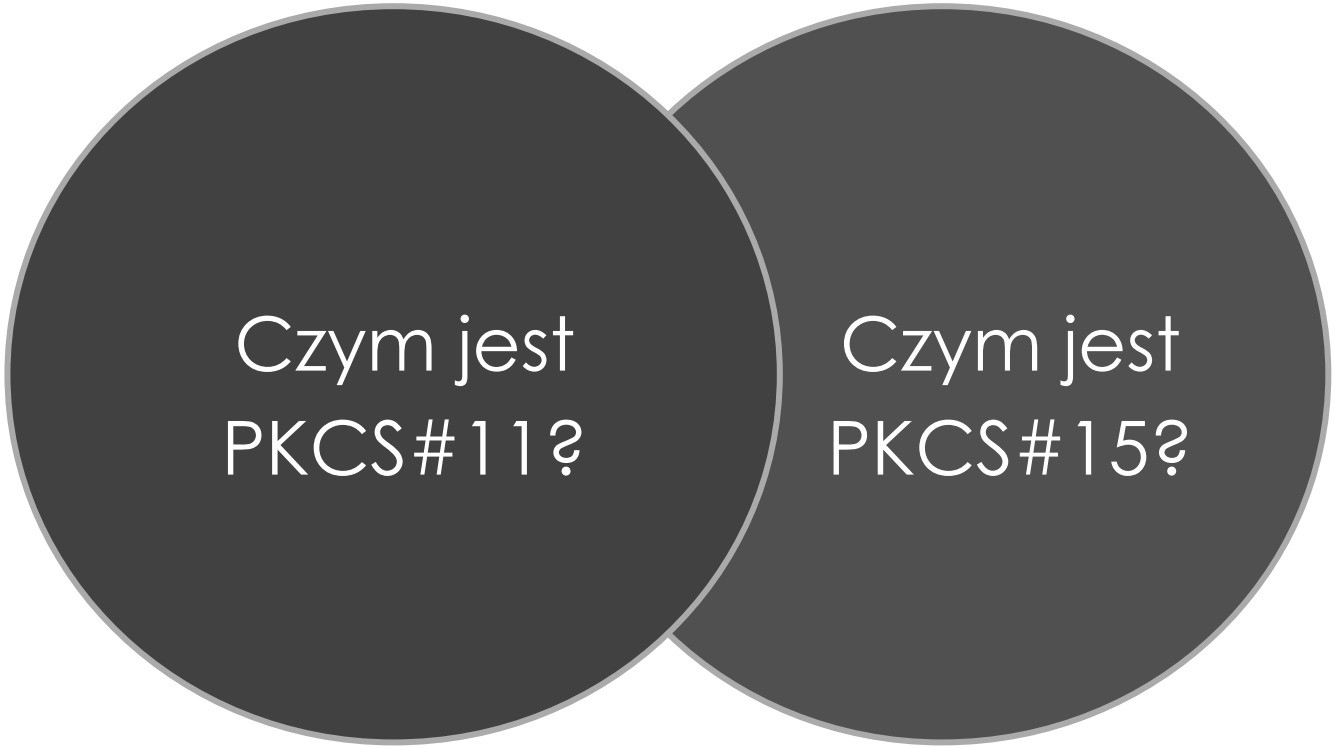
OpenSC

Zestaw narzędzi programowych i bibliotek do pracy z kartami inteligentnymi posiadającymi funkcje kryptograficzne

