Tecnologia de segurança para Web Services

AUTOR1[1], AUTOR2[1]

[1] ORGANIZACAO
EMAIL

Resumo

Este artigo apresenta os resultados do levantamento de normas de segurança para a plataforma de Web Services, que são uma proposta das grandes empresas de informática para uma plataforma que permita aos sistemas de informação dar melhor resposta a desafios de negócio. As normas são as especificações técnicas que definem a plataforma, cobrindo os seus vários requisitos, neste caso de segurança. A partir deste levantamento, é possível enquadrar investigação aprofundada e apoiar decisões de escolha de tecnologia em organizações que desejem tornar mais seguros os seus sistemas de informação estruturados em serviços.

1 Introdução

Os *Web services* são uma tecnologia para construir sistemas de informação empresariais com maior flexibilidade, reutilização e interoperabilidade; proposta por empresas líderes em tecnologias de informação – Microsoft, IBM, Sun Microsystems e Oracle – que disponibilizam diversas normas e ferramentas. Uma aplicação baseada em serviços pretende ser *mais ágil*, ou seja, pretende ter *menores custos de adaptação*, tendo em conta as circunstâncias da sua utilização e a constante evolução dos requisitos de negócio ao longo do tempo [1].

Do ponto de vista técnico, um serviço Web define uma interface funcional que pode ser invocada remotamente para dar acesso a recursos informacionais. A figura 1 apresenta uma visão de funcionamento dos serviços [2].

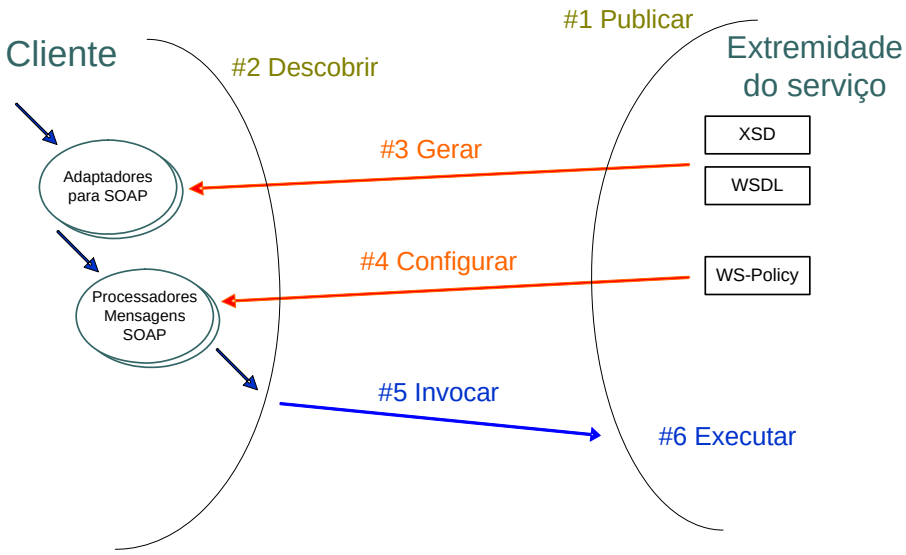

Fig. 1. Fases de interacção de um cliente com o Web Service [2].

As fases de interacção do cliente com o serviço são a *descoberta* (publicar, descobrir), a *vinculação* (gerar, configurar) e a *invocação* (invocar, executar). Em todo este processo, a meta-informação, na forma de *contratos*, desempenha um processo central. O *contrato de dados* descreve os tipos de dados. O *contrato de interface* descreve as mensagens e permite ao cliente gerar código para converter os seus tipos de dados para a representação das mensagens. Adicionalmente, o *contrato de política* vai configurar processadores de mensagens e bibliotecas de suporte a requisitos não funcionais, como a segurança.

A segurança é particularmente relevante nos Web Services, pois estes têm que ser seguros para que possam de facto ser usados em aplicações com valor significativo para as organizações. Este artigo apresenta um levantamento de normas de segurança, sendo especialmente detalhadas a WS-SecurityPolicy, para especificação de contrato de política de segurança, e a SAML, para credenciais de autenticação e autorização.

2 Plataforma de Web Services

A figura 2 [3] resume a plataforma de Web Services, de forma global e independente do fornecedor de tecnologia. Em cada categoria são agrupadas normas que resolvem problemas relacionados. As categorias são: Representação de dados, Transporte, Mensagem, Contrato, Descoberta, Segurança, Mensagens Fiáveis, Transacções, Processos de negócio, Gestão e Interoperabilidade.

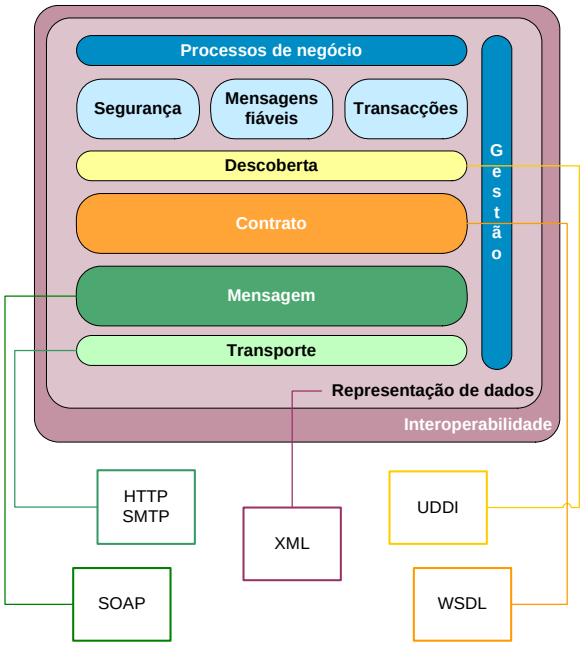

Fig. 2. Categorias de normas de serviços Web. Adaptado de [3].

