STUDIA INFORMATICA 2002
Volume 23 Number 2B (49)

Jacek BAGIŃSKI, Piotr GAŁUSZKIEWICZ
SOTEL - Centrum Inżynierii Bezpieczeństwa Systemów Komputerowych
i Telekomunikacyjnych

OCHRONA DOKUMENTÓW ELEKTRONICZNYCH W INFRASTURKTURZE KLUCZA PUBLICZNEGO

Streszczenie. Artykuł zawiera krótki opis architektury PKI i interfejsu CryptoAPI oraz ich wykorzystanie do zabezpieczania dokumentów elektronicznych przy użyciu certyfikatów w standardzie X.509. Zamieszczono przykład procesu realizacji podpisu elektronicznego za pomocą funkcji CryptoAPI w programie SecOffice.

ELECTRONIC DOCUMENTS PROTECTION IN PUBLIC KEY INFRASTRUCTURE

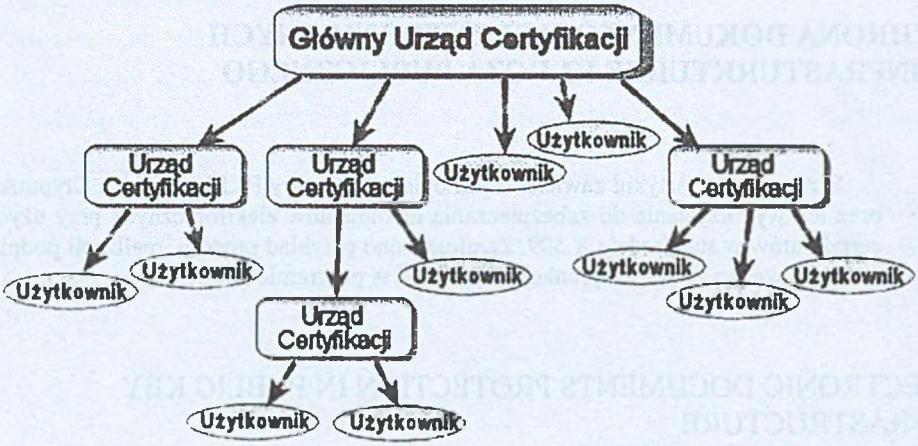
Summary. The article contains a short description of PKI architecture and CryptoAPI, and its usage for electronic documents protection with X.509 certificates. Example of process of digital signature by means of CryptoAPI functions in SecOffice application has been presented.

1. Infrastruktura Klucza Publicznego (PKI)

Podstawą zapewnienia bezpieczeństwa informacji w środowisku rozproszonym jest kryptografia asymetryczna (z kluczem jawnym zwanym inaczej publicznym). Wynika to z konieczności przekazywania kluczy w niezaufanym medium. Wiarygodność kluczy jawnych można zapewnić stosując tzw. certyfikaty cyfrowe. Są to struktury zawierające klucz jawny, których autentyczność jest potwierdzona przez podpis specjalnej jednostki certyfikującej. Jednostki te wraz z ich użytkownikami tworzą Infrastrukturę Klucza Publicznego (*ang. PKI – Public Key Infrastructure*). Jest to hierarchiczna struktura składająca się z urzędów certyfikacji (*ang. CA – Certification Authority*) oraz użytkowników lub aplikacji wykorzystujących certyfikaty (rys. 1).

Z urzędami certyfikacji związane są takie jednostki, jak urzędy rejestracji (*ang. RA – Registration Authority*), których zadaniem jest wymiana informacji pomiędzy użytkownikiem a urzędem certyfikacji (m.in. weryfikacja danych użytkownika zgłaszającego żądanie wystawienia certyfikatu) oraz repozytoria przechowujące klucze, certyfikaty i listy unieważnionych certyfikatów (*ang. CRL - Certificate Revocation List*).

Urzędy certyfikacji działają pod nadzorem Głównego Urzędu Certyfikacji (*ang. Root CA*). Stanowi on centralny punkt zaufania w strukturze.



Rys. 1. Struktura PKI
Fig. 1. PKI

Zadaniem CA jest zapewnienie wiarygodności użytkowników końcowych oraz urzędów certyfikacji niższego poziomu.

Struktura PKI zapewnia realizację takich głównych grup usług, jak uwierzytelnianie poprzez zastosowanie algorytmów podpisów elektronicznych oraz zapewnienie poufności poprzez zastosowanie algorytmów wymiany kluczy szyfrujących.

Certyfikaty pozwalają na zapewnienie *poufności* i *integralności* danych, np. przesyłki pocztowej, danych do transakcji czy dokumentów elektronicznych oraz *autentyczności* i *niezaprzeczalności* ich autorów, nadawców lub odbiorców.

