Segurança em Sistemas de Micropagamentos Eletrônicos

Alexandre M. Braga Delano M. Beder Ricardo Dahab
Cecília M. F. Rubira

Universidade Estadual de Campinas
Instituto de Computação
Caixa Postal 6176
13081-970 Campinas SP
Fone/fax:+55+19+2393115
e-mail:{972314,delano,rdahab,cmrubira}@dcc.unicamp.br

Sumário

A criptografia é usada para proporcionar segurança em aplicações de comércio eletrônico na Internet e na geração de dinheiro eletrônico destas aplicações, em particular, nos sistemas de micropagamentos eletrônicos: pagamentos de valor muito baixo feitos muito rapidamente e a freqüências altas. No comércio eletrônico com distribuição *on-line* de produtos, transferências de valores e de produtos devem ser tratadas como transações atômicas. Nestes sistemas, mecanismos de recuperação de erros por avanço e por retrocesso são combinados para prevenir fraudes. *PayPerClick* é um sistema para venda e distribuição *on-line* de publicações na Internet baseado em micropagamentos.

Palavras-chave: criptografia, comércio eletrônico, micropagamento, orientação a objetos.

Abstract

Cryptography is used to offer security and also to generate electronic currency for electronic commerce applications over the Internet, especially for those using micropayment schemes. In such schemes, payments involving very low amounts are done quickly and frequently. An important aspect of electronic commerce systems with on-line distribution of goods is that value and goods transferences must be atomic transactions. In such systems, forward and backward error recovery are combined to prevent fraud. *PayPerClick* is a tool for electronic sale and on-line distribution of publications based on micropayments.

Key words: cryptography, electronic commerce, micropayment, object orientation.

1 Introdução

A tecnologia para pagamentos eletrônicos seguros pela Internet foi herdada da época em que a segurança de informações interessava somente aos militares, mas está sendo usada hoje para proporcionar segurança, anonimato e privacidade aos usuários de comércio eletrônico. A criptografia tem papel importante não somente na solução de problemas de segurança, mas também na geração de dinheiro eletrônico, em particular, nos micropagamentos eletrônicos [2, 3, 9, 12, 13]. O texto a seguir está organizado da seguinte forma. A Seção 2 trata dos aspectos criptográficos e transações críticas do pagamento eletrônico. Na Seção 3, os detalhes da geração de moedas para micropagamentos são descritos. O *PayPerClick*, um *software* para venda e distribuição *on-line* de publicações na Internet, é descrito brevemente na seção 4. Conclusões e trabalhos futuros são apresentados na Seção 5.

2 Criptografia, Transações e Pagamento Eletrônico

Comércio eletrônico é qualquer forma de transação de negócios na qual as partes interagem eletronicamente, em oposição ao intercâmbio físico ou contato físico direto [16]. Três entidades estão associadas a uma transação comercial eletrônica: recebedor, pagador e, opcionalmente, uma instituição financeira intermediária. O pagamento eletrônico usado em comércio eletrônico pode ser classificado em quatro categorias [15]: dinheiro eletrônico, cheque eletrônico, transferência eletrônica de fundos e cartão de crédito.

No comércio eletrônico com pagamento eletrônico e distribuição *on-line* de produtos, compras envolvem duas transações críticas: transferência de produtos e transferência de valores. A primeira faz parte da segunda e ambas devem ser transações atômicas. O conceito de transação atômica originou-se de pesquisas em gerência de banco de dados. Atualmente, ambientes de programação distribuída transacionais têm reforçado a tese que motivou o uso de ações atômicas em ambientes distribuídos: ambientes transacionais oferecem uma abstração poderosa e eficiente para a programação de sistemas distribuídos tolerantes a falhas.

Transações atômicas [5] têm três propriedades que ajudam a diminuir a complexidade da programação distribuída: (*i*) seriação: garante que a execução de programas concorrentes que compartilham objetos é livre de interferência, isto é, uma execução concorrente é equivalente a alguma execução na qual acessos a objetos compartilhados ocorrem de forma serial. A implementação de um protocolo para o controle de concorrência é necessário a fim de garantir esta propriedade de transações atômicas. (*ii*) atomicidade: garante que uma transação atômica termina somente em um de dois estados: (*a*) normal, no qual é validada e produz o resultado desejado; (*b*) anormal, na qual não produz resultados porque foi abortada. Técnicas de recuperação de erros por retrocesso [11] são utilizadas na implementação dos mecanismos que garantem esta propriedade. Falhas típicas incluem a parada de uma estação de trabalho ou falhas de comunicação. (*iii*) permanência de efeito: garante que qualquer resultado produzido por uma transação atômica não será desfeito devido a falhas. É implementada através de um armazenamento dos resultados em memória estável. Um protocolo de controle de término da transação atômica (*commit protocol*) garante que todos os objetos modificados

dentro de uma transação atômica têm o seu estado escrito em memória estável, no caso de *commit*, ou que as modificações não são gravadas, no caso de aborto.

