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

ISSN 1413-215X
BT/PCS/0221

μP: Uma Solução de Micropagamentos

Pedro Ancona Lopez Mindlin
Tereza Cristina Melo de Brito Carvalho

São Paulo - 2002

O presente trabalho é um resumo da dissertação de mestrado apresentada por Pedro Ancona Lopez Mindlin, sob orientação da Profa. Dra. Tereza Cristina Melo de Brito Carvalho: "μP: Uma Solução de Micropagamentos", defendida em 05/03/2002, na EPUSP.

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

FICHA CATALOGRÁFICA

Mindlin, Pedro Ancona Lopez
μP : uma solução de micropagamentos / P.A.L. Mindlin, T.C.M.B. Carvalho. – São Paulo : EPUSP, 2002.
13 p. – (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia de Computação e Sistemas Digitais, BT/PCS/0221)

1. Comércio 2. INTERNET I. Melnikoff, Selma Shin Shimizu II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais III. Título IV. Série
ISSN 1413-215X CDD 380.1
004.6

μP: uma solução de micropagamentos

Pedro Ancona Lopez Mindlin
Escola Politécnica da Universidade de São Paulo
Departamento de Computação e Sistemas Digitais
e-mail: pmindlin@larc.usp.br

Tereza Cristina Melo de Brito Carvalho
Escola Politécnica da Universidade de São Paulo
Departamento de Computação e Sistemas Digitais
e-mail: carvalho@larc.usp.br

Resumo

O desenvolvimento do comércio eletrônico brasileiro possibilita o surgimento de uma nova tendência: a venda de informações em formato digital; esse tipo de produto apresenta as características de envolver baixos valores e requerer transações rápidas, que são inviáveis para os sistemas tradicionais de pagamento eletrônico. Os micropagamentos surgem como alternativa nesse cenário, possibilitando a realização de transações eletrônicas rápidas e de baixo custo. Este trabalho apresenta um sistema de micropagamentos chamado μ**P**, que se caracteriza pela geração, em uma instituição central, de fichas eletrônicas que são compradas e distribuídas entre os clientes e facilmente verificadas — e portanto aceitas como pagamento — por vendedores de comércio eletrônico. O μ**P** se diferencia por utilizar apenas um conjunto de fichas que permitem realizar compras em qualquer vendedor cadastrado.

Abstract

The development of electronic commerce in Brazil has led to the arrival of a new trend: the distribution of digital information; this kind of product is characterized by involving low values and requiring fast transactions that are not suitable to existing electronic payment systems. Micropayment systems come as an alternative, allowing for the implementation of such transactions at low costs. This work introduces μP, a micropayments system based on central generation of electronic coins that are bought by and distributed among clients and easily verified — and thus accepted as payment — by electronic commerce vendors. It differentiates itself by generating a single group of tokens that can be used for shopping among all of these vendors.

Palavras-chave

Comércio eletrônico, micropagamentos

I. Introdução

O desenvolvimento do comércio eletrônico tem levado ao desenvolvimento concomitante de sistemas de pagamento que lhe sejam aplicáveis. Muitos sistemas já existem e estão hoje em funcionamento, tais como os sistemas de cartões de crédito e alguns sistemas de dinheiro eletrônico. Esses sistemas, contudo, apresentam sempre ao menos um de dois problemas: não se aplicam a pagamentos de valores muito pequenos ou apresentam latência muito grande para cada transação.

Um sistema de micropagamentos propõe-se a resolver ambas as questões — apresentando latência muito baixa e aplicando-se a transações de valores pequenos — e ao mesmo tempo manter os requisitos necessários de segurança e escalabilidade necessários a sistemas de pagamento que trafegam em redes abertas.

A possibilidade de realização de transações de valores muito pequenos abre caminhos interessantes para o comércio eletrônico: passa a ser possível a venda de produtos de informação cujo preço é da ordem de centavos, tais como imagens, acessos individuais a páginas de informação e notícias individuais de jornais em texto integral.

Podemos perceber em sistemas de dinheiro eletrônico e, em especial, em sistemas de micropagamentos, a mesma característica de todos os sistemas financeiros e bancários: a garantia de pagamento por uma instituição forte. No início do século o dinheiro em circulação em um país, incluindo o Brasil, era apenas a garantia de uma quantidade correspondente em ouro guardada nos cofres centrais; teoricamente, um cidadão poderia apresentar ao Banco Central uma cédula e sacar o correspondente em ouro (a emissão de novas cédulas sem o acréscimo correspondente em ouro corresponde então à desvalorização das próprias cédulas — o início da inflação). Mais tarde o lastro em ouro foi trocado pelo lastro em dólares americanos, mas o princípio continua inalterado. Da mesma forma, um sistema de dinheiro eletrônico nada mais é que a garantia de um lastro de dinheiro em troca de uma outra moeda circulante, agora em forma eletrônica, que é mais adequada ao comércio eletrônico por poder ser distribuída em sua forma natural através das redes de computadores.

A moeda de um sistema de micropagamentos tem, como se poderia esperar, valor e custo muito baixos. O valor tem que ser baixo para que se possa fazer o pagamento de transações de valor muito baixo — o que possibilita a compra de produtos e serviços que não poderiam ser comercializados de outra forma. O custo tem que ser baixo — menor que o valor — para que o processo todo seja economicamente viável para todas as partes envolvidas. No caso específico do trabalho desenvolvido neste artigo, a moeda é um valor de 20 *bytes* obtido através de uma função de *hash*, como veremos mais tarde.

Os sistemas de micropagamento, assim como muitos dos sistemas tradicionais de pagamento, baseiam-se na existência de três entidades: o usuário ou cliente, o vendedor e o agente. O usuário é a pessoa que está fazendo a compra; ele deve ser assinante de um serviço, portador de um cartão ou ter algum outro vínculo com o sistema de pagamento através do qual ele depositará o dinheiro ou pagará a conta. O vendedor é aquele que está vendendo o produto ou serviço; ele também deve ser cadastrado no sistema de micropagamentos para poder efetuar o processamento de cada venda e receber mais tarde o total de seu faturamento. O agente, por fim, é a entidade financeira que dá sustentação ao sistema, cadastrando usuários e vendedores e garantindo o pagamento em dinheiro em contrapartida à apresentação da moeda eletrônica já gasta.

Outra característica interessante de alguns sistemas de micropagamento é a sua maleabilidade com relação às fraudes. Como os valores transacionados são muito baixos e o custo da segurança é relativamente alto, muitos sistemas de micropagamento são permissivos a uma determinada taxa de incidência de fraudes; supondo que essa taxa seja estatisticamente baixa, esses sistemas

são desenhados para arcar com os prejuízos de fraudes até um determinado limite, o que influencia diretamente no baixo custo por transação que eles oferecem.

II. Comércio Eletrônico

Os produtos mais adequados para a venda *on-line* são aqueles para os quais o valor adicionado pelo processo eletrônico de compra é comparável ao valor do próprio produto: um CD ou livro, por exemplo, em oposição a uma casa, cujo alto valor não justifica a pequena economia gerada pelo comércio eletrônico. Segundo Albertin [1], "a regra geral é que os consumidores *on-line* estão mais interessados em realizar compras com melhor informação e mais rapidamente do que necessariamente obter melhor preço". Isso deixa claro que, quanto mais se aproxima do nível de informação pura, distanciando-se do mundo material, mais adequada é a utilização do comércio eletrônico como base do modelo de negócio. Assim, a venda de informações puramente digitais como imagens, áudio e vídeo, cuja entrega ao comprador se dá através do mesmo meio que ele utiliza para a escolha do produto e para a realização do pagamento — o computador — é a que melhor se adequa ao comércio eletrônico.

Nossa transação de comércio eletrônico ficou agora reduzida a uma troca bilateral de informações: a informação financeira para um lado e informação-produto para o outro. A troca de informação financeira é o assunto principal deste trabalho. Essa troca, também chamada forma ou sistema de pagamento, pode acontecer de diversas maneiras, que serão vistas a seguir.

A. *Sistemas de pagamento*

No comércio tradicional brasileiro existem muitas formas de pagamento vigentes, algumas mais utilizadas que as outras. A taxa de utilização de algumas formas de pagamento está diretamente ligada a vários fatores: facilidade de uso, custo por transação, segurança e outros. Algumas delas se aplicam fácil e diretamente ao comércio eletrônico, algumas precisam de adaptações e outras simplesmente não se aplicam, mas apresentam formas análogas.

a) dinheiro: A primeira e mais comum forma de pagamento é o dinheiro. O dinheiro é a forma impressa ou cunhada em metal de uma garantia de valor assegurada pelo Estado, e é portanto de aceitação obrigatória em qualquer forma de transação financeira no país. É também a forma mais barata de pagamento, incorrendo em custo zero por transação. Embora possua a vantagem de ser universal e barato, o dinheiro tem alguns inconvenientes:

- sua propriedade é de seu portador, não havendo como estabelecer ou provar outra forma de propriedade que não a posse física — isso gera sérios problemas quanto à segurança de se portar dinheiro;
- é totalmente fungível, ou seja, não há diferença prática alguma entre cédulas do mesmo valor — isso também dificulta a comprovação de sua propriedade;
- o valor das cédulas é quantizado — 1, 2, 5, 10 etc. — e seu número é limitado, o que cria problemas para a troca de valores intermediários (troco).

A utilização do dinheiro no comércio eletrônico é impossível, pois o seu valor está associado intrinsecamente à cédula ou moeda. Não é possível transformar uma cédula ou uma moeda em informação, embora seja possível enviar informações sobre a posse das mesmas. A troca financeira feita em dinheiro, no entanto, deve ser realizada mecanicamente através do transporte físico das cédulas ou moedas.

b) cheque: Outra forma de pagamento muito comum no Brasil é o cheque. Trata-se de uma ordem de pagamento garantida por uma instituição privada na qual o usuário mantém uma conta. O cheque é pessoal e intransferível, e só vale se assinado pelo seu titular, o que lhe confere um grau de segurança bem maior que o do dinheiro (embora os bancos não possam conferir todas as assinaturas de cheques de baixo valor). Outra vantagem do cheque é a possibilidade de ele ser preenchido com qualquer valor, o que facilita pagamentos de maior vulto ou de valores quebrados em relação às cédulas e moedas de dinheiro. O cheque, no entanto, também apresenta algumas desvantagens: seu custo é relativamente alto, restringindo seu uso a camadas sociais de maior poder aquisitivo e a pagamentos de maior valor; pode ser roubado e preenchido com valores quaisquer por terceiros, valendo-se do fato de que nem todas as assinaturas são conferidas e pode ser emitido sem a real existência de fundos que lhe garantam o pagamento. Como a verificação é feita *a posteriori*, os comerciantes incorrem no risco de arcar com o prejuízo.

O cheque também não pode ser utilizado diretamente no comércio eletrônico por sua natureza; assim como as cédulas e moedas de dinheiro, o cheque é feito de papel e não pode ser transmitido eletronicamente. Seu formato, entretanto, é facilmente transformado para a forma eletrônica, pois nada mais é que uma ordem de pagamento assinada. Se lembrarmos da existência das assinaturas feitas com certificados digitais essa transformação se revelará imediatamente.

c) cartão de crédito: Outra forma de pagamento muito utilizada é o cartão de crédito, que se baseia na garantia de pagamento da instituição administradora do cartão. Das três formas de pagamento discutidas até agora, é a única que se baseia em crédito ao invés de débito; a administradora não requer um depósito anterior ao gasto, e sim cobra posteriormente a conta dos gastos. O cartão de crédito, no entanto, é ainda mais caro que o cheque, e seu uso está limitado a uma parcela ainda menor da população.