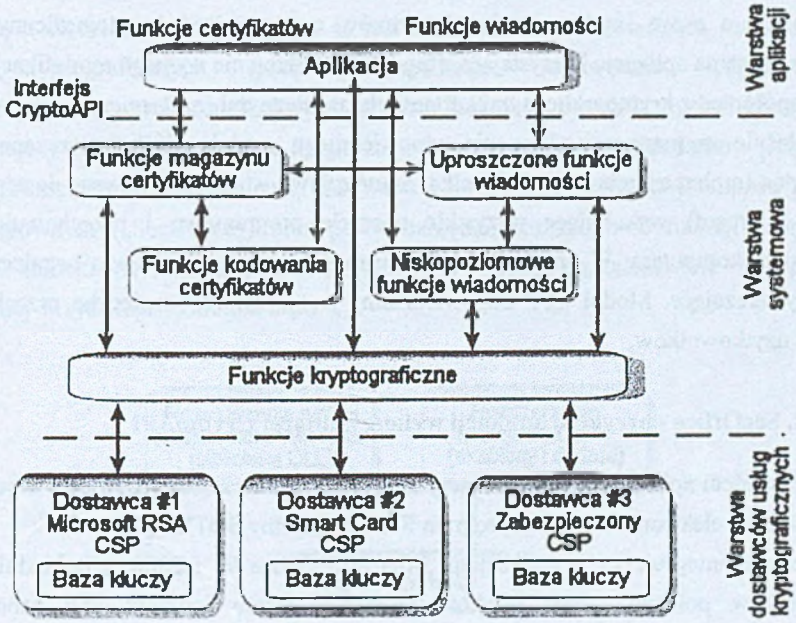
Można to osiągnąć np. przy wykorzystaniu interfejsu CryptoAPI. Jest on szeroko stosowany przez aplikacje i producentów urządzeń czy oprogramowania kryptograficznego dla systemu Windows.

2. Wykorzystanie certyfikatów w aplikacjach

2.1. CryptoAPI

Jest to interfejs wbudowany w każdy 32-bitowy system rodziny Windows. Udostępnia on programiście zestaw funkcji, dzięki którym możliwa jest implementacja wybranych operacji kryptograficznych ([1, 2]).

Dzięki dodaniu do CryptoAPI zestawu funkcji operujących na standardowych certyfikatach X.509 struktura ta łatwo integruje się z infrastrukturą klucza publicznego. Za pomocą funkcji CryptoAPI możliwe jest generowanie żądań, wystawianie i zarządzanie certyfikatami i listami certyfikatów odwołanych (CRL). Certyfikaty zgrupowane są w tzw. magazynach certyfikatów. Każdy magazyn jest zbiorem certyfikatów przeznaczonych do określonego celu albo pochodzących z określonego źródła, np. z katalogu Active Directory albo osobistej książki adresowej.



Rys. 2. Architektura CryptoAPI
Fig. 2. CryptoAPI architecture

Trójwarstwową strukturę CryptoAPI przedstawia rysunek 3 [1]. Możemy tu wyróżnić:

- warstwę aplikacji (*ang. application layer*),
- warstwę systemową (*ang. system layer*),
- warstwę dostawców usług kryptograficznych (*ang. cryptographic service provider layer*).

W najwyższej warstwie modelu działa aplikacja korzystająca z funkcji kryptograficznych. Stykiem aplikacji z systemem operacyjnym jest druga warstwa (systemowa), czyli zestaw funkcji CryptoAPI udostępniających wszelkie potrzebne operacje kryptograficzne oraz operacje na certyfikatach. Dzielą się one na kilka grup, takich jak: podstawowe funkcje kryptograficzne, funkcje zarządzania certyfikatami czy funkcje do kodowania i dekodowania wiadomości. Najniżej położoną warstwą jest samo jądro CryptoAPI, czyli tzw. dostawcy usług kryptograficznych (*ang. CSP – Cryptographic Service Provider*). Moduły te wykonują właściwe operacje szyfrowania, podpisywania i generacji kluczy. Stanowią one pewną formę systemowych sterowników implementujących funkcje kryptograficzne.