Projekt

Sandra Bocian
131907
Internet Przedmiotów
Informatyka, semestr II

Podstawowe pytania



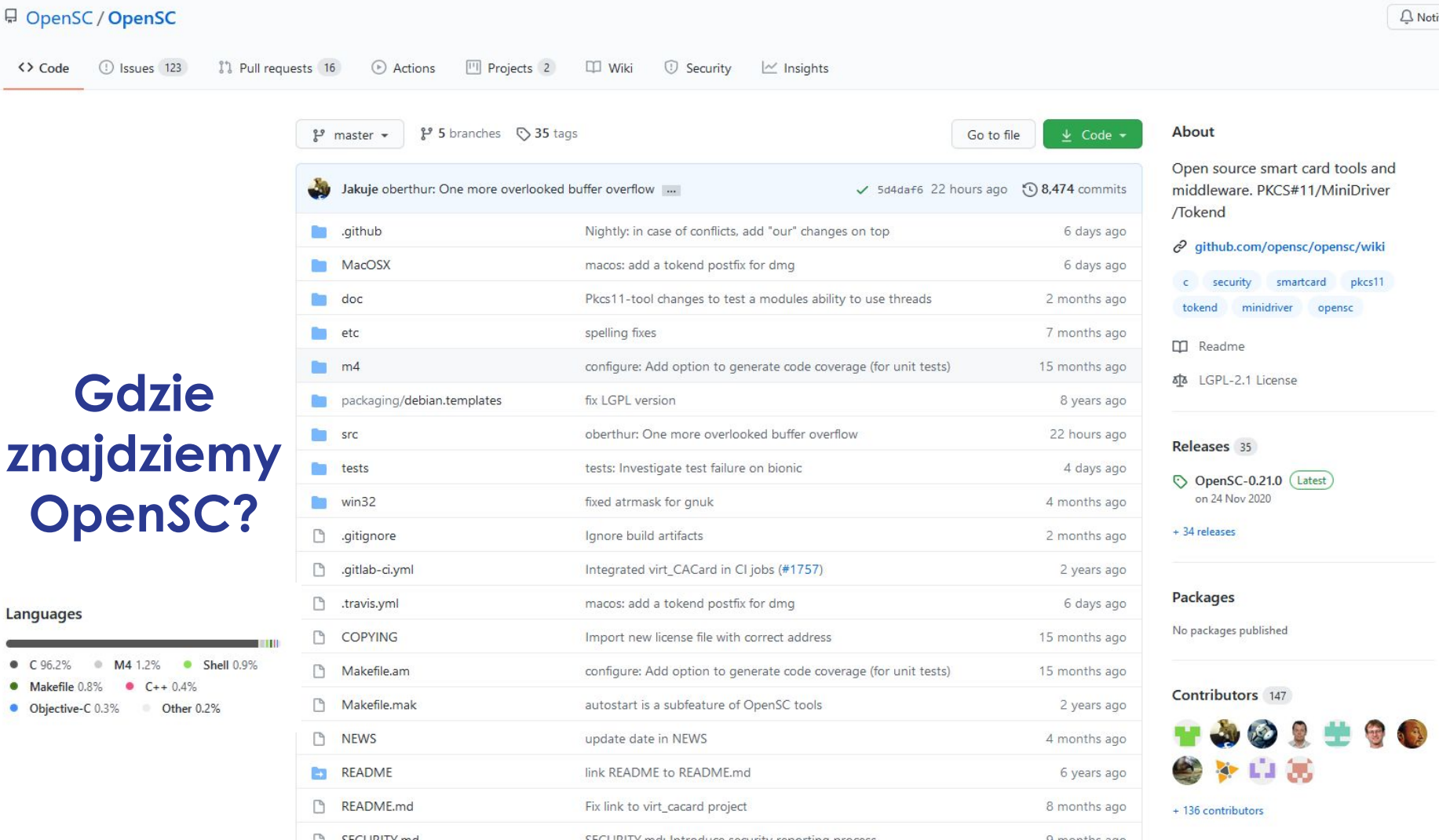
Słownik cd.

- **PC/SC** - standard dostępu do karty inteligentnej,
- **OpenCT** - własne oprogramowanie pośrednie napisane przez programistów,
- **CT-API i CT-BCS** (przestarzałe) - implementacja prostego interfejsu API, wysyłanie poleceń.

Czym jest OpenSC?

- Zestaw bibliotek i narzędzi do pracy z kartami inteligentnymi.
- Głównie dla kart obsługujących operacje kryptograficzne.
- Implementuje standardowe API do kart inteligentnych.

Gdzie znajdziemy OpenSC?



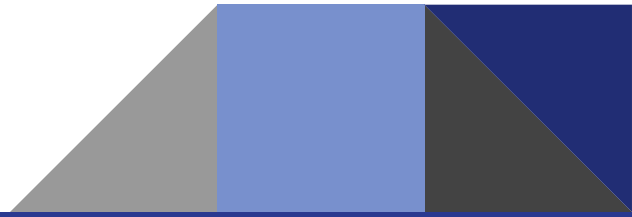
Nowe wersje

- Wydania mniej więcej raz w roku.
- Nie są wydawane mikroaktualizacje.

Version	Supported
0.20.0	✅
< 0.20.0	❌

Ogólne ulepszenia - OpenSC-0.21.0

- **eDo** - nowy sterownik z początkową obsługą polskiej karty eID (e-dowód, eDO).
- **pkcs11-tool** - ustawia SHA256 jako domyślny dla szyfrowania OAEP
- **CardOS** - poprawia wykrywanie różnych konfiguracji CardOS 5.
- **TCOS** - dodaje brakujące certyfikaty szyfrowania.

