STUDIA INFORMATICA 2003
Volume 24 Number 2B (54)

Michał KOZIELSKI
Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki

# STANDARDY BEZPIECZEŃSTWA DLA XML

**Streszczenie**. Opracowanie przedstawia najnowsze standardy bezpieczeństwa związane z coraz powszechniej wykorzystywanym językiem XML. W artykule uzasadniono potrzebę zastosowania nowych standardów odpowiadających strukturze dokumentów XML. Przedstawiona została również krótka charakterystyka wybranych standardów oraz perspektywy ich zastosowań.

**Słowa kluczowe**: XML, bezpieczeństwo.

# XML SECURITY STANDARDS

**Summary**. The paper presents the latest security standards connected with eXtensible Makup Language (XML), which is increasingly popular in use. The need of introduction the new standards corresponding with the XML document structure is justified in the article. The paper includes also a short characteristic of the selected standards and the perspectives of their implementation application.

**Keywords**: XML, security, XML Encryption, XML Digital Signature.

## 1. Język XML i jego zastosowania

Język XML (ang. *eXtensible Markup Language*) służy do strukturalnego opisu danych za pomocą znaczników. Zalety tego języka sprawiły, że w ciągu kilku ostatnich lat XML stał się ogromnie popularny i powszechnie wykorzystywany. XML znalazł zastosowanie przy tworzeniu stron WWW, dokumenty XML są wykorzystywane przez protokół SOAP, stworzony na potrzeby usług sieciowych. Liczba przesyłanych danych w postaci dokumentów XML stała się tak znacząca, że powstały komercyjne bazy danych dedykowane specjalnie do przechowywania dokumentów XML (ang. *native XML databases*).

Główne zastosowanie XML to przesył danych w środowiskach heterogenicznych, czyli na przykład w sieci Internet. Powszechny dostęp do sieci ułatwia podsłuch informacji oraz celowe manipulacje przesyłanymi przez nią danymi. Jeżeli zadania realizowane poprzez sieć mają być wiarygodne, musi zostać zapewnione bezpieczeństwo przesyłanych danych, również zawartych w dokumentach XML.

## 2. Wymagania bezpieczeństwa w sieci Internet

Do podstawowych wymagań bezpieczeństwa związanych z przesyłaniem danych w sieci należą [10, 11]:

- poufność – czyli zapewnienie, że tylko odbiorca może przeczytać przesyłane dane,
- autentyfikacja – czyli umożliwienie identyfikacji strony przesyłającej dane,
- nienaruszalność – czyli uniemożliwienie niezauważalnej zmiany danych,
- niezaprzeczalność – czyli zapewnienie, że przesyłane dane pochodzą od ich autora.

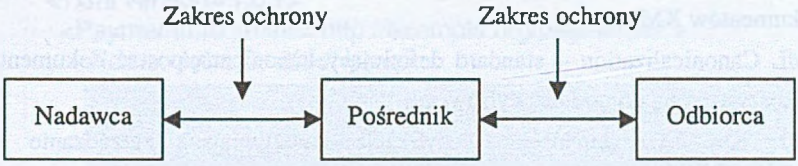
Omówione w kolejnych punktach artykułu standardy zapewniają wymienione, fundamentalne warunki bezpieczeństwa dla dokumentów XML.

## 3. Celowość wprowadzania nowych zabezpieczeń

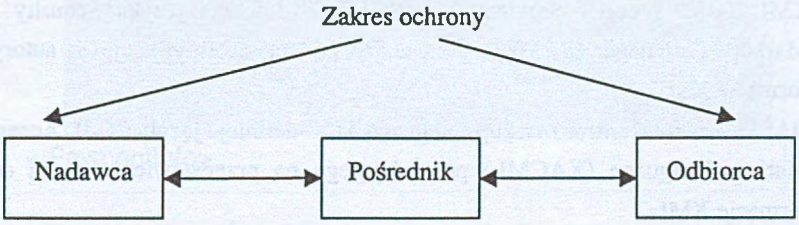
W celu zabezpieczenia przesyłanych danych można wykorzystać istniejące już zabezpieczenia, takie jak SSL/TLS lub protokół IPSec. Zabezpieczenia te chronią jednak jedynie logiczny kanał przesyłu danych [6] (ang. *point-to-point configuration*) (rys. 1).