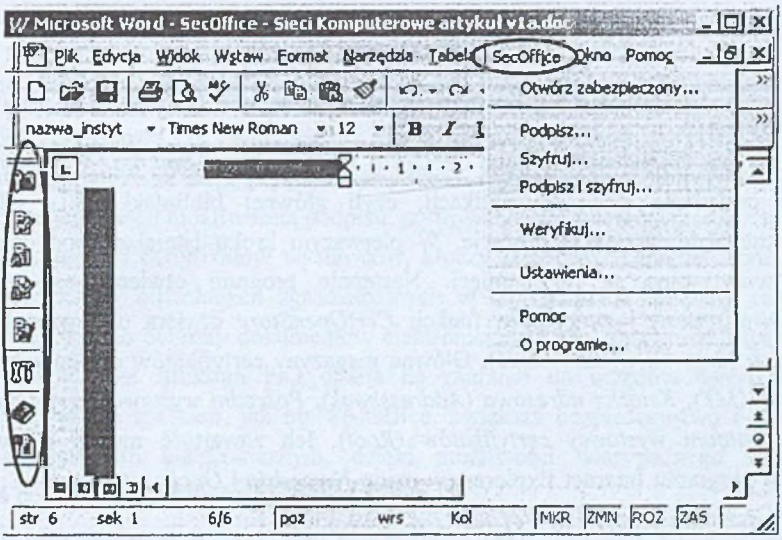
Zaletą tej architektury jest możliwość wymiany i wyboru rożnych dostawców, dzięki czemu uzyskuje się niezależność od konkretnej realizacji i możliwość rozszerzania CryptoAPI o nowe implementacje algorytmów czy urządzeń kryptograficznych w taki sposób, że sama aplikacja korzystająca z tego mechanizmu nie wymaga modyfikacji.

Współpraca z kryptograficznymi kartami elektronicznymi (*ang. smart cards*) realizowana jest właśnie za pomocą podłączenia odpowiedniego modułu CSP dostarczanego wraz z konkretną implementacją karty i czytnika. Standardowo w systemie zawsze dostępne są CSP firmy Microsoft wykonujące wszystkie operacje programowo i przechowujące klucze w pamięci komputera. W przypadkach nie wymagających szczególnego bezpieczeństwa są one wystarczające. Moduł CSP odpowiedzialny jest także za bezpieczne przechowywanie kluczy użytkowników.

2.2. SecOffice - przykład aplikacji wykorzystującej CryptoAPI

Przykładem aplikacji wykorzystującej certyfikaty w standardzie X.509 do zabezpieczania dokumentów elektronicznych jest program SecOffice firmy SOTEL [3].

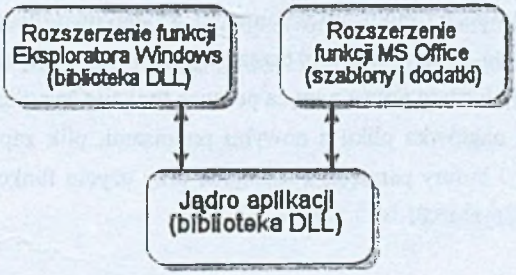
Program integruje się z aplikacjami Microsoft Office 97 i 2000, a dokładniej z Word i Excel (w polskiej wersji językowej), dzięki czemu możliwe jest zabezpieczenie edytowanego dokumentu (podpisanie bądź zaszyfrowanie) z poziomu tych programów (rys. 3).



Rys. 3. Menu i przyciski SecOffice zintegrowane z MS Word
Fig. 3. Menu and toolbar of SecOffice integrated with MS Word

Głównym elementem aplikacji (rys. 4) jest biblioteka DLL zawierająca implementację operacji kryptograficznych (podpis, weryfikacja, szyfrowanie, odszyfrowanie) zrealizowanych przy użyciu standardowego zestawu funkcji kryptograficznych CryptoAPI.

Integrację z MS Office uzyskano dzięki zastosowaniu szablonów i dodatków (*ang. Add-ins*), które rozszerzają funkcjonalność MS Word i Excel, tzn. tworzona jest nowa pozycja w menu głównym („SecOffice") oraz pasek narzędzi. Rozszerzenie funkcji dostępnych w menu kontekstowym (Eksplorator Windows) zostało zrealizowane za pomocą rejestrowanej w systemie biblioteki DLL.



Rys. 4. SecOffice - struktura aplikacji
Fig. 4. SecOffice - application structure

2.3. Podpis elektroniczny - realizacja z wykorzystaniem CryptoAPI

Na podstawie programu SecOffice poniżej zostanie opisana operacja podpisu przy użyciu funkcji CryptoAPI.

Załóżmy, że użytkownik z menu *SecOffice* wybiera opcję *Podpisz*. W tym momencie następuje odwołanie do jądra aplikacji, czyli głównej biblioteki DLL, gdzie dalej wykonywane będą wszystkie operacje. W pierwszym kroku istniejące pod dokumentem podpisy wczytywane są do pamięci. Następnie program otwiera sesję CryptoAPI (*CryptAquireContext*) i za pomocą funkcji *CertOpenStore* otwiera odpowiedni magazyn certyfikatów (*ang. Certificate Store*). Główne magazyny certyfikatów dostępne w systemie to *Osobisty* (*MY*), *Książka adresowa* (*Addressbook*), *Pośredni wystawcy certyfikatów* (*CA*) i *Główni zaufani wystawcy certyfikatów* (*Root*). Ich zawartość można sprawdzić np. za pomocą programu Internet Explorer (w menu *Narzędzia* | *Opcje internetowe...*, zakładka *Zawartość*, przycisk *Certyfikaty*).