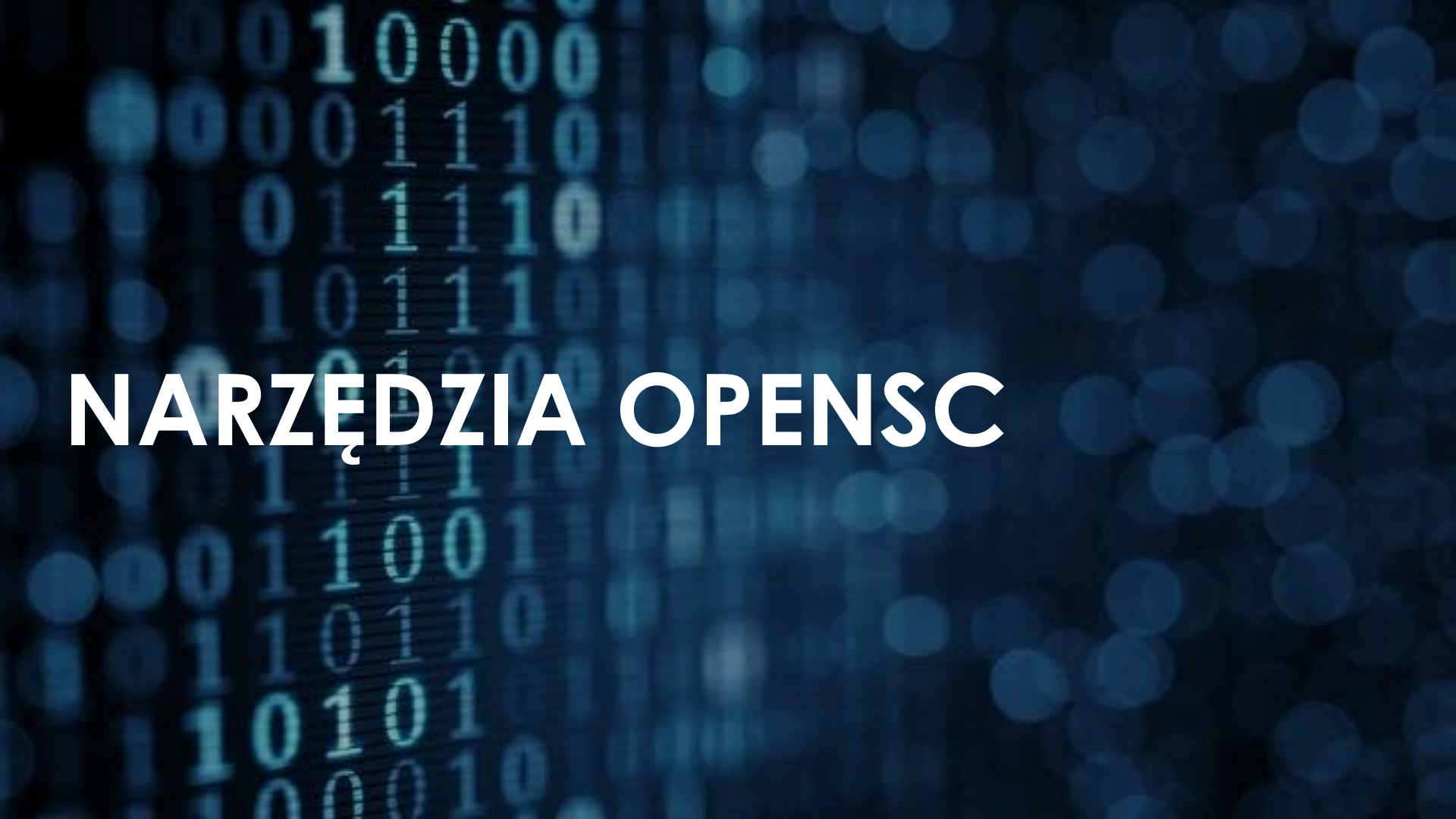


Proces rozwoju OpenSC

- Rozwój koordynuje lista opensc-devel.
- Można pomóc przeglądając problemy na Github i naprawiając je.
- Poprawki można przesyłać za pomocą prośby Github lub na listę e-mai.
- Należy testować nowe funkcje np. wykorzystując testy jednostkowe.

Obsługa kart inteligentnych (OpenSC)

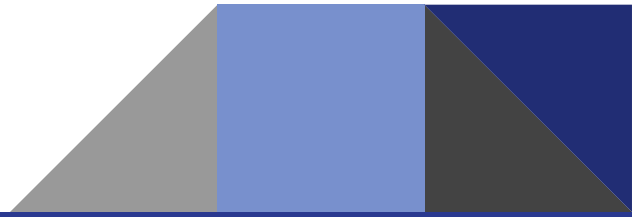
- W przypadku pustych kart OpenSC ma kod do inicjalizacji karty w formacie PKCS#15.
- Zgodnie z ogólną zasadą OpenSC obsługuje tylko karty z systemem plików i funkcje kryptograficzne (RSA).