W przesyle mogą uczestniczyć węzły pośredniczące, które wykorzystują lub przetwarzają przesyłane dane. Węzeł pośredniczący łączy dwa bezpieczne kanały logiczne, a końcowy odbiorca przesyłu musi całkowicie ufać pośrednikowi oraz przetwarzanym przez niego danym.

Pełniejsze bezpieczeństwo gwarantuje mechanizm chroniący przesyłane dane na całej ich trasie (ang. *end-to-end configuration*) (rys. 2), czyli nie tylko na poziomie warstwy transportowej, lecz również na poziomie warstwy aplikacji. Pozwala on na ochronę danych w przypadku braku zabezpieczeń, takich jak SSL/TLS lub protokół IPSec oraz na wykorzystanie własności języka XML, tak jak zostanie to przedstawione w punktach 4.1 i 4.2.




Rys. 1. Ochrona danych w konfiguracji *point-to-point*
Fig. 1. Point-to-point data security configuration




Rys. 2. Ochrona danych w konfiguracji *end-to-end*
Fig. 2. End-to-end data security configuration

Istnieje więc potrzeba zastosowania mechanizmów bezpieczeństwa pozwalających na ochronę danych XML na całej przestrzeni pomiędzy nadawcą i odbiorcą. Mechanizmy te powinny również być zgodne z istniejącymi specyfikacjami języka XML, tak aby narzędzia przetwarzające przesyłane dane operowały na poprawnych dokumentach XML. Przedstawione i opisane poniżej standardy nie tylko spełniają powyższe warunki, lecz również tworzą nowe możliwości zabezpieczania danych oraz ich przetwarzania dzięki wykorzystaniu struktury dokumentów XML.

## 4. Istniejące standardy bezpieczeństwa XML

Standardy dotyczące mechanizmów bezpieczeństwa związanych z językiem XML tworzone są głównie przez dwie organizacje: World Wide Web Consortium (W3C) oraz Organization for the Advancement of Structured Information Standards (OASIS) [10, 11]. Dodatkowo korporacje informatyczne (np. IBM, Microsoft), rozwijające technologie związane z językiem XML (np. usługi sieciowe), przedstawiają własne propozycje polityki ochrony danych. Poniżej wymienione zostały standardy zaproponowane przez organizacje W3C i OASIS.

Standardy W3C:

- XML Encryption – standard definiujący mechanizm szyfrowania dokumentów XML,

- XML Digital Signature – standard definiujący mechanizm podpisu cyfrowego dokumentów XML,
- XML Canonicalization – standard definiujący kanoniczną postać dokumentu XML wykorzystywaną do podpisu cyfrowego,
- XML Key Management – specyfikacja umożliwiająca zarządzanie kluczem publicznym przy szyfrowaniu i podpisie cyfrowym.

Standardy OASIS:

- XML-Based Security Services TC (SSTC) – definicja języka Security Assertion Markup Language (SAML), pozwalającego na autentyfikację i autoryzację w formacie XML,
- OASIS Access Control Markup Language TC – definicja języka XML Access Control Markup Language (XACML) pozwalającego na przedstawienie zasad dostępu w formacie XML,
- OASIS Digital Signature Services TC – definicja technik wspierających przetwarzanie i weryfikację podpisu cyfrowego.

