Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych
Instytut Informatyki

Rok akademicki 2011/2012

Praca dyplomowa inżynierska

Cezary Guz

TEST ĄąŻżŹźĆćŚśÓŁłĘę

Opiekun pracy:
Tytuł Imię i Nazwisko

Ocena .....

.....
Podpis Przewodniczącego
Komisji Egzaminu Dyplomowego

Specjalność: Informatyka –
Inżynieria oprogramowania
i systemy informacyjne

Data urodzenia: 1 stycznia 1980 r.

Data rozpoczęcia studiów: 1 października 2002 r.

Życiorys

Nazywam się . . . .

. . . . . .
podpis studenta

Egzamin dyplomowy

Złożył egzamin dyplomowy w dn. . . . . .

Z wynikiem . . . . . .

Ogólny wynik studiów . . . . . .

Dodatkowe wnioski i uwagi Komisji . . . . . .

. . . . . .

Streszczenie

Praca ta prezentuje . . .

Słowa kluczowe: *słowa kluczowe.*

Abstract

Title: *Thesis title.*

This thesis describes . . .

Key words: *key words.*

Spis treści

1. Wprowadzenie

Szblon ten jest propozycją składu pracy dyplomowej inżynierskiej lub magisterskiej. Poniżej znajdują się przykłady pozwalające na szybkie zapoznanie się z podstawowymi elementami dokumentu takimi jak tablice, rysunki, wyliczenia itp.

Przed oddaniem tego dokumentu prawidłowo wypełnić jego początkowe strony tj.:

— stronę tytułową: rocznik, typ (magisterska/inżynierska), imię i nazwisko autora, tytuł, imię i nazwisko promotora pracy,
— życiorys: data urodzenia, datę rozpoczęcia studiów, zdjęcie i życiorys autora,
— streszczenie oraz słowa kluczowe w języku polskim i angielskim.

Szczegółowe opcje klasy `mwrep`, którą wykorzystuje ten dokument, opisane są w dokumentacji.

1.1. Przykład pierwszy

Pozostałe pliki (w głównej gałęzi katalogu 5015) reprezentują logikę struktury PKCS #15 oraz certyfikaty (pliki 4545, 4546, 4547). Jej szczegółowy opis zamieszczono w [4, 130-140]. W tablicy 1.1 zebrano podstawowe dane o wszystkich plikach. Rozmiar niektórych plików może być różny dla innych danych wejściowych. Dotyczy to w szczególności certyfikatów.

Należy przyjąć, że aplikacja PKCS #15 w karcie zajmuje do 6kB. W karcie *Cryptoflex 32K* pozsostaje więc około 26 kB możliwych do wykorzystania przez inne aplikacje. W szczególności mogą to być kolejne aplikacje PKCS #15 o innym profilu zastosowań.

Założenia

Dany jest zbiór pewnych maszyn. Każda z nich charakteryzuje się pewnym typem i lokalizacją. Maszyny złożone są z pewnych modułów.

Każda z maszyn raportuje do systemu centralnego zdarzenia jakie na niej zachodzą. Należą one do jednej z kategorii:

— normalne zdarzenie
— błąd – zdarzenie to zawiera opis zgłaszanego błędu (moduł); wyróżniamy błędy krytyczne (maszyna nie działa) i ostrzeżenia (np. brakuje zasobów dla pewnego modułu)
— interwencja – o kategorii lokalnej (np. maszyna sama się naprawiła) lub zdalnej (wymagana interwencja człowieka)

Na maszynach zachodzą pewne transakcje, których przebieg raportowany jest w postaci normalnych zdarzeń (chyba, że w trakcie pojawi się błąd).

Zdarzenia przechowywane są w bazie danych. Jest to jedna tabela, w której zapisane są dane określające maszynę, data i czas zdarzenia oraz jego opis.

Tablica 1.1. Wykaz plików karty *Cryptoflex 32K* z aplikacją PKCS #15

FID[a]	Rozmiar[b] (w bajtach)	Rodzaj pliku	Podstawowe prawa dostępu[c]
3F00	–	DF[d]	–
0011	27	EF[e]	R: NEV, U: AUT
0002	8	EF, TR[f]	R: ALW, U: NEV
5015	–	DF	–
4401[g]	255	EF, TR	R: ALW, U: AUT
4402	255	EF, TR	R: ALW, U: AUT
5031	255	EF, TR	R: ALW, U: AUT
5032	33	EF, TR	R: ALW, U: AUT
4545	849	EF, TR	R: ALW, U: AUT
4546	847	EF, TR	R: ALW, U: AUT
4547	849	EF, TR	R: ALW, U: AUT
4946	127	EF, TR	R: ALW, U: AUT
4B01, 4B02	–	DF	–
0000	16	EF, CHV[h]	R: NEV, U: CHV1 | AUT
3045, 3047	–	DF	–
0012	326	EF, PRVK[i]	R: NEV, U: AUT, C: CHV1
1012	330	EF, PUBK[j]	R: ALW, U: AUT, C: ALW
2F00	127	EF, TR	R: ALW, U: AUT