As normas de *Representação de dados* XML [4] e XML Schema [5] permitem especificar formatos de documentos.

As normas de *Interoperabilidade* centram-se na compatibilidade das implementações, esclarecendo especificações que não são suficientemente claras. A WS-I [6] é a principal organização que define perfis de interoperabilidade.

As normas de *Transporte* permitem estabelecer um canal de comunicação entre o cliente e o prestador do serviço. A comunicação pode ser síncrona ou assíncrona. Os transportes mais comuns são HTTP [7] e SMTP [8].

As normas de *Mensagem* definem a estrutura das unidades de comunicação e as formas como são trocadas entre serviços. O SOAP [9] define mensagens em XML, separando o cabeçalho com dados extensíveis para a plataforma, do corpo com dados para as aplicações.

As normas de *Contrato* permitem a descrição do serviço. A interface do serviço é especificada com WSDL [10]. A política do serviço, ou seja, os requisitos que têm que ser cumpridos pelo cliente e pelo prestador do serviço para que a interacção entre ambos possa acontecer, são definidos com WS-Policy [11].

As normas de *Descoberta* definem formas de publicar e pesquisar serviços. A UDDI [12] define um directório. Os serviços podem também fazer a sua auto-descrição, através de WS-MetadataExchange [13].

As normas de *Segurança* especificam mecanismos de protecção para os serviços. A WS-Security [14] especifica a segurança de mensagens SOAP, enquanto que a

WS-SecurityPolicy [15] especializa a WS-Policy para políticas de segurança. Estas normas e outras relacionadas são detalhadas mais à frente.

As normas de *Mensagens Fiáveis* têm como objectivo dar fiabilidade à comunicação de forma independente do transporte, em aspectos como a entrega garantida, a eliminação de repetições e a correcta ordenação. Existem duas propostas concorrentes: a WS-Reliability [16] da Oracle e Sun; e a WS-ReliableMessaging [17] da IBM e Microsoft.

As normas de *Transacções* permitem ter semânticas bem definidas para os resultados de várias interacções entre serviços com recursos informacionais distribuídos. Para isso assumem modelos de faltas temporárias e recuperáveis para as máquinas e comunicações. Existem dois enquadramentos de normas concorrentes para transacções em serviços Web: a WS-Coordination [18] da Microsoft e IBM e a WS-CompositeApplicationFramework [19] da Oracle e Sun.

As normas de *Processos de negócio* satisfazem a necessidade de ferramentas mais próximas do negócio. A WS-BPEL [20] da Microsoft e IBM, é baseada em orquestração, sendo o processo representado por um grafo, em que os nós são actividades e os arcos são fluxos de controlo e informação, permitindo a composição de serviços. A WS-CDL [21], patrocinada pela Oracle, faz a coreografia de processos de forma declarativa, especificando pré-condições e pós-condições para a execução de actividades, podendo a forma concreta como o processo é executado variar, desde que as condições continuem a ser satisfeitas.

Neste momento o esforço em normas de *Gestão* está centrado na gestão dos dispositivos computacionais e das redes de dados que suportam serviços, existindo duas normas concorrentes: a WS-Management [22] da Microsoft e a WS-DistributedManagement [23] da IBM.

2 Segurança de Web Services

Tradicionalmente a segurança de uma aplicação informática é representada com os conceitos abstractos de: agente, acção e recurso. O *agente*[1] pode ser uma pessoa, organização ou programa informático. A *acção* é a funcionalidade. O *recurso* são os dados e outros meios necessários à acção. Os mecanismos primitivos de segurança, representados na figura 3, pretendem garantir a autenticação dos agentes, a autorização e não-repúdio das acções e a disponibilidade, integridade e confidencialidade dos recursos.

[1] Do inglês, *principal*.

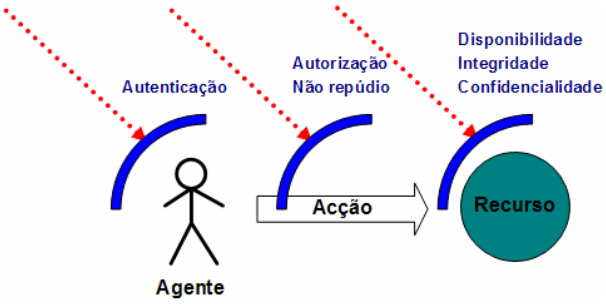

Fig. 3. Protecção de agente, acção e recurso de uma aplicação informática [2].

No caso dos serviços, a representação com agente, acção e recurso é demasiado simplista pois a interacção acontece sobre uma rede não segura e existem, pelo menos, dois agentes. O *agente do cliente* faz a invocação enviando um *pedido* pela rede. O *agente do serviço* recebe a mensagem no servidor e vai executar acções sobre recursos. No fim, é enviada a *resposta*. Os principais problemas a resolver pela segurança de serviços são apresentados na figura 4 e são: a protecção das mensagens, o controlo de acessos e a flexibilidade de configuração.

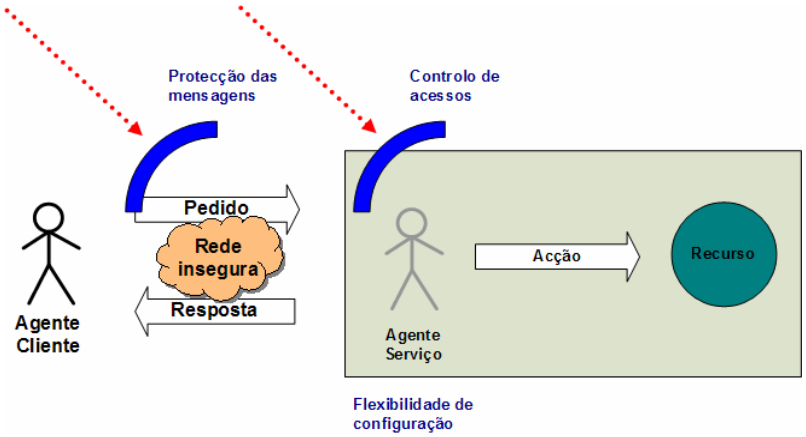

Fig. 4. Problemas a resolver na segurança de serviços [2].

