Kazienko P.: *Elastyczna ochrona w elektronicznej wymianie danych*. NetWorld, grudzień 12/2003, s. 64-69.

Elastyczna ochrona w elektronicznej wymianie danych

Przemysław Kazienko
kazienko@pwr.wroc.pl, http://www.pwr.wroc.pl/~kazienko

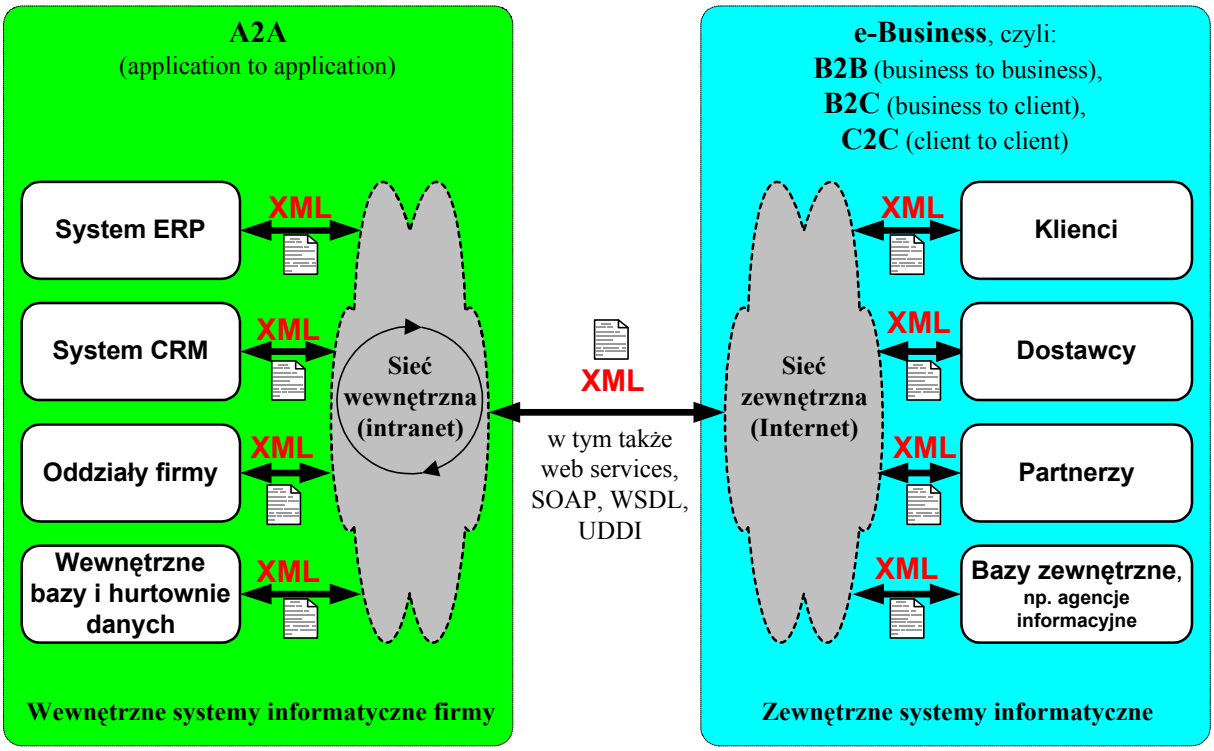
Język XML jest już powszechnie wykorzystywany w serwisach internetowych oraz w elektronicznej wymianie danych. Nowe standardy bezpieczeństwa dodatkowo rozszerzają zakres jego możliwości.

Elektroniczna wymiana danych (EDI - *Electronic Data Intercharge*) to przede wszystkim możliwość kontaktowania się w sposób elektroniczny między współpracującymi firmami (organizacjami), niezależnie od zainstalowanego w tych firmach oprogramowania i niezależnie od wykorzystywanych formatów danych. Wymiana elektroniczna pomiędzy partnerami handlowymi ma kilka istotnych zalet:

- oszczędność czasu i pieniędzy (dokumenty przekazywane są szybciej),
- zminimalizowane ryzyko błędu (nie jest konieczne powtórne „ręczne" wprowadzanie danych do systemów komputerowych),
- natychmiastowe retransmisje — dzięki elektronicznej, zrozumiałej dla kontrahenta postaci dokumentu możliwa jest natychmiastowa odpowiedź na dokument, np. potwierdzająca zamówienie, po weryfikacji ze stanami magazynowymi,
- polepszona gospodarka towarami — dzięki szybszej i pełniejszej informacji można lepiej zarządzać stanami magazynowymi,
- powiększenie rynku odbiorców — jest to istotne dla firm, które chciałyby współpracować z kontrahentami stosującymi EDI; niektórzy z nich wymagają od swoich kooperantów dokumentów w formie elektronicznej.