Następny etap podpisywania dokumentu to wybór w okienku dialogowym programu certyfikatu użytkownika z kluczem prywatnym, za pomocą którego wykonany będzie podpis. Klucz ten wyszukiwany jest w magazynie (funkcja *CertFindCertificateInStore*).

Wśród istniejących już pod dokumentem podpisów wyszukiwany jest ten (o ile istnieje), który był dodany wcześniej za pomocą tego samego certyfikatu. Jeżeli taki podpis istnieje, będzie on zastąpiony, jeśli nie - dodany zostanie nowy podpis.

Wybrany przez użytkownika certyfikat jest sprawdzany pod kątem ważności, ewentualnego odwołania (sprawdzenie CRL wystawcy certyfikatu) oraz weryfikowany jest certyfikatem wystawcy (funkcja *CertGetIssuerCertificateFromStore*).

Następnie za pomocą funkcji *CryptAcquireCertificatePrivateKey* program „podłącza się" i pobiera uchwyt do klucza prywatnego tego certyfikatu.

Za pomocą algorytmu SHA-1 tworzony jest cyfrowy skrót (funkcja *CryptHashData*) z zawartości podpisywanego pliku lub dokumentu. Dodatkowo uwzględniane są inne dane, jak np. czas czy komentarze dodane przez użytkownika. Skrót ten podpisywany jest wybranym wcześniej kluczem prywatnym za pomocą funkcji *CryptSignHash*.

Po sformowaniu nagłówka pliku z nowymi podpisami, plik zapisywany jest na dysku, zwalniane są zasoby i bufory pamięci, a następnie przy użyciu funkcji *CryptReleaseContext* zamykana jest sesja CryptoAPI.

3. Podsumowanie

Opisana w rozdziale 2.3 procedura nie w pełni prezentuje możliwości wykorzystania certyfikatów cyfrowych w aplikacjach kryptograficznych przy użyciu CryptoAPI. Użytkownik czy aplikacja korzystająca z certyfikatu, czyli końcowe ogniwo infrastruktury klucza publicznego, poza możliwością podpisu, szyfrowania czy weryfikacji, musi mieć także możliwość pobierania certyfikatów wystawców, kluczy publicznych innych użytkowników czy list certyfikatów odwołanych zgromadzonych w repozytoriach urzędów certyfikacji. Zadaniem aplikacji do ochrony dokumentów elektronicznych jest zapewnienie tych właśnie funkcji. Hierarchiczna struktura PKI oparta na zaufaniu do urzędów certyfikacji oraz wykorzystanie takich aplikacji, jak np. SecOffice, zwiększa bezpieczeństwo i pozwala na zaufanie dokumentom elektronicznym, dzięki możliwości wiarygodnego sprawdzenia właściciela certyfikatu oraz weryfikacji jego klucza.

Najsłabszym ogniwem w tej strukturze jest, jak zwykle, człowiek - użytkownik i jego klucz prywatny, którego ochrona leży w jego gestii.

LITERATURA

1. Microsoft Platform SDK, „CryptoAPI System Architecture".
2. Microsoft Platform SDK, „CSP Architectural Overview".
3. http://www.sotel.com.pl
4. http://www.polcert.pl
5. http://www.signet.pl
6. http://www.sotel.com.pl/karty/SecOffice_Instrukcja.zip - SecOffice. Program do szyfrowania i podpisywania dokumentów elektronicznych. Instrukcja użytkownika.
7. Białas A.: Bezpieczeństwo sieci komputerowych. Podręcznik Wyższej Szkoły Informatyki i Zarządzania w Bielsku-Białej. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2001.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 4 kwietnia 2002 r.

Abstract

The fundamental method to assure information security in distributed environment is public key cryptography. This article presents one of way of electronic documents protection by using CryptoAPI functions with digital certificates.

The first part of article shortly describes public key infrastructure and its elements (fig. 1). Next chapter presents CryptoAPI architecture (fig. 2) which includes functionality for hashing, encrypting and decrypting data, for authentication using digital certificates, and for managing certificates in certificate stores. Chapter 2.2 shortly presents SecOffice application. Its integration with Microsoft Office 97 and Microsoft Office 2000 (fig. 3) and Windows Explorer allow in simple way to sign or encrypt electronic documents. Figure 4 shows SecOffice architecture. On the basis of SecOffice, example process of digital signing of document with CryptoAPI functions use was shown.