Transações atômicas fornecem um mecanismo de tolerância a falhas com recuperação de erros por retrocesso de estado, mas há aplicações que requerem também recuperação de erros por avanço de estado. Recuperação de erros por retrocesso retorna o sistema para um estado prévio livre de erros sem requerer nenhum conhecimento dos erros. Recuperação de erros por avanço (geralmente esquemas de tratamento de exceções) é baseada no uso de dados redundantes e repara o sistema através da análise do erro detectado e colocando-o num estado correto. Na atomicidade de transferência de produtos, o dinheiro eletrônico deve ser transferido atomicamente e esta ação deve obrigar à transferência do produto [6].

Técnicas criptográficas são usadas de dois modos em sistemas de pagamento eletrônico: (*i*) como serviços criptográficos primitivos na implementação dos protocolos de segurança; (*ii*) na adaptação de protocolos já existentes em aplicações de comércio eletrônico. Dois exemplos são os sistemas de micropagamentos baseados em cadeias de *hash* [13, 9, 3], usadas anteriormente para autenticação de *passwords* em canais de comunicação inseguros [10], e aqueles baseados em sistemas para controle de cotas de uso dos recursos em um sistema distribuído [12].

Sistemas de dinheiro eletrônico são divididos em duas categorias, de acordo com o compromisso entre segurança, desempenho e valor das transações: (*i*) os baseados em funções de *hash* e (*ii*) os baseados em assinaturas digitais. No primeiro caso, os sistemas são usados em micropagamentos e baseiam-se em cadeias de *hash* propostas por Lamport [10]. Alguns sistemas também usam assinaturas digitais e certificação digital de chave pública [13, 3, 9]. Estes sistemas não garantem anonimato e o gasto repetido da mesma moeda é inibido com moedas específicas do cliente, do vendedor ou da transação. Dos sistemas baseados em assinaturas digitais, aqueles baseados em assinaturas cegas garantem o anonimato e a privacidade do usuário [7, 14].

3 Micropagamentos

Micropagamentos são pagamentos de valor muito baixo feitos muito rapidamente [4] e a freqüências altas. Esquemas de micropagamentos com muitos pagamentos repetidos tratam um número muito grande de transações em intervalos de tempo relativamente pequenos e usam protocolos criptográficos de segurança simples e computacionalmente eficientes. Moedas eletrônicas são elementos de cadeias de *hash* gerados da seguinte forma: uma semente aleatória, x, é escolhida e a cadeia $A_0, A_1, \ldots, A_{n-1}$ é computada recursivamente, como segue:

$$A_0(x) = x$$
$$A_{i+1}(x) = h(A_i(x)) \text{ e } A_i \neq A_j \text{ para } i \neq j,$$

onde h é uma função de *hash* unidirecional e com baixíssima probabilidade de colisões.

As moedas $A_0, \ldots, A_{n-1}$ permitem até n micropagamentos de um valor fixo v já estabelecido. Antes de qualquer pagamento, o pagador envia ao recebedor A_n e v de modo autêntico. O pagador garante que A_n é de fato o extremo de uma cadeia de *hash* usada em

pagamentos subseqüentes. Os micropagamentos são efetuados pelo envio, na ordem inversa, $A_{n-1}, A_{n-2}, \ldots, A_0$, de elementos da cadeia para o recebedor. A verificação dos pagamentos é feita pela reconstituição parcial de elementos da cadeia a partir do pagamento anterior, A_{n-i}, até o atual A_{n-j}, $i < j$, como abaixo:

$$A_{n-i} = h_0(h_1(h_2(...h_{k-1}(A_{n-j})))), \text{ onde } k = j - i.$$

O valor do pagamento é o produto vk. A_{n-j} assume o papel do pagamento anterior na próxima verificação. O recebedor envia para a instituição financeira intermediária a cadeia $A_{n-i}, \ldots, A_{n-j}$ para ser reembolsado de kv em dinheiro real. Com esse esquema simplificado é possível inibir furtos. O pagador não pode gerar a mesma moeda duas vezes, porque a seqüência de moedas é verificada em relação ao pagamento anterior e deve ocorrer pelo menos uma iteração da função de *hash* sobre o pagamento atual. O recebedor não pode forjar moedas e receber mais que o devido, uma vez que não é capaz de gerar elementos da cadeia para diante a partir do registro do último pagamento. Por outro lado, um intruso tem a possibilidade de fraudar o sistema de três formas: (*a*) ganhando acesso aos registros temporários de pagamentos mantidos pelo vendedor, (*b*) interceptando a comunicação entre pagador e recebedor pelo monitoramento do canal de comunicação ou observando a execução do programa de verificação das moedas e (*c*) obtendo acesso ao sistema pela obtenção ou descobrimento da senha.

A terceira forma só pode ser eliminada com uma forma de identificação baseada em características físicas intrínsecas ao usuário. Fraudes a partir da leitura de registros dos pagamentos ou pela observação da verificação não são possíveis pelos mesmos motivos citados acima. Um intruso nas condições do segundo caso tem a possibilidade de fraudar se conseguir uma moeda cuja posição na cadeia esteja à frente daquela correntemente registrada pelo recebedor como último pagamento e tiver a oportunidade de gastá-la. Esta tal moeda será válida de acordo com as verificações. Esta situação é possível após uma queda do sistema seguida de uma recuperação de erros por retrocesso, considerando pagamentos como transações atômicas, do seguinte modo: o intruso está monitorando e copiando constantemente os pagamentos em trânsito de um usuário. Eventualmente, o sistema do recebedor cai antes que o pagamento novo seja verificado e registrado. Neste caso, ocorrendo uma recuperação de erros por retrocesso, o sistema volta para o estado estável anterior, mas o intruso possui uma cópia de uma moeda que, de acordo com os registros do vendedor, ainda não foi gasta.

Supondo que o intruso possa personificar o pagador, esta fraude pode ser inibida de duas formas. Primeira, solicitando ao pagador cuja transação de pagamento não foi completada uma cadeia nova de moedas. Segunda, tirando vantagem das propriedades de sincronização da comunicação com segurança baseada em cadeias de *hash* [10]. Por exemplo, o último pagamento registrado contém a moeda A_i e o pagamento interceptado pelo intruso contém a moeda A_{i-m}, para $m > 0$. No primeiro pagamento após a queda do sistema, daquele pagador cuja transação de pagamento não foi completada, o vendedor pode requisitar mais um pagamento compulsório. Assim, o recebedor receberá a moeda A_{i-x}, para $x \geq m$, e exigirá a moeda A_{i-x-1}, possuida somente pelo pagador verdadeiro. A fraude é inibida e a perda eventual da moeda A_{i-x-1} pode ser compensada por descontos em compras subseqüentes.

Estas soluções combinam os esquemas de recuperação de erros por avanço e retrocesso. Recuperação de erros por avanço do estado inibe fraudes nas transferências de valores em sistemas de micropagamentos porque moedas interceptadas por intrusos, antes de quedas do sistema, podem ser invalidadas mais facilmente que na recuperação por retrocesso. Mas a detecção de transações incompletas e a comunicação adicional podem ter custo computacional alto em relação ao prejuízo com a fraude. Se quedas forem raras e os valores dos pagamentos muito baixos, o sistema pode tolerar furtos pequenos e pouco freqüentes.

4 Exemplo de Aplicação: *PayPerClick*

Um *software* para venda e distribução *on-line* de publicações acessíveis a partir de *web browsers* e baseado em micropagamentos eletrônicos, o *PayPerClick*, foi implementado usando técnicas de orientação a objetos. Um esquema de micropagamento baseado em cadeias de *hash*, semelhante ao apresentado em [13], foi usado. O projeto apresenta uma arquitetura composta por quatro subsistemas: transações, porta-moedas eletrônicos, controladores de vendas e composições recursivas de hiperdocumentos. A linguagem Java foi usada com o JDK 1.1.2. O *framework* criptográfico Java (JCA) [8] foi usado nas operações de geração das moedas eletrônicas, assinaturas digitais e verificação dessas. O algoritmo de assinatura digital usado foi o DSA com chaves de 512 bits. As assinaturas digitais possuem apêndice. As moedas eletrônicas foram geradas pela função de *hash* SHA sem chave. O hotjava 1.1b2 foi o *web browser* usado e o recurso de *applets* assinadas, integrado ao *browser*, proporcionou a capacidade de verificação da integridade e autenticidade do código da *applet*, tornando-o confiável. O Java *Remote Method Invocation* (RMI) [1] foi usado na implementação da comunicação baseada no modelo cliente-servidor entre os módulos do consumidor e do vendedor da aplicação.