Jeszcze kilka lat temu powszechnymi standardami EDI były ANSI X.12 (Ameryka Północna) oraz UN/EDIFACT (reszta świata). Rozwiązania te mają jednak pewne wady: są drogie i bardzo skomplikowane — „komunikaty" EDI czyli dokumenty miały bardzo złożoną, sztywną strukturę. Obecnie coraz większa część elektronicznej wymiany wykorzystuje język XML. Istnieją międzynarodowe inicjatywy na rzecz wykorzystania XML w elektronicznej wymianie danych, jak na przykład ebXML (*electronic business XML*), BizTalk Framework, OAGIS (*The Open Applications Group Integration Specification*), BASDA (*The Business Application Software Developers Association*), czy specjalizowane RosettaNet (przemysł elektroniczny), OTA - turystyka (*The OpenTravel Alliance*). Trwają nawet prace nad adaptacją standardu X.12 do formatu XML.

Wymiana EDI może dotyczyć nie tylko podmiotów gospodarczych, ale także urzędów (np. wymiana pomiędzy urzędem skarbowym a podatnikiem czy szpitalem a Narodowym Funduszem Zdrowia), organizacji niekomercyjnych czy nawet pojedynczych osób (np. wymiana danych w systemach aukcyjnych). Przekazywanie danych może się odbywać nie tylko między odrębnymi instytucjami, ale również w ramach tej samej organizacji (np. komunikacja pomiędzy oddziałami lub różnymi wykorzystywanymi systemami) lub nawet w obrębie jednego systemu. Ważnym przykładem są usługi web services, w których dane pomiędzy niezależnymi systemami / komponentami przesyłane są za pomocą protokołu SOAP (*Simple Object Access Protocol*). Przesyłane dane mają oczywiście format XML.



Rys. 1. Elektroniczna wymiana danych z zastosowaniem języka XML

Bezpieczeństwo elektronicznej wymiany danych

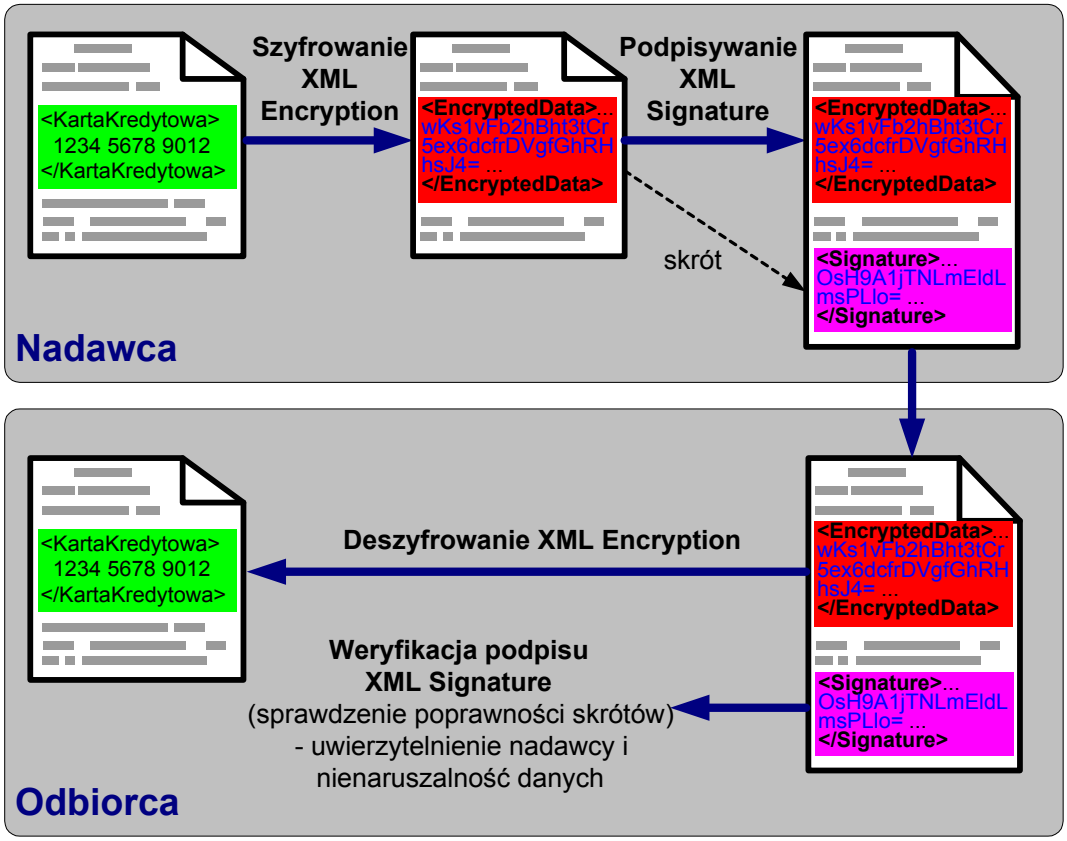
Ważnym aspektem przesyłania danych jest ich bezpieczeństwo. Ma to szczególne znaczenie w przypadku wykorzystywania ogólnodostępnych kanałów transmisji jakie oferuje sieć Internet. Projektant takiej wymiany ma do dyspozycji typowe rozwiązania: wirtualne sieci prywatne (VPN - *Virtual Private Network*), wirtualne sieci lokalne (VLAN - *Virtual Local Area Network*), protokół SSL (*Secure Socket Layer*) stosowany często w komunikacji poprzez HTTP, czy S/MIME (*Secure Multipurpose Internet Mail Extensions*) wykorzystywany zwłaszcza dla zabezpieczania listów poczty elektronicznej. Metody te są jednak ogólne tzn. traktują dokument XML jak każdy inny i chronią całą jego zawartość. Dodatkowo VPN, czy SSL dotyczą wyłącznie transmisji i po jej zakończeniu informacja o zabezpieczeniach (np. podpis elektroniczny) ginie. Oprócz tego uwierzytelniane w nich są najczęściej systemy (serwery, stacje) a nie autorzy wiadomości (użytkownicy), co może mieć istotne znaczenie np. w procesach sądowych.