[a] ang. *file identifier* – identyfikator pliku

[b] rozmiar plików zawierających certyfikaty może być inny (w zależności od umieszczonego certyfikatu)

[c] stosowane oznaczenia: R (ang. *read*) – odczyt, U (ang. *update*) – zmiana, C – operacje kryptograficzne, NEV (ang. *never*) – nigdy, CHV1 (ang. *cardholder verification*) – kod uwierzytelniający użytkownika, ALW (ang. *always*) – zawsze, AUT (ang. *authenticate*) – uwierzytelnienie z użyciem klucza

[d] ang. *dedicated file* – plik dedykoway, katalog

[e] ang. *elementary file* – plik elementarny

[f] ang. *transparent* – struktura transparentna, przeźroczysta

[g] identyfikatory podano bez pełnych ścieżek, zobacz rysunek 1.1

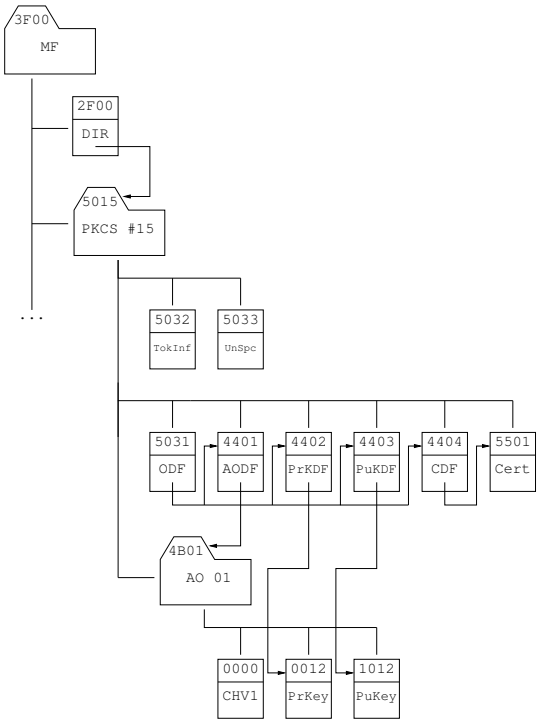
[h] ang. *cardholder verification* – kody służące do uwierzytelnienia użytkownika

[i] ang. *private key* – klucz prywatny

[j] ang. *public key* – klucz publiczny

1.1.1. Przykład drugi

Przykładowa struktura aplikacji zgodnej z PKCS #15 została zaprezentowana na rysunku 1.1. Kolejne elementy systemu plików odzwierciedlają instancje obiektów z danymi zdefiniowanymi w PKCS #15. Ich szczegółowa budowa, określona z użyciem notacji ASN.1 (ang. *abstract syntax notation number 1*) przedstawiona jest w samej normie.



Rysunek 1.1. Przykładowa struktura aplikacji PKCS #15

Twierdzenie 1.1.1. *Niech $x_1, x_2, x_3, \ldots$ będą dowolnymi zmiennymi o wartościach należących do zbiorów $X_1, X_2, X_3, \ldots$.*

Uwaga. Zdanie jest spełnione wyłącznie w dziedzinie D_1.

Dowód Twierdzenia 1.1.1. Załóżmy, że twierdzenie 1.1.1 nie jest prawdziwe. Wtedy zachodzi:

$$G(t) = L\gamma!\, t^{-\gamma} + t^{-\delta}\eta(t) \tag{1.1}$$

□

1.2. Przykład trzeci

W ostatnim dziesięcioleciu, wraz z silnym rozwojem aplikacji i urządzeń wykorzystujących algorytmy kryptograficzne, pojawiło się szereg problemów związanych z uniwersalnością zapisu danych wykorzystywanych podczas tych operacji. Jedna z amerykańskich firm, będąca liderem na rynku biznesowych zastosowań kryptografii, postanowiła opracować własne formuły zapisu informacji kryptograficznych. Brak dyskusji nad proponowanymi zaleceniami (w przeciwieństwie do głosowania

nad normami ISO/IEC) pozwolił na szybkie ogłoszenie początkowych wersji dokumentów oraz ich wdrożenie. Pomysł przyjął się i dzięki temu powstały normy przemysłowe dotyczące kryptografii.

