Segurança para Televisão Digital: uso de *WS-Security* para o consumo de *Web services* seguros

Lilissanne Marcelly de Sousa[1], Paulo Roberto de Lira Gondim[1], Manoel Campos da Silva Filho[2]
lilissanne@gmail.com, pgondim@unb.br, mcampos@ifto.edu.br

[1]Universidade de Brasília, Brasil
Campus Universitário Darcy Ribeiro, Faculdade de Tecnologia, Dep. de Engenharia Elétrica – CEP: 70910-900
Brasília/DF – Brasil
[2] Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, Brasil
AE 310 SUL, Avenida LO 05, s/n. Plano Diretor Sul. Coordenação de Informática – CEP:77.021-090
Palmas/TO – Brasil

Resumo: *Este artigo propõe o emprego de segurança fim-a-fim na comunicação e no consumo de serviços disponibilizados por Web services seguros existentes, através de aplicativos para TV digital interativa. É considerado o emprego de protocolos de comunicação padronizados como HTTP e SOAP, considerando versões específicas, compatíveis com o middleware adotado no Sistema Brasileiro de Televisão Digital. Para garantir a segurança na comunicação e no consumo de serviços seguros, é proposto o uso de padrões que garantam a segurança fim-a-fim, como WS-Security, XML Encryption e XML Signature.*

Abstract: *This paper proposes the use of end-to-end security in the communication and consumption of services provided by secure Web services, through applications for interactive digital TV. It is considered the use of standardized communication protocols like HTTP and SOAP, considering specific versions, compatible with the middleware adopted in the Brazilian System of Digital Television. To ensure secure communication and consumption of insurance services, we propose the use of standards to ensure end-to-end security such as WS-Security, XML Encryption and XML Signature.*

Palavras-chave: *TV Digital, Web service, segurança, SOAP, WS-Security*.

1. Introdução

A interatividade advinda com a televisão digital (TVD) permite a participação do telespectador, ensejando um novo paradigma e levando a uma oferta de serviços mais diversificada, que inclui entretenimento, comércio eletrônico (*T-commerce*), *T-banking*, e *T-learning*, dentre outros.

Entretanto, os serviços interativos oferecidos aos usuários criam demandas de segurança relacionadas a aspectos como autenticação e confidencialidade. As questões de segurança em ambiente digital estão em ampla discussão. A norma ABNT NBR15605 parte 2 [ABNT, 2008], que trata dos mecanismos de segurança para aplicativos para TVD, ainda está em fase de construção, o que pode ajudar a compreender que há muito que se discutir e realizar em relação ao tema.

A norma ABNT NBR 15605-1 [ABNT, 2008c] trata do controle de cópias no âmbito da TVD, e tem como escopo a especificação de mecanismos para o combate à pirataria dos conteúdos de alta-definição. Em seu trabalho, [Costa, 2009] fornece algunas contribuições à proteção de direitos e à autenticação de aplicativos para TVD. Recentemente, o CPqD (Centro de Pesquisa e Desenvolvimento em Telecomunicações) publicou algunas recomendações de segurança para a TV digital interativa brasileira [CPqD, 2012]. Portanto, a segurança em TVD não é um tema exaustivo e está em constante discussão.

Este artigo propõe o uso do *WS-Security* para garantir a segurança no nível de mensagem (segurança fim-a-fim) para o consumo de *Web services* seguros, através de aplicações para TV digital interativa, utilizando o canal de retorno. Para isso, é considerado o uso de especificações de segurança padronizadas, tais como *XML Encryption* e *XML Signature*, e de protocolos de comunicação padronizados, como os protocolos HTTP e SOAP, que serão descritos no artigo, com foco nas versões produzidas para atender, de forma específica, as peculiaridades do middleware Ginga em sua versão declarativa (Ginga-NCL). Dessa forma, a proposta vai ao encontro da crescente tendência de integração e convergência entre Web e TV, com vistas a garantir a segurança nesta integração.

O trabalho está assim organizado: na seção 2 são apresentados os trabalhos relacionados. Na seção 3 é apresentada uma breve descrição referente a *Web services*. Na seção 4 é apresentada uma comparação entre a segurança ponto-a-ponto e a segurança fim-a-fim. Na seção 5 é apresentado o padrão *WS-Security* como alternativa para a segurança fim-a-fim. Na Seção 6 é apresentada a proposta do uso do *WS-Security* para a TV Digital. Na seção 7 são apresentadas as conclusões.