Innym rozwiązaniem jest zastosowanie nowych — ostatecznie zatwierdzonych w 2002 r. — standardów bezpieczeństwa specjalizowanych dla języka XML, czyli podpisów elektronicznych (XML Signature) oraz szyfrowania (XML Encryption). Standardy te — jak cała koncepcja języka XML — są bardzo elastyczne. Z ich pomocą można w szczególności:

- zabezpieczyć niemal dowolny fragment dokumentu XML, np. tylko numer karty kredytowej w informacji o płatności lub cenę hurtową w katalogu produktów, pozostawiając resztę dokumentu niechronioną;
- zabezpieczyć kilka fragmentów jednym mechanizmem np. podpisać wszystkie poprawki naniesione przez redagującego (przydatne zapewne przy redagowaniu ustaw i przekazywaniu ich drogą elektroniczną) czy zaszyfrować wszystkie ceny z katalogu jednym kluczem;
- zastosować wiele rożnych zabezpieczeń (podpisów i szyfrów) dla tych samych danych chronionych. np. kilka podpisów złożonych pod uchwałą przez wszystkie upoważnione osoby, czy zaszyfrować dane dla kilku różnych odbiorców;
- stosować różne, nawet własne, algorytmy kryptograficzne oraz ich różne parametry;
- zabezpieczać dane binarne np. podpisywać obrazki lub programy;
- dowolnie rozdzielać lub łączyć dane zabezpieczone i dane źródłowe w tym powoływać się na dane rozproszone w sieci, np. podpisać zdalną stronę WWW zaś klucz niezbędny do weryfikacji podpisu umieścić na jeszcze innym serwerze.

Ponieważ zarówno podpisy XML Signature jak i szyfry XML Encryption mają postać elementów XML więc można je łączyć z innymi dokumentami XML.

Szyfrowanie można dowolnie łączyć z podpisywaniem tzn. najpierw szyfrować, potem podpisywać - patrz rysunek (rozwiązanie najlepsze) lub odwrotnie — sposób gorszy, gdyż weryfikacja podpisu musi być poprzedzona deszyfrowaniem XML Encryption. W obu przypadkach podpisywać można albo element zabezpieczany przez XML Encryption (np. kartę kredytową), większą część dokumentu (np. dane o płatności zawierające w sobie numer karty) lub nawet cały dokument (zamówienie z informacją o płatności).



Rys. 2. Szyfrowanie i podpisywanie za pomocą XML Encryption i XML Signature

Podpis elektroniczny czyli XML Signature

Podpis elektroniczny XML Signature gwarantuje odbiorcy (są to typowe cechy podpisu elektronicznego):