O perfil de interoperabilidade WS-I para segurança [24] clarifica que a segurança de serviços deve preocupar-se com os ataques de falsificação e repetição de mensagens, ficando de fora ataques a chaves e algoritmos criptográficos, a redes e servidores, o uso de canais encobertos e os erros de programação das aplicações.

As secções seguintes apresentam: os mecanismos necessários, o modelo conceptual, as normas e as implementações disponíveis para serviços seguros.

2.1 Mecanismos necessários

Tendo em conta os problemas a resolver, os mecanismos de segurança genericamente necessários para serviços são os seguintes [25]:

- Autenticação:
  - Directa;
  - Com intermediário;
- Protecção de mensagens:
  - Confidencialidade;
  - Origem e integridade dos dados;
- Autorização de acesso a recursos:
  - Baseada em credencial de autenticação:
    - Controlo de acesso e capacidades;
    - Personificação e delegação;
  - Baseada em credencial de autorização;
- Protecção de domínios de segurança:
  - Validação de mensagens;
  - Detecção de mensagens repetidas;
  - Filtragem de excepções.

A *autenticação* pode ser efectuada directamente ou com intermediário. Na autenticação directa o cliente e o serviço participam numa relação de confiança entre si com um modelo de duas entidades, que lhes permite trocar e validar chaves, que são válidas nesse domínio de segurança. Na autenticação com intermediário uma entidade terceira confiada está entre o cliente e o serviço.

No que respeita à *protecção de mensagens*, para garantir a confidencialidade, é necessário cifrar os dados, total ou parcialmente; para garantir a origem e integridade dos dados, é necessário assinar os dados. A assinatura digital com chaves assimétricas permite a diferenciação entre chave pública e chave privada e, deste modo, o não-repúdio de acções. Se apenas se pretender assegurar a integridade dos dados, pode utilizar-se uma assinatura com uma chave simétrica partilhada entre as partes.

A *autorização* de acesso a recursos pode ser baseada em autenticação prévia, sendo depois verificado se existe permissão. A autorização com personificação consiste em assumir temporariamente uma outra identidade. Na autorização com delegação, existe também uma personificação, mas para um propósito bem definido e não se perde a referência da verdadeira identidade do sujeito. Na delegação, a identidade do agente do cliente fica registada nas acções que são efectuadas pelo agente do serviço. O uso de uma credencial de autorização permite aceder a uma dado recurso, sem que exista uma autenticação prévia.

Para a *protecção de domínios de segurança* a nível aplicacional pode ser usado um nó intermediário, colocado na fronteira e que intervém ao nível das mensagens SOAP dos serviços, tal como ilustrado na figura 5.

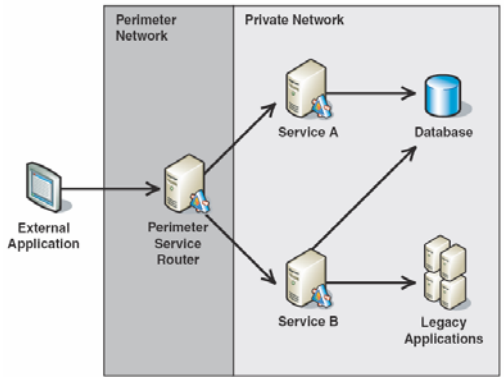

Fig. 5. Protecção da fronteira de um domínio de segurança por um nó intermediário [25].

O intermediário pode efectuar:

- A validação de mensagens – confirmando que estão bem formadas;
- A detecção de mensagens repetidas – com validações temporais e mantendo um registo temporário dos pedidos mais recentes, para evitar ataques por repetição de mensagens;
- A filtragem de excepções – garantindo que, em casos de erro, não é devolvida informação interna sensível do serviço.

2.2 Modelo conceptual

O modelo base de serviços seguros foi proposto inicialmente pela IBM e Microsoft [26] e faz actualmente parte da norma OASIS. O modelo descreve a forma como se processam as invocações que se pretendem seguras.

Cada serviço tem uma política que define as condições de acesso e os *tokens de segurança* exigidos. Por exemplo, um serviço pode exigir que uma mensagem que recebe seja cifrada e que contenha um token com utilizador e senha. Se a mensagem chegar sem provas suficientes, o serviço pode ignorar ou rejeitar a mensagem.

A figura 6 ilustra o modelo de serviços seguros. As setas representam comunicação entre extremidades de serviços.

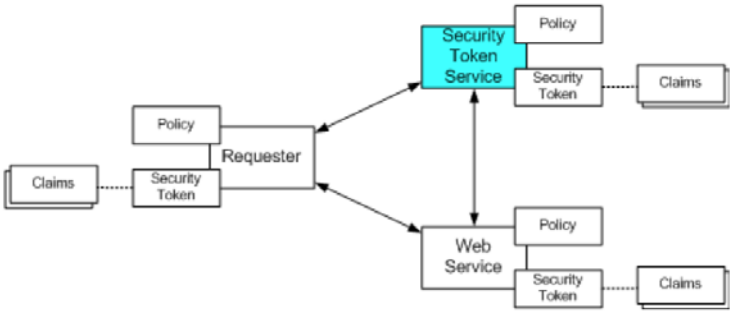

Fig. 6. Modelo de base para Web Services seguros [26].

