um vendedor, este necessita de *scripVendor* daquele vendedor. 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* daquele vendedor, o usuário pode enviar o *scripVendor* como forma de pagamento ao vendedor. O vendedor

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

A segurança do esquema *MilliCent* decorre se todas as transações entre as partes (*broker*, vendedor e usuário) forem protegidas, usando os três segredos, que envolvem desde a produção, gasto e validação do *scrip* (ver Seções 5.2.2, 5.2.3 e 5.2.4). 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.

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, concatenando-se o respectivo *Master Scrip Secret* ao corpo do *scrip* e, em seguida, aplica-se uma função *hash*, que assegura a integridade e previne 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 *MilliCent* 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. O vendedor já obteve seu benefício ou lucro quando este vendeu os *scrips* ao *broker* (ver seção 5.3.1).

- *Não é necessário o esquema de criptografia de chave pública* – O protocolo *MilliCent* 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 isto, a segurança do esquema *MilliCent* faz uso de criptografia relativamente barata, aplicando-se uma função *hash* e 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

Existem diversos Sistemas de Pagamento Eletrônico em vários estados de desenvolvimento, desde propostas no âmbito acadêmico até sistemas em fase comercial. Muitos destes sistemas compartilham objetivos e propriedades similares. A análise a seguir, compara pontos similares e divergentes do *PayWord* com *MicroMint* e do *PayWord* com *MilliCent*. Em seguida, é realizada uma comparação das propriedades desejáveis de um Sistema de Pagamento Eletrônico.

6.1 *PayWord* versus *MicroMint*

Segue 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*. O certificado digital possui uma data de vencimento e deve ser renovado pelo *broker* que verifica a validade da conta do usuário. Com este 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 que são 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 vendedor*. Uma série encadeada de *PayWord* é autenticada pelo compromisso de pagamento, onde contém w_0 o certificado que identifica o usuário junto ao vendedor (ver Seção 3.2). Também, a moeda eletrônica *MicroMint* é produzida especificamente para um usuário e vendedor em particular (ver Figura 4.2).

Segue alguns pontos onde o *PayWord* difere do *MicroMint*:

- *PayWord utiliza o esquema de criptografia de chave pública*. Apesar do esquema de criptografia de chave pública ser considerado caro, 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). Desta forma, o custo deste esquema 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 são vendidas no mês seguinte aos usuários.

- *PayWord é baseado no esquema de crédito*. O *broker* do *MicroMint*, envia mensalmente ao usuário um número específico de moedas eletrônicas e debita da conta do usuário o valor monetário referente às moedas compradas. No caso do *PayWord*, somente é debitado depois que o vendedor obter 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. Por outro lado, o *broker MicroMint* necessita de um enorme espaço de armazenamento, para a produção mensal da moeda eletrônica *MicroMint*.

- *Espaço de armazenamento exigido no vendedor*. O vendedor *PayWord* necessita 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*

Segue 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 uma função *hash* com uma chave secreta e as mensagens são autenticadas aplicando 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, que é representado pelo conjunto de valores $\{x_1, x_2, ..., x_k\}$, onde $h(x_1) = h(x_2) = ... = h(x_k) = y$.

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

Segue alguns pontos onde o *MilliCent* difere do *MicroMint*:

- *Não é o broker que é proprietário do scrip e sim o vendedor*. A descentralização da geração ou produção do *scrip* no *MilliCen*t resulta em uma redução de tempo comparado com o processo equivalente do *MicroMint*.

- *O broker é necessário ser envolvido a cada pagamento junto a um novo vendedor*. Caso um usuário não possua o apropriado *scripVendor* do vendedor, é necessário a interação prevista na seção 5.3.1.3, para atender múltiplos *brokers*. No *MicroMint* somente uma transação é necessária entre o usuário e o *broker* durante a produção da moeda eletrônica, independente da quantidade de vendedores 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 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 somente pelo vendedor ou *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***
Macro–Pagamento	Ö			
Micro–Pagamento		Ö	Ö	Ö
Atomicidade da Transação	Ö	Ö	Ö	
Transação Segura	Ö	Ö	Ö	Ö
Privacidade da Identidade do Usuário	Ö	Ö	Ö	Ö
Privacidade do Usuário vs. Compra	Ö			
Prevenção de Falsificação	Ö	Ö	Ö	Ö
Detecção de Gasto Duplicado	Ö	Ö	Ö	Ö
Escalabilidade	Ö	Ö	Ö	Ö
Operação *On-line*	Ö			Ö
Operação *Off-line*		Ö	Ö	
Solução por *Software*	Ö	Ö	Ö	Ö

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

Capítulo 7

Conclusões Finais

Podemos imaginar bilhões de números de cartões de crédito sendo enviados por redes de comunicação inseguras como a Internet para viabilizar o pagamento eletrônico de transações de pequeno valor monetário?

Neste texto, foram analisadas três possíveis propostas que podem evitar este cenário. Transações com cartão de crédito para pagamentos de operações de pequeno valor são de custo alto, inseguras e lentas. O esquema de Micro-pagamento, com proposta da moeda eletrônica e envolvimento entre os usuários, vendedores e *broker*, apresenta-se como sendo uma melhor solução. Contudo, diversos métodos de pagamento deverão coexistir lado a lado na Internet.

A utilização de Sistemas de Pagamento Eletrônico será fortemente influenciada por sua mobilidade, facilidade de uso, confiabilidade, segurança e custo reduzido. Estes Sistemas serão úteis quando muitas pessoas os estiverem usando no comércio de produtos e serviços. Para isso, é muito importante que eles possuam intercâmbio no nível monetário e tecnológico.

Referências Bibliográficas

[1] *Ronald L. Rivest and Adi Shamir,* PayWord and MicroMint: Two simple micropayment schemes (1996).

[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. Gahthier and 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, Switzeland, 1992.

[6] *J. Daemen and V. Rijmen,* AES Proposal: Rijndael, First Advanced Encryption Standard (AES) Conference, Ventura, CA, 1998.