5 Conclusões e Trabalhos Futuros

O uso comercial da Internet faz surgir várias possibilidades novas de negócios, assim como contextos novos para os problemas de segurança de informações. Segurança baseada em criptografia, técnicas de tolerância a falhas e mecanismos de recuperação de erros são partes fundamentais no desenvolvimento de *software* para comércio eletrônico baseado em micropagamentos eletrônicos e distribuição *on-line* de produtos. De fato, o *software* para comércio eletrônico só possui algum valor com a aplicação correta e combinada destas técnicas. Tarefa nada trivial e muito suscetível a erros. Em particular, na implementação de porta-moedas eletrônicos, somente a correção do modelo matemático de geração das moedas não garante a segurança da aplicação. Os serviços criptográficos disponíveis, tais como funções de *hash* e algoritmos de assinatura digital, devem não somente ser bons, mas também implementados de forma segura. *Software* no qual a segurança é crítica deve ser capaz de escolher entre algoritmos e implementações diferentes baseando-se na disponibilidade de recursos, mas sem comprometer a qualidade dos serviços de segurança.

Referências

[1] Java Remote Method Invocation. http://java.sun.com/products/jdk/rmi/index.html.

[2] Netbill: an Internet Commerce System Optimized for Network Delivered Services. IEEE CompCon Conference, Março 1995. http://www.ini.cmu.edu:80/netbill/pubs.html.

[3] Ross Anderson, Charalampos Manifavas e Chris Sutherland. Netcard — A Practical Eletronic Cash Scheme. http://www.cl.cam.ac.uk/users/rja14/.

[4] N. Asokan, Philippe A. Janson, Michael Steiner e Michael Waidner. The State of the Art in Eletronic Payment Systems. *IEEE Computer*, páginas 28–35, Setembro 1997.

[5] P. A. Bernstein, V. Hadzilacos e N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addisson-Wesley Publishing Company, 1987.

[6] L. Jean Camp, Marvin Sirbu e J. D. Tygar. Token and Notational Money in Electronic Commerce. *Usenix Workshop on Electronic Commerce*, New York, NY, Julho 1995.

[7] David Chaum. Security Without Identification: Card Computers to Make Big Brother Obsolete. *Communications of the ACM*, 28(10), Outubro 1985.

[8] Mary Degeforde. Java Cryptography Architecture API Specification and Reference. http://java.sun.com/products/JDK1.1/docs/guide/security/CryptoSpec.html, Fevereiro 1997.

[9] Steve Glassman, Mark Manasse, Martí Abadi, Paul Gauthier e Patrick Sobalvarro. The Millicent Protocol for Inexpensive Electronic Commerce. http://www.millicent.digital.com.

[10] Leslie Lamport. Password Authentication with Insecure Communication. *Communications of the ACM*, 24(11):770–772, Novembro 1981.

[11] P.A. Lee e T. Anderson. *Fault Tolerance: Principles and Practice*. Springer-Verlag, 2a. edição, 1990.

[12] B. Clifford Neuman e Gennady Medvinsky. Requirements for Network Payment: The Netcheque Perspective. *IEEE Compcon' 95*, San Francisco, Março 1995.

[13] Ronald L. Rivest e Adi Shamir. Payword and Micromint: Two Simple Micropayment Schemes. http://theory.lcs.mit.edu/rivest/~rivest/publications.html. Maio 1996.

[14] Bruce Schneier. *Applied Cryptography — Protocols. Algorithms . and Source Code in C*. John Wiley and Sons. 2a. edição. 1996.

[15] Kiyoon Sung e Jae Kyu Lee. Analysis and Design of the Internet Based Payment System. Dissertação de Mestrado. Department of Management Engineering - Graduate School of Management - Korea Advanced Institute of Science and Technology, 1996. Disponível em http://www.dcc.unicamp.br/~cripto/artigos/analysis.design.pay.system.html.

[16] Paul Timmers. Eletronic Commerce — An Introduction. http://www.cordis.lu/esprit/src/ecomint.html, Maio 1996.