Um *framework* para prover comunicação segura em aplicativos de TV Digital

Alexandro Bordignon, Valter Roesler

Centro de Ciências Exatas e Tecnológicas – Universidade do Vale do Rio dos Sinos (UNISINOS). São Leopoldo – RS – Brasil

alebordig@yahoo.com.br, roesler@unisinos.br

Resumo. *A TV Digital chega como um novo canal de comunicação e meio de disponibilizar novos serviços por meio da televisão, como o tv-mail e tv-commerce. Entretanto, ao mesmo tempo em que essa nova tecnologia facilita a vida das pessoas, também requer que um conjunto de processos e protocolos seja desenvolvido em prol da garantia de segurança neste ambiente. Com base nessas necessidades, este artigo analisa as ferramentas e protocolos de segurança disponíveis para comunicação segura de dados por meio da TV Digital, apresentando também um framework implementado com o objetivo de facilitar o desenvolvimento de aplicativos que necessitem tal funcionalidade.*

1. Introdução

Serviços *on-line* via Internet como o acesso a entretenimento com propriedade autoral, troca de mensagens instantâneas privadas e transações financeiras são algumas das notáveis revoluções que a tecnologia da informação proporcionou na última década do século 20 [SCHWALB, 2004]. Com o advento da adesão da Televisão Digital Interativa (TVDI) por vários países, neste momento também em fase de implantação pelo Brasil, um novo canal se torna disponível para o fornecimento destes serviços.

Em se tratando de TV Digital, a interatividade poderá ser efetuada através de várias tecnologias, como Cable Modem, ADSL, celular (GPRS, EDGE), wimax, wi-fi e assim por diante. Visto que o sinal de retorno passa pela Internet até chegar no servidor de aplicação (meio inseguro), se faz necessário também que um conjunto de processos e protocolos seja desenvolvido em prol da garantia de segurança neste ambiente. O protocolo TLS (*Transport Layer Security*) [DIERKS e ALLEN, 1999] é disponibilizado na TV Digital Interativa para prover comunicação segura[1] nesse canal [ETSI, 2003] e foi esse o protocolo utilizado também na modelagem do *framework*.

Para que o desenvolvimento de aplicações seja efetuado de forma independente de plataforma, utiliza-se diversas APIs (*Application Program Interfaces*) que abstraem a complexidade do hardware e do sistema operacional. Tais APIs são denominadas, genericamente, de "middleware" [MORRIS e SMITH-CHAIGNEAU, 2005]. Na área de televisão digital, soluções proprietárias de middleware foram disponibilizadas durante vários anos por organizações como OpenTV, NDS, Canal+, PowerTV e Microsoft. Atualmente, existe também um mercado para middleware de padrão aberto,

[1] Entende-se por comunicação segura medidas que assegurem confidencialidade, autenticidade e integridade da informação em sistemas e canais de telecomunicação [CESG Memorandum No.1. v.1.2, 1992 apud IEEE, 1993].

onde a MHP (*Multimedia Home Platform*) [DVB-MHP, 2006] do projeto DVB[2] [DVB, 2006], ambiente baseado em XML ou Java, é o padrão atualmente mais aceito neste mercado [MORRIS e SMITH-CHAIGNEAU, 2005], [JUCÁ, 2006]. Para a implementação do *framework* de segurança apresentado neste artigo, optou-se por padrões abertos, no caso a MHP, enfocando a utilização da interatividade através do Java, pois foi neste modelo que o *framework* foi desenvolvido. Os recursos de segurança da MHP são descritos na seção 2.

Visando facilitar a implementação de aplicativos que necessitem segurança no canal de interatividade, foi desenvolvido um *framework* de segurança, cuja descrição é apresentada na seção 3. A seção 4 apresenta um exemplo de aplicação utilizando o *framework*, e a seção 5 traz as considerações finais sobre o trabalho desenvolvido.

2. Segurança na plataforma MHP

Para que seja possível garantir a segurança na comunicação de dados é necessário também que exista a garantia de que a aplicação recebida pelo receptor seja confiável. Conforme também cita Rescorla (2001), "a proteção contra ataques onde uma das pontas do sistema está sobre controle do atacante é extraordinariamente difícil, senão impossível".

