Medindo o Custo de Serviços de Segurança em Sistemas de Informação Orientados a Comércio Eletrônico

Rodrigo Thiesen Flores [1], Vinicius Gadis Ribeiro [2]
[1] Dell Computer do Brasil, GDC Brazil, Av. Bento Gonçalves, 95A, Porto Alegre, RS, Brasil
rodrigo_thiesen@dell.com
[2]Centro Universitário Ritter dos Reis, Faculdade de Sistemas de Informação,
Rua Orfanotrófio, 555, Porto Alegre, RS, Brasil
vinicius@uniritter.edu.br

Resumo

Este artigo apresenta a análise do resultado da condução de um conjunto de experimentos, no sentido de se quantificar qual o custo tempo que um sistema de comércio eletrônico demanda, para prover os serviços de segurança necessários para que os usuários considerem um sistema confiável. Para tanto, foram empregadas três máquinas em situação de laboratório, e foi realizada a tomada de tempos, considerando-se aplicações simples que efetuavam comunicação de dados - simulando as mensagens entre as partes componentes de um sistema de comércio eletrônico. Buscou-se empregar as tecnologias atuais para prover serviços de segurança, e realizar medições com a ausência/presença desses serviços. Os resultados obtidos nos experimentos - considerando-se as condições de laboratório - possibilitam afirmar que o impacto, em termos de tempo, não é significativo.

Palavras-chave: Comércio Eletrônico, Segurança Computacional, Análise de Desempenho.

Abstract

This paper presents the analysis of some experiments on processing and communication times among machines that simulate an electronic commerce system. The object of this work is measure the impact to using security technologies in electronic commerce systems. The results of the experiments allow to assert the impact of the use of these technologies is minimum, considering the experiments conditions.

Key-words: Electronic Commerce, Computer Security, Performance Analysis.

1 Introdução

O projeto de desenvolvimento da Internet tinha como objetivo permitir que pesquisadores e comunidade científica de diversos locais no mundo pudessem interagir entre si, possibilitando a realização de pesquisa – embora, em seu embrião, tivesse caráter militar. Assim, não havia a necessidade de qualquer preocupação com a segurança das informações que transitavam pela rede, em razão do clima de confiança existente – razão pela qual os protocolos de comunicação não requeriam implementação de serviços serviços de segurança, exceto o serviço de integridade. Posteriormente, iniciou-se o uso da rede para fins comerciais. Necessidades comerciais incorreram, em um primeiro momento, em trocas de documentos eletrônicos – "electronic data interchange", ou EDI. Com a evolução da Internet nos últimos anos, propagaram-se diversas facilidades, proporcinando condições para o estabelecimento de comércio eletrônico (ou "e-commerce"). Essa nova funcionalidade popularizou-se rapidamente, devido aos baixos custos e grande dinamismo em operações. O comércio eletrônico atualmente pode ser divido em três tipos básicos, para fins de estudos: o Negociante Para Negociante – NPN ou, em inglês, "business-to-business" (B2B) -, Negociante Para Consumidor – NPC ou, em inglês, "business-to-consumer" (B2C) - e o Consumidor Para Consumidor – CPC, ou similar em inglês, "consumer-to-consumer" (C2C)[1, 2, 8].

Os usuários estão se adaptando muito bem à facilidade de contratar serviços ou fazer compras pela "web" [3, 12]. Porém, há a preocupação de garantir outros serviços de segurança que não a a integridade das transações, uma vez que, da mesma forma que cresce o comércio eletrônico, cresce o número de tentativas de invasões de sítios e servidores, com o intuito de apropriação indevida de recursos ou de informação, sob

várias formas[4, 15]. Essas formas são divididas em dois tipos de ataques: passivo (monitoramento de dados na transmissão, cópia da transmissão, etc.) ou ativo (interceptação e alteração da transmissão, inserção de dados falsos, etc.)[15]. Os ataques passivos são difíceis de serem detectados porque não alteram nenhuma informação, não permitindo a rápida identificação do que ocorreu. Entretanto, também é difícil prevenir-se de alguns dos ataques ativos, porque adversários estão sempre adquirindo informações para invadir um sistema ou rede, identificando novas vulnerabilidades dos sistemas, ou explorando antigas falhas. Nesse caso, obtêm-se dados para prevenção apenas após a realização do ataque.

