Desenvolvimento de um pacote de classes para garantir uma comunicação segura

Mateus Presotto Balen, Alexandro M. S. Adário

URI – Campus de Erechim
Av. Sete de setembro 1621, 99700-000 Erechim, RS
mateus_balen@hotmail.com, adario@uricer.edu.br

Introdução

Uma das maiores preocupações em sistemas distribuídos é a segurança dos dados que trafegam na rede [VIE 2004]. A cada dia cresce o número de entidades (indivíduos ou grupos organizados) com objetivos de invasão, interceptação ou modificação de mensagens e ainda fabricação mensagens falsas [CER 2006]. Como a área de segurança computacional é muito ampla e complexa, há inúmeras soluções para a troca de mensagens numa aplicação cliente/servidor. Cada uma delas está sujeita a falhas e a questões particulares de implementação.

De forma geral, a solução de segurança e privacidade das mensagens pode ser baseada na rede ou na aplicação. Caso a segurança seja implementada na rede, a aplicação somente estará segura caso conheça a rede na qual suas mensagens trafegam; sendo que, em geral, essa solução apresenta melhor desempenho em termos de latência de comunicação e tamanho/número de pacotes. Por outro lado, se a segurança estiver na aplicação, não é preciso preocupar-se com o local em que é executada, mas isso afeta o desempenho, pois requer uma solução de alto nível para criptografia e autenticação.

Baseado nessas considerações, este projeto visa implementar um pacote de classes Java, baseado em padrões de projeto, garantindo segurança nas comunicações a uma aplicação, independente da rede utilizada. A avaliação dessa solução, baseou-se em testes com SSL (*Secure Socket Layer*) e com a Extensão de Criptografia Java (JCE) para a escolha do melhor método para a criação do pacote.

Desenvolvimento

Antes de desenvolver o pacote de classes, foi testada a segurança provida nas comunicações por sockets através de SSL e de JCE. Ambas são eficientes e garantem a segurança das comunicações via sockets, e possuem um conjunto bem distinto de vantagens e desvantagens.

O SSL é um protocolo não proprietário, incluído na maioria dos navegadores web. Ele cifra e decifra mensagens usando chaves assimétricas (pública mais privada) ou simétricas (chave única), e possui certificados digitais para autenticar transações. Esse protocolo está integrado à linguagem Java através da API JSSE (*Java Secure Socket Extension*) [SUN 2004]. Esta API é eficiente na criptografia de mensagens, mas não foi usada na implementação do pacote de classes, por basear a segurança de comunicação nos protocolos da rede.

A JCE, parte da API JCA (*Java Cryptography Architecture*) usa o conceito de provedores de segurança. Essa extensão fornece suporte à criptografia simétrica, como DES, e assimétrica, como RSA. Juntamente com a SDK Java é fornecido um provedor, chamado SunJCE, o qual suporta criptografia simétrica. É possível obter provedores de criptografia assimétrica gratuitos, como o Bouncy Castle (BC).

Os testes com SSL e JCE, possibilitaram verificar um grande conjunto de dificuldades apresentadas por cada um deles. Após uma bateria de testes e análise das classes presentes nos dois pacotes (JSSE e JCE), definimos através de modelagem UML um novo pacote de classes, seguido o padrão de projeto Façade [MET 2004], para simplificar e esconder detalhes de implementação usando uma abordagem com padrões de projeto.

O objetivo principal é facilitar ao engenheiro ou programador de sistemas a inclusão de criptografia nas suas comunicações via rede. Entretanto, o pacote não fica restrito somente à comunicação em rede, podendo ser utilizado para gerar arquivos criptografados, aplicações de email e ainda bancos de dados. Os testes preliminares demonstraram que, ao contrário do que se poderia pensar, a criptografia na aplicação não afeta consideravelmente o desempenho, uma vez que o protocolo SSL também adiciona camadas extras ao TCP/IP, tornando-o mais lento.

Conclusão

Este trabalho definiu e implementou um pacote de classes Java utilizando a API JCE, que implementa no nível da aplicação os algoritmos de criptografia [SUN 2000]. Essa abordagem oferece um diferencial em relação ao tradicional uso do protocolo SSL (em Java, disponível através da API JSSE), que garante segurança nos protocolos de rede. Com isso o desenvolvedor pode assegurar que a comunicação entre cliente e servidor de uma aplicação Java é segura independente da rede na qual esteja trafegando.

Referências

[CER 2006] CERT. Estatísticas dos Incidentes Reportados ao CERT.br. Disponível em: http://www.cert.br/stats/incidentes/. Último acesso em Out/2006.

[LEG 2006] The Legion of the Bouncy Castle. Disponível em: http://www.bouncycastle.org. Último acesso em Out/2006

[MET 2004] METSKER, Steven John. Padrões de Projeto em Java . São Paulo: Bookman, 2004.

[SUN 2000] SUN, Microsystems. Java Cryptography Extension 1.2.2 - API Specification & Reference. 2000. Disponivel em: http://java.sun.com/products/jce/doc/guide/API_users_guide.html>. Acessado em: Out/2006

[SUN 2004] SUN, Microsystems. Java Secure Socket Extension (JSSE) - Reference Guide. 2004. Disponivel em: http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html>. Acessado em: Out/2006.

[VIE 2004] VIEGA, John; MESSIER, Matt. Security is Harder than You Think. Queue, v. 2, n. 5, p 60-65. 2004.