

Kontenery z rodziny -AdES

Advanced Digital Electronic Signatures



Wstęp	3
Rozwiązania techniczne	3
Wnioski	4
Cechy formatów	4
PAdES	4
XAdES	4
CAAdES	5
Dodatkowe konkluzje	6

Wstęp

W dyskusji nt. standardu EZD pojawiał się pomysł stworzenia formatu danych umożliwiającego zawarcie danych dot. dokumentów ze spraw, odwzorowanie powiązań między nimi (w formie ograniczonego Merkle DAG) i zawarcie podpisów. Format mógłby być czysto binarny lub oparty np. o Bencode. Jednym z problemów do rozwiązania było zawarcie typu MIME w dokumentach (jako że bez niego nie wiadomo, jak zinterpretować ciąg bajtów). Istnieje jednak szereg standardów kryptograficznych, które pozwalają na rozwiązanie części (a być może wszystkich) związanych z tym problemów, dochowując przy tym zgodności z uregulowaniami unijnymi.

[Dyrektywa Parlamentu Europejskiego i Rady 1999/93/WE z dnia 13 grudnia 1999 r. w sprawie wspólnotowych ram w zakresie podpisów elektronicznych](#) definiuje podpis elektroniczny jako „dane w formie elektronicznej dodane do innych danych elektronicznych lub logicznie z nimi powiązane i służące jako metoda uwierzytelnienia”. Aby podpis elektroniczny został uznany za zaawansowany, musi on dodatkowo spełnić następujące wymagania:

1. przyporządkowany jest wyłącznie podpisującemu;
2. umożliwia ustalenie tożsamości podpisującego;
3. stworzony jest za pomocą środków, które podpisujący może mieć pod swoją wyłączną kontrolą; i
4. jest tak powiązany z danymi, do których się odnosi, że każda późniejsza zmiana danych jest wykrywalna;

Dyrektywa ta została zastąpiona przez rozporządzenie eIDAS.

Rozwiązania techniczne

Trzy standardy z rodziny “-AdES” (Advanced Digital Electronic Signatures) powstały w celu spełnienia prawnych wymogów określonych wyżej:

- [PAdES](#) — pozwala na zapakowanie podpisów do PDFa.
- [XAdES](#) — oparty o XML-Dsig.
- [CAAdES](#) — oparty o [Cryptographic Message Syntax](#).

Dwa ostatnie są dość podobne do siebie; w tej sytuacji najciekawszy wydaje się być CAAdES. Wybrane (niekoniecznie unikalne) cechy:

- Pozwala zastosować kodowanie DER, które jest nieciągłe.
- Zawiera atrybut content-reference, który stanowi “a link from one SignedData to another. It may be used to link a reply to the original message to which it refers, or to incorporate by reference one SignedData into another.” W naszym przypadku można by go użyć do odwołania się do wcześniejszego dokumentu w aktach.
- Zawiera atrybut “content-identifier”, który “provides an identifier for the signed content, for use when a reference may be later required to that content”.
- Pozwala na zawarcie znaczników czasu zarówno sygnatury, jak i podpisywanej zawartości (do zbadania, czy jest to kompatybilne z KSI).
- Pozwala na zawarcie nie tylko podpisu, ale i danych go uwierzytelniających (certyfikaty, odpowiedzi OCSP itp.).
- Obsługuje różne algorytmy podpisu.