O cartão de crédito não precisa existir fisicamente para a realização de uma compra. Através de métodos como a Transação Eletrônica Segura (SET ou *Secure Electronic Transaction* — ver [18]) ou de protocolos como o SSL é possível transmitir o número do cartão de crédito entre usuário, vendedor e administradora, completando a transação de forma *on-line*, o que viabiliza essa forma de pagamento para ser utilizada imediatamente no comércio eletrônico.

d) boleto bancário: Existe ainda um último sistema de pagamento muito utilizado no Brasil, o boleto bancário. Esse é um sistema de cobrança bancária tradicional que foi adaptado ao comércio eletrônico de três formas diferentes:

1) o comprador pede o produto, espera pela chegada do boleto pelo correio, paga no banco e envia o comprovante ao vendedor ou espera que ele confirme o pagamento. Esse é o método mais lento e antigo, já praticamente não utilizado;
2) o comprador pede o produto, imprime o boleto, paga no banco e envia o comprovante ou aguarda a confirmação. Esse método é mais ágil, mas ainda depende do tempo necessário para ir ao banco e para a confirmação do pagamento;
3) o comprador pede o produto e paga o boleto através de um serviço bancário *on-line*. Esse é o método mais rápido, porém a confirmação de recebimento do pagamento pelo vendedor ainda pode demorar de forma proibitiva.

B. *Micropagamentos*

Os micropagamentos são uma nova categoria de sistema de pagamento, proposta inicialmente por volta de 1992 e que se beneficiou muito do desenvolvimento de novas aplicações para funções de *hash* como SHA [10] e MD5 [13] por volta de 1996, com as propostas de Rivest [15], Pedersen [11] e Anderson [3]. Essas propostas serão analisadas mais detalhadamente na seção III.

Micropagamentos, como o próprio nome já indica, são pagamentos de valores muito pequenos. O problema que os sistemas tentam resolver é como fazer esses pagamentos a custos ainda mais baixos que os já baixos valores dos pagamentos que realizam. Os sistemas de micropagamentos são intrinsecamente eletrônicos, pois se destinam originalmente para uso apenas em comércio eletrônico. Assim, sua adaptação a esse meio é perfeita.

Um dos maiores problemas que esses sistemas têm que resolver, como conseqüência de sua natureza totalmente eletrônica, é a interface com o mundo financeiro real. A solução desse problema muitas vezes é deixada pelos autores para outros protocolos.

Os sistemas de micropagamentos envolvem normalmente três entidades: o usuário, que é aquele que quer realizar uma compra, normalmente uma pessoa física diante de um navegador da Internet; o vendedor, que é a empresa ou pessoa que está vendendo seu produto ou serviço através da rede e o agente, que é a instituição — normalmente uma instituição financeira — responsável pelo sistema, por emitir as fichas e/ou garantir o seu pagamento, por manter as contas de usuários e vendedores atualizadas, enfim, por manter tudo funcionando. Alguns sistemas apresentam também um quarto elemento, um agente aglutinador de pagamentos, que funciona como um atravessador para evitar os custos associados a baixos volumes de transações.

Os sistemas de micropagamentos são normalmente classificados em duas categorias: *on-line* e *off-line*.

Sistemas *on-line* são aqueles em que o agente participa diretamente da transação, autorizando a compra no mesmo momento em que ela é realizada. Esse tipo de sistema é mais seguro pois a possibilidade de ocorrência de fraudes é menor, uma vez que o agente pode verificar instantaneamente a autenticidade do pagamento que está sendo feito. Por outro lado, a necessidade de verificação de cada pagamento por parte de uma entidade central traz três desvantagens: o atraso causado por essa verificação pode ser grande, tornando-se um fator de inviabilidade do sistema; a carga computacional sobre a entidade central pode ser muito grande se o número de usuários do sistema crescer demasiadamente e a entidade central passa a ser um ponto único de falha.

Os dois últimos problemas enumerados acima podem ser contornados com um maciço investimento em *hardware*, para a obtenção de um sistema altamente escalável e redundante. Nesse caso, apenas o problema da latência restaria — o que já é suficiente para que a concorrência dos sistemas *off-line* se torne marcante.

Sistemas *off-line* são aqueles em que o agente não participa diretamente de cada transação. Seu papel é, normalmente, restrito a uma fase inicial antes da realização das compras — confecção de fichas, assinatura de certificados que garantem crédito e outras funções inicializadoras — e ao posterior pagamento aos vendedores das fichas gastas e eventuais cobranças aos usuários que se fizerem necessárias. A verificação da autenticidade da transação é feita diretamente pelo vendedor, com posterior revalidação por parte do agente.

Esse tipo de sistema tem a vantagem de apresentar transações mais rápidas, pois a latência do envio das mensagens ao agente é eliminada. A desvantagem é que a descentralização — o objetivo principal desse tipo de sistema — possibilita uma maior ocorrência de fraudes.

A maioria dos sistemas de micropagamento funciona com base no princípio de que se pode gerar eletronicamente uma moeda de forma única. A moeda é, na verdade, uma mensagem composta por um valor principal e outros campos que, possivelmente, autenticam o valor principal. É importante ressaltar que o valor a que nos referimos aqui não é o valor financeiro da moeda, mas o valor eletrônico que tem a propriedade de ser único. Esse valor geralmente é escolhido como sendo um número muito grande, da ordem de 2^{160}, que é o tamanho médio da saída produzida por funções como SHA. Valores dessa ordem são importantes por dois motivos: são grandes o suficiente para que sejam custosamente forjados por ataques de força bruta e são pequenos o suficiente para permitir fácil manipulação para a verificação de sua autenticidade.

As fraudes em sistemas de micropagamentos podem acontecer de várias formas, e variam bastante com a forma específica de cada sistema. Alguns tipos de fraudes, contudo, são comuns: a criação de fichas falsas e o gasto múltiplo de uma ficha, principalmente em vendedores diferentes. As medidas tomadas para resolver esses problemas têm formas semelhantes: para impedir a criação de fichas falsas são empregadas funções de mão única e assinaturas digitais em diversas combinações na criação das fichas e para impedir o gasto múltiplo são empregados esquemas de gerenciamento de conta, normalmente através do agente, mesmo que de forma tardia.

Os sistemas de micropagamentos permitem o surgimento ou a viabilização econômica de uma série de aplicações no comércio eletrônico, como por exemplo o acesso a publicações científicas, a notícias indiviuais de jornais ou a páginas pagas da Internet [20].

O projeto SCDSID (Sistema de Controle e Distribuição Segura de Imagens e Documentos, ver [17]) prevê a distribuição segura de documentos com controle de acesso. Cada documento tem propriedades associadas que podem ser controladas separadamente: visualização, impressão e cópia. O controle dessas propriedades poderia fazer com que um usuário pudesse visualizar uma imagem sem no entanto poder imprimi-la ou copiá-la, ou poderia permitir um número reduzido de visualizações e/ou impressões. O controle desses parâmetros poderia se dar através de micropagamentos: ao escolher uma imagem, o usuário envia uma ficha para cada visualização que quer fazer da imagem, e mais cinco fichas para cada impressão que quiser fazer. Se quiser fazer uma cópia da imagem ele pode então enviar cem fichas, por exemplo.

O movimento em torno dos esquemas de micropagamento foi muito grande nos meados da década de 1990 nos Estados Unidos, quando os primeiros esquemas foram propostos e o comércio eletrônico estava se tornando uma realidade estável. Desde então muitas tentativas comerciais foram feitas, mas poucas sobreviveram a longo prazo. O recrudescimento da Internet e das empresas .com no início do século XXI fez com que essas idéias voltassem ao papel.

Nos últimos anos o micropagamento tem voltado às pautas dos pesquisadores (veja [8]), e nós acreditamos que existem possibilidades de esses esquemas funcionarem no Brasil, uma vez que o nosso mercado para o comércio eletrônico se desenvolveu mais tarde e está agora em uma fase de maturidade suficiente para aceitar mudanças mais ousadas em seu meio de funcionamento.

III. Trabalhos Relacionados

A pesquisa relacionada à área de micropagamentos surgiu durante a segunda metade da década de 1990, beneficiada pelo surgimento de algoritmos de mão única, também conhecidas como funções de *hash* como as funções SHA [10] e MD5 [13]. Essas funções foram inicialmente utilizadas em esquemas de controle de acesso por Haller [4]).

A. *Sistemas originalmente propostos*

Millicent: O primeiro trabalho importante surgido na área de micropagamentos foi o Millicent [7]. Esse trabalho propõe o uso de uma mensagem chamada *scrip* ou título para a realização de compras de pequenos valores, tipicamente abaixo de 50 centavos de dólar.

O título é uma mensagem contendo: a identificação de seu proprietário, a identificação do vendedor associado, o seu valor, uma data de expiração, um número serial, uma série de propriedades que descrevem o proprietário e um certificado digital que autentica o conteúdo acima.

Para realizar uma compra, o cliente primeiro obtém um título de um agente. Esse título é usado então para adquirir, do agente, títulos específicos para cada vendedor. Os títulos são trocados entre agente e vendedor e depois repassados ao cliente. Uma vez de posse do título do vendedor, o cliente o utiliza para realizar a compra. O título é enviado para o vendedor, que deduz o valor da compra do valor do título e reenvia o título com outro valor. Dessa forma, a manutenção da conta do cliente junto ao vendedor é feita dentro do próprio título.

O Millicent se baseia em um modelo de segurança simplificado, justificado pelo fato de as transações e os títulos serem de valores muito baixos. Segundo os autores, os títulos devem ser tratados como moedas de baixo valor — daquelas que não nos importamos por perder ou manter uma contabilidade estrita. O roubo de títulos dessa natureza seria então algo não rentável, e o roubo de muitos títulos seria facilmente detectado e combatido. Essa filosofia é compartilhada por quase todos esquemas de micropagamentos, incluindo o μ**P**.

O Millicent inspirou muitos outros trabalhos, tais como o NetCard [3], o Tick Payments [11], o PayWord [15] e o iKP [5], que são semelhantes e contemporâneos, e também o SVP [19], baseado em verificação *off-line*, e seus desdobramentos.

Payword: O PayWord [15] é um esquema baseado em crédito em que o usuário recebe do agente um certificado digital contendo seu nome, endereço IP, chave pública e outras informações. De posse desse certificado o usuário pode criar e assinar seqüências de valores de *hash* específicas para cada vendedor.

A cada novo dia, quando o usuário acessa pela primeira vez o *site* do vendedor, ele cria uma nova seqüência de valores de *hash* e envia ao vendedor apenas o último desses valores, assinado digitalmente com o certificado que recebeu do agente. Para cada compra subseqüente o usuário envia ao vendedor o valor imediatamente anterior. Lembrando que a função de *hash* é uma função de mão única, sabemos que é impossível ao vendedor descobrir qual o próximo valor a ser enviado pelo usuário. Quando esse valor é recebido, basta aplicar sobre ele a função de *hash* e se chegará ao valor anterior, autenticando assim o pagamento.

Ao final de cada dia, o vendedor envia ao agente um relatório contendo os certificados recebidos de cada usuário e a última ficha gasta, a partir dos quais o agente pode fazer o pagamento que lhe é devido em um só lote.

A principal vantagem do PayWord é o fato de a autenticação de cada pagamento ser feita através da aplicação de uma função de *hash*, que é muito rápida[1]. Sua principal desvantagem está no fato de o usuário ter que gerar assinaturas digitais para cada novo vendedor que visitar a cada dia; essa carga pode ser muito pesada se o número de vendedores visitados for grande.