Quando o emissor da mensagem não cumpre as condições exigidas e não tem capacidade de as gerar por si, pode tentar obter tokens contactando Serviços de Tokens de Segurança (STS). Estes podem, por sua vez, exigir um outro conjunto de condições. Desta forma, os STS podem ser intermediários de segurança entre diferentes domínios, aplicando-se um modelo de confiança de três ou mais entidades.

Este modelo genérico – com políticas, condições e tokens – abstrai vários modelos mais específicos, tais como: segurança baseada em identidade, listas de controlo de acesso, capacidades, etc. O modelo permite também o uso conjunto de tecnologias existentes, como certificados de chave pública X.509 [27], bilhetes de chave partilhada Kerberos [28] e resumos criptográficos de senhas.

2.2 Normas

As normas de segurança para Web Services tiram partido da composição de protocolos. As mensagens SOAP [9] são extensíveis, permitindo incorporar informação de segurança em cabeçalhos. A figura 7 apresenta as normas de segurança enquadradas na plataforma.

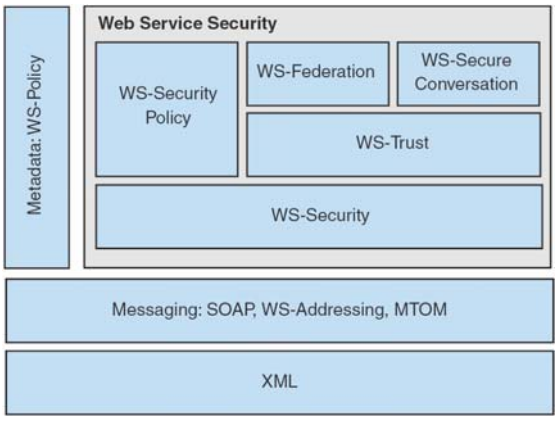

Fig. 7. Normas de segurança para Web Services [25].

A *XML-Signature* [29] e *XML-Encryption* [30] são duas normas que definem como assinar e cifrar documentos XML, respectivamente. Ambas podem ser aplicadas selectivamente a partes da mensagem ou a conteúdos externos referenciados nos documentos. A diferença mais notória entre ambas é que elemento de assinatura (Signature) *referencia* o que está a ser assinado enquanto que o elemento de dados cifrados (EncryptedData) *contém* o que está a ser cifrado [31].

A *WS-Security* [14] permite proteger a mensagem SOAP e transportar tokens de segurança, definindo o modelo base de segurança descrito anteriormente.

A WS-Security permite as seguintes operações sobre a mensagem SOAP:

- Acrescentar data e identificação à mensagem;
- Enviar tokens de segurança no cabeçalho;
- Usar XML-Signature para assinar toda ou parte da mensagem e enviar a assinatura no cabeçalho;

- Usar XML-Encryption para cifrar toda ou parte da mensagem;
- Enviar chaves criptográficas ou referências no cabeçalho.

Existem actualmente duas versões de Web Services Security: a 1.0 (2004) e a 1.1 (2006). A versão 1.1 estende a 1.0 ao nível dos XML Schemas, corrigindo alguns problemas, e acrescentando tokens de segurança para Kerberos, SAML e REL.

A *WS-Trust* [32] define um modelo de confiança com operações para adquirir, emitir, renovar e validar tokens de segurança e formas de criar novas relações de confiança através de serviços intermediários.

A *WS-SecureConversation* [33] permite que dois serviços estabeleçam uma sessão segura entre si para trocarem várias mensagens. Assim, é possível tirar partido de segredos partilhados para derivar chaves simétricas de sessão e ter segurança de forma mais eficiente e robusta.

A *WS-Federation* [34] define federações de serviços para partilhar informação sobre identidade, atributos, autenticação e autorização entre diferentes domínios de confiança.

A *WS-SecurityPolicy* [15] é uma especialização da WS-Policy para definir políticas de segurança que descrevem a forma como as mensagens devem ser tornadas seguras. A política pode aplicar-se a mensagens individuais, a operações ou a toda a extremidade do serviço.

As asserções WS-SecurityPolicy referem-se às funcionalidades de segurança de WS-Security, WS-Trust e WS-SecureConversation. Podem também referir segurança no transporte, como HTTPS.

As asserções descrevem os algoritmos de cifra suportados, as partes das mensagens a assinar ou cifrar, os tokens de segurança que são aceites pelo serviço e outros. Por exemplo, uma política pode especificar que a mensagem seja assinada com uma chave de certificado digital, e outra pode ditar que a autenticação seja feita com a chave de um bilhete Kerberos.

O exemplo 1 apresenta a WS-SecurityPolicy de um serviço. Esta política descreve apenas uma configuração, mas poderia conter diferentes alternativas.

Exemplo 1. WS-SecurityPolicy [15].

```
(01)    <wsp:Policy>
(02)        <sp:SymmetricBinding>
(03)            <wsp:Policy>
(04)                <sp:ProtectionToken>
(05)                    <wsp:Policy>
(06)                        <sp:KerberosV5APREQToken
sp:IncludeToken=".../IncludeToken/Once" />
(07)                    </wsp:Policy>
(08)                </sp:ProtectionToken>
(09)                <sp:SignBeforeEncrypting />
(10)                <sp:EncryptSignature />
(11)            </wsp:Policy>
(12)        </sp:SymmetricBinding>
(13)        <sp:SignedParts>
(14)            <sp:Body/>
(15)            <sp:Header
Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
(16)        </sp:SignedParts>
(17)        <sp:EncryptedParts>
(18)            <sp:Body/>
(19)        </sp:EncryptedParts>
(20)    </wsp:Policy>
```

A linha (01) indica que se trata de uma política e que todas as asserções nela contidas têm que ser satisfeitas. A linha (02) indica um vínculo de segurança de criptografia simétrica. A linha (04) indica o token de segurança a usar para protecção e na linha (06) é especificado o token Kerberos V5 APREQ, que deve ser usado por ambas as entidades para a protecção da troca de mensagens. A linha (09) indica que as assinaturas devem ser geradas a partir dos dados em claro em vez do texto cifrado. A linha (10) indica que a assinatura deve ser cifrada. As linhas (13) a (16) indicam que partes da mensagem devem ser abrangidas pela assinatura principal, neste caso o soap:Body indicado pela linha (14) e todos os cabeçalhos SOAP dentro do espaço de nomes do WS-Addressing, indicado pela linha (15). As linhas (17) a (19) indicam que partes da mensagem devem ser cifradas; neste caso apenas o soap:Body, indicado pela linha (18).