NARZĘDZIA OPENSC

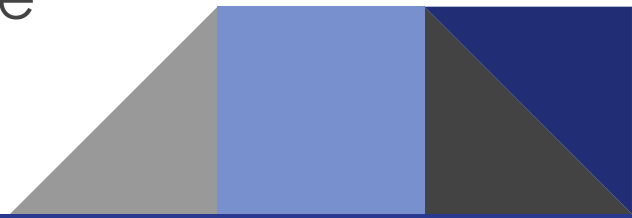
Narzędzia niskiego poziomu

- **OpenSC-tool** - podstawowe narzędzie testowe. Niezbędne do funkcjonowania OpenSC.
- **OpenSC-explorer** - narzędzie do przeglądania inteligentnej karty. Zamienne z narzędziem pkcs15-tool.



Narzędzia wysokiego poziomu

- **Pkcs15-init** - narzędzie do edycji karty inteligentnej i generowania kluczy.
- **Pkcs15-tool** - narzędzie do przeglądania karty inteligentnej.
- **Pkcs15-crypt** - narzędzie do obsługi funkcji kryptograficznych.
- **Pkcs11-tool** - narzędzie wykorzystujące moduł OpenSC PKCS#11.



Narzędzia specyficzne dla karty

- **Cardos-info** - pokazuje informacje o kartach Siemens CardOS/M4.
- **Cryptoflex-tool** - narzędzie do działania z kartami cryptoflex.
- **PivTool** - narzędzie do obsługi funkcji kryptograficznych.
- **Pkcs11-tool** - administrowanie kartami PIV.

1136 lines (1047 sloc) | 44.2 KB

```
# Configuration file for OpenSC
# Example configuration file

# NOTE: All key-value pairs must be terminated by a semicolon.

# Default values for any application
# These can be overridden by an application
# specific configuration block.
app default {
        # Amount of debug info to print
        #
        # A greater value means more debug info.
        # Default: 0
        #
        #debug = 3;

        # The file to which debug output will be written
        #
        # Special values 'stdout' and 'stderr' are recognized.
        # Default: stderr
        #
        # debug_file = @DEBUG_FILE@

        # PKCS#15 initialization / personalization
        # profiles directory for pkcs15-init.
        # Default: @PROFILE_DIR_DEFAULT@
        #
        # profile_dir = @PROFILE_DIR@;
```

opensc.conf

Przykładowe opcje konfiguracji

- debug = num;
- debug_file = nazwa_pliku;
- profile_dir = nazwa pliku;
- enable_default_driver = bool;
- card_drivers = name ...;

```
# debug_file = @DEBUG_FILE@

# PKCS#15 initialization / personalization
# profiles directory for pkcs15-init.
# Default: @PROFILE_DIR_DEFAULT@
#
# profile_dir = @PROFILE_DIR@;

# Disable pop-ups of built-in GUI
#
# Default: false
# disable_popups = true;

# Enable default card driver
# Default card driver is explicitly enabled for the 'opensc-explorer' and 'opensc-tool'.
#
# Default: false
# enable_default_driver = true;

# List of readers to ignore
# If any of the strings listed below is matched in a reader name (case
# sensitive, partial matching possible), the reader is ignored by OpenSC.
# Use `opensc-tool --list-readers` to see all currently connected readers.
#
# Default: empty
# ignored_readers = "CardMan 1021", "SPR 532";
```

Kompilacja

- Pobierz kod źródłowy
  git clone https://github.com/OpenSC/OpenSC.git OpenSC-code

- Typowa instalacja LINUX (zainstalowanie OpenSC w /usr i umieszczenie pliku konfiguracyjnego w /etc/opensc):

```
tar xfvz opensc-a.b.c.tar.gz
cd opensc-a.b.c
./bootstrap
./configure --prefix=/usr --sysconfdir=/etc/opensc
make
sudo make install
```