A Internet é uma rede pública que faz o transporte de mensagens[15]. Dado o seu projeto inicial, o controle de certos serviços de segurança eram limitados, restringindo-se, principalmente, à garantia de integridade de dados. Logo, em aplicações orientadas a comércio eletrônico na Internet, pode ocorrer exposição de informações, funcionamento inadequado do sistema, espionagem, ataques diversos, seqüestro de senhas e outros tipos de danos. Não apenas o Internet está exposta: as redes privadas também estão sujeitas a essas invasões. Outros fatores que afetam a segurança no comércio eletrônico são a má administração da rede, ou dos sistemas sem controle, ou do monitoramento da segurança. Como forma de minimizar a ocorrência desses ataques e invasões, foram criados e aperfeiçoados diversos mecanismos destinados a prover outros serviços de segurança, dentre os quais a criptografia e os protocolos criptográficos, que definem regras para o transporte de mensagens pela rede de computadores.

A relevância do trabalho aqui proposto está ligada diretamente ao impacto, no desempenho, de se utilizar tecnologias de segurança no comércio eletrônico, uma vez que todas as transações eletrônicas nesse segmento exigem serviços de segurança - como privacidade, integridade, autenticidade e não-repúdio[14, 16]. Há a necessidade de se dotar tais serviços; contudo, sabe-se qual será o impacto no desempenho desses sistemas? Haverá aumento significativo no custo tempo de operações?

O presente trabalho conduziu um experimento[14, 17], considerando os dados coletados na execução de um protótipo cliente/servidor desenvolvido com base nas principais tecnologias de segurança para o comércio eletrônico e a conseqüente análise do impacto do uso das mesmas. Encontra-se dividido nas seguintes seções: a seção 2 trata da metodologia com a qual foi conduzido o experimento, bem como o funcionamento do protótipo desenvolvido; a seção 3 apresenta os resultados obtido e, na seção 4, são apresentadas as considerações finais, limitações e trabalhos futuros.

2 Metodologia

O experimento foi conduzido sob condições de laboratório - isto é, embora tendo sido considerados protocolos empregados pela Internet, os tempos de espera e outros fatores poderiam demandar análises mais complexas, possibilitando que esse trabalho possa ser conduzido, brevemente, em condições de campo.

A seguir é apresentada a arquitetura do protótipo e breve descrição de seu funcionamento. O protótipo fora desenvolvido em arquitetura cliente - em número de dois - e servidor, tendo sido possível verificar como funcionou a troca de mensagens e o comportamento e eficácia de algumas tecnologias de segurança empregadas - considerando-se uma rede exclusiva. Os equipamentos da rede tinham a mesma configuração: 15 computadores PC Pentium 4 2.2GHz, com 256 MB de memória RAM e disco rígido de 40 GB, com conexão via "switch", com placas de rede com protocolo Fast Ethernet 100Mbps. As considerações sobre o tempo já levam em conta o tempo de latência dessa sub-rede.

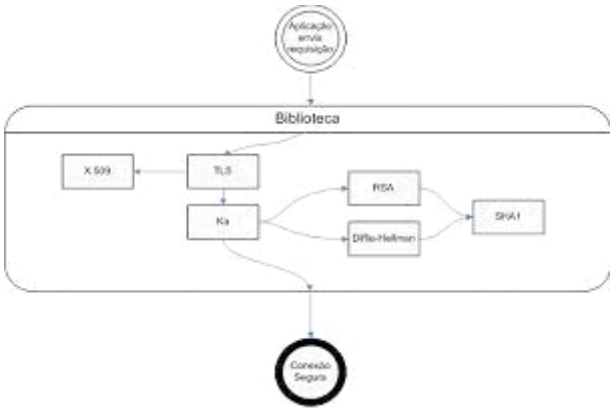

Figura 1: Arquitetura do protótipo desenvolvido

Para a arquitetura acima referida, foi desenvolvido um protótipo em "Object Pascal" - Delphi 5.0 - para a plataforma Windows; tendo sido executado no ambiente de rede local para a coleta dos dados, conforme já citado. Conforme pode-se inferir, essa arquitetura propõe o emprego de chaves públicas, gerência das últimas, cifragem e decifragem de mensagens, assim como assinatura e verificação das mensagens, vindo, assim, a fornecer os serviços de segurança[16] de autenticação, não repúdio, e privacidade – tão essenciais quando se trata de comércio eletrônico.

A sub-seção a seguir apresenta o funcionamento do protótipo, assim como de seus componentes clientes e servidor.