Tal como o exemplo ilustra, existem duas secções principais na definição da política: o vínculo de segurança e a protecção.

A secção da política que define o *vínculo de segurança* (security binding) pode ser: criptografia simétrica (SymmetricBinding), criptografia assimétrica (AsymmetricBinding) e segurança no transporte (TransportBinding).

As sub-asserções disponíveis para detalhar a configuração são: AlgorithmSuite (algoritmos criptográficos a utilizar), Layout (ordenação dos elementos de segurança), IncludeTimestamp (incluir marca temporal na mensagem), EncryptBeforeSign (cifrar antes de assinar), EncryptSignature (cifrar assinatura), ProtectTokens (incluir os tokens de segurança na assinatura) e OnlySignEntireHeadersAndBody (assinar apenas corpo e cabeçalhos que não os de segurança).

As asserções de *tokens de segurança* identificam quais são os tokens aceites e qual o processamento que lhes deve ser dado. Os tokens podem ser: HttpsToken, UsernameToken, X509Token, KerberosToken, SamlToken e RelToken. Por exemplo, o token HttpsToken indica o uso de HTTPS, podendo também especificar se o certificado cliente é obrigatório.

A propriedade de processamento dos tokens mais importante é a TokenInclusion, que indica como o token de segurança deve ser usado: Never (nunca), Once (uma vez), AlwaysToRecipient (sempre para o receptor), Always (sempre).

A *SAML (Security Assertion Markup Language)* [35] permite representar factos de segurança em formato XML. A especificação define um protocolo de comunicação e um formato de asserções, que são independentes um do outro. O *protocolo de comunicação* permite obter e validar asserções. As *asserções* permitem expressar autenticação, atributos e autorização de agentes com identidade num domínio de segurança. Os exemplos seguintes ilustram a estrutura das asserções SAML para uma autenticação, atributos e autorização, respectivamente.

Exemplo 2. Asserção SAML de autenticação.

```
(01)    <Assertion>
(02)        <Conditions NotBefore="2006-07-22T12:02:00Z" NotOnOrAfter="2006-
07-22T13:02:00Z">
(03)            <AudienceRestrictionCondition>
(04)                <Audience>http:/www.example.com/Members</Audience>
(05)            </AudienceRestrictionCondition>
(06)        </Conditions>
(07)        <Advice>
(08)            <AssertionIDReference>id</AssertionIDReference>
(09)            <Assertion>...</Assertion>
(10)        </Advice>
(11)        <AuthenticationStatement AuthenticationMethod="urn:ietf:rfc:2246"
(12)                                 AuthenticationInstant="2006-07-
22T12:02:00Z">
(13)            <Subject>
(14)                <NameIdentifier
Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
(15)                user@example.com
(16)                </NameIdentifier>
(17)            </Subject>
(18)        </AuthenticationStatement>
(19)        <ds:Signature>...</ds:Signature>
(20)    </Assertion>
```

Esta asserção comunica que o utilizador com o endereço de correio electrónico user@example.com foi autenticado com sucesso. A linha (01) indica que se trata de uma asserção. As linhas (02) a (06) indicam as condições, que definem limites temporais e os destinatários da asserção. As linhas (07) a (10) contém informação adicional, podendo referir outras asserções, etc. As linhas (11) a (18) indicam que se trata de uma autenticação, que neste caso foi realizada com um certificado cliente SSL. Finalmente, a linha (19) contém a assinatura da entidade emissora da asserção de autenticação, para garantir a autenticidade e integridade.

Exemplo 3. Asserção SAML de atributos de utilizador.

```
(01)    <Assertion>
(02)        <AttributeStatement>
(03)            <Subject>
(04)                <NameIdentifier
Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
(05)                user@example.com
(06)                </NameIdentifier>
(07)            </Subject>
(08)            <Attribute AttributeName="PaidStatus"
(09)                       AttributeNamespace="http://company.com">
(10)                <AttributeValue>PaidUp</AttributeValue>
(11)            </Attribute>
(12)            <Attribute AttributeName="CreditLimit"
(13)                       AttributeNamespace="http://company.com">
(14)                <AttributeValue xsi:type="my:type">
(15)                    <my:amount currency="EUR">500.00<my:amount>
(16)                </AttributeValue>
(17)            </Attribute>
(18)        </AttributeStatement>
(19)        <ds:Signature>...</ds:Signature>
(20)    </Assertion>
```

Esta asserção informa que o utilizador user@example.com tem as contas em dia e diz qual o limite de crédito que tem disponível. Os atributos são específicos a cada aplicação, que os qualifica num espaço de nomes próprio.

A linha (01) indica que se trata de uma asserção. A linha (02) indica que se trata da asserção de atributos. As linhas (03) a (07) especificam o sujeito a que se referem os atributos. As linhas (08) a (11) indicam o valor do atributo ‘PaidStatus’. As linhas (12) a (17) indicam o valor do atributo ‘CreditLimit’. A linha (19) é a assinatura da entidade emissora da asserção.