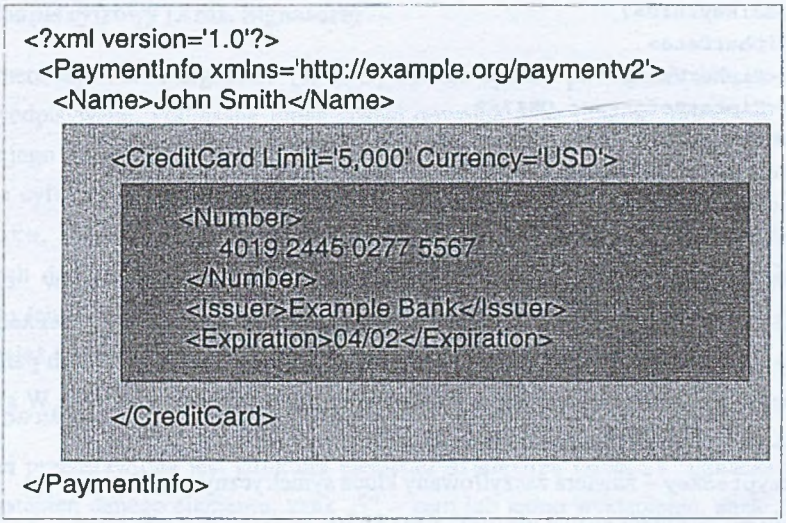
W kolejnych podpunktach przedstawione zostaną dwa, zaproponowane niedawno przez W3C, standardy, które spełniają fundamentalne wymagania bezpieczeństwa wymienione w punkcie 2. Standardy XML Encryption i XML Digital Signature zapewniają poufność, nienaruszalność oraz niezaprzeczalność przesyłanych danych. Podstawową cechą i zaletą obydwu standardów jest możliwość ich selektywnego wykorzystywania do różnych części dokumentów XML. Stosując poniższe standardy do dokumentu XML, otrzymamy poprawny dokument XML.

Jeżeli niektóre elementy (jawne) przesyłanego dokumentu wymagałyby przetworzenia przez węzeł pośredniczący, wtedy można zaszyfrować pozostałe części zgodnie z poniższymi standardami. Pośrednik przesyłu będzie mógł operować na poprawnym dokumencie XML, mając dostęp jedynie do jego jawnej części.

### 4.1. Szyfrowanie dokumentów XML (XML Encryption)

Szyfrowanie dokumentów XML z wykorzystaniem rekomendacji XML Encryption [4, 5, 7] pozwala na zapewnienie poufności danych podczas przesyłania dokumentu.

Szyfrowanie może zostać przeprowadzone na różnych poziomach granulacji dokumentu (np. dla całego dokumentu lub niektórych jego elementów) (rys. 3), przy wykorzystaniu różnych metod i algorytmów szyfrowania (kluczem symetrycznym, kluczem asymetrycznym).




Rys. 3. Szyfrowanie na różnych poziomach granulacji dokumentu (cały dokument, element, zawartość elementu)
Fig. 3. Encryption at the different document granularity levels (a whole document, an element, an element content)

Dokument, którego elementy zostały zaszyfrowane różnymi kluczami, może być przesłany do różnych odbiorców posiadających różne klucze, pozwalające na odczytanie różnych części dokumentu. Metoda ta może zostać wykorzystana, gdy wytworzenie wielu dokumentów adresowanych do konkretnych odbiorców jest nieefektywne. Algorytm takiego szyfrowania przy założeniu możliwości tylko jednokrotnego szyfrowania każdego elementu przedstawiony został przez Bertino i Ferrari [2].

### *4.1.1. Struktura zaszyfrowanego dokumentu*

W wyniku zaszyfrowania dokumentu XML elementy szyfrowane zostają zastąpione przez element `EncryptedData`. Strukturę tego elementu przedstawia poniższa definicja, w której „?" oznacza zero lub jedno wystąpienie, „+" oznacza jedno lub więcej wystąpień, „*" oznacza zero lub więcej wystąpień danego elementu.

```
<EncryptedData Id? Type? MimeType? Encoding?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievalMethod>?
    <ds:*>?
```

```
    </ds:KeyInfo>?
    <CipherData>
      <CipherValue>?
      <CipherReference URI?>?
    </CipherData>
    <EncryptionProperties>?
  </EncryptedData>
```

Element `CipherData` zawiera zaszyfrowane dane umieszczone w elemencie `CipherValue` lub odwołuje się do nich poprzez URI zawarte w elemencie `CipherReference`.

Zaszyfrowany dokument, jak i podpis cyfrowy, przedstawiony w następnych paragrafach, mogą zawierać opcjonalną informację o kluczu zawartą w elemencie `KeyInfo`. W skład tego elementu mogą wchodzić między innymi takie elementy, jak:

- `EncryptedKey` – zawiera zaszyfrowany klucz symetryczny,
- `KeyName` – zawiera nazwę identyfikującą klucz,
- `KeyValue` – zawiera pojedynczy klucz publiczny,
- `RetrievalMethod` – zawiera odniesienie do klucza poprzez podany adres URI, gdy klucz nie został dołączony do dokumentu.