2.1 - Funcionamento do protótipo

A arquitetura acima propõe a utilização TLS, X.509, RSA/Diffie-Hellman, SHA-1, entretanto, este trabalho fez algumas restrições na utilização das tecnologias. Por exemplo, os algoritmos que empregam criptografia que este protótipo trabalha são RSA/Diffie-Hellman e SHA-1. O X.509 não foi implementado, porque não foi possível encontrar um servidor de certificado digital disponível para realizar a implementação desse servidor. Entretanto, prevendo

futuros trabalhos, o servidor já dispõe de uma classe preparada para trabalhar com o X.509.

- TLS: ou "Transport Layer Security", é um protocolo criptográfico que buscar prover comunicação segura na Internet. Foi derivado do Secure Sockets Layer – ou SSL -, da Netscape, e padronizado pela Internet Engineering Task Force. Após o estudo teórico do mesmo, foi implementado baseando-se na RFC2246. O TLS foi empregado em todas as mensagens com ou sem segurança e sem a necessidade do uso de todos os recursos descritos na RFC[5, 6, 7, 9].
- X.509: Padrão que serve de base para a infraestrutura de chaves públicas pela "Internet Engineering Task Force". Esse padrão foi considerado no presente trabalho, mas não implementado foi estudada essa tecnologia; entretanto, conforme já citado, não foi implementada por questão de custo econômico. As RFCS que tratam do X.509 são as RFC2459 e RFC3039 [18].
- RSA/Diffie-Hellman: são clássicos esquemas de chave pública, necessários para fornecer os serviços de privacidade e não-repúdio. Ambos os esquemas foram implementados, a fim de obter o serviço de privacidade na troca de mensagens. O servidor fornece dois parâmetros ("g" e "n") para os clientes que conectarem ao servidor. O cliente informa uma chave privada ("x/y") e o sistema calcula a chave pública que é trocada entre os clientes. Depois da troca, é calculada a chave pública comum entre os dois clientes. A chave pública é empregada no momento de cifrar a mensagem, enquanto que a chave privada é usada para decifrar a mensagem no momento em que há a troca de mensagens entre os clientes[11, 17].
- SHA-1: é um esquema para assinatura digital, necessário para fins de prover o serviço de autenticação, ou validar uma mensagem. A validação, no protótipo desenvolvido, é necessário para trabalhar a mensagem que foi cifrada pelo cliente 1 e decifrada pelo cliente 2. Quando o cliente 2 recebe a mensagem, ele decifra a mensagem e, aplicando o SHA-1 na mensagem decifrada, pode realizar a comparação das assinaturas - caso forem iguais à mensagem, então esta é considerada válida[14].

Deve ser lembrado que, durante a condução do presente estudo, não se havia ainda notícias de ataques a essa última tecnologia. Há trabalhos que indicam possíveis ataques ao RSA e, recentemente, ao SHA-1. Contudo, tais algoritmo foram escolhidos para o desenvolvimento do protótipo, em função da simplicidade, o que permite manter o foco do trabalho nas questões de desempenho e sua medição.

2.1.1 O Módulo Servidor

O módulo servidor opera da seguinte forma: o servidor disponibiliza alguns parâmetros para os clientes, faz o gerenciamento da troca de mensagens entre os clientes e possibilita salvar o "log" em um arquivo do tipo texto.

Primeiramente o servidor é inicializado, para que sejam informados parâmetros - no caso, os valores de "g" e "n" e escolher o tipo de criptografia que será usada pelos clientes. Se nenhum tipo de segurança foi selecionado, então a troca mensagens será feita de modo padrão - Diffie-Hellman. Esse servidor também trabalha como um "midleware" entre os clientes, porque ele recebe e envia mensagens de dados e confirmação. Os parâmetros "g" e "n" devem respeitar a definição do método Diffie-Hellman, podendo o valor n ser empregado, também, no método RSA - também respeitando as suas definições.

Existem três funcionalidades que se pode controlar a interface do módulo servidor, conforme se pode observar na figura 2 posterior:

- **Começar**: usado para que o servidor verifique se os campos 'g' e o 'n' estão preenchidos, se o 'g' é primo, estabelecer conexão com os clientes e fazer troca de mensagens entre os clientes. Se o servidor é inicializado corretamente, gera-se a mensagem no rodapé - "Executando...".
- **Parar**: usado para parar o serviço do servidor, ou seja, parar a comunicação entre os clientes. Nesse caso, a mensagem mostrada no rodapé é "Desconectado".
- **CHK**: usado para salvar todas as mensagens que estão no "log" da tela para um arquivo texto.