Exemplo 4. Asserção SAML de autorização.

```
(01)    <Assertion>
(02)        <Conditions NotBefore="2006-07-22T12:02:00Z" NotOnOrAfter="2006-
        07-22T13:02:00Z">
(03)        </Conditions>
(04)        <AuthorizationDecisionStatement
        Resource="http://www.company.com/info"
(05)                                        Decision="Permit">
(06)            <Subject>
(07)                <NameIdentifier
        Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
(08)                user@example.com
(09)                </NameIdentifier>
(10)            </Subject>
(11)            <Action
        Namespace="urn:oasis:names:tc:SAML:1.0:action:rwedc">Read</Action>
(12)        </AuthorizationDecisionStatement>
(13)        <AuthorizationStatement
        Resource="http://www.company.com/register.cgi"
(14)                                    Decision="Permit">
(15)            <Subject>...</Subject>
(16)            <Action
        Namespace="urn:oasis:names:tc:SAML:1.0:action:rwedc">Execute</Action>
(17)        </AuthorizationStatement>
(18)        <ds:Signature>...</ds:Signature>
(19)    </Assertion>
```

Esta asserção afirma que o utilizador user@example.com está autorizado a consultar a página http://www.company.com/info e que pode submeter o formulário http://www.company.com/ register.cgi. Ambas as decisões foram permitir (Permit), mas poderiam ter sido negar (Deny) ou indeterminadas (Indeterminate).

A linha (01) indica que se trata de uma asserção. A linha (02) indica a validade temporal da asserção. As linhas (04) a (12) autorizam a leitura do recurso. As linhas (13) a (17) autorizam a execução do programa que processa o formulário. A linha (18) contém a assinatura da entidade emissora da asserção de autorização.

As asserções podem ser anexadas a mensagens e podem ser aceites em diferentes domínios de segurança, desde que, directa ou indirectamente, se confie no emissor. Qualquer entidade pode emitir asserções. Cabe depois ao receptor da asserção, verificá-la e decidir se confia no seu conteúdo ou não. O mecanismo mais usual para atribuir confiança a asserções é a assinatura digital.

A *XACML (eXtensible Access Control Markup Language)* [36] especifica regras de acesso à informação que permitem: o controlo fino de actividades autorizadas, a análise de características do cliente, a autorização baseada em classes de actividades e a introspecção de conteúdos.

A *REL (Rights Expression Language)* [37] especifica formas de proteger os direitos de autor de conteúdos acessíveis através de serviços.

A *XKMS (XML Key Management Specification)* [38] define protocolos para distribuir e registar chaves públicas, adequadas ao uso na XML-Signature e na XML-Encryption. A XKMS dá acesso à autoridade de certificação, o que possibilita operações de confirmação e revogação de chaves.

2.3 Implementações disponíveis

As principais implementações disponíveis de segurança para serviços são:

- WSE 3 (Web Services Enhancements 3) para Microsoft Dot Net 2 [39];
- WSS4J (Web Services Security for Java), sobre Apache Axis2, para Java [40];
- XWSS (XML and Web Services Security), sobre JAX-WS 2, disponível no pacote JWSDP 2.0, para Java [41].

A tabela 1 indica quais as normas suportadas por cada implementação.

Tabela 1. Normas suportadas nas principais implementações de serviços seguros.

Fornecedor	Implementação	Normas suportadas
Microsoft	**WSE 3**: Dot Net Framework 2.0, Visual Studio 2005, Web Services Enhancements 3.0	WS-Security: Username, X.509, Kerberos WS-Secure Conversation, WS-Trust SAML[2]
Apache	**WSS4J**: Apache Axis2, Rampart module of Web Services Security for Java (WSS4J)	WS-Security: Username, X.509 WS-Policy SAML
Sun Microsystems	**XWSS**: Java Web Services Developer Pack 2.0, XML and Web Services Security 2.0	WS-Security: Username, X.509 SAML

3 Conclusão

Os serviços têm demonstrado ser um tópico de interesse para a comunidade científica, existindo actualmente três conferências internacionais anuais:

- ICWS (IEEE International Conference on Web Services), que se realiza desde 2004;
- ECOWS (IEEE European Conference on Web Services), que se realiza desde 2005;
- NWeSP (International Conference on Next Generation Web Services Practices), que se realiza também desde 2005.

A nível de modelo de aplicações seguras estruturadas em serviços, Gutierrez [42] deparou-se com o problema de existirem diversas normas de serviços mas não existir uma metodologia de desenvolvimento para aplicá-las. No seu artigo propõe um processo que acompanha todo o ciclo de desenvolvimento, com possibilidade de seguir o mapeamento dos requisitos na arquitectura e depois na tecnologia utilizada. A figura 8 resume a abordagem proposta.

[2] O ambiente Microsoft suporta experimentalmente as asserções SAML, mas não o protocolo de comunicação.

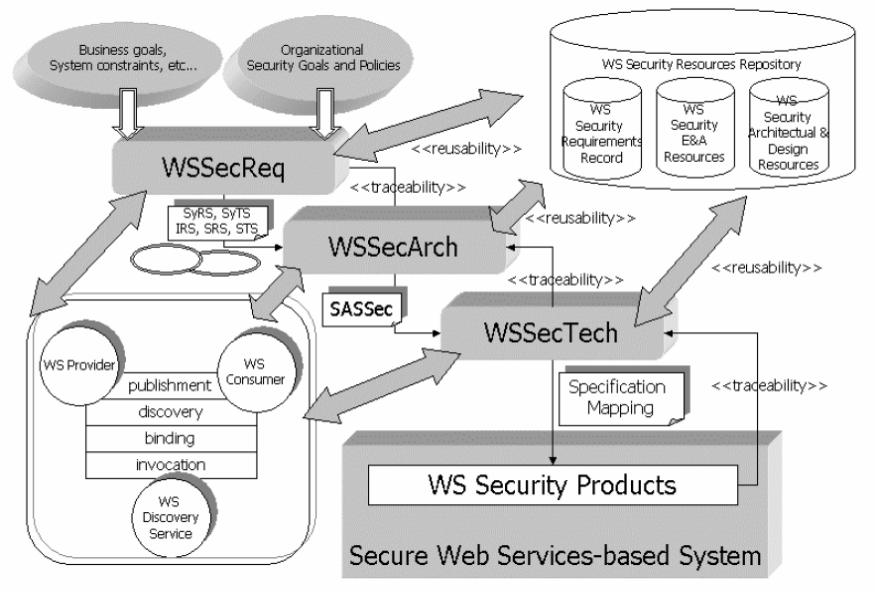

Fig. 8. Processo de desenvolvimento para serviços seguros [42].