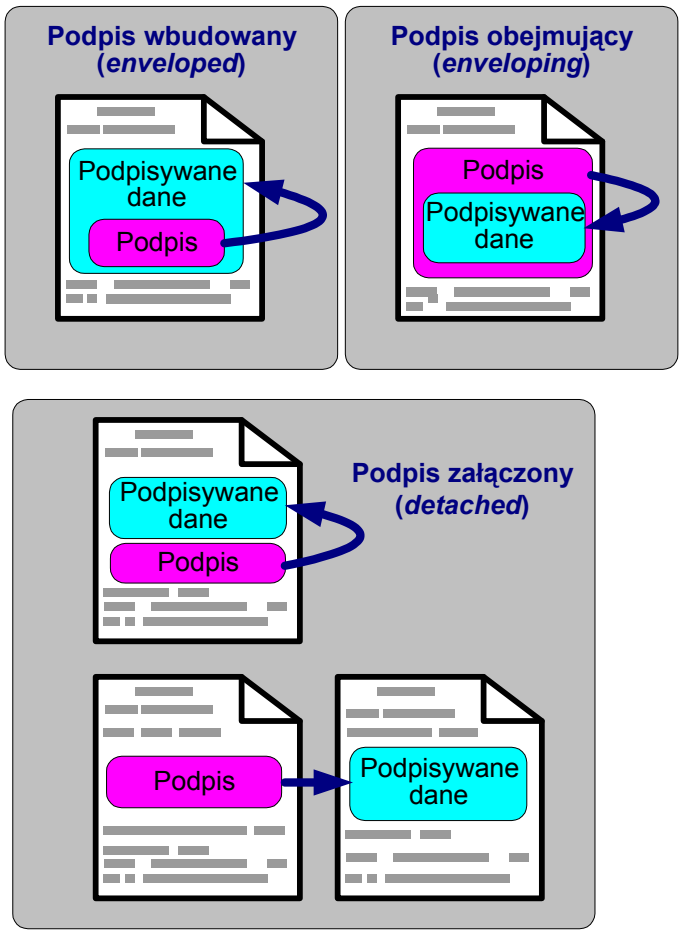
- nienaruszalność podpisywanych danych — dane nie zostały zmienione
- to, że dane podpisał ten, kto miał dostęp do klucza prywatnego skojarzonego z kluczem publicznym wskazanym w podpisie — potwierdzenie autorstwa podpisu czyli uwierzytelnienie nadawcy

Podpis nie zapewnia jednak poufności danych - dane źródłowe są jawne. W celu zagwarantowania tego, że dane mogą być odczytane tylko przez odbiorcę należy te dane — korzystając z XML Encryption — zaszyfrować wygenerowanym, losowym kluczem sesyjnym, który zostanie zaszyfrowany kluczem publicznym odbiorcy.
Podpis XML Signature nie ma postaci binarnej lecz jest tekstowym elementem XML — w trakcie podpisywania i weryfikacji następuje odpowiednie przekodowywanie znaków za pomocą algorytmu *Base64*. Podobne kodowanie jest stosowane wtedy, gdy dane podpisywane są binarne, co umożliwia podpisywanie np. pliku dźwiękowego czy uruchamialnego programu.
Podpisywane dane są wskazywane przez identyfikator URI (zalecanym jest schemat HTTP), w którym można wykorzystać wyrażenia XPointer oraz XPath. Dzięki temu możliwe jest podpisywanie niemal dowolnych fragmentów dokumentu XML.
Podpisy XML można dowolnie rozmieszczać względem podpisywanych danych, tj.:

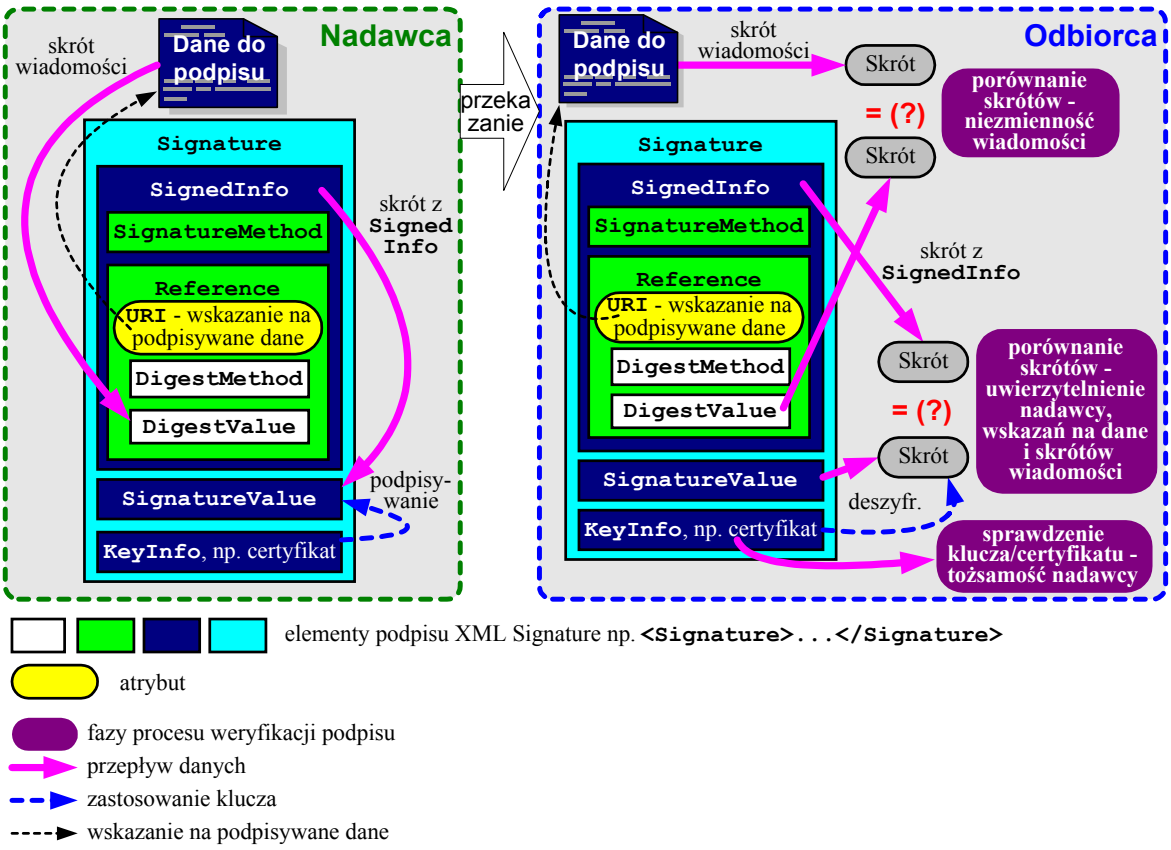
- podpis może być wbudowany (*enveloped*) w podpisywany element; podczas weryfikacji podpisu niezbędne jest wtedy odpowiednie usunięcie danych związanych z podpisem,
- podpis może zawierać w sobie podpisywane dane (*enveloping*),
- podpis jest oddzielony od danych podpisywanych (*detached*) i umieszczony w innym miejscu tego samego lub innego dokumentu.

Ten ostatni sposób umożliwia m.in. tworzenie podpisów wiadomości, które są przechowywane na zdalnych serwerach. W takim przypadku do odbiorcy może być przekazywany jedynie sam podpis, zawierający wskazanie na zdalny, podpisany zasób, np. stronę WWW lub plik dostępny przez FTP. Rozwiązanie to można także

zastosować do zgrupowania w osobnym dokumencie wszystkich podpisów złożonych dla poszczególnych dokumentów pewnej kolekcji.