A seguir, é apresentada a interface do módulo servidor:

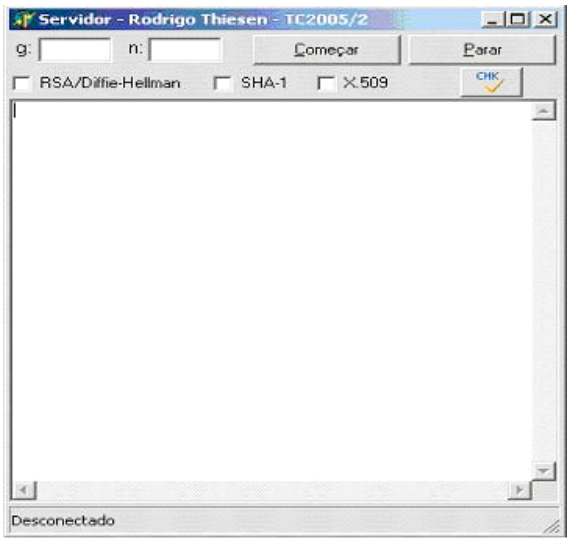

Figura 2: Interface do módulo servidor.

2.1.2 O Módulo Cliente

Com relação ao módulo cliente, seuas funcionalidades englobam o estabelecimento da conexão com o servidor, geração das chaves (pública e privada), envio e recebimento de mensagens e manutenção de um "log" em um arquivo tipo texto.

Quando da inicialização do módulo cliente, é necessário informar dois parâmetros: computador escolhido e nome, para que se possa conectar ao servidor. Quando ativado o botão CHK ao lado do nome, é estabelecida a conexão com o servidor e se recebe uma mensagem de resposta.

Após receber os parâmetros "g" e "n" e apresentar ao usuário as possíveis tecnologias de segurança (TLS, RSA, SHA-1 e X.509) que podem ser utilizadas, é apresentada a mensagem no rodapé: "Conectado a <servidor> - Versão <protocolo TLS>". Se estiver somente marcado o TLS, pode-se enviar a mensagem sem fazer a troca de chaves, porque o servidor escolheu a forma que os clientes estão trabalhando sem nenhuma outra opção de uso de tecnologia de segurança. Caso contrário, deve-se colocar a chave privada (x/y) e pressionar "Calcula". O cliente calcula a chave pública que será enviada para o outro cliente através do método criptográfico RSA, utilizando a chave privada, e os parâmetros "g" e "n". O resultado desse cálculo é enviado para o servidor, que envia para outro cliente.

O servidor recebe a mensagem e aguarda que o próximo cliente requisite a mensagem para que ele possa enviá-la. Ao receber a mensagem, o cliente calcula através do RSA para obter as chaves pública e privada que serão utilizadas por ele. A estrutura da mensagem de resposta contém as seguintes informações:

- Tipo da mensagem = "0"
- Versão do protocolo = "1.0"
- Endereço do cliente = endereço "host"
- Valor de "G" = número ímpar
- Valor de "N" = qualquer número
- RSA = Informa se será usado ou não pelos clientes
- SHA-1 = Informa se será usado ou não pelos clientes
- X.509 = Não implementando.

Nesse momento, os clientes e o servidor estão prontos para fazer a troca de mensagens com segurança, de acordo com as tecnologias adequadas que foram selecionadas no servidor. Quando um dos usuários digitar a mensagem e acionar o botão "Enviar Mensagem", o sistema aplicará todas as tecnologias de segurança que o servidor tiver escolhido - o que permite analisar o impacto individual de cada uma delas, considerando cada opção escolhida.

Se estiver utilizando RSA/Diffie-Hellman e SHA-1, o sistema vai cifrar a mensagem, utilizando a chave pública e aplicar assinatura digital na mensagem original, para que o usuário que receber a mensagem, possa verificar a autenticidade e a integridade da mesma.

Detalhes podem ser visualizados na figura 3, que apresenta a interface cliente do protótipo.