Firma **RSA Data Security, Inc.**, bo o niej mowa, zaproponowała szereg zaleceń związanych z interfejsem dla kryptografii z kluczem publicznym. Znane są one pod ogólną nazwą PKCS (ang. *Public Key Cryptography Standards*). Wnioskując jedynie po ogólnym tytule można odnieść wrażenie, że zalecenia objęły jedynie algorytmy asymetryczne (w których występuje para kluczy - jawny, zwany publicznym oraz tajny, zwany prywatnym). Twórcy poruszyli również tematykę związaną z szerokim zastosowaniem tych algorytmów, dzięki czemu zalecenia zawierają praktycznie wszystkie najważniejsze i najaktualniejsze informacje dotyczące praktycznych zastosowań kryptografii.

Lista dokumentów z serii PKCS jest następująca:

— PKCS #1: *RSA Cryptography Standard* – zawiera opis algorytmu RSA zarówno w odniesieniu do podpisu cyfrowego jak i kopert cyfrowych[1]
— PKCS #3: *Diffie-Hellman Key Agreement Standard* – opisuje sposób implementacji algorytmu uzgadniania kluczy metodą Diffiego-Hellmana
— PKCS #5: *Password-Based Cryptography Standard* – zawiera opis metody bezpiecznej wymiany kluczy prywatnych
— PKCS #6: *Extended-Certificate Syntax Standard* – opisuje budowę certyfikatów klucza publicznego X.509
— PKCS #7: *Cryptographic Message Syntax Standard* – jest to abstrakcyjny opis danych, które podlegają operacjom kryptograficznym
— PKCS #8: *Private-Key Information Syntax Standard* – zawiera abstrakcyjny opis dotyczący składowania kluczy prywatnych (w formie jawnej i zaszyfrowanej) wraz z zestawem atrybutów
— PKCS #9: *Selected Attribute Types* – zawiera definicję atrybutów związanych z certyfikatami, podpisami cyfrowymi i kluczami prywatnymi
— PKCS #10: *Certification Request Syntax Standard* – opisuje format żądania certyfikacyjnego
— PKCS #11: *Cryptographic Token Interface Standard* – opisuje abstrakcyjny interfejs programisty dla różnych typów urządzeń kryptograficznych
— PKCS #12: *Personal Information Exchange Syntax Standard* – zawiera opis formatu zapisu danych kryptograficznych przez aplikacje
— PKCS #13: *Elliptic Curve Cryptography Standard* – zawiera opis algorytmów opartych na krzywych eliptycznych
— PKCS #14: *Pseudo Random Number Generation* – zawiera opis algorytmów związanych z generacją liczb pseudolosowych[2]
— PKCS #15: *Cryptographic Token Information Format Standard* – opisuje sposób zapisu danych w żetonach kryptograficznych (takich jak karty procesorowe)

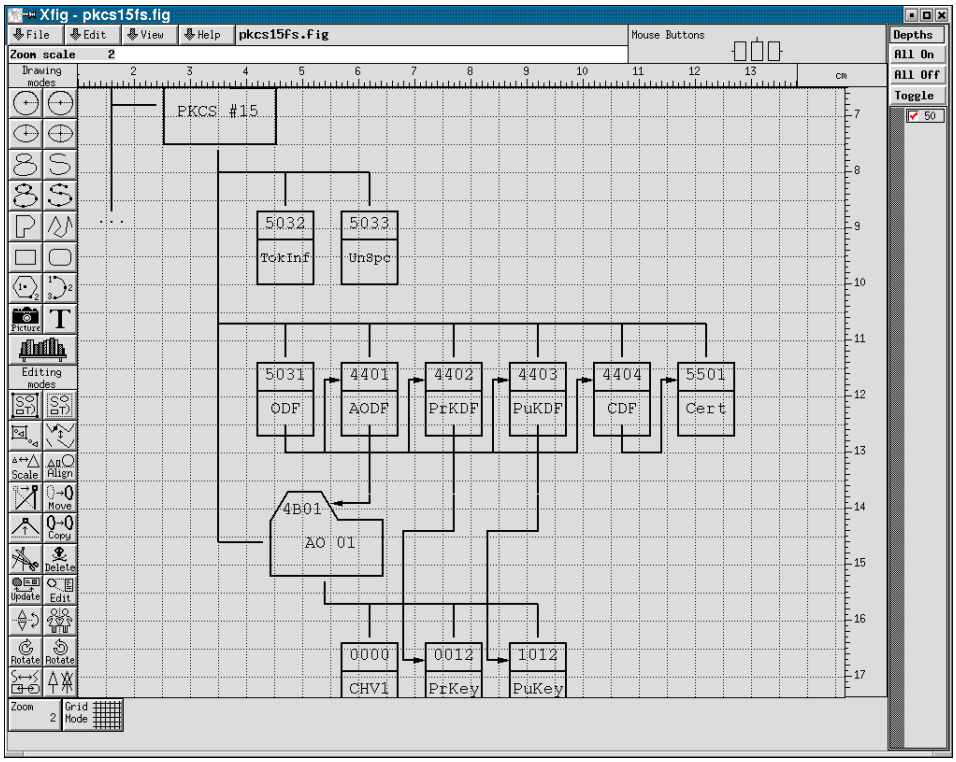
Wszystkie publikacje dostępne są pod adresem internetowym `http://www.rsasecurity.com/rsalabs/`.