- Oprócz pliku podpisywany jest też content type.
- Pozwala oprócz tożsamości podpisującego zawrzeć też jego rolę (przydatne zwłaszcza, gdy jedna osoba ma wiele ról w jednym urzędzie).
- [Ma inne zalety.](#)

W systemie EZD można sobie wyobrazić wykorzystanie tego formatu na przynajmniej dwa sposoby:

1. Powiedzieć, że wszystko, co wchodzi w skład metryki sprawy (vide [rozporządzenie Ministra Administracji i Cyfryzacji z dnia 6 marca 2012 r. w sprawie wzoru i sposobu prowadzenia metryki sprawy](#)), zostaje opakowane w taki kontener — przy czym część dokumentów będzie tworzył urzędnik, a część ewentualnie serwer, podpisując się własną pieczęcią. Do ewentualnego zbadania, czy urzędnik mógłby tworzyć dokumenty w formacie PAdES, na co zapewne pozwala mu Adobe Reader — czy te formaty są interoperatywne. Najlepiej pewnie, gdyby każdy dołączany element metryki odwoływał się do elementu poprzedniego (po funkcji skrótu). Migracja ze starych systemów powinna być dość prosta — serwer musiałby po prostu elementy metryk opakować w kontenery i poopatrywać swoim podpisem.
2. Powiedzieć, że w kontenery opakowywane są dokumenty mające znaczenie prawne i tworzone przez urzędników, a metryka sprawy jest zabezpieczana w inny sposób (choć można sobie wyobrazić ją w ten sposób, że serwer będzie we własnym zakresie tworzył łańcuch kontenerów, odwołujący się niekiedy do dokumentów wytworzonych przez urzędników).

Wnioski

Być może nie warto tworzyć własnych formatów danych na potrzeby standardu EZD, tylko oprzeć się na istniejących rozwiązaniach, które pozwalają spełnić szeroki zakres potrzeb. Trzy opisane wyżej formaty są same w sobie obszerne, a także odwołują się do innych standardów, nie wspominając o kwestii zgodności z przepisami prawa. Być może przy ich wyborze warto wspomóc się wiedzą dostępną na rynku, w firmach takich, jak [Ticons](#) (lub w COI).

Cechy formatów

PAdES

Format ogranicza podpisywane dokumenty do PDFów.

XAdES

XAdES ma analogiczne jak CAdES warianty: XAdES-T, XAdES-C, XAdES-A, ...

- linkowanie pomiędzy dokumentami: w XAdES odbywa się inaczej niż w CAsES. W kontenerze występuje sekcja references w której załączamy referencje do obiektów (URI) wraz z ich skrótami kryptograficznymi. Nie ma wyróżnionego jednego dokumentu, który jest tym głównym dokumentem pod którym się podpisujemy.

- referencja URI może odwoływać się do części pliku (przykładem niech będzie URI "<https://www.w3.org/TR/xmlsig-core/#sec-Reference>")
- referencja URI może się odwoływać do fragmentów dokumentu, którego część stanowi - w ten sposób realizowane jest pokrywanie podpisem metadanych
- jak spośród listy References odróżnić właściwy podpisany dokument od powiązań, które tylko dostarczają kontekstu do podpisywanego dokumentu? A może wstawianie tam powiązań jest nadużyciem i należy robić to w inny sposób?
- współpraca z KSI: podobnie jak w CAdESie. Powinno się udać, bo znacznik czasu nie ma dokładnie specyfikowanej postaci.
- content-hints: opis tego co jest w dokumentach do których się odnosimy, oraz ich content-type można zawrzeć w elemencie DataObjectFormat
- czy można dodawać dowolne metadane: nie znaleziono jasnego zezwolenia na dodawanie właściwości, ale też zakazu. Z punktu widzenia technicznych możliwości można takie metadane dodawać i kontener ciągle będzie czytelny.