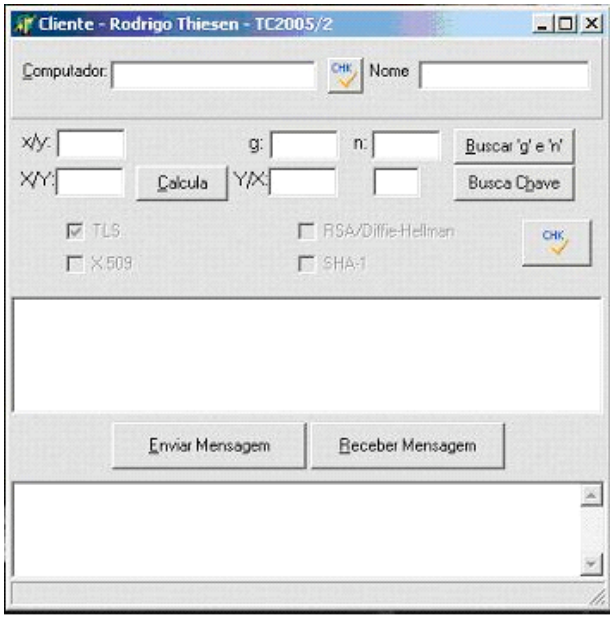

Figura 3: Interface dos módulos cliente.

Quando acionado o botão "Receber mensagem", o sistema busca a mensagem e a decifra, utilizando a chave privada e, se necessário, aplica a assinatura digital na mensagem decifrada para poder validar a assinatura da mensagem que havia recebido. O conteúdo que o cliente recebe contém a mensagem cifrada e a assinatura digital quando solicitado, sendo essa assinatura digital da mensagem recebida que é comparada com a assinatura da mensagem decifrada, para fins de verificação.

A estrutura dos dados recebidos pelo servidor é a seguinte:

- Tipo de mensagem: "1"
- Endereço do cliente: endereço do "host"
- Chave Pública: Resultado do cálculo

Outras informações relevantes sobre a comunicação o cliente/servidor:

- foi utilizada a tecnologia WinSock (API) do Delphi para estabelecer conexão. Nessa API, é realizada rotina de "handsacke" para estabelecer a conexão entre o servidor e o cliente, e todas as trocas de mensagens entre os clientes e o servidor.
- esse protótipo pode ser executado numa máquina ou em mais de uma. Quando utilizado numa máquina, é necessário definir, no campo "Computador" dos usuários, a expressão "localhost".
- a biblioteca de assinatura digital foi implementada por Dave Barton, e está liberada para ser utilizada em outros trabalhos acadêmicos.
- sempre é necessário ter ativos, no mínimo, dois clientes e um servidor para que se possa

utilizar o protótipo, para que se possa adquirir dados para posterior análise.

A vantagem de se empregar os módulos clientes descritos na forma acima é que o emprego de tal módulo tanto pode representar um consumidor, quanto um negociante.

A próxima seção apresenta os resultados obtidos na experimentação.

3 Resultados Obtidos

As condições de experimentação foram as seguintes, considerando as três situações exploradas para posterior análise de dados:

- no primeiro caso, cada protótipo cliente/servidor foi executado sem nenhum tipo de segurança.
- no segundo caso, cada protótipo cliente/servidor foi executado, utilizando Diffie-Hellman/RSA na segurança da troca de mensagens.
- e no terceiro e último caso, cada protótipo cliente/servidor foi executado na combinação de Diffie-Hellman/RSA e SHA-1 para troca de mensagens com segurança.

O TLS foi utilizado nos três casos: observou-se mínimo impacto - provavelmente, por já vir incorporado em diversas tecnologias. Em todos os casos, foram utilizadas as mesmas mensagens para tornar possível a avaliação do impacto no desempenho.

Para fazer essa verificação no protótipo, foram definidas trinta e quatro mensagens, que têm tamanhos diferentes, para que se possa coletar o tempo utilizado por cada processo e, além disso, se possa visualizar as mensagens com criptografia e assinatura digital. Cada mensagem foi submetida quarenta vezes, e os tempos apresentados nas tabelas é o tempo médio para cada variável tempo. Assim, por exemplo, o tempo mínimo apresentado na tabela representa o tempo mínimo médio, considerando a média dos tempos mínimos das quarenta execuções. Para detalhes dessas informações, ver apêndice B disponível em endereço eletrônico nas considerações finais do presente trabalho.

Em todos os três casos, foi usada a mesma informação nos parâmetros, para que se tenha o mesmo cenário nas três verificações. Após executar os testes, nos três casos, foi possível analisar os tempos de cada mensagem e identificar qual o impacto dessas tecnologias na troca de mensagens.