MicroMint: O MicroMint[2] [15] é um esquema muito interessante em que o agente gera moedas eletrônicas que podem ser trocadas livremente entre usuários e vendedores, e são garantidas contra falsificação. A tecnologia por trás dessas moedas está em produzir colisões de funções de *hash*, ou seja, encontrar k valores que produzam a mesma saída quando fornecidos como entrada a uma dessas funções. É claro que, para encontrar esses valores, é preciso executar essa função em quantidade de força bruta, e isso só é possível, em um tempo prático, com grande investimento em *hardware*. Uma vez feito o investimento, no entanto, o custo por ficha decresce ao ponto de sua produção ser economicamente viável. A falsificação, porém, continua sendo muito cara e a verificação de autenticidade das fichas é muito simples, dependendo apenas de se aplicar k vezes a função de *hash*. Uma grande vantagem do MicroMint sobre outros esquemas é o fato de não envolver assinaturas digitais, que são computacionalmente custosas, em momento algum. Embora seja simples e interessante, esse esquema é impraticável em bases experimentais, fugindo muito do escopo deste trabalho.

B. *Propostas de melhoramentos*

Alguns esquemas de micropagamentos foram propostos a partir de 1997 como melhoramentos sobre as propostas originais apresentadas acima. Entre eles estão: o protocolo NetCents [12], um melhoramento proposto sobre o MilliCent; os protocolos S, P e UPayWord [9], melhoramentos sobre o esquema PayWord; o esquema definido em [6], que usa a idéia de envio probabilístico das fichas gastas em um vendedor para o agente e os esquemas MR1, MR2 e MR3 [8], que são propostas de melhoramento sobre o esquema apresentado por Rivest em [14].

IV. Arquitetura

A. *Opções*

O desenvolvimento da arquitetura do μ**P** foi um processo iterativo de muitas fases. Muitos problemas surgiram no decorrer da solução de outros, e as soluções, por sua vez, normalmente levavam a outros problemas. Muitos problemas de engenharia apresentam essa característica; o desejável, porém, é que o processo seja convergente, ou seja, que lentamente o número de problemas seja menor que o número de soluções. Esse foi, felizmente, o caso do μ**P** . Esta seção descreve a arquitetura do sistema, mostrando como chegamos a cada uma das soluções.

A primeira decisão a ser tomada foi qual esquema subjacente seria utilizado na implementação do sistema. Alguns esquemas, como impunham a necessidade de utilização de *hardware* dedicado e foram descartados imediatamente; o uso *hardware* dedicado, embora geralmente apresente melhor desempenho, encarece muito o projeto e é inviável para uma implementação experimental como a que estávamos nos propondo a fazer. A saída foi, então, buscar um esquema que fosse totalmente implementado em *software*.

Existe ainda outro motivo para implementar o sistema totalmente em *software*: o desenvolvimento da tecnologia de semicondutores, com a conseqüente queda nos preços dos *chips* de uso geral, pode levar sistemas de software mais rapidamente a patamares

[1] Aproximadamente 100 vezes mais rápida que a verificação de uma assinatura RSA e 10.000 vezes mais rápida que a geração de uma assinatura RSA, segundo os autores.

[2] *Mint* denomina o lugar em que se forjam moedas.

elevados de eficiência do que é possível fazer com sistemas de *hardware*. Embora estes sejam intrinsecamente mais rápidos, a opção por *hardware* não-dedicado (microprocessadores como Athlon e Pentium) pode se beneficiar de seu rápido desenvolvimento e de sua grande flexibilidade.

Entre os esquemas totalmente baseados em *software* estudados, o mais simples é o PayWord [15]. Esse esquema é o mais claramente descrito por seus autores e o de mais simples implementação, pois consiste de rotinas curtas e diretas. Além disso o PayWord é, ao lado do MilliCent, um dos esquemas mais estudados na literatura. Por esses motivos decidimos iniciar a implementação pelo esquema PayWord.

Uma das opções cruciais no início de qualquer implementação é a da tecnologia a ser utilizada. Existem inúmeras formas e linguagens de programação com as quais se pode construir um sistema, e essa escolha pode afetar fatores como o tempo de desenvolvimento, o número de pessoas envolvidas no projeto e também o resultado final, uma vez que cada tecnologia tem as suas limitações. No caso do μ**P** essa escolha teve como ponto central a questão da distribuição: o sistema deveria ser construído para funcionar livremente através da Internet, sobre as múltiplas plataformas que nela se encontram, especialmente nos sistemas operacionais Microsoft Windows, Linux e MacOS, que são os principais sistemas em número de usuários no mundo. Por esse motivo foi escolhida a tecnologia Java; essa tecnologia oferece as seguintes vantagens:

- portabilidade entre as plataformas para as quais exista uma máquina virtual (existem máquinas virtuais para Microsoft Windows, Linux e MacOS);
- possibilidade de desenvolvimento uniforme de módulos cliente, servidor e programas *standalone* utilizando a mesma linguagem de programação e até as mesmas classes, o que reduz muito o tempo de desenvolvimento;
- disponibilidade de vasta documentação, exemplos e listas de discussão na Internet.

Escolhida a tecnologia, passou-se agora para a etapa de adaptar o esquema PayWord para a mesma. Nesse ponto surgiu a primeira e fundamental diferença entre o μ**P** e o esquema de Rivest: o Payword exige a computação de uma assinatura digital no usuário para cada talão de fichas emitido, e isso significa uma assinatura digital para cada vendedor para cada dia, pois um talão só tem validade por um dia em um vendedor. O cálculo de uma assinatura digital nessa quantidade pode não parecer muito se assumirmos que um usuário não deve visitar muitos vendedores por dia (essa é a hipótese dos autores do PayWord), mas nossa decisão foi de eliminar a necessidade dessa assinatura por dois motivos:

1) mesmo sendo ocasional, o cálculo de uma assinatura digital no usuário pode ser um grande problema, pois no ambiente distribuído da Internet não se pode contar com capacidade de processamento nas máquinas clientes; a diversidade encontrada nesses equipamentos é muito grande, e um bom projeto deve contemplar ao máximo essa diversidade;
2) a tecnologia Java apresenta séria restrição para o cálculo de assinaturas digitais nos clientes. Isso se deve ao fato de a versão da máquina virtual distribuída com o navegador mais popular do mercado, o Microsoft Internet Explorer, ser de uma versão que ainda não tinha as classes necessárias para o cálculo de assinaturas digitais. A versão mais atual da máquina virtual tem essas classes, mas seu *download* pode levar um tempo proibitivo em lugares com conexão lenta. O fato de ser distribuída uma máquina virtual antiga com um navegador moderno se deve exclusivamente a disputas de mercado entre a Microsoft e a Sun Microsystems. Assim, para que o sistema abarcasse o maior número de clientes possíveis, as *applets* desenvolvidas não poderiam calcular assinaturas digitais.

O próximo aspecto a ser discutido na arquitetura do μ**P** foi um ponto levantado como fraco no esquema PayWord: a necessidade de criação de um conjunto independente de fichas para cada vendedor. É certo que essa forma de funcionamento facilita bastante a implementação do sistema, como veremos a seguir, mas ela também apresenta uma desvantagem: implica no cálculo de um novo conjunto de fichas para cada vendedor visitado a cada dia, o que pode ser muito custoso computacionalmente para ser executado em *applets* nos computadores dos usuários.

Assim, a primeira idéia que surgiu para o μ**P** foi a idéia de fazer o que se chamou um sistema de micropagamentos específicos: o sistema seria desenhado para funcionar diretamente ligado a um vendedor; o conjunto de fichas geradas seria único e somente poderia ser gasto com aquele vendedor. Esse seria um sistema de implementação ainda mais fácil, e poderia ser utilizado como forma de pagamento em sistemas como o SCDSID ou outros sistemas que necessitassem de uma forma específica de controle de pagamentos. Seu funcionamento se basearia no conceito de um cartão telefônico pré-pago: compra-se da empresa um número de créditos que se utilizará então somente nos aparelhos daquela empresa. Essa idéia funciona bem se a empresa que está vendendo os créditos tem força suficiente de mercado para que se saiba que todos ou quase todos os créditos comprados serão gastos mais cedo ou mais tarde. No entanto ao pensarmos em uma pequena empresa ou mesmo pessoa física vendendo conteúdo digital na Internet, a idéia de compra de créditos específicos cai por terra: dificilmente um consumidor compraria créditos que só poderia utilizar em um *site* da Internet se não tivesse a certeza de visitar muito freqüentemente esse *site*.

A partir desse problema surgiu então a idéia de fazer o que se chamou de um sistema de micropagamentos genéricos: um sistema em que apenas um conjunto de fichas fosse criado, que pudesse ser utilizado em qualquer um dos vendedores cadastrados. Um esquema desse tipo se assemelha muito à emissão de uma moeda por parte do Estado: a força do emissor determina a aceitação ou não aceitação da moeda; a força da moeda depende da situação econômica do emissor; a utilização da moeda pelo público depende dos dois fatores anteriores e de um mais sutil, a confiança. Fica evidente que um sistema de micropagamentos genéricos é mais ousado e complexo, mas também que era a única alternativa que o projeto μ**P** poderia tomar para manter-se coerente com suas propostas. O sistema de micropagamentos específicos, no entanto, ainda poderia ser útil no caso de vendedores com grande força de mercado que quisessem utilizar o μ**P** em suas operações; ficou decidido, assim, por criar um sistema híbrido que possibilitasse as duas configurações, de acordo com o desejo do vendedor ao se cadastrar.

O sistema de micropagamentos genéricos imaginado funcionaria da seguinte forma: um único conjunto de fichas seria gerado para cada usuário no momento de seu cadastramento no sistema. Quando visitasse vendedores cadastrados o usuário poderia enviar as fichas para realizar compras. As fichas enviadas teriam então que ser verificadas localmente em cada vendedor. Como as fichas são valores de saída de uma função de *hash*, elas têm que ser verificadas através da aplicação da função sobre elas e posterior comparação com valor subseqüente. O importante nesse ponto é notar que o valor subseqüente na cadeia gerada já tem que estar armazenado no vendedor para que ele possa realizar a comparação. Assim ficou criado o problema de como enviar aos vendedores a ficha raiz (que é na verdade a última ficha da cadeia de *hash*) através da qual os pagamentos seriam autenticados.

A primeira abordagem para a solução desse problema foi a da criação de um certificado do vendedor, que seria enviado a todos os vendedores no momento da criação do certificado do usuário. Assim, se o certificado do usuário contivesse uma seqüência de n

elevados de eficiência do que é possível fazer com sistemas de *hardware*. Embora estes sejam intrinsecamente mais rápidos, a opção por *hardware* não-dedicado (microprocessadores como Athlon e Pentium) pode se beneficiar de seu rápido desenvolvimento e de sua grande flexibilidade.

Entre os esquemas totalmente baseados em *software* estudados, o mais simples é o PayWord [15]. Esse esquema é o mais claramente descrito por seus autores e o de mais simples implementação, pois consiste de rotinas curtas e diretas. Além disso o PayWord é, ao lado do MilliCent, um dos esquemas mais estudados na literatura. Por esses motivos decidimos iniciar a implementação pelo esquema PayWord.

Uma das opções cruciais no início de qualquer implementação é a da tecnologia a ser utilizada. Existem inúmeras formas e linguagens de programação com as quais se pode construir um sistema, e essa escolha pode afetar fatores como o tempo de desenvolvimento, o número de pessoas envolvidas no projeto e também o resultado final, uma vez que cada tecnologia tem as suas limitações. No caso do μ**P** essa escolha teve como ponto central a questão da distribuição: o sistema deveria ser construído para funcionar livremente através da Internet, sobre as múltiplas plataformas que nela se encontram, especialmente nos sistemas operacionais Microsoft Windows, Linux e MacOS, que são os principais sistemas em número de usuários no mundo. Por esse motivo foi escolhida a tecnologia Java; essa tecnologia oferece as seguintes vantagens:

- portabilidade entre as plataformas para as quais exista uma máquina virtual (existem máquinas virtuais para Microsoft Windows, Linux e MacOS);
- possibilidade de desenvolvimento uniforme de módulos cliente, servidor e programas *standalone* utilizando a mesma linguagem de programação e até as mesmas classes, o que reduz muito o tempo de desenvolvimento;
- disponibilidade de vasta documentação, exemplos e listas de discussão na Internet.