Jeżeli w przedstawionym na rysunku 3. dokumencie XML zostanie zaszyfrowana wyróżniona najciemniejszym kolorem zawartość elementu `CreditCard`, dokument ten może mieć następującą postać:

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD'>
    <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
     Type='http://www.w3.org/2001/04/xmlenc#Content'>
      <CipherData>
        <CipherValue>A23B45C56</CipherValue>
      </CipherData>
    </EncryptedData>
  </CreditCard>
</PaymentInfo>
```

Rekomendacja XML Encryption pozwala na szyfrowanie danych, które nie mają postaci dokumentu XML. W takim przypadku po ich zaszyfrowaniu powstaje dokument, którego korzeniem jest element `EncryptedData`.

### 4.2. Podpis cyfrowy (XML Signature)

Rekomendacja XML Signature [1, 4, 5] opisuje sposób powiązania klucza z danymi, które są podpisywane. Podpisane mogą zostać dowolne dane binarne, dokument XML lub niektóre z jego elementów.

Podpis cyfrowy oraz odpowiednie informacje z nim związane zawarte są w elemencie `Signature`. Element ten może być przesyłany lub przechowywany niezależnie od podpisanych danych (ang. *detached document*), a może być dołączony do podpisywanego dokumentu (ang. *enveloped signature*) jako jeden z jego elementów. Element `Signature` może również zawierać podpisane dane w sobie (ang. *enveloping signature*).

#### *4.2.1. Struktura podpisanego dokumentu*

Poniżej przedstawiona jest struktura elementu `Signature` (znak „+" oznacza jedno lub więcej wystąpień danego elementu, znak „?" – zero lub jedno wystąpienie, znak „*" – zero lub więcej wystąpień danego elementu).

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI? >
      (<Transforms>)?
      <DigestMethod>
      <DigestValue>
    </Reference>)+
  </SignedInfo>
  <SignatureValue>
 (<KeyInfo>)?
 (<Object ID?>)*
</Signature>
```

Element `CanonicalizationMethod` informuje o metodzie wykorzystanej do uzyskania postaci kanonicznej podpisywanego dokumentu. Dokument przed podpisaniem oraz przed weryfikacją podpisu należy sprowadzić do postaci kanonicznej, gdyż XML pozwala na pewną dowolność (np. stosowanie pojedynczych lub podwójnych apostrofów). W przesyłanym dokumencie mogą nastąpić zmiany podczas przetwarzania go, na przykład, przez parsery XML stworzone przez różnych producentów. Otrzymany dokument może więc być logicznie identyczny z wysłanym, jednak funkcja haszująca da, z jego postaci niekanonicznych, dwa różne skróty.

Dokładny opis postaci kanonicznej XML jest zawarty w propozycji standardu „Canonical XML" [3] przedstawionej przez organizację W3C.

Następnym obowiązkowym elementem jest `SignatureMethod`, gdzie podany jest algorytm podpisu.

Kolejny element, `Reference`, może wystąpić wielokrotnie, gdyż w dokumencie może zostać podpisanych na przykład wiele różnych elementów. W skład elementu `Reference` wchodzą następujące elementy:

- `Transforms` – zawiera rodzaj transformacji, jakiej został poddany podpisywany element (np. wyszukanie elementu za pomocą języka XPath),
- `DigestMethod` – zawiera wskazanie na wykorzystaną funkcję haszującą,
- `DigestValue` – zawiera skrót uzyskany za pomocą funkcji haszującej.

Właściwy podpis cyfrowy (czyli skrót zaszyfrowany kluczem tajnym podpisującego) zawarty jest w elemencie `SignatureValue`.

## 5. Zastosowania przedstawionych mechanizmów oraz perspektywy ich rozwoju

Opis danych za pomocą języka XML jest powszechnie wykorzystywany i znajduje coraz poważniejsze zastosowania. Wymagają one dużej wiarygodności, a implementacja standardów bezpieczeństwa związanych z XML jest coraz bardziej nagląca. Problem bezpieczeństwa w sieci jest jednak niezwykle rozległy, o czym świadczy liczba standardów, które zostały stworzone.