Essa verificação foi feita em dois momentos: no primeiro momento, utilizando uma máquina com plataforma Windows 2000 e, no segundo, utilizando três máquinas com plataforma Windows XP. Nesse último caso, uma máquina era o servidor e as outras duas operaram como máquinas clientes.

Um ponto importante é que o servidor também é utilizado como um "middleware", fornecendo os parâmetros g e n, e repassando as mensagens de um cliente para outro. Nos testes, um cliente somente enviava as mensagens e outro só recebia as mesmas.

Como já citado, três foram as situações exploradas na experimentação, cujas condições se seguem.

Caso 1 - Protótipo Sem Serviço de Segurança

Nesse experimento, o cliente/servidor não utilizou nenhuma opção de emprego de tecnologia de segurança, a fim de obter o maior desempenho nas trocas de mensagens, servido como controle. Essa verificação foi feita para se obter um tempo base médio e apresentar, de forma objetiva, o impacto do Diffie-Hellman/RSA e SHA-1 no desempenho da troca de mensagens (comércio eletrônico). Os "logs" dos módulos clientes e do módulo servidor, nessa situação, estão divididos em máquina local e máquinas em rede. Para detalhes dessas informações, ver apêndice C disponível em endereço eletrônico nas considerações finais do presente trabalho.

Caso 2 - Protótipo com Diffie-Hellman/RSA

Já nesse experimento, o cliente/servidor utilizou o Diffie-Hellman/RSA na configuração de segurança, para fazer a criptografia das mensagens. Nesse caso, foi necessária a troca de chaves entre os clientes para gerar a chave pública comum entre eles e gerar a chave privada de cada um deles. Os "logs" dos dados das máquinas de clientes e do servidor desse caso estão divididos em máquina local e máquinas em rede. Para detalhes dessas informações, ver apêndice D disponível em endereço eletrônico nas considerações finais do presente trabalho.

Caso 3 - Protótipo com Diffie-Hellman/RSA e SHA-1

Finalmente, nesse experimento, o cliente/servidor utilizou Diffie-Hellman/RSA e SHA-1 na configuração de segurança, para fazer a criptografia e assinatura digital das mensagens. Nesse caso, também foi necessário realizar a troca das chaves entre os clientes para gerar a chave pública comum entre eles e gerar a chave privada de cada um deles. Os "logs" dos clientes e do servidor desse caso estão divididos em máquina local e máquinas em rede. Para detalhes dessas informações, ver apêndice E disponível em endereço eletrônico nas considerações finais do presente trabalho.

Os testes foram executados com mensagens que têm tamanhos que variam de 10 bytes até 1339 bytes, em dois ambientes: numa máquina com Windows 2000 e numa rede com três máquinas, com Windows XP. Para detalhes dessas informações, ver apêndice B disponível em endereço eletrônico nas considerações finais do presente trabalho.

Os dados coletados nestes testes foram analisados e parametrizados. Para detalhes dessas informações, ver

apêndice F disponível em endereço eletrônico nas considerações finais do presente trabalho. A seguir, são apresentados dois quadros que contém os tempos máximo, mínimo e médio das mensagens enviadas e recebidas para os três casos descritos acima. A unidade de medida de tempo utilizada foi milésimos de segundos.

Tabela 1: Resultados do teste em Rede (tempo em milésimos de segundos)

TE	Caso 1			Caso 2			Caso 3		
	Mensagem enviada	Mensagem recebida	**Total**	Mensagem enviada	Mensagem recebida	**Total**	Mensagem enviada	Mensagem recebida	**Total**
Mín.	62	16	**93**	62	31	**93**	62	31	**93**
Máx.	125	47	**157**	79	63	**141**	93	140	**218**
Méd.	73	36	**109**	74	46	**119**	74	50	**123**

Fonte: Elaborado pelos autores, 2006

Analisando a tabela 1 do teste rede, é possível observar que o tempo mínimo foi igual nos casos dois e três, em relação ao caso 1 que não possuía configuração de qualquer tecnologia de segurança e que realizou o protocolo de modo mais rápido - como seria de se esperar. O tempo máximo mostrou que, cada vez que era incorporada uma tecnologia de segurança, aumentava o tempo de envio e recebimento de uma mensagem. Já o tempo médio mostrou que os casos 2 e 3 têm praticamente o mesmo tempo para envio e recebimento de mensagens. Contudo, o caso 3 emprega outra tecnologia de segurança, além da empregada no caso 2: o SHA-1. Assim, pode-se inferir, em um primeiro momento, que a tecnologia SHA-1 não incorre em impacto sensível com a tecnologia de medição empregada para o tempo.