CAdES

- kontener wymusza podpis: nie da się stworzyć dokumentu CAdES, który nie jest podpisany
- czy da się dodawać dowolne metadane: tak (RFC 5126 4.3.1 "Optional signed attributes may be added to the CAdES-BES, including optional signed attributes defined in CMS (RFC 3852 [4]), ESS (RFC 2634 [5]), and the present document.")
- jak linkować między dokumentami
 - RFC 2634 2.11 - linkujemy za pomocą contentIdentifier (który powinien (SHOULD) być nadawany zgodnie z RFC 2634 2.7 - ale nie musi)
 - czy można linkować z CAdESa do np. XAdESa - tak, można o ile dokument, który wskazujemy ma wygenerowany contentIdentifier
 - Jeżeli dokument chciałby odwołać się do innego dokumentu po sumie kontrolnej, to czy da się to pogodzić z atrybutem atrybutami content-reference i content-identifier? W jaki sposób?
 - Tak, da się pogodzić. Standard wskazuje sugerowany sposób obliczania content-identifier (coś w stylu identyfikator autora + losowy token), jednak można użyć jakiegoś innego algorytmu, którego wynikiem jest OCTET STRING. Ważne jest to, że ten wybór jest dokonywany przy obliczaniu content-identifier. Jako content-reference używamy content-identifier dokumentu, do którego się odwołujemy. Gdy dostaniemy przesyłkę z zewnątrz to prawdopodobnie musimy użyć content-identifera, który jest w niej już policzony. (Dlaczego "prawdopodobnie": content-identifier nie musi być objęty podpisem - w takim przypadku, oraz gdy nie ma żadnego zewnętrznego opakowania zapewniającego spójność, można go zmienić.)
- czy timestampowanie jest kompatybilne z KSI
 - wprost: nie
 - serwer usługi KSI musi wypełniać protokół TSP (RFC 3161, Time-Stamp Protocol)

- nie wszystkie punkty wymienione w rozdziale 2.1. Requirements of the TSA (Time Stamping Authority) (RFC 3161) są spełniane przez serwer KSI
- wygląda na to, że aby wypełniać protokół TSP nie trzeba być TSA
- w największym uproszczeniu protokół wygląda tak: wysyłamy serwerowi skrót dokumentu, on nam odsyła znacznik czasu
- content-hints: opisane w RFC 2634 2.9 - mało strukturalny opis (UTF8String) co znajduje się w środku wielokrotnie zapakowanych kontenerów
 - Pytanie: czy opakowanie w CAdES może skorzystać z atrybutu content-hints do zawarcia informacji o dokumencie PDF?
 - Tak, można. Atrybut content-hints zawiera w sobie wskazanie jakiego typu jest opakowana rzecz oraz tekstowe opisanie (UTF8String) tej rzeczy.

CAdES daje możliwość dodawania kolejnych opakowań w miarę wykonywania operacji na dokumencie (RFC 5126 4):

- podpisujący tworzy dokument CAdES-BES albo CAdES-EPES (treść, metadane, podpis)
- dokument jest znakowany czasem - powstaje dokument CAdES-T (... , znacznik czasu)
- do dokumentu są dodawane dane podpisującego (potrzebne do walidacji podpisu) (tutaj jest to trochę niezrozumiałe) - powstaje dokument CAdES-C
- do dokumentu dodawane są jeszcze inne dane i kolejny znacznik czasu - powstaje dokument w jednym z formatów
 - CAdES-X Long
 - CAdES-X Type 1
 - CAdES-X Type 2
 - CAdES-X Long Type 1 or 2
- dokument może być zarchiwizowany - powstaje dokument CAdES-A (... , znacznik czasu)

Dodatkowe konkluzje

- CAdES bez modyfikacji nie nadaje się do podpisywania encji bazodanowych (modyfikację tu rozumiemy w ten sposób, że różne części podpisanego dokumentu przechowywane są w różnych polach, np: dane, metadana 1, metadana 2, ..., podpis, znacznik czasu, ...)
 - wygląda na to, że kontenerów dES powinniśmy używać tylko do eksportu/importu/innej komunikacji
- CAdES specyfikuje szczegółowo jak przekształca się dokument poddawany różnym procesom - część z nich znajduje odzwierciedlenie w EZD np: znakowanie czasem, archiwizacja
- możliwość dodawania metadanych jest dużym plusem - np. można dodać pole specyfikujące więcej niż jednego rodzica dokumentu i budować DAGi
- natknęliśmy się na propozycję standardu Evidence Record Syntax (ERS) (RFC 4998)