Escolhida a tecnologia, passou-se agora para a etapa de adaptar o esquema PayWord para a mesma. Nesse ponto surgiu a primeira e fundamental diferença entre o μ**P** e o esquema de Rivest: o Payword exige a computação de uma assinatura digital no usuário para cada talão de fichas emitido, e isso significa uma assinatura digital para cada vendedor para cada dia, pois um talão só tem validade por um dia em um vendedor. O cálculo de uma assinatura digital nessa quantidade pode não parecer muito se assumirmos que um usuário não deve visitar muitos vendedores por dia (essa é a hipótese dos autores do PayWord), mas nossa decisão foi de eliminar a necessidade dessa assinatura por dois motivos:

1) mesmo sendo ocasional, o cálculo de uma assinatura digital no usuário pode ser um grande problema, pois no ambiente distribuído da Internet não se pode contar com capacidade de processamento nas máquinas clientes; a diversidade encontrada nesses equipamentos é muito grande, e um bom projeto deve contemplar ao máximo essa diversidade;
2) a tecnologia Java apresenta séria restrição para o cálculo de assinaturas digitais nos clientes. Isso se deve ao fato de a versão da máquina virtual distribuída com o navegador mais popular do mercado, o Microsoft Internet Explorer, ser de uma versão que ainda não tinha as classes necessárias para o cálculo de assinaturas digitais. A versão mais atual da máquina virtual tem essas classes, mas seu *download* pode levar um tempo proibitivo em lugares com conexão lenta. O fato de ser distribuída uma máquina virtual antiga com um navegador moderno se deve exclusivamente a disputas de mercado entre a Microsoft e a Sun Microsystems. Assim, para que o sistema abarcasse o maior número de clientes possíveis, as *applets* desenvolvidas não poderiam calcular assinaturas digitais.

O próximo aspecto a ser discutido na arquitetura do μ**P** foi um ponto levantado como fraco no esquema PayWord: a necessidade de criação de um conjunto independente de fichas para cada vendedor. É certo que essa forma de funcionamento facilita bastante a implementação do sistema, como veremos a seguir, mas ela também apresenta uma desvantagem: implica no cálculo de um novo conjunto de fichas para cada vendedor visitado a cada dia, o que pode ser muito custoso computacionalmente para ser executado em *applets* nos computadores dos usuários.

Assim, a primeira idéia que surgiu para o μ**P** foi a idéia de fazer o que se chamou um sistema de micropagamentos específicos: o sistema seria desenhado para funcionar diretamente ligado a um vendedor; o conjunto de fichas geradas seria único e somente poderia ser gasto com aquele vendedor. Esse seria um sistema de implementação ainda mais fácil, e poderia ser utilizado como forma de pagamento em sistemas como o SCDSID ou outros sistemas que necessitassem de uma forma específica de controle de pagamentos. Seu funcionamento se basearia no conceito de um cartão telefônico pré-pago: compra-se da empresa um número de créditos que se utilizará então somente nos aparelhos daquela empresa. Essa idéia funciona bem se a empresa que está vendendo os créditos tem força suficiente de mercado para que se saiba que todos ou quase todos os créditos comprados serão gastos mais cedo ou mais tarde. No entanto ao pensarmos em uma pequena empresa ou mesmo pessoa física vendendo conteúdo digital na Internet, a idéia de compra de créditos específicos cai por terra: dificilmente um consumidor compraria créditos que só poderia utilizar em um *site* da Internet se não tivesse a certeza de visitar muito freqüentemente esse *site*.

A partir desse problema surgiu então a idéia de fazer o que se chamou de um sistema de micropagamentos genéricos: um sistema em que apenas um conjunto de fichas fosse criado, que pudesse ser utilizado em qualquer um dos vendedores cadastrados. Um esquema desse tipo se assemelha muito à emissão de uma moeda por parte do Estado: a força do emissor determina a aceitação ou não aceitação da moeda; a força da moeda depende da situação econômica do emissor; a utilização da moeda pelo público depende dos dois fatores anteriores e de um mais sutil, a confiança. Fica evidente que um sistema de micropagamentos genéricos é mais ousado e complexo, mas também que era a única alternativa que o projeto μ**P** poderia tomar para manter-se coerente com suas propostas. O sistema de micropagamentos específicos, no entanto, ainda poderia ser útil no caso de vendedores com grande força de mercado que quisessem utilizar o μ**P** em suas operações; ficou decidido, assim, por criar um sistema híbrido que possibilitasse as duas configurações, de acordo com o desejo do vendedor ao se cadastrar.

O sistema de micropagamentos genéricos imaginado funcionaria da seguinte forma: um único conjunto de fichas seria gerado para cada usuário no momento de seu cadastramento no sistema. Quando visitasse vendedores cadastrados o usuário poderia enviar as fichas para realizar compras. As fichas enviadas teriam então que ser verificadas localmente em cada vendedor. Como as fichas são valores de saída de uma função de *hash*, elas têm que ser verificadas através da aplicação da função sobre elas e posterior comparação com valor subseqüente. O importante nesse ponto é notar que o valor subseqüente na cadeia gerada já tem que estar armazenado no vendedor para que ele possa realizar a comparação. Assim ficou criado o problema de como enviar aos vendedores a ficha raiz (que é na verdade a última ficha da cadeia de *hash*) através da qual os pagamentos seriam autenticados.

A primeira abordagem para a solução desse problema foi a da criação de um certificado do vendedor, que seria enviado a todos os vendedores no momento da criação do certificado do usuário. Assim, se o certificado do usuário contivesse uma seqüência de n

fichas, o certificado do vendedor conteria a ficha $n+1$ e informações sobre o usuário a quem pertenceria aquela ficha. Ao realizar a primeira compra, o usuário enviaria ao vendedor a ficha n; sobre ela o vendedor aplicaria uma vez a função de *hash* e obteria então a ficha $n+1$, que já estaria armazenada localmente. Se as fichas fossem iguais a compra estaria autorizada, e a ficha n seria então armazenada no vendedor. Na próxima compra, o usuário enviaria a ficha $n-1$, cujo *hash* é a ficha n, que agora também está armazenada e que portanto autentica a ficha $n-1$ enviada, e assim por diante.

Esse esquema parece funcionar muito bem, mas o que acontece se a ficha $n-1$ for enviada a outro vendedor, no qual a ficha n não foi gasta e portanto não está armazenada? A solução para esse problema seria aplicar duas vezes a função de *hash* sobre a ficha $n-1$, chegando então à ficha $n+1$ que consta do certificado do vendedor e que está, portanto, armazenada localmente. Assim, a ficha agora tinha que estar acompanhada por um número, a que se chamou ordem da ficha, que representa o número de vezes que a função de *hash* tem que ser aplicada sobre aquela ficha para que se chegue à ficha raiz. Ao receber uma ficha bastaria ao vendedor aplicar k vezes a função de *hash* sobre ela, onde k é o número de ordem enviado junto com a ficha, e compará-la com a ficha raiz constante no certificado daquele usuário.

Embora essa questão parecesse resolvida, persistia ainda um problema: a transmissão dos certificados a todos os vendedores no momento da geração de um certificado de usuário. Os certificados de usuários são normalmente gerados a intervalos de tempo razoáveis, mas existem alguns fatores que podem levar à geração de um número considerável de certificados:

1) o número de usuários do sistema pode simplesmente crescer, e com isso o número de certificados iniciais gerados também cresceria;
2) os usuários podem perder seus certificados (apagando-os por engano, ou tendo que formatar seus discos, por exemplo), e isso não pode significar que perderam os créditos pelos quais já pagaram. Assim, é preciso haver uma possibilidade de re-geração de certificados, o que pode aumentar ainda mais o número de certificados gerados;
3) se um usuário possui um certificado com p ficha ainda não gastas e pretende realizar uma compra de $p+q$ fichas, ele precisará adquirir ao menos mais q fichas. Isso significa a geração de um novo certificado, contendo agora ao menos $p+q$ fichas. Esse fato pode ocorrer com freqüência, levando a um crescimento ainda maior no número de certificados gerados.

Podemos perceber agora que o número de certificados gerados por dia pode ser muito grande. Se considerarmos que a aceitação do sistema crescerá, então devemos supor que o número de vendedores cadastrados também crescerá. O número de envios de certificados de vendedores será igual ao número de certificados de usuários gerados multiplicado pelo número de vendedores cadastrados. Esse custo de comunicação pode ser proibitivamente alto.

Uma alternativa para realizar essa comunicação seria o uso de *multicast*, ou seja, apenas um envio da mensagem que alcançaria todos os vendedores. Há, no entanto, duas razões para que não se possa utilizar essa tecnologia nesse caso: a comunicação tem que ser segura (o que não é possível em *multicast*) e a recepção do certificado tem que ser confirmada, pois um vendedor sem certificado não poderia autenticar a compra de um cliente válido, o que seria uma falha inadmissível do sistema.

A alternativa escolhida para a solução do problema de comunicação foi a seguinte: ao invés de enviar certificados contendo a ficha raiz aos vendedores no momento da geração dos certificados dos usuários, a ficha raiz seria enviada junto com a ficha de compra a cada transação. Ao receber uma ficha para compra, o vendedor recebe junto com ela a ficha raiz, aplica sobre a ficha de compra k vezes a função de *hash* e compara com a ficha raiz que acabou de receber. Para que isso funcione, no entanto, a ficha raiz tem que apresentar uma garantia: a assinatura digital do agente. Ao gerar o certificado do usuário, o agente gera um pequeno certificado de vendedor contendo a última ficha da seqüência, assina esse certificado e o insere como mais um campo no certificado do usuário. Quando vai fazer uma compra, o usuário envia a próxima ficha a ser gasta e esse certificado, que contém a ficha raiz. Ao receber a mensagem o vendedor verifica a assinatura digital do certificado que contém a ficha raiz através da chave pública do agente e pode assim autenticar a compra.

O inconveniente da solução apresentada acima é o fato de que o vendedor tem que verificar uma assinatura digital a cada transação. Esse fator certamente aumenta o atraso na conclusão de uma compra, mas ainda não é tão grande quanto a criação de uma assinatura digital. Além disso, pode-se assumir, como em [8], que o desenvolvimento tecnológico de *hardware* e *software* fez com que essa operação seja hoje mais acessível e que o equipamento dos vendedores é, em média, bastante superior em capacidade ao dos usuários, podendo suportar essa operação com razoável eficiência.

Proteção contra fraudes: O próximo problema a ser definido na arquitetura do μ**P** foi o seu esquema de proteção contra fraudes. Do ponto de vista do agente, que é quem vai pagar aos vendedores pelas fichas gastas apresentadas por eles, existem dois tipos de fraudes que podem ocorrer: a fraude do vendedor e a fraude do usuário.

A fraude do vendedor consiste na possibilidade de o vendedor criar e apresentar ao agente uma ficha válida que não tenha sido gasta em seu *site*. Essa fraude se tornou possível a partir do ponto em que as fichas podem ser gastas seqüencialmente em vendedores diferentes. É computacionalmente inviável ao vendedor descobrir a ficha n a partir da ficha $n+1$, pois $n+1$ corresponde a $h(h^n(x))$, onde h é a função de *hash*, escolhida justamente por ser uma função de mão única. No entanto, analisemos a seguinte situação:

1) o usuário gasta a ficha n com o vendedor A;
2) o usuário gasta a ficha $n-1$ com o vendedor B;
3) sabendo o valor da ficha $n-1$, B pode facilmente descobrir o valor da ficha n e apresentar essa ficha como tendo sido gasta em seu *site*.