A seguir, são apresentados os resultados dos tempos obtidos no teste 2, em situação de máquina local.

Tabela 2: Resultados do teste - local (tempo em milésimos de segundos)

TEM	Caso 1			Caso 2			Caso 3		
	Mensagem enviada	Mensagem recebida	**Total**	Mensagem enviada	Mensagem recebida	**Total**	Mensagem enviada	Mensagem recebida	**Total**
Mín.	60	30	**90**	60	30	**90**	60	30	**90**
Máx.	80	40	**120**	81	131	**201**	81	130	**210**
Méd.	67	34	**101**	70	41	**111**	69	41	**109**

Fonte: Elaborado pelos autores, 2006

Analisando a tabela 2 do teste local, é possível verificar que o tempo mínimo foi igual nos três casos. Os valores de tempo máximo mostraram que, cada vez que era incorporada uma tecnologia de segurança, aumentava o tempo de envio e recebimento de uma mensagem – como esperado. Já os valores de tempo médio mostram que os casos 2 e 3 têm praticamente o mesmo tempo para envio e recebimento de mensagens; entretanto, o caso 3 emprega outra tecnologia de segurança, além da empregada no caso 2: o SHA-1.

Na próxima seção, são apresentadas as considerações do presente trabalho, assim como limitações e trabalhos futuros.

5 Considerações Finais e Limitações

Para a realização do presente trabalho, fez-se um estudo sobre algumas tecnologias de segurança para troca de mensagens representando transações comerciais, e foi empregado um protótipo desenvolvido em "Object Pascal" para coletar dados que possibilitassem a análise do impacto do uso das seguintes tecnologias: TLS, RSA/Diffie-Hellman e SHA-1.

A análise dos resultados apresentados nas tabelas 1 e 2 possibilitou constatar que a utilização de segurança é viável em termos de desempenho e necessária para que haja privacidade, confiabilidade, autenticidade e integridade das partes que utilizarão as aplicações para Comércio Eletrônico. Entretanto, é preciso levantar algumas considerações: certamente, para realizar uma implementação em condições de campo, devem ser empregados algoritmos otimizados para criptografia, assinatura digital e certificados digitais; ademais, lembra-se que é imprescindível que a rede e máquinas encontrem-se corretamente configuradas para as comunicações necessárias.

Uma limitação que se deve destacar é que o trabalho foi conduzido exclusivamente em situação de laboratório. Certamente, ao se realizar experimento em campo, poder-se-á obter resultados diversos daqueles aqui apresentados. Há ainda, como limitações desse trabalho, o tamanho máximo das mensagens (em torno de 1.400 bytes), e o número de clientes que o servidor comportou (limitado a 2). Alguns fatores que provavelmente vieram a afetar o tempo de envio e recebimento de mensagens foram:

- a pequena capacidade da memória dos computadores envolvidos em cifrar e decifrar as mensagens – visto que, dependendo do tamanho usado para as chaves (privada ou pública) é necessário o emprego de mais memória, para que não haja perda de desempenho; e
- os algoritmos de criptografia, assinatura digital e certificação – a lógica aplicada no desenvolvimento desses algoritmos pode afetar no desempenho, visto que implementações

comerciais que já empregam tais algoritmos empregam técnicas de otimização dos mesmos.

Uma outra limitação que deve ser considerada é o não emprego da tecnologia X.509, por motivos de não disponibilidade, no momento da condução do experimento, de um servidor público que viabilize a utilização dessa tecnologia.

Em termos de consumo de tempo, a tecnologia SHA-1 não se apresentou onerosa. Contudo, deve-se ter em vista reservas, visto haver recentes comunicações técnicas sobre ataques a essa tecnologia.

De forma geral, considerando-se a análise dos dados obtidos, pode-se concluir que o emprego dessas tecnologias não incorreu em demanda de tempo sensível ao usuário - dadas as condições de laboratório.