2. Trabalhos relacionados

O WSS4J [Apache, 2008] fornece uma implementação *open source* do *WS-Security*, com o uso de bibliotecas de código aberto, como o *Apache XML Security*. Ele fornece recursos para a construção de *Web services* seguros e interoperáveis, através do uso do padrão *WS-Security*.

O projeto *GlassFish* para a plataforma Java.Net [GlassFish, 2006] também fornece uma implementação da especificação do *WS-Security*. Ele incorporou uma implementação inicial da Sun, o JWSDP, que já fornecia o *WS-Security* para a segurança de *Web services*.

A norma ABNT NBR 15605-2 (em preparação) [ABNT, 2008] define os mecanismos de autenticação dos receptores, dos dispositivos externos e de usuários, além das questões de segurança do canal de interatividade, bem como autenticação de aplicativos. Segundo [Costa, 2009], os estudos e análises do SBTVD (Sistema Brasileiro de Televisão Digital) fase I apresentam a segurança de mensagens como a mais adequada para o desenvolvimento do soquete de comunicação segura.

Em [Macedo et al, 2010] os autores fazem uma análise dos mecanismos de controle de acesso que utilizam a tecnologia WS, para determinar como as mensagens de requisições e autorizações de acesso devem ser formadas, transportadas e processadas, de modo a inibir ataques contra a confidencialidade da informação gerenciada. Utiliza as especificações *WS-Security*, *WS BPEL* e *WS Policy*. O trabalho possui foco somente em *Web services*.

3. Web services

Um *Web service* (WS), conforme [W3C, 2004], é uma aplicação identificada por uma URI (*Uniform Resource Identifier*) e que possui interfaces bem definidas e descritas em XML. Também pode ser entendido como uma unidade lógica de aplicação na qual sua funcionalidade pode ser reutilizada sem a preocupação de como a mesma é implementada, e acessada através de protocolos padrões da Internet [Erradi e Maheshwari, 2005].

Os *Web services* podem ser considerados como base para uma implementação de arquitetura orientada a serviços (SOA), e adotam os seguintes padrões baseados em XML para viabilizar as operações de publicação, localização e invocação de um serviço [Curbera, 2002]:

- WSDL (*Web services Description Language*) - padrão para a definição de interface de serviço;
- UDDI (*Universal Description, Discovery and Integration*) - padrão de descoberta de serviços;
- SOAP (*Simple Object Access Protocol*) - padrão de trocas de mensagens que oferece suporte à comunicação entre os serviços.

Para o consumo de serviços oferecidos por *Web services* existentes, através de aplicações de televisão digital interativa, faz-se necessário o uso de protocolos de comunicação padrões (como TCP, HTTP e SOAP) para garantir a interoperabilidade com outros sistemas.

Entretanto, para o consumo de determinados serviços, como *T-commerce* e *T-Banking*, questões de segurança precisam ser consideradas. Este trabalho apresenta a segurança de mensagens como a solução mais adequada para esse fim.

4. Segurança ponto-a-ponto versus segurança fim-a-fim

O modo ponto-a-ponto promove a segurança durante o tráfego da mensagem entre dois nós consecutivos de processamento SOAP [Rahaman et al, 2006], conforme observado na Figura 1. Neste modo de segurança, quando a mensagem é recebida e reencaminhada por um nó intermediário de processamento SOAP, que está acima da camada de transporte, a confidencialidade, a integridade e/ou outro princípio de segurança aplicado à mensagem podem ser perdidos. A criptografia é aplicada a todos os dados dos pacotes de comunicação, incluindo informações de protocolo e a mensagem propriamente dita.

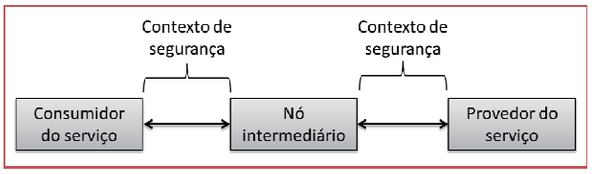

Figura 1. Configuração ponto-a-ponto. Adaptado de [Rahaman et al, 2006].

Já no modo fim-a-fim, conhecido também como segurança no nível de mensagem, o contexto de segurança não se limita ao tempo em que a mensagem está em trânsito entre dois nós consecutivos [Rahaman et al, 2006]. Neste modo de segurança, não há decriptografia intermediária. Além disso, é possível selecionar partes da mensagem que de fato necessitam ser criptografadas. A Figura 2 representa o modo de segurança fim-a-fim.

