



# Certificate Authority CBZ - Dokumentacja kodu

Piotr Pazdan, Maksymilian Oliwa

Styczeń 2026

*II rok Cyberbezpieczeństwo, WIEiT*

## I Wprowadzenie

Niniejszy dokument stanowi udokumentowanie bazy kodu wchodzącego w skład głównego komponentu projektu, czyli programu **ca-cbz**. Program ten umożliwia generowanie żądań podpisu certyfikatu (ang. *certificate signing request, CSR*) oraz samych certyfikatów, zarówno tych podpisanych własnoręcznie (ang. *self-signed certificate*), jak i podpisanych za pomocą odrębnego certyfikatu urzędu certyfikacji.

Ze względu na obszerność standardów definiujących infrastrukturę klucza publicznego, program nie wspiera go całego. Limit czasowy projektu zmusił nas do selektywnego podejmowania decyzji na temat wspierania konkretnych schematów, szyfrów, rozszerzeń itd. Niemniej jednak zbiór obecnie zaimplementowanych mechanizmów wystarcza w pełni na efektywne użytkowanie i korzystanie z możliwości PKI.

## II Infrastruktura klucza publicznego

Ważną kwestią jest istotność samego konceptu programu. Aby zrozumieć jego przeznaczenie, należy być zaznajomionych z technicznym opisem mechanizmów PKI, głównie wcześniej wspomnianych certyfikatów. Zarysujmy zatem pewien zestaw kroków które musimy spełnić, aby móc korzystać z bezpieczeństwa które oferuje nam standard - za przykład weźmy obsługę protokołu HTTPS przez nasz serwer sieciowy.

Protokół HTTPS służy weryfikacji (uwierzytelnieniu) serwera komunikującego się z klientem. Aby takie uwierzytelnienie było możliwe, należy stworzyć pewien podmiot autoryzujący zaufane domeny odpowiednim certyfikatem. Rolę takiego podmiotu pełni urząd certyfikacji (ang. *certificate authority*), który w modelu łańcucha zaufania (ang. *chain of trust*) znajduje się na samej jego górze, tzn. jest zaufany z założenia.

**Uwaga:** w prawdziwym świecie w omawianym modelu mamy także do czynienia z tzw. pośrednimi urządzeniami certyfikacji (ang. *intermediate certificate authority*), które nie są zaufane z założenia - poświadczą za nimi urząd certyfikacji wyższego szczebla, a zaufane z założenia są główne urzędy certyfikacji (ang. *root certificate authority*) stojące najwyżej w hierarchii. W tym wprowadzeniu świadomie posłużymy się uproszczonym modelem zaufania, w którym urząd certyfikacji jest *de facto* głównym urządzeniem certyfikacji.

Urząd certyfikacji wystawia certyfikaty domenom, które:

1. dostarczą żądanie podpisu certyfikatu do urzędu
2. udowodnią własność domeny widniejącej w żądaniu (odbywa się przeważnie poprzez wysłanie wiadomości z linkiem autoryzującym na parę adresów mailowych powiązanych z daną domeną)

Po pomyślnym przejściu procesu weryfikacji, certyfikat jest wystawiany na określony czas (z reguły parę miesięcy/lat).

Program **ca-cbz** zawiera funkcjonalność wspierającą obie strony w tym przedsięwzięciu. Posiada on funkcjonalność potrzebną właścicielowi domeny ubiegającej się o certyfikat (generowanie żądań podpisu certyfikatu, *CSR*), jak i potrzebną urzędowi w celu wystawiania certyfikatów (generowanie certyfikatów podpisanych własnoręcznie, jak i zwykłych certyfikatów).

**Uwaga:** Program **nie wspiera** generowania kluczy kryptograficznych - w tym celu zalecamy użycie gotowego i szeroko stosowanego oprogramowania *OpenSSL*.

### III Implementacja prymitywów PKI

Dokumenty definiujące infrastrukturę klucza publicznego (RFC 5280, RFC 7468) zakładają, że pliki reprezentujące prymitywy tej infrastruktury (klucze kryptograficzne, *CSR*, certyfikaty) będą reprezentowane w formacie PEM, który jest powiązany z formatem DER, który to z kolei wykorzystuje format ASN.1. Poniżej krótki opis tych formatów:

- **ASN.1** - standard służący do opisu danych przeznaczonych do reprezentacji, kodowania lub dekodowania
- **DER** - jeden z wariantów kodowania ASN.1, tzn. konwersji abstrakcyjnych obiektów na ciąg bitowy (i *vice versa*)
- **PEM** - format ułatwiający transmisję danych zakodowanych w DER - przewiduje on zakodowanie danych DER poprzez Base64 oraz opatrzenie nagłówka oraz stopki pliku odpowiednimi etykietami (ang. *labels*), np. -----BEGIN CERTIFICATE-----, -----END CERTIFICATE-----

### IV Odgórny wzgląd na kod

Baza kodu programu `ca-cbz` składa się z zawartości katalogu `src` oraz pliku `Makefile`, służącego do kompilacji.

Wewnątrz katalogu `src` znajdziemy poniższą zawartość:

- **asn1** - katalog zawierający kod implementujący kodowanie prymitywów zdefiniowanych w standardzie ASN.1
- **encryption** - katalog definiujący prymitywy kryptograficzne, takie jak:
  - (a) szyfr *AES-CBC* w odmiennych wariantach długości klucza
  - (b) funkcja *HMAC* obliczająca kod uwierzytelnienia wiadomości
  - (c) funkcja *PBKDF2* - szeroko stosowana odmiana funkcji z rodziny KDF (ang. *key derivation function*) służąca do wyznaczania klucza odpowiedniej długości na podstawie hasła
- **hash** - katalog z zawartymi algorytmami wyznaczania skrótów wiadomości; znajdują się tu różne warianty algorytmów z rodziny *SHA*
- **pkcs** - najbardziej obszerny katalog; zawiera kod odpowiedzialny za implementację wszelkich mechanizmów powiązanych z *PKCS* (ang. *public-key cryptography standards*); znajduje się tu kod odpowiedzialny za zarządzanie kluczami kryptograficznymi, kodowanie *obiektów PKCS* zdefiniowanych w RFC 5280 oraz funkcje generujące i weryfikujące podpis cyfrowy.
- **tests** - katalog zawierający tzw. testy jednostkowe (ang. *unit tests*) weryfikujące sprawność różnych fragmentów kodu
- **utils** - katalog z funkcjonalnością niesprecyzowanego przeznaczenia; głównie funkcje wspomagające sterowaniem wejścia-wyjścia, implementacja kodowania Base64 oraz funkcje związane z czyszczeniem buforów pamięciowych
- **main.cpp** - serce całego programu; znajduje się tu funkcja wejściowa `main` wraz z funkcjami pomocniczymi