Em função disso, um *framework* de segurança foi inserido na MHP para garantir a integridade e origem confiável da aplicação [ETSI, 2003]. Esse se utiliza da arquitetura de chave pública, requerendo que todos os receptores MHP incluam um certificado raiz pré-instalado e todos os emissores de aplicações possuam seu próprio certificado de identidade emitido por uma Autoridade Certificadora reconhecida [DVB, 2005].

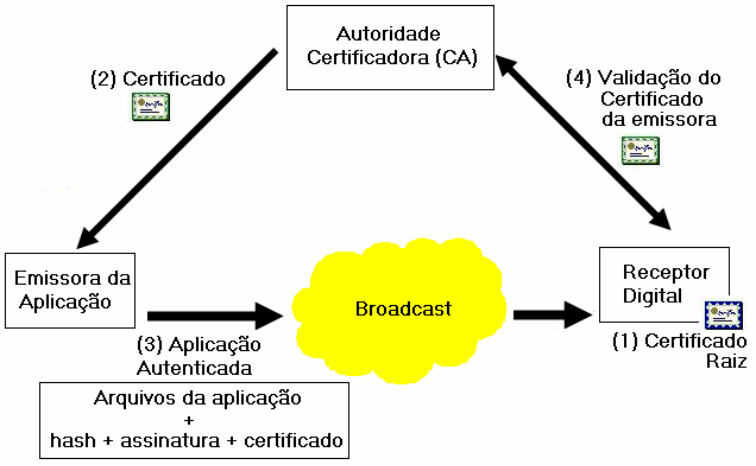

Figura 1. Arquitetura utilizada para autenticar as aplicações.

A Figura 1 ilustra o processo de autenticação das aplicações (que é o envio do certificado da emissora e assinatura digital da aplicação). Os receptores possuem um certificado raiz pré-instalado de uma Autoridade Certificadora reconhecida (1). Todas as emissoras também recebem seu certificado dessa Autoridade Certificadora (2).

[2] O DVB (Digital Vídeo Broadcast) é um consórcio de empresas com a finalidade de desenvolver especificações para a televisão digital. Mesmo com intenção inicial de abranger a Europa, atualmente as suas especificações são utilizadas mundialmente e possuem membros situados fora da Europa [MORRIS e SMITH-CHAIGNEAU, 2005].

Quando a emissora envia uma aplicação (3) ela deve enviar juntamente um arquivo hash correspondente aos arquivos enviados. Caso queira também autenticar essa aplicação (para obter permissão de acesso ao canal de interatividade, por exemplo, que só é disponibilizada a aplicações autenticadas), deve incluir no diretório raiz da aplicação também o certificado da emissora e o arquivo hash da aplicação assinado digitalmente [ETSI, 2003].

Já para o canal de interatividade, o padrão MHP escolheu o TLS (*Transport Layer Security*) [DIERKS e ALLEN, 1999] como protocolo para prover segurança [ETSI, 2003]. Esse protocolo pode ser disponibilizado para as aplicações de TVDI por meio da utilização da API JSSE (*Java Security Socket Extension*), segundo especificação SUN para o DVB [SUN, 2006].

Nos terminais MHP em que o canal de interatividade está disponível, esses deverão disponibilizar os seguintes algoritmos criptográficos [ETSI, 2003] [ETSI, 2005]:

- Algoritmo assimétrico RSA;
- Algoritmos hash MD5 e SHA-1;
- Algoritmos simétricos DES, 3DES e AES (obrigatório apenas no MHP v1.1.2).

3. O *framework* de segurança

O *framework* de segurança projetado envolve três tipos de arquivos, conforme ilustra a Figura 2: classes de fábrica de sockets seguros, arquivos de configuração e certificados adicionais para autenticação do servidor de aplicação, conforme descrição nos próximas subseções. Esses arquivos devem ser enviados juntamente com as aplicações que os utilizem, conforme previsto pela MHP [ETSI, 2003]. Além disso, são enviados os itens de autenticação da aplicação (hash + assinatura da emissora + certificado da emissora), conforme descrito anteriormente na seção 2.