A delegação em serviços, ou seja, a noção que a identidade do agente cliente deve ser usada para autorização e mantida no contexto de execução de serviços no âmbito de um processo de negócio, é um problema reconhecido para o qual estão a ser efectuadas várias propostas. Bengtsson [43] propõe que a orquestração segura de serviços seja baseada no registo, acompanhamento e autenticação das identidades assumidas pelos serviços ao longo da sua execução. Mello [44] propõe uma abordagem de federação entre domínios para a troca de informação de segurança para autenticação e autorização. Vecchio [45] propõe um sistema de gestão de credenciais para serviços centradas no utilizador, tendo realizado uma implementação multi-plataforma com WS-Security e WS-Trust que permite a emissão e troca de credenciais, sendo a delegação apontada como trabalho futuro relevante. Wang [46] propõe uma extensão à SAML, definindo asserções de delegação, cuja segurança é baseada em assinaturas XML com certificados X.509.

O principal contributo deste artigo é um levantamento da tecnologia de segurança para Web Services, que permite situar trabalhos de investigação que pretendam aprofundar o estudo das normas e das implementações disponíveis, recorrendo, por exemplo, a casos práticos de estudo, para aferir a sua utilidade e versatilidade a dar resposta aos requisitos de segurança. Além de ponto de partida para trabalhos de investigação, este artigo é também útil como elemento informativo em decisões de adopção de tecnologia, em organizações que procuram formas de tornar mais seguros os seus sistemas que fazem uso de Web Services.

Referências

1. Endrei, M.; Ang, J.; Arsanjani, A.; Chua, S.; Comte, P.; Krogdahl; Luo, M. & Newling, T., "Service-Oriented Architecture and Web Services", IBM RedBooks, 2004 http://publib-b.boulder.ibm.com/abstracts/sg246303.html?Open
2. AUTOR1., "Segurança de aplicações empresariais em arquitecturas de serviços", ORGANIZACAO, Setembro 2006
3. AUTOR1, "Em construção: uma análise ao estado actual da plataforma de Serviços Web para negócio electrónico", XATA2006, XML: Aplicações e Tecnologias Associadas, 2006
4. Bray, T.; Paoli, J.; McQueen, C.M.S.; Maler, E. & Yergeau, F., "Extensible Markup Language (XML) 1.0 (Third Edition)", W3C, 2004 http://www.w3.org/TR/2004/REC-xml-20040204
5. Fallside, D.C. & Walmsley, P., "XML Schema Part 0: Primer Second Edition", W3C, 2004 http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/
6. WSI, "Interoperability: Ensuring the Success of Web Services - an overview of WS-I", WS-I Web Site, 2005 http://www.ws-i.org/about/Default.aspx
7. Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P. & Lee, T.B., "Hypertext Transfer Protocol -- HTTP/1.1", IETF, 1999 http://www.w3.org/Protocols/rfc2616/rfc2616.txt
8. Klensin, J., "Simple Mail Transfer Protocol", IETF, 2001 http://www.ietf.org/rfc/rfc2821.txt
9. Gudgin, M.; Hadley, M.; Mendelsohn, N.; Moreau, J. & Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C, 2003 http://www.w3.org/TR/2003/REC-soap12-part1-20030624/
10. Booth, D. & Liu, C.K., "Web Services Description Language (WSDL) Version 2.0", W3C, 2005 http://www.w3.org/TR/2005/WD-wsdl20-primer-20050803
11. Schlimmer, J., "Web Services Policy Framework (WSPolicy) Version 1.2", Microsoft, IBM, VeriSign, Sonic Software, SAP, BEA Systems, 2006 http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf
12. Clement, L.; Hately, A.; von Riegen, C. & Rogers, T., "UDDI Version 3.0.2", OASIS, 2004 http://uddi.org/pubs/uddi-v3.0.2-20041019.htm
13. Curbera, F. & Schlimmer, J., "Web Services Metadata Exchange (WS-MetadataExchange)", MSDN, Microsoft, IBM, Computer Associates, SAP, BEA Systems, Sun Microsystems, webMethods, 2004 http://msdn.microsoft.com/ws/2004/09/ws-metadataexchange/
14. Anthony Nadalin, C.K., "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)", OASIS, 2004 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
15. Kaler, C. & Nadalin, A., "Web Services Security Policy Language (WS-SecurityPolicy) Version 1.1", Microsoft, IBM, VeriSign, RSA Security, 2005 http://www.ibm.com/developerworks/library/specification/ws-secpol/
16. Iwasa, K., "Web Services Reliable Messaging TC WS-Reliability 1.1", OASIS, 2004 http://docs.oasis-open.org/wsrm/ws-reliability/v1.1
17. Ferris, C. & Langworthy, D., "Web Services Reliable Messaging Protocol (WS-ReliableMessaging)", Microsoft, IBM, BEA, TIBCO Software, 2005 http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-ReliableMessaging.pdf
18. Feingold, M., "Web Services Coordination (WS-Coordination) Version 1.0", IBM, Microsoft, Hitachi, Arjuna Technologies, IONA, 2005 http://www.ibm.com/developerworks/library/specification/ws-tx/
19. Little, M., "Web Services Composite Application Framework (WS-CAF) version 1.0", Sun, Oracle, IONA, Arjuna, Fujitsu, 2003 http://www.arjuna.com/library/specs/ws_caf_1-0/WS-CAF-Primer.pdf