Potrzeba wiarygodności stosowanych technologii jest oczywista również dla korporacji informatycznych i wydaje się, że nowe przedstawione w tym artykule mechanizmy bezpieczeństwa, związane z dokumentami XML, znajdą zastosowanie chociażby w takich rozwiązaniach, jak usługi sieciowe. Pomimo wielości standardów i organizacji standaryzujących istnieje szansa na stworzenie w kolejnym kroku, opartej na przedstawionych standardach, spójnej polityki bezpieczeństwa dla Web Services. Szansa ta niestety maleje w związku z doniesieniami o pojawianiu się nowych, odrębnych specyfikacji tworzonych przez różne grupy korporacji [8].

## LITERATURA

1. Bartel M., Boyer J., Fox B., LaMacchia B., Simon E.: XML-Signature Syntax and Processing W3C Recommendation 12 February 2002, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, 12.02.2002, (21.11.2002).
2. Bertino E., Ferrari E.: Secure and Selective Dissemination of XML Documents, ACM Transactions on Information and System Security, Vol. 5, No. 3, 2002.
3. Boyer J.: Canonical XML Version 1.0 W3C Recommendation 15 March 2001, http://www.w3.org/TR/2001/REC-xml-c14n-20010315, 15.03.2001, (02.02.2003).
4. Djajadinata R.: Yes, you can secure your Web services documents, Part 1, http://www.javaworld.com/javaworld/jw-08-2002/jw-0823-securexml.html, 23.08.2002, (17.11.2002).
5. Hirsch F.: Getting Started With XML Security, http://home.earthlink.net/~fjhirsch/xml/xmlsec/starting-xml-security.html, 07.08.2002, (17.11.2002).
6. IBM Corporation and Microsoft Corporation, Security in a Web Services World: A Proposed Architecture and Roadmap, Version 1.0, http://www-3.ibm.com/software/solutions/webservices/wstk-info.html, 07.04.2002, (15.01.2003).
7. Imamura T., Dillaway B., Simon E.: XML Encryption Syntax and Processing W3C Proposed Recommendation 03 October 2002, http://www.w3.org/TR/2002/PR-xmlenc-core-20021003/, 03.10.2002, (17.11.2002).
8. Lange L.: Web Services: Pick A (Proprietary) Lock, http://www.techweb.com/tech/security/20030129_security, 29.01.2003, (07.02.2003).
9. Salz R.: Securing Web Services, http://webservices.xml.com/pub/a/ws/2003/01/15/ends.html 15.01.2003, (07.02.2003).
10. Stallings W.: Ochrona danych w sieci i intersieci w teorii i praktyce, WNT, Warszawa 1997.
11. Treese W.: XML, Web Services, and XML, ACM 1091-556/02/0900, (10.01.2003).

Recenzent: Dr inż. Arkadiusz Sochan

Wpłynęło do Redakcji 24 marca 2003 r.

**Abstract**

XML (eXtensible Markup Language) enables a structured data description. During the previous years it has been adopted for a growing number of applications. The popularity of XML in the Internet environment and its applications of high responsibility forced the introduction of security standards dedicated for this language.

The existing methods like SSL/TLS or IPSec guaranteed the security of the logical channel of a data transfer (point-to-point configuration) (fig. 1) whereas the end-to-end configuration (fig. 2) ensures security of the whole way of data between communicating parties. Another weakness of existing methods is the fact that they are not dedicated for XML.

Due to the growing requirements a large set of new security standards was introduced (mostly by World Wide Web Consortium (W3C) and Organization for the Advancement of Structured Information Standards (OASIS)). The XML Encryption and XML Signature standards proposed by W3C ensure the fundamental requirements of confidentiality and integrity of the transported data. These standards, described in the article, allow selective encryption and signature of a XML document (fig. 3) and give as a result a correct XML document.

The standards that are mentioned in this paper give a basis for the broaden security policies created by software corporations. The question is if these companies are able to create a well formed, cohesive security policy.

**Adres**

Michał KOZIELSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, mkoz@zeus.polsl.gliwice.pl .