Nessa situação, A e B apresentam ao agente a mesma ficha, pedindo em troca o seu valor em dinheiro. Se não houver nenhuma informação extra, o agente não terá como saber quem é o fraudador, e será obrigado a pagar ambas as fichas, arcando com o prejuízo.

A fraude do usuário, por sua vez, se baseia no fato de que o certificado do usuário, que contém todas as suas fichas e a ficha raiz assinada pelo agente, está armazenado localmente em seu computador. Para maior segurança, decidiu-se por armazenar esse certificado em forma criptografada. A criptografia utilizada é simétrica e a chave é derivada da senha do usuário. Isso impede que um usuário tenha acesso ao certificado do outro, mas não impede que um usuário mal-intencionado tenha acesso ao próprio certificado (desde conheça a forma de funcionamento do sistema, o algoritmo de criptografia utilizado, a forma de gerar a chave e o funcionamento geral do sistema) e introduza modificações que lhe sejam favoráveis, como por exemplo voltar o ponteiro que indica a próxima ficha a ser gasta; dessa forma o usuário pode gastar a mesma ficha n em vários vendedores diferentes. Nesse caso os vendedores apresentarão fichas que, em sua visão, foram legitimamente gastas em seus *sites*, obrigando o agente a honrar os pagamentos.

Outro tipo de fraude que o usuário pode cometer se baseia na geração de certificados: o usuário deve ter o direito de gerar um novo certificado em caso de perda. O novo certificado invalida o anterior e repõe as fichas não gastas. O problema, no entanto, é

que o gasto e a geração de fichas se dão um lugares diferentes, o primeiro nos vendedores e a segunda no agente. Assim, se o agente ainda não sabe que uma determinada ficha já foi gasta por um usuário que está pedindo um novo certificado, ele pode gerar o novo certificado contendo essa ficha, que será gasta de forma legítima.

A geração de fichas além das que já existem dentro de uma seqüência é computacionalmente inviável também para o usuário, pois implica em descobrir a inversa da função de *hash*, e a geração de uma nova cadeia de fichas esbarra na impossibilidade de criação de uma assinatura digital válida sobre a ficha raiz, que só pode ser feita com a chave privada do agente.

Para se proteger das fraudes possíveis, o μ**P** se baseia em duas premissas, da mesma maneira que [7]:

- os valores envolvidos são muito baixos e não compensam o trabalho e custo envolvidos em burlar o sistema;
- a repetição de pequenas fraudes pode ser facilmente detectada e o fraudador descoberto e punido.

Apesar de considerar as premissas acima, o desenvolvimento de uma arquitetura de micropagamentos tem que contar com um esforço para diminuição de fraudes já previstas; um sistema que prevê a ocorrência de fraudes, embora conceitualmente correto, pode ter dificuldade em angariar a confiança dos seus participantes principais, os usuários e os vendedores. Em termos de *marketing*, um sistema que admite fraudes pode facilmente se tornar um fracasso.

Para que o agente impeça alguns tipos de fraudes e detecte outros, é necessário que ele faça a verificação das fichas gastas nos vendedores. Essa verificação pode ser feita de quatro formas diferentes:

1) ***on-line*** - a cada transação, quando recebe a ficha do usuário, o vendedor pode consultar diretamente o agente a respeito da validade da ficha. Isso garante proteção total contra fraudes, uma vez que o agente sabe de todas as fichas gastas e pode negar a autorização para uma transação com ficha já gasta. Essa opção já havia sido descartada anteriormente, no entanto, por dois motivos: impõe um atraso muito grande a cada transação e introduz um ponto único de falha imediata para todo o sistema.
2) ***off-line* com pesquisa** - o agente consulta regularmente os vendedores acerca das fichas gastas em cada um deles, e repete essa consulta, restrita às fichas gastas por um usuário específico, quando esse usuário pede um novo certificado. Esse esquema não impede que as fraudes sejam feitas mas permite que o agente, através do cruzamento dos dados dos vendedores, detecte a sua ocorrência. Não seria possível, no entanto, diferenciar entre a geração de fichas pelos vendedores e o gasto duplo por parte dos usuários. O problema que esse esquema apresenta é a enorme carga computacional e de comunicação necessária para realizar a pesquisa, o que pode inserir um atraso muito grande na geração de certificados e também exigir que o agente possua um equipamento muito caro para fazer todo o processamento ao fim de cada dia.
3) ***quase on-line*** - a cada transação o vendedor envia para o agente informações sobre a compra realizada (usuário, ficha recebida, ordem da ficha, valor da compra e horário), porém *sem aguardar a autorização do mesmo*. Esse esquema proporciona o conhecimento imediato acerca das transações por parte do agente sem, contudo, implicar em atraso na autenticação da compra, ou seja, é imperceptível para o usuário. Sua desvantagem é que a quantidade de mensagens chegando no agente ao mesmo tempo pode ser muito grande nos horários de pico de compras; o tamanho das mensagens é, no entanto, muito pequeno e seu processamento é simples. Embora não impeça o duplo gasto de fichas por parte de usuários fraudulentos, esse esquema detecta esse tipo de fraude rapidamente, podendo então cancelar a validade do certificado do usuário, como veremos adiante, e possibilita maior facilidade na distinção entre esse tipo de fraude e a fraude do vendedor, com base no horário de recebimento das fichas.
4) **em lotes** - nesse esquema o vendedor envia as fichas gastas para o agente em lotes, em períodos fixos ou variáveis de tempo. A configuração do período de tempo entre as remessas permite grande flexibilidade ao sistema: se o tempo é curto o sistema funciona como na opção 3, com o agente recebendo informações a cada transação; se o tempo é mais longo os custos de comunicação e processamento são menores e, em contrapartida, a possibilidade de fraudes aumenta. A configuração do tempo entre os envios dos lotes pode ser feita de duas formas: manual ou dinâmica. Na configuração manual o intervalo de tempo é um parâmetro do sistema que pode ser editado manualmente em um arquivo de configuração, e na configuração dinâmica existe um algoritmo que determina esse intervalo de tempo de acordo com a movimentação atual de transações no vendedor, seguindo a idéia de [6].

A solução escolhida para o μ**P** foi a número 3, quase *on-line*, mas decidimos manter a possibilidade de realizar o envio das fichas em lotes, como na solução 4. A utilização de um esquema ou de outro depende apenas de um parâmetro configurável em tempo de execução nos vendedores. O agente, por sua vez, não vê diferença entre as duas formas, apenas processa as fichas recebidas através de uma *servlet* sem saber se essas fichas foram gastas imediatamente ou algum tempo antes.

CRL: Até agora discutimos como o agente pode saber da ocorrência de fraudes e precaver-se contra elas. Há dois aspectos, entretanto, que necessitam maior esclarecimento: como os vendedores podem se precaver contra fraudes e como o agente pode punir os vendedores ou usuários fraudulentos.

Os vendedores recebem fichas que vêm de certificados dos usuários, acompanhadas de fichas raiz assinadas que as autenticam. Cada certificado contém apenas uma ficha raiz. Existem alguns casos nos quais um certificado deixa de ser válido:

- quando o usuário cometeu uma fraude;
- quando o usuário requisitou um novo certificado, seja por ter perdido o antigo, seja para comprar mais fichas.

A requisição de um novo certificado, mesmo que o antigo não esteja expirado e contenha fichas não gastas, deve revogar (ou invalidar) automaticamente todos os certificados anteriores, de forma que haja apenas um certificado de micropagamentos genéricos válido a qualquer instante.

Quando um certificado é revogado, no entanto, ele pode ainda estar armazenado no computador de um usuário. A informação sobre a sua revogação está somente no banco de dados do agente. Como a ficha raiz que acompanha o certificado também está armazenada no computador do usuário, este ainda é capaz de realizar uma compra, pois o vendedor ainda não possui a informação de que o certificado não está válido e a ficha raiz é autêntica quando assinada pelo agente.

Assim, é preciso encontrar uma forma de avisar aos vendedores quando um certificado é revogado. Em sistemas de segurança os certificados revogados são mantidos em uma lista chamada CRL (*Certificate Revokation List* ou lista de revogação de certificados). A CRL, no entanto, deve ser acessada por todos os vendedores no momento da verificação de uma ficha. Para atingir esse objetivo foram estudadas três propostas:

1) o vendedor verifica a CRL junto ao agente a cada transação; essa solução é inviável pois implica em um atraso muito grande sobre a compra, diminuindo muito a eficiência do sistema;

2) o agente avisa todos os vendedores sobre a entrada de um novo certificado na CRL; essa solução implica em um custo muito grande de comunicação, que aumenta com o aumento do número de certificados gerados;
3) o vendedor verifica a CRL junto ao agente de forma periódica; essa é uma solução de compromisso entre o custo de comunicação e a imunidade a fraudes: quanto maior for o intervalo, menor será o custo e maior a probabilidade de que um certificado que já tenha sido revogado seja utilizado para uma transação. Essa foi a solução escolhida para o sistema **μP**.

Integração: O próximo passo a decidir sobre a arquitetura do **μP** foi a sua integração a outros sistemas; como fazer para que, ao final de um processo de compra, o produto seja realmente entregue ao usuário?

O processo de compra acaba quando o vendedor compara a ficha raiz com o resultado da n-ésima aplicação da função de *hash*. Esse resultado é binário: se forem iguais a compra é autorizada e, se não forem, é cancelada. Como quem controla a compra do lado do usuário é uma *applet*, ela pode receber de volta a resposta do vendedor e redirecionar o navegador do usuário para a página do produto, caso a transação tenha sido autorizada.

A página do produto, no entanto, não pode estar disponível abertamente, pois nesse caso bastaria a um usuário mal-intencionado descobrir esse endereço e utilizar a página sem passar pelo processo de pagamento. Assim, concluímos que o produto a ser vendido deve estar inserido em uma página dinâmica, que verifica se o usuário que está pedindo a página está ou não autorizado a obtê-la.

O problema que surge nesse ponto é como será implementada essa página dinâmica. Páginas desse tipo podem ser implementadas com as mais diversas tecnologias, e não é a intenção de projeto de micropagamentos restringir a forma de montagem de uma página de produto somente pelo fato de que ele tem que ser pago através desse sistema.

Assim, optamos por utilizar uma solução de integração que pode ser adequada de forma simples a qualquer implementação de página dinâmica: o banco de dados. Ao finalizar o processo de compra e verificar a autenticidade da ficha, o vendedor simplesmente gera uma autorização aleatória e insere essa autorização no banco de dados, em uma tabela específica e com validade de alguns segundos ou minutos. Dessa forma, a *applet* tem apenas que receber a autorização e redirecionar o navegador para a página do produto acrescentando ao cabeçalho HTTP a autorização recebida. A página dinâmica que fornece o produto ao usuário tem somente que consultar o banco de dados e verificar se a autorização é válida.

B. Funcionamento

A partir das opções listadas na seção anterior, pode-se resumir o **μP** através das seguintes características:

- é totalmente baseado em *software*;
- implementado com tecnologia Java em clientes e servidores;
- é capaz de realizar micropagamentos específicos ou micropagamentos genéricos, dependendo da vontade do vendedor sobre a forma sob a qual quer operar;
- realiza a autenticação da transação de forma *off-line*;
- tem proteção moderada contra fraudes, que funciona no esquema *quase on-line* ou em lotes;
- possibilita a integração com outros sistemas através de banco de dados.