Os apêndices citados no presente trabalho - os quais contém os dados coletados durante os experimentos - não poderiam ser acomodados no formato do presente trabalho, razão pela qual encontram-se integralmente disponíveis em:
http://paginas.terra.com.br/informatica/thiesenseguranca

Como sugestão de futuros trabalhos futuros, pode-se implementar esse trabalho em linguagens de ambientes de software livre – Kylix ou Java, por exemplo - para que se tenha a vantagem da portabilidade em outras plataformas e sistemas operacionais e há possibilidade de se desenvolver uma biblioteca, implementando certificação digital (X.509), e estudar outras tecnologias de segurança para analisar o impacto das mesmas, vindo-se a explorar outras situações.

Agradecimentos

Os autores agradecem o apoio do Centro Universitário Ritter dos Reis, da Dell Computers(GDC Brazil) e do trabalho de Dave Barton, cuja "home page" contém diversos recursos em Delphi - especificamente, a biblioteca que apoiou parte da implementação do protótipo, disponível na Internet por http em http://www.cityinthesky.co.uk/cryptography.html.

Referências

[1] ADAM, Nabil R.; DOMAGRACI, Oktay; GANGOPADHYAY, Aryya. *Electronic Commerce.* Upper Saddle River: Prentice Hall PTR, 1999.

[2] ADDIE, R.G. *e-Comerce Security Architecture*. Disponível na Internet por http em: http://www.researchindex.com. Acesso em 21.set. 2005.

[3] BRASIL. Ministério da Ciência e Tecnologia (MCT). Disponível na Internet por http em: http://www.mct.gov.br/Temas/info/Imprensa/Noticias_5/Comercio_5.htm. Acesso em 30.nov.2005.

[4] BERNSTEIN, Terry; TERRY, BHIMANI; Anish B.; SCHULTZ, Eugene. ***Segurança na Internet.*** Rio de Janeiro: Campus, 1997.

[5] BLAKE-Wilson, S.; NYSTRON, M.; HOPWOOD, D; MIKKELSEN, J. *RFC 3546 (Draft) Transport Layer Security (TLS) Extensions*. 2004. Disponível na Internet por http em: www.ietf.org/rfc. Acesso em 2.ago.2005.

[6] DIERKS, T.; ALLEN, C.*RFC 2246 The TLS Protocol Version 1.0*. 1999. Disponível na Internet por http em: www.ietf.org/rtc. Acesso em 21.abr.2005.

[7] DIERKS, Tim; RESCORLA, Eric. *RFC 2246 (Draft) The TLS Protocol Version 1.1*. 2004. Disponível na Internet por http em: www.ietf.org/rfc. Acesso em 21 fev. 2005.

[8] GARFINKEL, Simson; SPAFFORD, Gene. *Comércio e Segurança na WEB*. São Paulo: Market Press, 1999.

[9] HOLLWNBECK, S.. *RFC 3749 Transport Layer Security Protocol Compression Methods*. 2004. Disponível na Internet por http em: www.ietf.org/rfc. Acesso em 21.fev.2005.

[10] IBM. *iKP*. Disponível na Internet por http em: http://www.zurich.ibm.com/security/past-projects/ ecommerce/iKP.html. Acesso em 20.maio.2005.

[11] JONSSON, J.; KALISKI, B. *RFC 3447 Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*. 2003. Disponível na Internet por http em: www.ietf.org/rfc. Acesso em 20.maio.2005.

[12] MINOLI, Daniel: MINOLI, Emma. *Web Commerce Technology Handbook*. New York: McGraw-Hill, 1998.

[13] RIBEIRO, Vinicius G. ***Um estudo sobre os métodos de pesquisa utilizados em segurança computacional - criptografia.*** Porto Alegre: PPGC da UFRGS, 2000. (Trabalho individual 916).

[14] SCHNEIER, Bruce. ***Applied Cryptography: Protocols, Algorithms and Source Code in C***. New York: John Wiley, 1994.

[15] SMITH, Richard E. *Internet Cryptography*. Massachussets: Addison-Wesley, 1997.

[16] STALLINGS, William. ***Cryptography and Network Security***. Upper Saddle River: Prentice-Hall, 1998.

[17] STANLEY, William D. *Network Analysis with applications*. Upper Saddle River: Prentice Hall, 2002.

[18] TUECKE, S.; WELCH, V. E. *RFC 3820: Internet X.509 Public Key Infrastructure (PKI)*. 2004. Disponível na Internet por http em: www.ietf.org/rfc. Acesso 25.maio.2005.