Rys. 3. Umiejscowienie podpisu względem podpisywanych danych



Rys. 4. Proces podpisywania i weryfikacji podpisu XML Signature

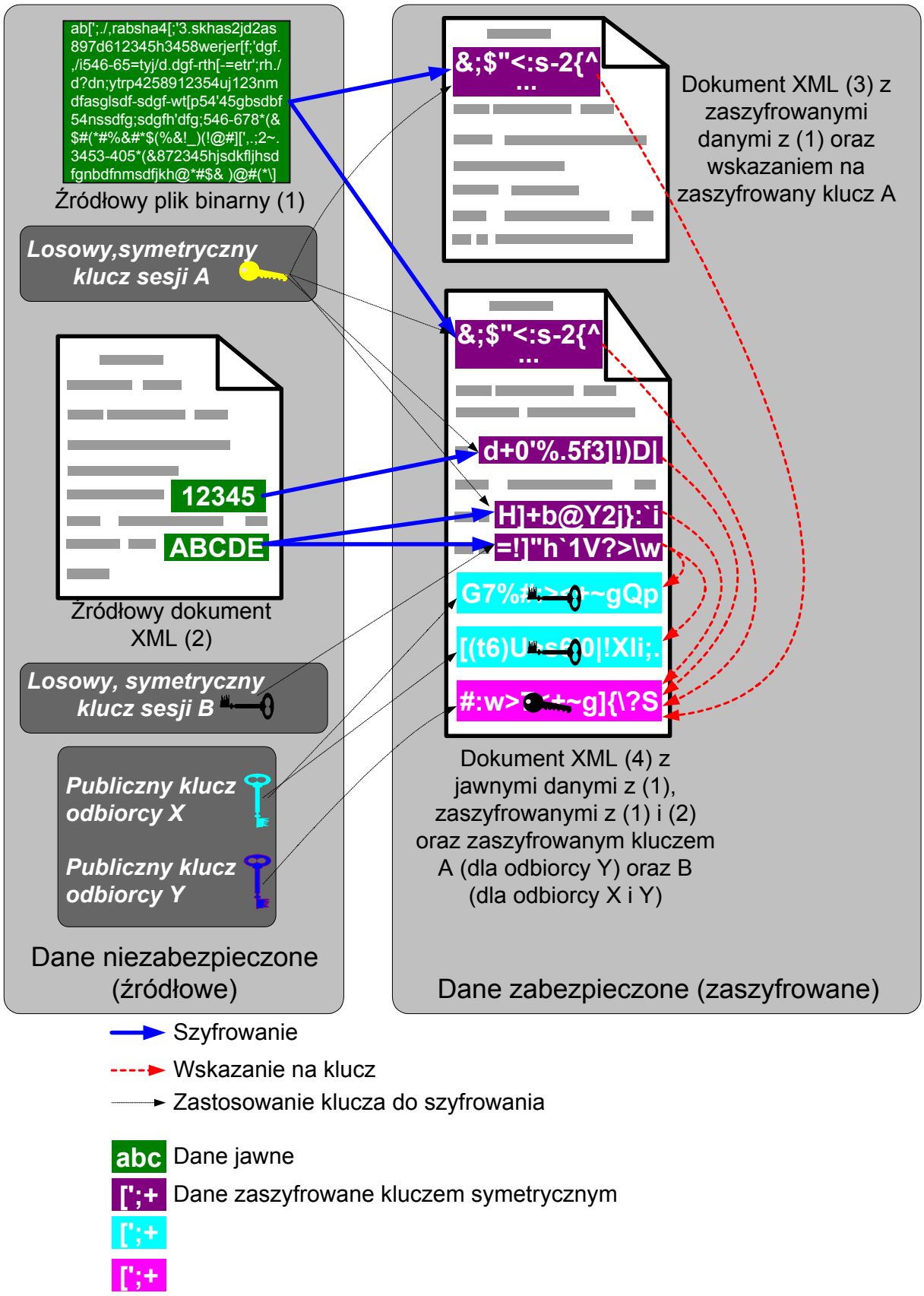
Do weryfikacji podpisu odbiorca musi posiadać klucz publiczny nadawcy. Może on być: zawarty wprost w samym podpisie (co jest oczywiście nie zalecane) lub odpowiednio wskazany, poprzez np. identyfikator certyfikatu X.509, nazwę lub identyfikator zrozumiały dla odbiorcy, adres URL.
Podpis XML Signature — podobnie jak większość obecnie dostępnych na rynku rozwiązań podpisu elektronicznego — nie zapewnia ochrony danych związanych z wysłaniem i otrzymaniem wiadomości tj. czasu oraz potwierdzenia wysłania i otrzymania. Gwarancje takie dają stemple czasowe, które jednak nie wchodzą w skład standardu XML Signature. Standard nie zawiera także żadnych mechanizmów ochrony długoterminowej, (większej niż kilka lat).
Cecha rozszerzalności podpisów XML Signature została wykorzystana przez European Telecommunications Standards Institute (ETSI), który zaproponował rozbudowanie tego standardu w postaci XML Advanced Electronic Signature (XAdES). Uwzględniono w nim m.in. stemple czasowe oraz ułatwiono realizację koncepcji podpisów długoterminowych. Polega to na dodawaniu, co jakiś czas — np. co dwa lata — dodatkowych podpisów cyfrowych wraz ze stemplami czasowymi, w których zastosowywane były by nowe algorytmy i ich parametry uznawane za bezpieczne w przyszłości.

Szyfrowanie — XML Encryption

Standard XML Encryption służy do tworzenia zaszyfrowanych wersji danych (zwykle całych elementów XML lub ich treści) w postaci pojedynczego elementu XML. Proces zabezpieczania polega najczęściej — możliwe są także inne scenariusze — na: wygenerowaniu symetrycznego klucza sesji, zaszyfrowaniu danych źródłowych i zaszyfrowaniu klucza sesji kluczem publicznym odbiorcy. W takim rozwiązaniu mamy pewność, że tylko odbiorca będzie miał dostęp do danych, gdyż tylko on posiada klucz prywatny skojarzony z wykorzystanym kluczem publicznym i tylko on może uzyskać dostęp do klucza sesji. Inny, mniej typowy, scenariusz to zastosowanie klucza symetrycznego uzgodnionego wcześniej z odbiorcą. Zaszyfrowana wersja danych zawiera wtedy jedynie wskazanie na odpowiedni klucz. W takim rozwiązaniu nie potrzebne jest stosowanie kryptografii asymetrycznej do wymiany klucza.