Componentes: O sistema **μP** é composto por cinco elementos:

1) a **ficha** é um valor de 20 *bytes* codificado em Base64 e que possui um número serial ou de ordem associado;
2) o **certificado** é um conjunto de fichas seqüenciais (cada ficha é o resultado da aplicação da função de *hash* sobre a ficha anterior; a ficha inicial é aleatória) e ordenadas, e contém os seguintes elementos:
   - um número serial;
   - uma data de expiração;
   - o nome do usuário ao qual pertence;
   - o nome do vendedor ao qual se destina (a palavra-chave *genérico* indica um certificado que pode ser gasto com qualquer vendedor que se cadastre nessa categoria);
   - o nome do agente;
   - o número total de fichas do certificado;
   - o valor individual das fichas;
   - um ponteiro indicando qual a próxima ficha a ser gasta;
   - a própria seqüência de fichas;
   - um valor de verificação de integridade do conteúdo, que nada mais é que um *hash* simples do mesmo e
   - a ficha raiz da seqüência, assinada digitalmente pelo agente para garantia de sua autenticidade e codificada em Base64;
3) o módulo do **usuário**;
4) o módulo do **vendedor** e
5) o módulo do **agente**.

e) Certificado: O certificado é gerado pelo agente a pedido do usuário e é armazenado no banco de dados do primeiro e no disco do segundo. O certificado é armazenado no disco local do usuário em forma criptografada, com a chave de criptografia simétrica sendo gerada a partir da senha do usuário.

A cada momento existe apenas um certificado válido por usuário por vendedor. Como o sistema **μP** permite a geração de certificados para micropagamentos genéricos ou específicos, pode haver certificados de micropagamentos específicos para cada vendedor cadastrado sob essa forma, além de um único certificado de micropagamentos genéricos.

f) Usuário: O usuário interage com o sistema **μP** em três momentos: o de seu cadastramento, o da compra e o do pedido de novo certificado.

O cadastramento é feito através do acesso direto a uma página do agente que fornece esse serviço, através de uma conexão segura via SSL. No momento do cadastramento o usuário fornece a senha com a qual serão criptografados os seus certificados.

Quando vai fazer uma compra o usuário tem que baixar e executar uma *applet* que tem as seguintes funções:

1) procurar no disco local pelo certificado do usuário;
2) decriptografar o certificado e verificar a sua integridade;
3) verificar se o certificado possui o número de fichas não-gastas necessário para a compra;

4) em caso positivo, enviar a próxima ficha — de acordo com o valor da compra — para o vendedor, receber deste uma autorização e redirecionar o navegador do usuário para a URL do produto, anexando a autorização; em caso negativo, redirecionar o navegador para uma página de pedido de novas fichas no *site* do agente;
5) enviar, junto com cada ficha enviada ao vendedor, a ficha raiz, para que o vendedor autentique a ficha junto a esta.

Se o usuário não está cadastrado no sistema μ**P** , não possui localmente o certificado ou não possui o número de fichas necessário para a compra, a AppletUsuario redireciona seu navegador automaticamente para o agente, onde ele tem a opção de comprar um novo certificado ou gerar novamente o certificado que já possui.

g) Vendedor: O vendedor interage com o sistema em três momentos: o de seu cadastramento, o da venda de um produto, quando recebe e autentica uma ficha, e o do envio de informações sobre movimentação para o agente.

O cadastro do vendedor é o momento em que ele fornece seus dados para o agente e define um parâmetro crucial: se quer utilizar o sistema de micropagamentos específicos ou o sistema de micropagamentos genéricos. Essa interação não foi implementada; por enquanto, é feita manualmente através da manipulação de registros no banco de dados do agente.

No momento da transação, quando recebe uma ficha — que vem sempre acompanhada da ficha raiz assinada pelo agente, o vendedor realiza as seguintes tarefas:

- verifica a assinatura digital da ficha raiz;
- se a ficha não é autêntica, cancela imediatamente a transação, enviando de volta para a *applet* o código -1;
- se a ficha raiz é autêntica, calcula k vezes a função de *hash* sobre a ficha enviada, onde k é o número de ordem da ficha, e compara o resultado com a ficha raiz. Se forem iguais a transação é aprovada e o número de autorização é gerado; em caso contrário a transação é cancelada;
- em caso de transação realizada, envia para o agente a ficha gasta e outras informações pertinentes. Esse envio é configurável, podendo ser imediato (maior custo e maior resistência a fraudes) ou em lotes (menor custo e menor resistência a fraudes);
- insere a transação realizada em sua tabela local de histórico.

O envio de informações de movimentação para o agente pode acontecer de forma *quase on-line* ou em lotes. No primeiro caso a informação é enviada para o agente imediatamente após a autorização de uma transação; no segundo caso, é enviada em intervalos de tempo determinados por um algoritmo no programa de envio, que depende basicamente da quantidade de movimentação financeira realizada no último intervalo de tempo.

h) Agente: O agente atua no sistema como elemento central: é ele o emissor das fichas, quem coleta o dinheiro dos usuários em troca delas no início do processo e quem paga os vendedores em troca das fichas gastas ao final do processo. Ao agente cabe manter um registro de gastos atuais de cada usuário, além de uma cópia dos certificados e fichas emitidos, para que possa verificar a ocorrência de fraudes e tomar as medidas cabíveis: revogação de certificados e suspensão de usuários ou vendedores fraudulentos.

O agente recebe a informação de cada ficha gasta de duas maneiras diferentes, que podem ser escolhidas na configuração do sistema.

A primeira maneira é o envio imediato, por parte do vendedor, de cada ficha gasta; essa maneira tem a vantagem de manter o agente sempre atualizado com relação às fichas gastas no sistema, diminuindo muito a possibilidade de fraudes, sob o custo de aumentar a carga de comunicação e de computação do servidor do agente. É importante ressaltar, no entanto, que o envio da ficha se dá independentemente da resposta da transação ao usuário. Essa funcionalidade foi implementada através de uma nova *thread* (linha de execução), o que possibilita que a resposta da transação para o usuário não seja atrasada por esse processo.

A segunda maneira é o envio das fichas gastas em lotes; de tempos em tempos (esse intervalo também é configurável) o vendedor envia um lote com as últimas fichas gastas. Esse esquema concentra o envio de fichas, diminuindo os custos de comunicação e computação do servidor do agente, mas tem a desvantagem de que o agente fica desatualizado sobre gastos de fichas enquanto esses gastos ainda não foram enviados.

A atualização do agente sobre os gastos é importante por dois motivos: para a geração de novos certificados e para a detecção de fraudes.

A geração de novos certificados pode se dar quando o usuário necessita de mais fichas do que possui para realizar uma compra ou quando o usuário perdeu o certificado (isso pode acontecer quando ele muda de computador ou formata seu disco, por exemplo). O agente leva sempre em conta as fichas já possuídas pelo usuário na hora de gerar um novo certificado, mesmo que ele tenha pago por mais fichas. Assim, se o usuário possui p fichas e comprou mais q fichas, seu novo certificado terá $p + q$ fichas. Como essa contagem considera apenas as fichas ainda não gastas, é importante que o agente saiba o mais cedo possível que uma ficha foi gasta.

A detecção de fraudes acontece quando uma ficha é gasta mais de uma vez, ou quando uma ficha não existente na base de dados é gasta. Em ambos os casos, o certificado ao qual aquela ficha pertence é colocado em uma lista de rejeição (CRL - *Certificate Revokation List*), que é consultada de tempos em tempos pelos vendedores. Também por esse motivo é importante que o agente saiba quando uma ficha foi gasta.

Processos: O sistema μ**P** apresenta os seguintes processos de funcionamento: cadastramento, geração de fichas, transação de compra de produto e informe e verificação de transações.

a) cadastramento: O sistema μ**P** não foi projetado originalmente para prover anonimidade aos usuários. Cada usuário que utiliza o sistema deve fornecer inicialmente seus dados, incluindo dados únicos como o CPF, que pode ser verificado junto à Receita Federal para impedir duplicações e falsidade ideológica.

Além de dados pessoais como nome, endereço e telefone, o usuário tem que fornecer também um nome de usuário (também conhecido como *login* ou *username* - uma palavra que o identifica unicamente no sistema) e uma senha, que será utilizada em todas as operações de compra ou geração de certificados para garantir a sua identidade e também na criptografia de seu certificado local, para garantir sua privacidade. Por esse motivo, o processo de cadastramento deve ser feito sobre uma conexão segura SSL.

O cadastramento deve também ser realizado pelos vendedores. Embora não esteja implementado, no cadastramento os vendedores devem informar ao agente, além de outros dados, se querem utilizar o sistema de micropagamentos específicos ou genéricos.

b) geração de fichas: O processo de geração de fichas acontece quando o usuário precisa de fichas para realizar uma compra e não as tem em número suficiente; isso pode se dar caso o usuário possua um certificado com número insuficiente de fichas, caso tenha perdido seu certificado ou caso ainda não possua um certificado.

O processo pode ser acompanhado abaixo:

- **pedido de fichas** Durante essa fase o usuário informa ao agente o seu nome de usuário, sua senha e o número de fichas que quer comprar.

- **seqüencia de pagamento** Essa fase envolve protocolos de pagamento externos ao sistema em questão; é a fase em que o usuário utiliza seu cartão de crédito, por exemplo, para pagar por todas as fichas que está adquirindo. Como essa fase não faz parte do escopo do projeto, ela não será detalhada. Para maiores informações podem ser consultadas as referências [18] e [2].
- **geração de fichas** A geração de fichas é feita pelo agente através dos seguintes passos:
  1) confere a senha fornecida pelo usuário com a que está armazenada no banco de dados;
  2) consulta no banco de dados o número de ordem k da próxima ficha não gasta;
  3) gera um número aleatório e a partir dele a seqüencia de fichas, com ordem começando em k;
  4) assina digitalmente a última ficha da sequência, gerando um certificado de vendedor, e insere esse certificado ao final do certificado do usuário;
  5) revoga o último certificado válido do usuário;
  6) insere o certificado gerado no banco de dados;
  7) disponibiliza o certificado para ser retirado pelo usuário em um diretório temporário.
- **entrega** A entrega do certificado é feita através da carga de uma *applet* que recupera o certificado disponibilizado na fase anterior e o grava no disco local do computador do usuário, em um local pré-determinado.

c) compra de produto: A compra de um produto é a principal etapa de funcionamento do sistema, e pode ser acompanhado pelos passos 1 a 8 abaixo:

1) A transação começa quando o usuário demonstra interesse em adquirir um produto, clicando em um *link* na página do vendedor. Supõe-se que o usuário chegou a essa página através de outros meios.
2) O sistema do vendedor retorna ao usuário uma página que contém a *applet* do sistema μ**P** . Essa *applet* é um o componente principal do módulo do usuário.
3) A *applet* busca o certificado local segundo as regras de nomenclatura, pede a senha ao usuário, decriptografa o arquivo e verifica se o certificado está válido e se possui o número necessário de fichas para fazer a compra. Em caso negativo, redireciona o navegador do usuário para uma página em que é oferecida a opção de compra de fichas ao usuário, passando assim ao passo A.
4) Caso o certificado seja válido, envia ao vendedor a ficha, sua ordem k, a ficha raiz assinada e outras informações pertinentes à compra: número serial do certificado, nome de usuário, nome do produto e valor da compra.
5) O vendedor verifica se o certificado não foi revogado, verifica a autenticidade da ficha raiz, calcula k vezes a função de *hash* sobre a ficha e compara o resultado com a ficha raiz.
6) Se os valores forem iguais, gera um número aleatório de autorização, insere esse número no banco de dados e o transmite de volta ao usuário; se forem diferentes, retorna ao usuário uma autorização de valor -1, indicando que a transação foi cancelada. Logo após o envio da autorização para o usuário, o vendedor transmite informações sobre a transação (nome do usuário, nome do vendedor, ficha gasta, ordem da ficha, valor da compra e número da autorização) para o agente.
7) De posse da autorização, a *applet* redireciona o navegador do usuário para a página do produto, submetendo conjuntamente o número da autorização.
8) A página dinâmica que fornece o produto ou serviço adquirido verifica a autorização recebida junto ao banco de dados; se ela corresponder ao produto correto e estiver ainda válida, o produto ou serviço é então enviado ao usuário.

d) informe e verificação de transações: As informações sobre cada transação podem ser enviadas imediatamente após a verificação das mesmas, como descrito acima, ou em lotes a intervalos de tempo definidos. A utilização de uma opção ou da outra depende de parâmetros configuráveis no vendedor.