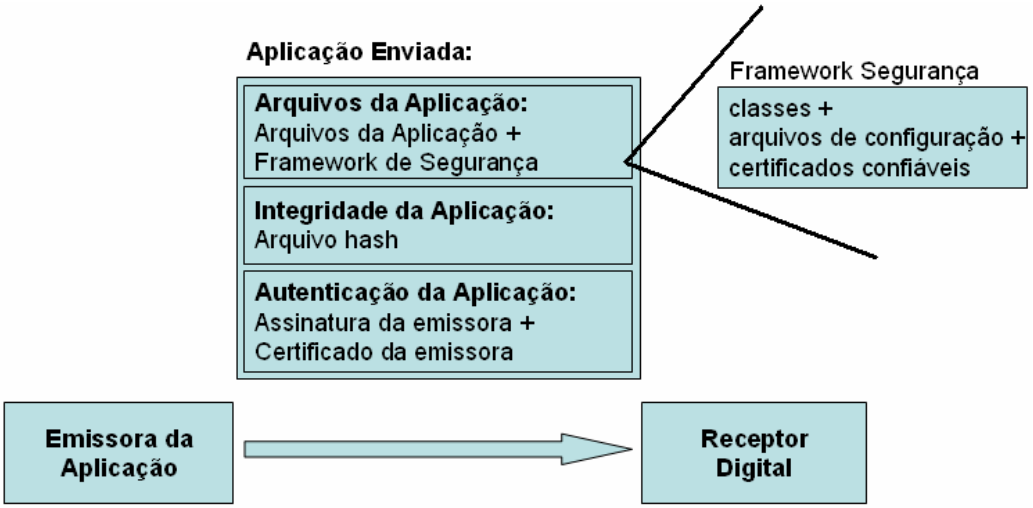

Figura 2. Arquivos do *framework* de segurança.

3.1. Classes e arquivos de configuração

A API desenvolvida consiste de duas classes principais (Figura 3): a classe que disponibilizará os sockets seguros no set-top box (*TvdiTlsSocketFactory*) e a classe que

disponibilizará os sockets seguros no servidor da aplicação (*TvdiTlsServerSocketFactory*).

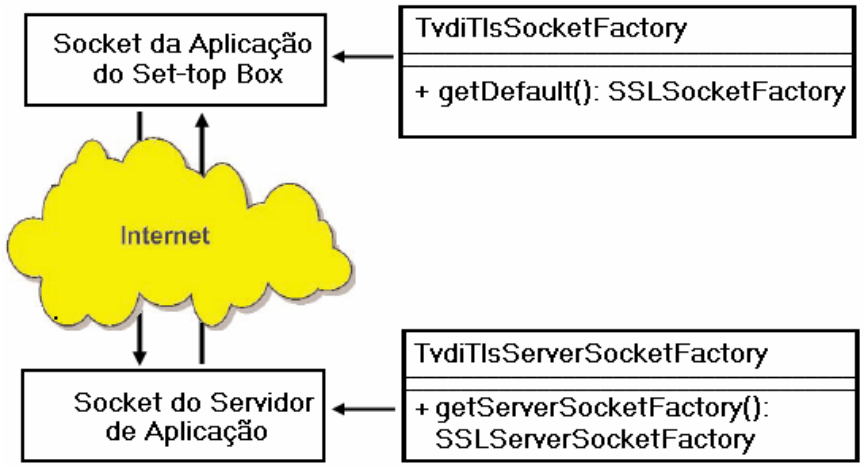

Figura 3: As aplicações do set-top box e do servidor utilizam as classes do *framework* de segurança para obter as fábricas e criar os sockets TLS.