Kompilacja Windows

- Zautomatyzowane kompilacje OpenSC z wiki OpenSC (wymagane posiadanie Mingw i automake.
- Działanie z OpenSc bez środowiska MingW.
  - Utwórz plik .rc

```
copy win32\versioninfo.rc.in win32\versioninfo.rc
copy win32\versioninfo-customactions.rc.in win32\versioninfo-customactions.rc
copy src\minidriver\versioninfo-minidriver.rc.in src\minidriver\versioninfo-minidriver.rc
copy src\pkcs11\versioninfo-pkcs11.rc.in src\pkcs11\versioninfo-pkcs11.rc
copy src\pkcs11\versioninfo-pkcs11-spy.rc.in src\pkcs11\versioninfo-pkcs11-spy.rc
copy src\tools\versioninfo-tools.rc.in src\tools\versioninfo-tools.rc
```

  - Ustaw wartości pól w każdym pliku .rc

```
@OPENSC_VERSION_MAJOR@        => 1
@OPENSC_VERSION_MINOR@        => 0
@OPENSC_VERSION_FIX@          => 0
@OPENSC_VERSION_REVISION@     => 0
@OPENSC_VS_FF_COMMENTS@       => ""
@OPENSC_VS_FF_COMPANY_NAME@   => "Your company name"
@PACKAGE_NAME@                => ""
@OPENSC_VS_FF_LEGAL_COPYRIGHT@ => "copyright string"
@OPENSC_VS_FF_PRODUCT_NAME@   => "product name"
```

Kompilacja Windows

- Utwórz plik .res z pliku .rc

```
rc.exe win32\versioninfo.rc
rc.exe win32\versioninfo-customactions.rc
rc.exe src\minidriver\versioninfo-minidriver.rc
rc.exe src\pkcs11\versioninfo-pkcs11.rc
rc.exe src\pkcs11\versioninfo-pkcs11-spy.rc
rc.exe src\tools\versioninfo-tools.rc
```

- Utwórz plik config.h

```
copy win32\winconfig.h.in win32\winconfig.h
```

- Zbuduj za pomocą nmake

```
nmake /f Makefile.mak
```

Bezpieczne przesyłanie wiadomości w OpenSC

iso7816.c

```
static struct sc_card_operations ops2 = {
        ....
        iso7816_set_security_env,
        is07816_restore_security_env,
        iso7816_give_challenge,
        iso7816_get_challenge (),
        ....
};
```

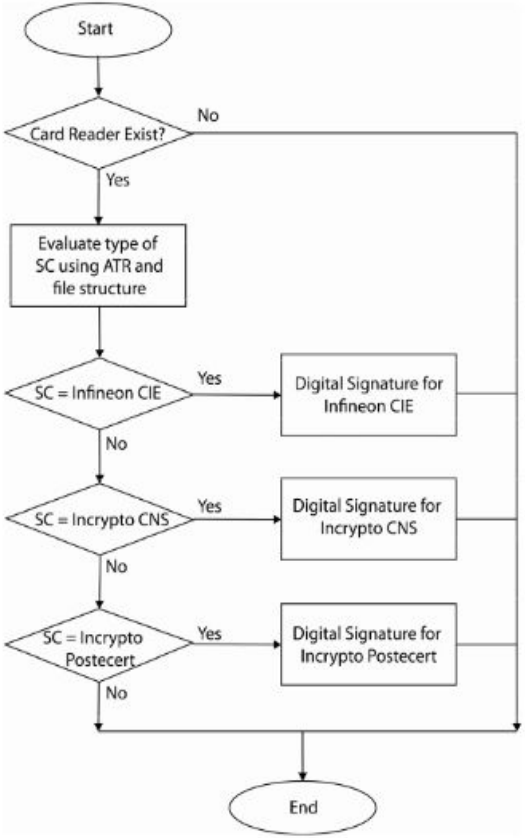
Struktura operacji SC

opensc.h

```
...

typedef struct sc_sm_context {
        int use_sm;
} sc_sm_context_t;

...

typedef struct sc_card {
        ...
        ...
        struct sc_card_operations *ops;
        struct sc_sm_context sm_ctx;
        ...
} sc_card_t;

...
```

Bezpieczne przesyłanie wiadomości do OpenSC

Ocena typu karty inteligentnej w OpenSC - problemy



Przykłady użycia OpenSC

- Generowanie kluczy:
  - RSA,
  - ECC,
  - AES.
- Dekodowanie
- Tworzenie inteligentnej karty z podpisem własnym.

.

Bibliografia

- https://github.com/OpenSC/OpenSC/wiki
- https://www.scirp.org/journal/paperinformation.aspx?paperid=23945
- https://developers.yubico.com/YubiHSM2/Usage_Guides/Using_OpenSC_pkcs11-tool.html
- https://securehomes.esat.kuleuven.be/~decockd/wiki/bin/view.cgi/Using/OpenSC

Dziękuję za uwagę!