Ponieważ możliwe jest wykorzystanie tego samego klucza szyfrującego (sesyjnego) wiele razy więc można nim osobno zabezpieczać różne fragmenty tego samego dokumentu lub nawet wielu dokumentów, np. numery kilku kart kredytowych. W takim przypadku wielokrotnie wykorzystany klucz jest jednorazowo, w bezpieczny sposób przekazywany odbiorcy np. zaszyfrowany pod koniec danego dokumentu a nawet tylko w jednym miejscu dla całej kolekcji dokumentów XML.

Gdy klucz sesji (i tylko klucz sesji) zaszyfrujemy osobno kluczami publicznymi kilku odbiorców, wtedy będziemy mogli zaszyfrować dane równocześnie dla wielu odbiorców — każdy z odbiorców odszyfruje klucz sesji zabezpieczony osobno dla niego. W przypadku, gdy zastosujemy dodatkowo kilka kluczy sesji przeznaczonych dla różnych danych, to będziemy mogli za pomocą odpowiedniej manipulacji kluczami sesji i ich przekazywaniem (szyfrowaniem) dla poszczególnych odbiorców uzyskać coś w rodzaju „listy kontroli dostępu" do poszczególnych fragmentów danych. Powyższy scenariusz jest trudno realizowalny za pomocą innych standardowych mechanizmów zabezpieczeń wykorzystywanych w elektronicznej wymianie danych, takich jak SSL czy S/MIME.



Rys. 5. Złożone szyfrowanie XML Encryption. Element 'ABCDE' jest w dokumencie (4) zaszyfrowany dwoma kluczami („A" i „B") i przeznaczony dla dwóch odbiorców („X" i „Y"). Pozostałe zaszyfrowane elementy są dostępne jedynie dla odbiorcy „X". Dokument (3) nie zawiera zaszyfrowanej postaci klucza sesji „A". Aby nadmiernie nie rozbudowywać rysunku pominięto odpowiednie znaczniki wymagane przez prawidłowa składnię XML.

Ponieważ zaszyfrowana dla pewnego odbiorcy „A" wersja danych ma postać elementu XML, więc element ten bez problemów może być jeszcze raz zaszyfrowany z wykorzystaniem innego klucza. Jeżeli klucz ten będzie zaszyfrowany kluczem publicznym innego odbiorcy „B", wtedy dostęp do jawnej wersji będzie możliwy jedynie przy zgodzie obu odbiorców. Najpierw deszyfracji będzie musiał dokonać odbiorca „B" a potem „A".

XML Encryption jest wprawdzie niezależny od podpisów elektronicznych jednak poprzez koncepcję przestrzeni nazw wykorzystuje XML Signature gdyż zapożycza z niego strukturę informacji o kluczu wykorzystanym do szyfrowania. W XML Encryption dodano przy tym — nie istniejący w XML Signature — mechanizm wymiany (szyfrowania) klucza. Koegzystencja obu standardów jest przykładem wielkiej przewagi koncepcji XML, w której można łączyć ze sobą różne języki.

Współdziałanie obu standardów widać także w trzecim standardzie zatwierdzonym pod koniec 2002 roku. Opisuje on sposób przekazania odbiorcy informacji o kolejności procesów szyfrowania i weryfikacji podpisu. Ma to znaczenie szczególnie wtedy, gdy podpisane XML Signature dane zostały jednocześnie zaszyfrowane XML Encryption. Dzięki temu odbiorca wie czy najpierw powinien zdeszyfrować dane a potem weryfikować podpis, czy odwrotnie. Proces może być jeszcze bardziej złożony wtedy, gdy zastosowane zostaną wielokrotne szyfrowania i podpisy, dokonane np. przez poszczególnych uczestników transakcji elektronicznej. Przyjrzyjmy się takiemu, przykładowemu procesowi. Sprzedawca szyfruje dla nabywcy ceny i podpisuje swoją ofertę (cały katalog). Nabywca, deszyfruje ceny, ogląda katalog i dołącza część zawierającą zamówienie: zaszyfrowane dla sprzedawcy dane o sobie i towarach oraz zaszyfrowaną dla banku informację o płatności (m.in. numer karty płatniczej); całość podpisuje i wysyła do e-banku. Bank weryfikuje podpis nabywcy, deszyfruje swoją część (płatności) i po zablokowaniu środków na koncie, dołącza potwierdzenie dokonania płatności, swój podpis i przekazuje sprzedawcy. Sprzedawca weryfikuje podpis banku i deszyfruje część przeznaczoną dla siebie (zamówienie). Bez standardu gwarantującego odpowiednią kolejność czynności przeprowadzenie takiego procesu z pomocą jednego, rozbudowywanego dokumentu byłoby bardzo kłopotliwe.