Um programa *standalone* é responsável por varrer a tabela de histórico do vendedor e verificar se há fichas não enviadas; em caso positivo ele as envia em um só lote. O tempo que esse programa levará até a próxima varredura pode ser fixo ou variável. No caso de intervalo variável, o tempo entre as varreduras pode ser definido através da quantidade de movimentação ocorrida no último período. Embora esse esquema atrase a varredura sempre em um período em relação à movimentação, ele pode ajudar a manter o agente mais atualizado em períodos prolongados de grande movimentação.

No agente o programa que recebe esses dados é uma *servlet* que pode receber as informações de fichas avulsas ou em lotes, permitindo que esse parâmetro seja configurável apenas nos vendedores.

C. Comparação

Após o processo de definição da arquitetura do μ**P** , observamos que o resultado final é consideravelmente diferente da arquitetura do PayWord, em que foi baseada. As diferenças entre os dois sistemas são mostradas resumidamente na tabela I.

V. Resultados

A avaliação do μ**P** será feita de acordo com a estrutura proposta em [16]. Essa estrutura prevê uma avaliação qualitativa de nove parâmetros, separados em três dimensões: a dimensão microeconômica, a dimensão tecnológica e a dimensão social.

Na dimensão microeconômica os autores levantaram três parâmetros: baixo custo de transação, transações atômicas e a base de usuários. Os custos de transação são um parâmetro muito difícil de avaliar, pois envolvem um estudo detalhado de todos os custos fixos e marginais envolvidos na implantação de um sistema; assim, serão considerados apenas os custos de computação e comunicação envolvidos em cada transação. Transações atômicas são um requisito importante de funcionamento do sistema, e significam que os pagamentos têm que ser completos; a troca de bens por pagamentos também deve acontecer, ou seja, não se pode dar pagamento sem receber um bem e não se pode receber um bem sem dar pagamento. A base de usuários, por fim, representa o alcance econômico do sistema; quanto maior o número de pessoas que utilizarem o sistema, mais útil ele será para essas pessoas.

Na dimensão tecnológica foram levantados quatro parâmetros: segurança, confiabilidade, escalabilidade e latência. Segurança implica em confidencialidade, integridade e autenticação de mensagens, pois os sistemas funcionam em redes abertas. Confiabilidade depende da disponibilidade do sistema ou do risco de uma tentativa de utilização do sistema em determinado momento ser frustrada por razões intrínsecas ao sistema. Escalabilidade é a capacidade de crescimento do sistema, mantendo os outros requisitos nos mesmos patamares. Latência é o atraso apresentado pelo sistema em cada transação, que deve ser mantido pequeno mesmo em horários de pico.

PayWord	μP
um certificado para cada vendedor para cada dia	certificado único para todos os vendedores com validade configurável; aceita também certificados específicos por vendedor, porém com validade estendida
uso de assinaturas digitais no computador do usuário	usuário não calcula nenhuma assinatura digital
verificação da ficha através de aplicações da função de *hash*	verificação da ficha através de aplicações da função de *hash* e de uma verificação de assinatura digital
fichas geradas pelo usuário	fichas geradas pelo agente
vendedor informa ao agente somente a última ficha gasta de cada seqüência	vendedor informa ao agente sobre todas as fichas gastas em seu *site*
adequado para relacionamento freqüente entre usuário e vendedor	adequado para relacionamento ocasional entre usuário e vendedor
agente compara fichas com certificados gerados pelo próprio usuário	agente compara fichas com base de dados central
baseado em crédito	baseado em debito (pré-pago)

TABELA I

COMPARAÇÃO ENTRE O μP E O PAYWORD

Na dimensão social foram considerados dois parâmetros: pagamentos inter-usuários (ou *peer-to-peer*) e o grau de anonimidade oferecido pelo sistema. Os pagamentos inter-usuários são importantes para que o sistema de pagamentos proposto possa atingir um grau maior de aceitação social, pois uma economia real não funciona somente entre pessoas e estabelecimentos comerciais. O grau de anonimidade é importante para que o sistema tenha maior aceitação do público, embora isso seja, segundo os autores, controverso e discutível.

Para a avaliação os autores propõem o estudo detalhado do sistema em questão, principalmente no que se refere ao processo de transação de compra, e a atribuição de conceitos que variam entre "−−" para requisito não atendido, "−" para requisito mal atendido, "o" para não decidido, "+" para requisito bem atendido e "++" para requisito muito bem atendido.

Passaremos agora à avaliação do μP segundo esses critérios:

Baixo custo de transação

Quando consideramos uma transação, podemos tranqüilamente ignorar o envio de informações sobre essa transação ao agente e nos concentrarmos apenas no que concerne à troca de informações entre usuário e vendedor. Fazemos isso porque a troca de informações se dá após a conclusão da transação e pode ser configurada para ser feita de forma independente em lotes.

Assim, o custo de uma transação, em termos computacionais, é o custo de uma verificação de assinatura digital e de i aplicações da função de *hash*, onde i é a diferença entre o número de ordem k da ficha recebida e o número de ordem j da ficha raiz. O número de aplicações da função de *hash* é portanto variável, fazendo com que o custo de cada transação não seja determinístico. Mais ainda, esse número pode variar com o próprio tamanho do certificado que o usuário comprou.

Apesar disso, a aplicação de uma função de *hash* é muito rápida, mesmo repetida muitas vezes. Um motivo para atrasar a transação, no entanto, é a verificação da assinatura digital. Por esse motivo, avaliamos o custo de transação de nosso sistema com um conceito "+".

Transações atômicas

O sistema μP foi desenhado e implementado de forma que a ficha não seja gasta sem que a autorização seja recebida. Se a **AppletUsuario** receber uma autorização igual a -1 ou não receber uma autorização, ela não atualizará o certificado do usuário, mantendo o estado da ficha como não gasta. A atomicidade da transação, no entanto, não prevê a entrega final do produto, que pode não acontecer por outros motivos, como por exemplo uma queda abrupta de conexão. A transação do μP acaba no momento da recepção da autorização. Assim, podemos considerar que apenas uma parte da transação é atômica, o que resulta em um conceito "o".

Base de usuários

A base de usuários do μP ainda não existe e não pode ser avaliada. O conceito adequado é, portanto, "o".

Segurança

O sistema μP não apresenta, na implementação experimental, recursos para prover autenticidade, integridade ou confidencialidade para as mensagens trocadas entre os módulos. A decisão de não prover esse recurso foi tomada para simplificar a implementação funcional do sistema, deixando para desenvolvimentos futuros e para outros protocolos (como SSL, por exemplo) a tarefa de prover esses recursos.

O controle de fraudes do sistema é ainda falho e depende de atuação humana para a detecção de alguns tipos de fraudes, como por exemplo a geração de fichas falsas por vendedores. Embora esse modelo tenha sido seguido para atingir uma solução de compromisso entre custo de implementação e segurança, a sua avaliação continua sendo negativa: o conceito atribuído é "−".

Confiabilidade

Como a transação é verificada apenas no vendedor, a confiabilidade do sistema fica distribuída entre os vários vendedores participantes. Alguns problemas podem ocorrer quanto à utilização de *applets* Java nos computadores dos usuários, mas tomaremos por base a situação em que elas são autorizadas e funcionam normalmente. Um ponto central de falhas surge no momento da compra de fichas, que no entanto não faz parte da transação de compra propriamente dita. Assim, vamos considerar o sistema como semi-distribuído e atribuir a ele um conceito "+".

Escalabilidade

Até este momento não foi possível realizar testes de escalabilidade do sistema implementado. Consideraremos então a escalabilidade teórica, que se baseia no fato de o agente centralizar todas as operações, exceto a transação. Assim, embora no aspecto da compra o sistema seja distribuído, ele é bastante concentrado no que tange à administração.

A base de dados do agente deve conter todas as fichas geradas para todos os certificados de todos os usuários para todos os vendedores (embora consideremos que a maioria dos vendedores aceitará o sistema de micropagamentos genéricos), além do histórico de todas as transações realizadas. A dependência de uma base de dados assim centralizada, que só pode ser acessada através de um único servidor, depõe contra a escalabilidade do sistema, deixando-o com um conceito "–".

Latência

A latência da transação é o tempo necessário para a verificação de uma assinatura digital e de um número variável de aplicações da função de *hash*, como descrito no item **Baixos custos de transação**. Embora seja um componente de atraso, a assinatura digital é, com a tecnologia atual, verificada em tempo razoável ([8]), embora muito maior que as aplicações de funções de *hash*. Assim, a latência média do μ**P** é determinada por essa verificação, o que resulta em um conceito "+".

Pagamentos inter-usuários

O μ**P** não apresenta a menor possibilidade de implantação de uma forma de pagamento inter-usuários. Pela forma como foi implementado, existe a necessidade de instalação de um conjunto de *servlets* e de uma base de dados para receber pagamentos, o que é inviável para esse tipo de pagamento. Assim, o conceito atribuído para esse requisito é "não se aplica".

Anonimidade

Da forma como está implementado, o sistema não tem nenhum grau de anonimidade: o usuário tem que se cadastrar, fornecendo até o seu CPF, para poder comprar as fichas. Existe, no entanto, uma possibilidade de implementação de anonimidade através da venda de cartões pré-pagos com nomes de usuário e senha pré-definidos para cada cartão. Embora seja um trabalho futuro, acreditamos que essa possibilidade pode ser um interessante valor a ser acrescentado ao sistema, e que eleva em um ponto o seu conceito quanto a esse quesito, que fica então sendo "–".

Resultados

Os resultados da avaliação analítica do sistema μ**P** podem ser vistos na tabela II. Uma tabela semelhante é apresentada em [16] para os sistemas DigiCash, First Virtual e SET. Uma comparação entre os conceitos atribuídos para esses trabalhos pelos autores e os conceitos aqui atribuídos ao μ**P** nos leva a uma avaliação extremamente positiva deste trabalho, posicionando-o como uma alternativa viável para sistemas desse tipo.

	Requisito	μ**P**
Microeconômicos	Baixos custos de transação	+
	Transações atômicas	o
	Base de usuários	o
Tecnológicos	Segurança	–
	Confiabilidade	+
	Escalabilidade	–
	Latência	+
Sociais	pagamentos inter-usuários	não se aplica
	Anonimidade	–

TABELA II

AVALIAÇÃO FINAL DO μ**P**

VI. Conclusão

O comércio eletrônico brasileiro está em fase de franca expansão, apesar dos percalços econômicos gerados por instabilidades externas e internas. Essa expansão leva a novas possibilidades de negócios, entre eles uma tendência intrínseca que leva à venda de informação em formato digital. Esse modelo de negócio apresenta características peculiares que requerem a implantação de um novo sistema de pagamentos que atenda aos requisitos de baixo custo e baixa latência.

Os sistemas de micropagamentos surgiram como resposta a essa tendência em países desenvolvidos e são agora uma aposta em países em desenvolvimento como o Brasil, em função da maturidade atingida pela nossa economia digital. Sistemas de micropagamentos apresentam as características necessárias para transações de baixos valores através de operações simples e razoavelmente seguras. O modelo de segurança de sistemas desse tipo é simplificado, representando uma solução de compromisso entre os custos de implementação e o nível de segurança necessário para as transações que realiza.