20. Thatte, S., “Business Process Execution Language for Web Services Version 1.1”, Microsoft, IBM, Siebel Systems, BEA, SAP, 2003 http://www.ibm.com/developerworks/library/specification/ws-bpel/
21. Kavantzas, N.; Burdett, D. & Ritzinger, G., “Web Services Choreography Description Language Version 1.0”, W3C, 2004 http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/
22. Geller, A., “Web Services for Management (WS-Management)”, Microsoft, Sun, Intel, AMD, Dell, 2004 http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-management1004.pdf
23. Sedukhin, I. & Vambenepe, W., “Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0 and Management Using Web Services (MUWS 1.0)”, OASIS, 2005 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm
24. Schwarz, J.; Hartman, B.; Nadalin, A.; Kaler, C.; Davis, M.; Hirsch, F. & Morrison, K.S., “Security Challenges, Threats and Countermeasures Version 1.0”, WS-I, 2005 http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0.pdf
25. Hogg, J.; Smith, D.; Chong, F.; Taylor, D.; Wall, L. & Slater, P., “Web Service Security Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0”, Microsoft, 2005
26. IBM & Microsoft, “Security in a Web Services World: A Proposed Architecture and Roadmap Version 1.0”, IBM, Microsoft, 2002 http://www.ibm.com/developerworks/library/specification/ws-secmap/
27. Housley, R.; Ford, W.; Polk, W. & Solo, D., “Internet X.509 Public Key Infrastructure”, IEFT, 1999 http://www.ietf.org/rfc/rfc2459.txt
28. J. Kohl, C.N., “The Kerberos Network Authentication Service (V5)”, IETF, 1993 http://www.ietf.org/rfc/rfc1510.txt
29. Eastlake, D.; Reagle, J. & Solo, D., “XML-Signature Syntax and Processing”, W3C, 2002 http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/
30. Eastlake, D. & Reagle, J., “XML Encryption Syntax and Processing”, W3C, 2002 http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/
31. Rosenberg, J. & Remy, D., “Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature and XML Encryption”, SAMS, 2004
32. Gudgin, M. & Nadalin, A., “Web Services Trust Language (WS-Trust)”, Microsoft, IBM, OpenNetwork, Layer 7, Computer Associates, VeriSign, BEA, Oblix, Reactivity, RSA Security, Ping Identity, VeriSign, Actional, 2005 http://www.ibm.com/developerworks/library/specification/ws-trust/
33. Gudgin, M. & Nadalin, A., “Web Services Secure Conversation Language (WS-SecureConversation)”, Microsoft, IBM, OpenNetwork, Layer 7, Computer Associates, VeriSign, BEA, RSA Security, Ping Identity, Actional, Computer Associates, 2005 http://www.ibm.com/developerworks/library/specification/ws-secon/
34. Kaler, C. & Nadalin, A., “Web Services Federation Language (WSFederation) Version 1.0”, OASIS, 2003 http://www.ibm.com/developerworks/library/specification/ws-fed/
35. Cantor, S.; Kemp, J.; Philpott, R. & Maler, E., “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS, 2004 http://xml.coverpages.org/SAML-core-20-CD-01.pdf
36. Moses, T., “eXtensible Access Control Markup Language (XACML) Version 2.0”, OASIS, 2005 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
37. DeMartini, T.; Nadalin, A.; Kaler, C.; Monzillo, R. & Baker, P.H., “Web Services Security Rights Expression Language (REL) Token Profile”, OASIS, 2004 http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf
38. Baker, P.H. & Mysore, S.H., “XML Key Management Specification (XKMS 2.0) Version 2.0”, W3C, 2005 http://www.w3.org/TR/2005/REC-xkms2-20050628/

39. Microsoft, “Microsoft Web Services Enhancements (WSE) 3.0 documentation”, 2005 http://msdn.microsoft.com/webservices/webservices/building/wse/default.aspx
40. Apache, “Securing SOAP Messages with WSS4J”, 2006 http://ws.apache.org/axis2/modules/rampart/1_0/security-module.html
41. Sun, “Java Web Services Developer Pack”, Sun Microsystems Web Site, 2006 http://java.sun.com/webservices/
42. Gutierrez, C.; Medina, E.F. & Piattini, M., “Web services enterprise security architecture: a case study”, SWS '05: Proceedings of the 2005 workshop on Secure web services, ACM Press, 2005, 10-19
43. Bengtsson, A. & Westerdahl, L., “Secure Choreography of Cooperating Web Services”, Proceedings of the Third European Conference on Web Services (ECOWS’05), IEEE Computer Society, 2005 http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=33570&arnumber=1595725&count=29&index=19
44. de Mello, E.R. & da Silva Fraga, J., “Mediation of Trust across Web Services”, Proceedings of the IEEE International Conference on Web Services (ICWS’05), 2005 http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=32665&arnumber=1530842&count=129&index=72
45. Vecchio, D.D.; Humphrey, M.; Basney, J. & Nagaratnam, N., “CredEx: User-Centric Credential Management for Grid and Web Services”, Proceedings of the IEEE International Conference on Web Services (ICWS’05), 2005 http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=32665&arnumber=1530793&count=129&index=25
46. Wang, J.; Vecchio, D.D. & Humphrey, M., “Extending the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services”, Proceedings of the IEEE International Conference on Web Services (ICWS’05), 2005