Standard ten może być przydatny także wtedy, gdy jeden dokument jest podpisywany przez kilka osób, np. wszystkich członków zarządu a każda kolejna osoba podpisuje dane wraz z podpisem poprzednika. Kłopotu w takiej sytuacji można oczywiście uniknąć stosując podpisywanie jedynie odpowiednich danych, z pominięciem podpisów poprzedników.

Problemy wykorzystania standardów bezpieczeństwa XML

Pierwszy problem jest związany z obsługą standardów. Oba standardy bardzo precyzyjnie definiują elementy obowiązkowe i opcjonalne. Jednak dostępne na rynku ich implementacje są bardzo różne i nie można mieć pewności, że wszystkie zastosowane u nadawcy mechanizmy będą rozumiane przez odbiorcę.

Kolejny problem wiąże się z kodowaniem znaków, w szczególności polskich liter. Otóż zalecanym standardem kodowania jest Unikod, jednakże dopuszcza się stosowanie innych standardów, np. Windows-1250, czy ISO Latin 2. Problem może wyniknąć np. przy szyfrowaniu lub podpisywaniu łącznie fragmentów wielu dokumentów XML, które są zakodowane w różnych standardach a fragmenty te nie zawierają w sobie deklaracji XML, czyli nie posiadają danych o sposobie kodowania. Przy deszyfracji odbiorca może błędnie odtworzyć lub zinterpretować uzyskane dane źródłowe (w złym standardzie).

Jeszcze inne zagadnienie jest związane z zaleceniami: „tylko to, co podpisane jest chronione", „tylko to, co widoczne powinno być podpisane" oraz „widzieć to, co zostało podpisane". Główny problem wynika ze stosowania do wizualizacji dokumentu XML transformacji XSLT. Można w nich, przy prezentacji, dowolnie przekształcić treść i strukturę dokumentu. Aby więc mieć pewność ochrony należy dokument podpisywać wraz ze związanymi z nim transformacjami XSLT lub ewentualnie odbiorca powinien stosować swoje, własne transformacje.

Kolejny problem to potencjalna możliwość ataków DoS (*Denial od Service*) na systemy odbiorcy, który deszyfruje zabezpieczone dokumenty XML. Polegają one na „zalaniu pracą" systemu deszyfrujacego, lub jego „zapętleniu", które jest skutkiem możliwej rekurencji. Zagrożenie to wynika z dużej elastyczności standardów XML, w tym także XML Encryption. Jeżeli do deszyfracji potrzebujemy zdeszyfrowanego klucza „A", który wymaga zdeszyfrowania klucza „B", to jeżeli klucz „B" korzysta z klucza „A" — mamy rekurencję. Inny scenariusz polega na umieszczeniu w podpisach lub szyfrach odesłań do zasobów sieciowych, które są bardzo duże (mocno obciążają zasoby odbiorcy) lub wielokrotnie odsyłają dalej, do kolejnych zasobów.

Ostatnie zagrożenie jest wspólne dla wszystkich metod kryptograficznych. Przekazywanie zaszyfrowanych dokumentów może skutkować przesyłaniem danych, które mogą stanowić zagrożenie. Typowe systemy zabezpieczeń — np. zapory ogniowe, systemy wykrywania włamań — nie są w stanie kontrolować zaszyfrowanych treści. Analogiczny problem dotyczy jednak także innych bezpiecznych systemów takich, jak: SSL, VPN, S/MIME czy PGP.

Podsumowanie

Wydaje się, że XML Signature oraz XML Encryption w dużej mierze zastąpią istniejące mechanizmy bezpieczeństwa stosowane powszechnie w wymianie danych. Zapewniają one integralność przesyłanych danych, potwierdzają autorstwo i poufność przekazywanych informacji. Jednakże najważniejsza ich cecha to elastyczność, dzięki której można niemal dowolnie chronić różne fragmenty danych (także binarnych), również dowolnie łącząc zabezpieczone dane z innymi danymi.
Pozwalają one także na długotrwałe przechowywanie zabezpieczonych dokumentów (czego nie dają zabezpieczenia komunikacji takie jak wirtualne sieci prywatne czy SSL) a nawet na dodatkową ich ochronę w przyszłości.
Przedstawione standardy bezpieczeństwa mają szczególne znaczenie w komunikacji wykorzystującej SOAP z załącznikami w postaci dokumentów XML (*SOAP Messages with Attachments*), co ma miejsce np. w usługach web services oraz komunikatach ebXML.
Pewną niedogodnością jest brak darmowych implementacji powyższych standardów, w szczególności typu open source. Biorąc jednak pod uwagę duże nimi zainteresowanie (nawet w Polsce istnieje już szereg implementacji komercyjnych), wydaje się że wersje darmowe wkrótce się pojawią.