O μ**P** é um sistema de micropagamentos experimental e totalmente funcional, desenvolvido a partir do sistema PayWord proposto em [15]. Sua característica principal é o fato de utilizar uma moeda eletrônica gerada em uma instituição central mas verificada de forma simples em outras localidades através do uso mínimo de operações computacionalmente intensas. O μ**P** se diferencia por utilizar apenas um conjunto de fichas que permitem realizar compras em qualquer vendedor cadastrado. Sua interface Web desenvolvida em Java possibilita uma distribuição ampla através dos diversos sistemas que compõem o universo de clientes e vendedores da Internet, realizando nessas máquinas um mínimo de operações computacionalmente intensas.

Embora fosse inicialmente baseado em um sistema existente, o μ**P** se aproveitou de idéias propostas em muitos sistemas e se mostrou, ao final do processo de criação e desenvolvimento, como um sistema novo e único, apresentando vantagens consideráveis sobre outros sistemas propostos, especialmente no que concerne à sua potencial adaptação à realidade do comércio eletrônico no Brasil.

Referências Bibliográficas

[1] Albertin, A. L. *Comércio Eletrônico*, 2a ed. Atlas, São Paulo, 2000.

[2] Albrecht, C., Malone, S., He, B., and Bombard, L. An analysis of ssl and set for electronic commerce.

[3] Anderson, R., Manifavas, C., and Sutherland, C. Netcard – a practical electronic cash system.

[4] Haller, N. M. The S/KEY one-time password system. In *Proceedings of the Symposium on Network and Distributed System Security* (1994), pp. 151–157.

[5] Hauser, R., Steiner, M., and Waidner, M. Micro-payments based on iKP. Tech. Rep. 2791 (# 89269), 1996. Relatório técnico.

[6] Jarecki, S., and Odlyzko, A. An efficient micropayment system based on probabilistic polling. *Lecture Notes in Computer Science 1318* (1997), 173–191.

[7] Manasse, M., Glassman, S., Abadi, M., Gauthier, P., and Sobalvarro, P. The millicent protocol for inexpensive electronic commerce, 1995.

[8] Micali, S., and Rivest, R. L. Micropayments revisited. In *CT-RSA* (2002), pp. 149–163.

[9] Mu, Y., Varadharajan, V., and Lin, Y.-X. New micropayment schemes based on paywords. In *Australasian Conference on Information Security and Privacy* (1997), pp. 283–293.

[10] NIST. Fips publication 180: Secure hash standard (shs)., Maio 1993. National Institute of Standards and Technology.

[11] Pedersen, T. P. Electronic payments of small amounts. In *Security Protocols Workshop* (1996), pp. 59–68.

[12] Poutanen, T., Hinton, H., and Stumm, M. NetCents: A lightweight protocol for secure micropayments.

[13] Rivest., R. The md5 message-digest algorithm., Abril 1992. Request For Comments: 1321.

[14] Rivest, R. L. Electronic lottery tickets as micropayments. In *Financial Cryptography* (Anguilla, British West Indies, 1997), R. Hirschfeld, Ed., Springer, pp. 307–314.

[15] Rivest, R. L., and Shamir, A. Payword and micromint: Two simple micropayment schemes. In *Security Protocols Workshop* (1996), pp. 69–87.

[16] Schmidt, C., and Müller, R. A framework for micropayment evaluation.

[17] Schweitzer, C. M., Mindlin, P. A. L., Margi, C. B., Ruggiero, W. V., and Carvalho, T. C. M. B. Sistema para controle de distribuição segura de imagens. Tech. rep., São Paulo, 2001. Relatório técnico LARC - PCS - EPUSP / Scopus.

[18] SET. Secure electronic transaction llc. Set Specification Books.

[19] Stern, J., and Vaudenay, S. SVP: A flexible micropayment scheme. In *Financial Cryptography* (1997), pp. 161–172.

[20] Yen, S.-M., and Ku, P.-Y. Improved micro-payment scheme. Tech. Rep. TR-98-2, 1998.

BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

BT/PCS/01/07 – Um Mecanismo para Distribuição Segura de Vídeo MPEG – CÍNTIA BORGES MARGI, GRAÇA BESSAN, WILSON VICENTE RUGGIERO

BT/PCS/0108 – A Dependence-Based Model for Social Reasoning in Multi-Agent Systems – JAIME SIMÃO SICHMAN

BT/PCS/0109 – Ambiente Multilinguagem de Programação – Aspectos do Projeto e Implementação – APARECIDO VALDEMIR DE FREITAS, JOÃO JOSÉ NETO

BT/PCS/0110 – LETAC: Técnica para Análise de Tarefas e Especificação de Fluxo de Trabalho Cooperativo – MARCOS ROBERTO GREINER, LUCIA VILELA LEITE FILGUEIRAS

BT/PCS/0111 – Modelagem ODP para o Planejamento de Sistemas de Potência – ANIRIO SALLES FILHO, JOSÉ SIDNEI COLOMBO MARTINI

BT/PCS/0112 – Técnica para Ajuste dos Coeficientes de Quantização do Padrão MPEG em Tempo Real – REGINA M. SILVEIRA, WILSON V. RUGGIERO

BT/PCS/0113 – Segmentação de Imagens por Classificação de Cores: Uma Abordagem Neural – ALEXANDRE S. SIMÕES, ANNA REALI COSTA

BT/PCS/0114 - Uma Avaliação do Sistema DSM Nautilus –MARIO DONATO MARINO, GERALDO LINO DE CAMPOS

BT/PCS/0115 – Utilização de Redes Neurais Artificiais para Construção de Imagem em Câmara de Cintilação – LUIZ SÉRGIO DE SOUZA, EDITH RANZINI

BT/PCS/0116 – Simulação de Redes ATM – HSU CHIH WANG CHANG, WILSON VICENTE RUGGIERO

BT/PCS/0117 – Application of Monoprocessed Architecture for Safety Critical Control Systems – JOSÉ ANTONIO FONSECA, JORGE RADY DE ALMEIDA JR.

BT/PCS/0118 – WebBee – Um Sistema de Informação via WEB para Pesquisa de Abelhas sem Ferrão – RENATO SOUSA DA CUNHA, ANTONIO MOURA SARAIVA

BT/PCS/0119 – Parallel Processing Applied to Robot Manipulator Trajectory Planning – DENIS HAMILTON NOMIYAMA, LÍRIA MATSUMOTO SATO, ANDRÉ RIYUITI HIRAKAWA

BT/PCS/0120 – Utilização de Padrão de Arquitetura de Software para a Fase de Projeto Orientado a Objetos – CRISITINA MARIA FERREIRA DA SILVA, SELMA SHIN SHIMIZU MELNIKOFF

BT/PCS/0121 – Agilizando Aprendizagem por Reforço Através do uso de Conhecimento sobre o Domínio – RENÊ PEGORARO, ANNA H. REALI COSTA

BT/PCS/0122 – Modelo de Segurança da Linguagem Java Problemas e Soluções – CLAUDIO MASSANORI MATAYOSHI, WILSON VICENTE RUGGIERO

BT/PCS/0123 – Proposta de um Agente CNM para o Gerenciamento Web de um Backbone ATM – FERNANDO FROTA REDÍGOLO, TEREZA CRISTINA MELO DE BRITO CARVALHO

BT/PCS/0124 – Um Método de Teste de software Baseado em Casos Teste – SÉRGIO RICARDO ROTTA, KECHI HIRAMA

BT/PCS/0201 – A Teoria Nebulosa Aplicada a uma Bicicleta Ergométrica para Fisioterapia – MARCO ANTONIO GARMS, MARCO TÚLIO CARVALHO DE ANDRADE

BT/PCS/0202 – Synchronization Constraints in a Concurrent Object Oriented Programming Model – LAÍS DO NASCIMENTO SALVADOR, LIRIA MATSUMOTO SATO

BT/PCS/0203 – Construção de um Ambiente de Dados sobre um Sistema de Arquivos Paralelos – JOSÉ CRAVEIRO DA COSTA NETO, LIRIA MATSUMOTO SATO

BT/PCS/0204 – Maestro: Um Middleware para Suporte a Aplicações Distribuídas Baseadas em Componentes de Software – CLÁUDIO LUÍS PEREIRA FERREIRA, JORGE LUÍS RISCO BECERRA

BT/PCS/0205 - Sistemas de Automação dos Transportes (ITS) Descritos Através das Técnicas de Modelagem RM-OPD (ITU-T) e UML (OMG) – CLÁUDIO LUIZ MARTE, JORGE LUÍS RISCO BECERRA, JOSÉ SIDNEI COLOMBO

BT/PCS/0206 – Comparação de Perfis de Usuários Coletados Através do Agente de Interface PersonalSearcher – GUSTAVO A. GIMÉNEZ LUGO, ANALÍA AMANDI, JAIME SIMÃO SICHMAN

BT/PCS/0207 – Arquitetura Reutilizáveis para a Criação de Sistemas de Tutorização Inteligentes – MARCO ANTONIO FURLAN DE SOUZA, MARIA ALICE GRIGAS VARELLA FERREIRA

BT/PCS/0208 – Análise e Predição de Desempenho de Programas Paralelos em Redes de Estações de Trabalho – LIN KUAN CHING, LIRIA MATSUMOTO SATO

BT/PCS/0209 – Previsões Financeiras Através de Sistemas Neuronebulosos – DANIEL DE SOUZA GOMES, MARCO TÚLIO CARVALHO DE ANDRADE

BT/PCS/0210 – Proposta de Arquitetura Aberta de Central de Atendimento – ANA PAULA GONÇALVES SERRA, MOACYR MARTUCCI JÚNIOR

BT/PCS/0211 – Alternativas de Implementação de Sistemas Nebulosos em Hardware – MARCOS ALVES PREDEBON, MARCO TÚLIO CA.RVALHO DE ANDRADE

BT/PCS/0212 – Registro de Imagens de Documentos Antigos – VALGUIMA VICTORIA VIANA ODAKURA MARTINEZ, GERALDO LINO DE CAMPOS

BT/PCS/0213 – Um Modelo de Dados Multidimensional – PEDRO WILLEMSENS, JORGE RADY DE ALMEIDA JUNIOR

BT/PCS/0214 – Autômatos Adaptativos no Tratamento Sintático de Linguagem Natural – CÉLIA YUMI OKANO TANIWAKI, JOÃO JOSÉ NETO

BT/PCS/0215 – Fatores e Subfatores para Avaliação da Segurança em Software de Sistemas Críticos – JOÃO EDUARDO PROENÇA PÁSCOA, JOÃO BATISTA CAMARGO JÚNIOR

BT/PCS/0216 – Derivando um Modelo de Projeto a Partir de um Modelo de Análise, com Base em Design Patterns J2EE – SERGIO MARTINS FERNANDES, SELMA SHIN SHIMIZU MELNIKOFF

BT/PCS/0217 – Domínios Virtuais para Redes Móveis Ad Hoc: Um Mecanismo de Segurança – LEONARDO AUGUSTO MARTUCCI, TEREZA CRISTINA DE MELO BRITO CARVALHO

BT/PCS/0218 – Uma Ferramenta para a Formulação de Consultas Baseadas em Entidades e Papéis – ANDRÉ ROBERTO DORETO SANTOS, EDIT GRASSIANI LINO CAMPOS

BT/PCS/0219 – Avaliação de Performance de Arquiteturas para Computação de Alto Desempenho – KARIN STRAUSS, WILSON VICENTE RUGGIERO

BT/PCS/0220 – BGLsim: Simulador de Sistema Completo para o Blue Gene/L – LUÍS HENRIQUE DE BARROS CEZE, WILSON VICENTE RUGGIERO