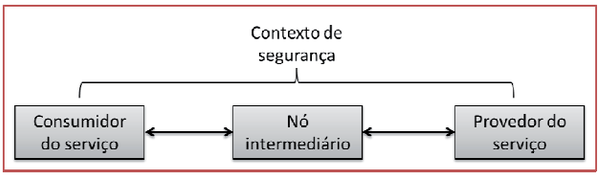

Figura 2. Configuração fim-a-fim. Adaptado de [Rahaman et al, 2006].

Na próxima seção, é apresentado o *WS-Security* como alternativa para a segurança fim-a-fim.

5. WS-Security (WSS) e segurança fim-a-fim

O *WS-Security* é uma especificação definida pela OASIS [OASIS, 2006] que descreve melhorias na mensagem SOAP (*Simple Object Access Protocol*) para fornecer segurança no nível de mensagem (segurança fim-a-fim), através de integridade, confidencialidade e autenticação de mensagem. [Seely, 2002] define *WS-Security* como um padrão criado sobre um conjunto de outros padrões e especificações já existentes. Portanto, o *WS-Security* oferece uma forma de se conjugar soluções existentes.

WSS utiliza *tokens* de segurança com informações de autenticação, por exemplo X.509, que podem ser incluídas na mensagem SOAP [Thompson, 2003]. Também utiliza os padrões *XML Encryption* e *XML Signature* para criptografia e assinatura digital. O *XML Encryption* é o padrão W3C [W3C, 2002] que define como criptografar dados e como representar o resultado de forma estruturada em um documento XML, a fim de garantir a confidencialidade do mesmo. Tem a finalidade de fornecer segurança fim-a-fim para aplicações com necessidade de troca de dados XML de forma segura. Ele permite criptografar somente um elemento específico de um documento XML. O *XML Signature* é padronizado pela W3C [W3C, 2008] e especifica um processo para

geração e validação de assinaturas digitais expressas em XML, para garantir a integridade e autenticação de um documento XML.

A Figura 3 mostra a estrutura de uma mensagem SOAP com *WS-Security*, em que o cabeçalho é utilizado para transportar informações relacionadas à segurança, incluindo o elemento *UsernameToken*, que contém os elementos *Username* e *Password*, criptografados por meio do *XML Encryption*, para garantir a confidencialidade dos mesmos. O elemento *wsse:Security* é o elemento raíz.

```
1. <soapenv:Envelope ....>
2.  <soapenv:Header>
3.   <wsse:Security ....>
4.          <wsse:UsernameToken wsu:Id="1">
5.            <wsse:Username>
6.              <xenc:EncryptionData>9CgAwIBAG...
7.              </xenc:EncryptionData>
8.            </wsse:Username>
9.            <wsse:Password>
10.             <xenc:EncryptionData>tJZc0...
11.             </xenc:EncryptionData>
12.           </wsse:Password>
13.         </wsse:UsernameToken>
14.  </wsse:Security>
15. </soapenv:Header>
16. <soapenv:Body>
17.   .....
18. </soapenv:Body>
19. </soapenv:Envelope>
```

Figura 3. Estrutura do *WS-Security* [OASIS, 2006].

A próxima seção fala sobre o uso do *WS-Security* para prover segurança no nível de mensagem no ambiente da TV digital interativa.

6. WS-Security para TV Digital

A implementação do *WS-Security* para TV Digital pressupõe o uso do protocolo SOAP para esse ambiente, visto que o *WS-Security* é uma extensão de SOAP. O uso de protocolos de comunicação padrões (como TCP, HTTP e SOAP) é necessário para o consumo de serviços oferecidos por *Web services* existentes, através de aplicações de televisão digital interativa, a fim de garantir a interoperabilidade com outros sistemas. Foram implementados os módulos NCLua HTTP e NCLua SOAP, desenvolvidos em linguagem Lua que implementam HTTP e SOAP, respectivamente, para o sub-sistema Ginga-NCL do middleware Ginga do SBTVD (Sistema Brasileiro de Televisão Digital) [Filho, 2011]. Dessa forma, a convergência entre Web e TV se torna possível.