A classe *TvdiTlsSocketFactory* retorna uma fábrica de sockets TLS para a aplicação que está sendo executada no set-top box. Uma instância da fábrica de sockets deve ser obtida pelo método estático *getDefault()*. Essa classe necessita do arquivo de configuração "*seguranca.cliente.properties*" no diretório raiz da aplicação. Esse deverá conter o nome da base de certificados confiáveis e a senha para acessar este arquivo, como demonstra o código abaixo, onde "*servidor.truststore*" é um exemplo de nome para a base de certificados confiáveis (*truststore)*:

truststore=servidor.truststore

senha=<senha da truststore>

Neste caso, não é problema a senha ser transmitida em formato aberto, pois essa base não contém qualquer informação sigilosa, mas apenas a base de certificados confiáveis. No entanto o Keytool (ferramenta utilizada para gerar os certificados adicionais), descrito a seguir, obriga que seja informada uma senha para a criação e leitura da truststore, tornando-se necessária também no arquivo de configuração.

A classe *TvdiTlsSeverSocketFactory* retorna uma fábrica de sockets TLS para o servidor da aplicação. Uma instância da fábrica de server sockets deve ser obtida pelo método estático *getServerSocketFactory()*. Essa necessita que o arquivo de configuração "*seguranca.servidor.properties*" seja criado no diretório raiz da aplicação. Esse conterá o nome da base contendo as chaves público/privadas do servidor e a senha para acesso a esta base. O código abaixo ilustra um exemplo, onde "*servidor.keystore*" é um exemplo de nome para a base de sockets TLS:

keystore=servidor.keystore

senha=<senha da keystore>

Destaca-se aqui a importância deste arquivo estar em um servidor seguro, pois a senha da keystore é armazenada em texto legível. Uma alternativa disponível para as aplicações que utilizam esta API é, ao invés de informar no arquivo de propriedades, solicitar a senha para o usuário e atribui-la na propriedade de sistema

"*br.unisinos.tvdi.tls.senhakeystore*". Abaixo um exemplo, onde <*senha keystore*> deve ser substituída pela senha solicitada ao usuário:

System.setProperty("br.unisinos.tvdi.tls.senhakeystore", "<senha keystore>")

3.2. Certificados Confiáveis (truststore)

Embora a especificação MHP disponibilize certificados raiz pré-instalados e um processo para autenticar as aplicações (conforme descrito anteriormente), essa não descreve a possibilidade das aplicações utilizarem esses mesmos certificados para prover segurança também para as conexões TLS realizadas.

Por este motivo, optou-se por criar certificados adicionais, gerados no servidor, que serão utilizados pelas aplicações que necessitem de conexão segura pelo canal de interatividade. A partir desses certificados do servidor, será gerada uma base de certificados confiáveis (truststore). A truststore deverá então ser enviada juntamente com a aplicação.

Como a entidade que gera os certificados é também a responsável por verificá-los depois no receptor, a confiabilidade destes certificados pode ser gerenciada pelo próprio *framework*, não necessitando de uma Autoridade Certificadora.

Para geração dos certificados, optou-se por utilizar a ferramenta Keytool [SUN, 2002] da SUN. O Keytool é um utilitário que permite aos usuários gerar e administrar seus pares de chaves público/privada e associar certificados a estas para permitir a autenticação e garantir a integridade das informações, utilizando-se de assinaturas digitais [SUN, 2002].

4. Exemplo de aplicação utilizando o *framework* seguro

Visando testar o *framework* desenvolvido, criou-se uma aplicação simples, que utiliza o *framework* para enviar mensagens no modo seguro e no modo não seguro. A Figura 4 ilustra a interface com o usuário, que é responsável pela solicitação dos parâmetros para o usuário e os enviar para o servidor de aplicação. O servidor é o responsável por implementar o processamento da mensagem e confirmar o envio da mensagem para o usuário, conforme o propósito do serviço em questão:

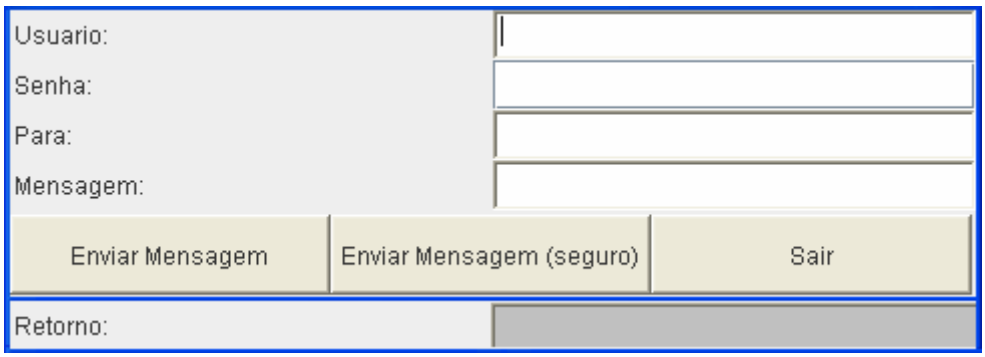

Figura 4. Tela da aplicação que executará no terminal MHP – Solicita usuário, senha, destino e a mensagem a ser enviada. Permite enviar a mensagem de forma normal (aberta) ou segura.

Mesmo sendo esse um protótipo bastante simples e criado apenas com a finalidade de validar a troca de informações de forma segura utilizando o *framework*

desenvolvido, muitas aplicações disponíveis atualmente utilizam este modelo de aplicação. Pode ser citado como exemplo o envio de e-mails e de mensagens de texto para celular.

5. Considerações finais

Este artigo apresentou um *framework* desenvolvido visando facilitar a geração de aplicativos seguros para transmissão no canal de interatividade da TV Digital. A contribuição deste artigo compreende a definição dos mecanismos necessários para se obter segurança, bem como na implementação do *framework* definido.

6. Referências

DIERKS, T.; ALLEN, C. *The TLS Protocol Version 1.0*. IETF RFC 2246. The Internet Society, 1999. Disponível em: http://www.ietf.org/rfc/rfc2246.txt. Acesso em: 31 de maio de 2006.

DVB. *Digital Vídeo Broadcasting Project*. 2003-2006. Site oficial do projeto DVB. Disponível em: http://www.dvb.org. Acesso em: 15 de junho de 2006.

DVB. Digital Video Broadcasting. *Atualização MHP/OCAP/GEM*. 2005. 2p. Disponível em: www.dvb.org/documents/white-papers/MHP-update.April%2005.final.Portuguese.pdf. Acesso em: 15 de junho de 2006.

DVB-MHP. *Digital Vídeo Broadcasting Multimedia Home Plataform*. 2003-2006. Site oficial da plataforma MHP. Disponível em www.mhp.org. Acesso em 15 de junho de 2006.

ETSI - European Telecommunications Standards Institute. Digital Vídeo Broadcasting (DVB). *Multimedia Home Platform (MHP) Specification 1.1.1*. v. 1.1.2. Rev.1. ETSI, França: 2003.

ETSI - European Telecommunications Standards Institute. Digital Vídeo Broadcasting (DVB). *Multimedia Home Platform (MHP) Specification 1.1.2. Draft*. ETSI, França: 2005.

IEEE POSIX P1003.6 SECURITY WORKING GROUP. Consolidated Security Glossary. 1993. Disponível em: http://csrc.nist.gov/posix/*framework*_wg/glossary.asc. Acesso em: 9 de julho de 2006.

JUCÁ, P.; COELHO, A.; DUARTE, R.; FERRAZ, C. *Comparação entre o Desenvolvimento de Aplicações MHP e Open TV*. 24º Simpósio Brasileiro de Redes de Computadores, Workshop de TV Digital. Curitiba, Pr. Junho de 2006.

MORRIS. S.; SMITH-CHAIGNEAU, A. *Interactive TV Standards.* ELSEVIER: EUA, 2005.

RESCORLA. E. *SSL and TLS: Designing and Building Secure Systems.* Upper Saddle River, NJ: Addison-Wesley, 2001. 499p.

SCHWALB, E. M. *iTV Handbook Technologies and Standards*. New Jersey: Prentice Hall, 2004. 724p.

SUN Microsystems. JDK Tools and Utilities. *Keytool - Key and Certificate Management Tool*. 2002. Disponível em: http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html. Acesso em 15 de junho de 2006.

SUN Microsystems. Sun Developer Network (SDN). *The Sun Specifications for DVB*. 1994-2006. Especificações e APIs disponíveis para a plataforma MHP. Disponível em: http://java.sun.com/products/specformhp/. Acesso em: 07 de junho de 2006.