[1] dokumenty o identyfikatorach #2 i #4 zostały połączone w #1
[2] aktualnie dokument ten jest opracowywany

1.3. Przykład czwarty

Xfig[3] i *Dia*[4] to aplikacje wspomagające tworzenie rysunków w grafice wektorowej. Pierwsza z nich przeznaczona jest głownie do tworzenia obrazów z prostych elementów takich jak linie, prostokąty, okręgi, łuki. *Dia*, wzorowane na *Visio* firmy *Microsoft*, posiada wiele bibliotek graficznych i jest najbardziej pomocne do tworzenia skomplikowanych diagramów np. w UML. Obie aplikacje obsługują szeroki wachlarz formatów plików graficznych, co pozwala na wykorzystanie rysunków stworzonych z ich pomocą w wielu programach służących do edycji lub składu dokumentów.

Rysunek 1.2 obrazuje program *Xfig* podczas pracy nad jednym z rysunków wykorzystanych w publikacji [4].



Rysunek 1.2. *Xfig*

1.4. Przykład piąty

Po utworzeniu pliku konfiguracyjnego można przystąpić do przygotowania katalogów służących do przechowywania danych CA (zgodnie z wcześniej założonymi nazwami w *openssl.cnf*). Tworzymy katalog *ca*, a w nim katalogi *certs*, *crl*, *private*, *newcerts* oraz pliki *serial.txt* (z wpisem 01) i *index.txt*. Plik *openssl.cnf* należy umieścić na tym samym poziomie co katalog *ca*. Oczywiście możliwe jest zupełnie inne

[3] http://www.xfig.org/

[4] http://www.gnome.org/projects/dia/, http://dia-installer.sourceforge.net/

zorganizowanie sposobu przechowywania danych w CA, jednak musi ono odpowiadać wcześniej przyjętym założeniom w pliku konfiguracyjnym.

Następnym krokiem przy tworzeniu CA jest wygenerowanie pary kluczy oraz autocertyfikatu (w przypadku, gdy certyfikat dla centrum nie będzie poświadczany przez inne centrum certyfikacji) dla centrum certyfikacji.

```
# generacja klucza prywatnego RSA o długości 4096 bitów
# do pliku cakey.pem (klucz w formie jawnej)
openssl genrsa -out ca/private/cakey.pem 4096 -config openssl.cnf

# utworzenie autocertyfikatu centrum (cacert.pem) o strukturze
# X.509 i formacie PEM dla klucza jawnego związanego z kluczem
# tajnym cakey.pem
openssl req -new -x509 -days 1825 -key ca/private/cakey.pem -out
    ca/cacert.pem -config openssl.cnf
```

Po tych operacjach CA jest gotowe do pracy.

1.5. Przykład szósty

Oto przykładowy wydruk:

```
1  import a.b.c;

   // komentarz

5  /*
    * Komentarz...
    */
   public class A
   {
10   int A;
     int B; // zmienna

     A()
     {
15     A=1;
       /* to jest komentarz */
     }

     public void metodaA(int i)
20   {
       for (int a=i; a<100; ++a)
       {
         short sw=(short)a;
         // ...
25     }
     }
   }
```

Wydruk 1.1. Przykładowy wydruk

A oto wydruk wpleciony w tekst...

```
1  /* ta funkcja oblicza a+b */
   int sum(int a, int b)
```

```
{
  int suma=0;

  suma=a+b;

  return suma;
}
```

. . . i tekst za kodem.

Bibliografia

[1] Michael D. Ernst. *Dynamically Discovering Likely Program Invariants*. Ph.D., University of Washington Department of Computer Science and Engineering, Seattle, Washington, 2000.

[2] Michael D. Ernst. Daikon Invariant Detector User Manual. 2005.

[3] Gajek Lesław, Kałuszka Marek. *Wnioskowanie statystyczne - modele i metody*. Wydawnictwa Naukowo-Techniczne, wydanie trzecie, Warszawa 1993, 1996.

[4] Piotr Nazimek. *Inżynieria programowania kart inteligentnych*. Warszawa, 2005.

[5] Benjamin Jack R., Cornell C. Allin. *Rachunek prawdopodobieństwa, statystyka matematyczna i teoria decyzji dla inżynierów*. Wydawnictwa Naukowo-Techniczne, wydanie pierwsze, Warszawa 1977.

[6] Łukaszek Władysław. *Podstawy statystycznego opracowania pomiarów*. Wydawnictwo Politechniki Śląskiej, wydanie trzecie, Gliwice 1995.