O NCLua HTTP baseia-se em alguns dos principais recursos do protocolo HTTP/1.0. Utiliza protocolo TCP da forma como especificado na norma [ABNT, 2008b]. Este módulo possui funções que permitem a geração de requisições e tratamento de respostas. Os seguintes recursos foram implementados: autenticação básica; download de arquivos; requisições GET e POST; passagem de cabeçalhos HTTP e definição de *User-Agent*; separação automática dos dados do cabeçalho e do corpo da resposta de uma requisição [Filho, 2011]. Já o NCLua SOAP implementa as principais funcionalidades do protocolo SOAP nas versões 1.1 e 1.2. Este módulo gera a requisição SOAP e utiliza o NCLua HTTP para o transporte da mensagem.

Para garantir a segurança da mensagem SOAP, a proposta considera o módulo NCLua WS-Security, que vem sendo implementado em linguagem Lua, que incorpora ao cabeçalho da mensagem SOAP informações relacionadas à segurança da mensagem. Utiliza o *XML Encryption* e *XML Signature* para criptografar os elementos necessários e acrescentar ao cabeçalho SOAP informações referentes a assinatura da mensagem. A Figura 4 exibe uma arquitetura de segurança simplificada para o consumo de *Web services* através de aplicações interativas.

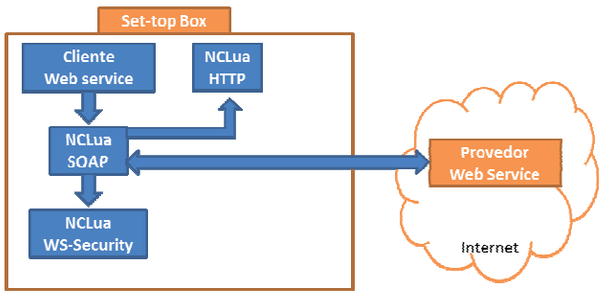

Figura 4. Arquitetura de segurança simplificada.

Alguns desafios são encontrados no desenvolvimento do módulo de segurança NCLua WS-Security. Segundo [Braga e Restani, 2010], não há em Ginga-NCL e nem em Lua uma biblioteca criptográfica completa, totalmente funcional. Existem os pacotes de software LuaCrypto, que implementa uma interface Lua para funções criptográficas disponíveis no receptor, e LuaMD5, uma biblioteca criptográfica simples para scripts Lua. Entretanto, ambos oferecem acesso apenas a algoritmos de *hash* ou algoritmos de criptografía fracos (DES de 56 bits), o que se torna um limitante quanto à aplicabilidade destas bibliotecas. Há também a biblioteca LuaSec, uma fachada para estabelecimento de conexões SSL via OpenSSL, mas não oferece acesso via API às funções criptográficas do OpenSSL [Braga e Restani, 2010].

Está em desenvolvimento a API LuaTV [Brandão et al, 2010], que inclui uma API de extensão de segurança Lua (NCLua Security), que oferece três módulos funcionais *signature*, *digest* e *cipher*, que irá permitir o aperfeiçoamento da implementação do módulo NCLua *WS-Security*. O módulo *signature* da API LuaTV Security oferece métodos para geração e verificação de assinaturas digitais. O módulo *digest* provê a facilidade de geração de *message digests* utilizando quaisquer implementações de algoritmos para *hashing*. O módulo *cipher* disponibiliza funções para a cifragem e decifragem de dados baseado em chaves simétricas e assimétricas.

7. Conclusões

O uso da segurança fim-a-fim foi considerada como a mais adequada para garantir a segurança no consumo de *Web services* através de aplicações para a TV digital interativa. Outrossim, foi considerado o uso de especificações de segurança padronizadas, tais como *WS-Security*, *XML Encryption* e *XML Signature*, e de

protocolos de comunicação padronizados, como os protocolos HTTP e SOAP.

Um cenário de integração e convergência entre Internet e TV digital foi aqui considerado, de forma a permitir que um maior número de serviços já existentes possa ser disponibilizado a telespectadores, e com vistas a garantir a segurança no consumo desses serviços.

Referências bibliográficas

[ABNT, 2008] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15605-2. Televisão digital terrestre – Tópicos de Segurança – Parte 2: Mecanismos de Segurança para Aplicativos. Rio de Janeiro, em construção. 2008.

[ABNT, 2008b] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15606-2. Televisão digital terrestre–Codificação de dados e especificações de transmissão para radiodifusão digital Parte 2: Ginga-NCL para receptores fixos e móveis– Linguagem de aplicação XML para codificação de aplicações, 2008.

[ABNT, 2008c] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15605-1. Televisão digital terrestre – Tópicos de Segurança – Parte 1: Controle de Cópias. Rio de Janeiro. 2008.

[Apache, 2008] Apache WSS4J. The Apache Software Foundation. http://www.apache.org. 2008.

[Braga e Restani, 2010] BRAGA, A. M., RESTANI, G. S. Hacking Ginga: uma avaliação de segurança da plataforma de aplicações interativas da TV digital brasileira. X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. 2010.

[Brandão et al, 2010] BRANDÃO R. R. de M.; BATISTA, C. E. C. F.; SOARES, L. F. G.; SOUZA FILHO, G. L. Extended features for the Ginga-NCL environment - Introducing the LuaTV API. Proceedings of the 19th International Conference on Computer Communication Networks (2nd Workshop on Multimedia Computing and Communications). 2010.

[Costa, 2009] COSTA, L. C. de P. Segurança para o sistema brasileiro de televisão digital: contribuições à proteção de direitos autorais e à autenticação de aplicativos. Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos. São Paulo, 2009.

[CPqD, 2012] Centro de Pesquisa e Desenvolvimento em Telecomunicações. Recomendações em Segurança da Informação para TVDi. Disponível em: http://www.cpqd.com.br. Acessado em 20 ago 2012.

[Curbera, 2002] CURBERA, F., et al. Unraveling the Web Services Web: an introduction to SOAP, WSDL and UDDI. IEEE Internet Computing, v. 6, n. 2, p. 86-93, 2002. ISSN 1089-7801.

.

[Erradi e Maheshwari, 2005] ERRADI, Abdelkarim; MAHESHWARI, Piyush. A Broker-Based Approach for Improving Web Services Reliability. ICWS, pp.355-362, IEEE International Conference on Web Services (ICWS'05), 2005.

[Fernando, 2006] FERNANDO, R. Setting up keystores for a cliente and a service. WSO2. 2006. Disponível em: http://wso2.org/library/174. Último acesso: 20 jan 2012.

[Filho, 2011] FILHO, M. C. da S. Arquitetura orientada a serviços para comércio eletrônico no Sistema Brasileiro de TV Digital. Dissertação de Mestrado em Engenharia Elétrica, Publicação 439/2011, Departamento de Engenharia Elétrica, Universidade de Brasília – UnB, Brasília-DF, 2011.

[GlassFish, 2006] GlassFish. http://glassfish.java.net/

[Macedo et al, 2010] MACEDO, R. T., MOZZAQUATRO, B. A., NETO, L. D. B., NUNES, R. C. Uma Arquitetura de Segurança para Mecanismos de Controle de Acesso Baseados em Serviços Web. X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. 2010.

[OASIS, 2006] OASIS. Web services security. 2006. Disponível em: http://docs.oasis-open.org/wss/v1.1/ Acessado em 12 jan. 2012.

[Rahaman et al, 2006] RAHAMAN, M.A.; SCHAAD, A.; RITS, M. Towards secure SOAP message exchange in a SOA. In: Proceedings of the 3rd ACM workshop on Secure web services. Nova Iorque: ACM, 2006, 77-84.

[Rodrigues, 2011] RODRIGUES, D. Um estudo comparativo das especificações de segurança aplicadas a uma arquitetura orientada a serviços. Dissertação de mestrado. USP – São Carlos. 2011.

[Seely, 2002] SEELY, S. XML and Web Services Security: Understanding WS-Security. Microsoft Corporation, 2002. Disponível em: http://msdn.microsoft.com/en-us/library/ms977327.aspx. Acesso em: 12 abr. 2012.

[Thompson, 2003] THOMPSON, S. Implementing WS Security. 2003. Disponível em: http://www.ibm.com/developerworks/webservices/library/WS-Security.html. Acessado em 20 dez. 2011.

[W3C, 2002] W3C. XML Encryption Syntax and Processing. 2002. Disponível em: http://www.w3.org/TR/xmlenc-core/. Acessado em 18 dez 2011.

[W3C, 2004] W3C. Web Services Architecture. 2004. Disponível em: http://www.w3.org/TR/ws-arch/. Acessado em: 18 dez. 2011.

[W3C, 2008] W3C. XML Signature Syntax and Processing (Second Edition), 2008. Disponível em: http://www.w3.org/TR/xmldsig-core/. Acessado em 12 jan 2012.