



# Laboratorium 1 - Podstawowe zasady posługiwania się modułem TPM



*To jest laboratorium ćwiczeniowe. Należy je wykonać w czasie trwania zajęć. Zadanie to nie powinno zająć więcej czasu niż czas trwania laboratorium. Jeśli zadanie zostanie zakończone wcześniej, to można kontynuować prace dotyczące poprzedniego laboratorium lub pracy semestralnej.*



Autorzy konspektu: Łukasz Cierocki, Mateusz Kłos, Jerzy Pejaś

## Metody zaliczenia:

Jako zaliczenie niniejszego laboratorium przewiduje się przygotowanie z niego sprawozdania. Terminem dostarczenia sprawozdania jest przesłanie go do kolejnych zajęć laboratoryjnych.

## Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z procedurami inicjowania oraz podstawowymi funkcjami TPM dostępnymi w systemie Linux. Obejmują one wszystkie ważne działania wykonywane bezpośrednio we współpracy z układem TPM, takie jak jego włączanie, lub czytanie rejestrów konfiguracji platformy (PCR). Podczas ćwiczenia zbadane zostaną także inne funkcjonalności modułu TPM.

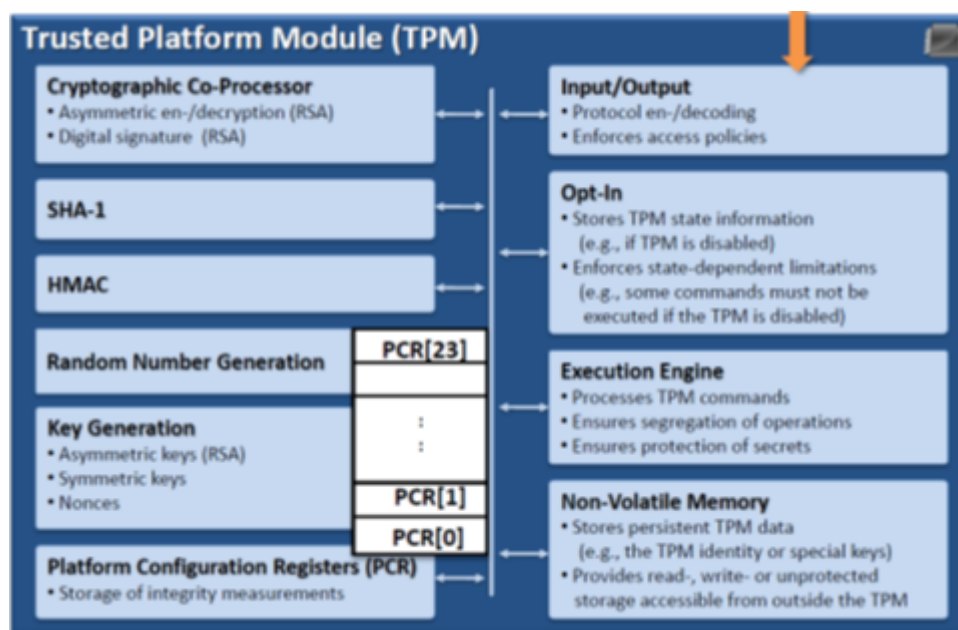
## Efekty

Po ukończeniu ćwiczenia studenci powinni posiadać wiedzę dotyczącą podstaw obsługi TPM.

# Wstęp

Zaufany model obliczeniowy i technologia do jego realizacji zostały zaproponowane przez Trusted Computing Group (TCG) w 2003 roku. Jego głównym składnikiem jest Trusted Platform Module (TPM), który jest scalonym mikroukładem bezpieczeństwa montowanym na płycie głównej w większości nowych komputerów/laptopów klasy PC. Obecna implementacja jest w rzeczywistości koprocesorem kryptograficznym zapewniającym sprzętowe generowanie liczb losowych i niewielki zestaw funkcji kryptograficznych (generowanie kluczy, podpisywanie, szyfrowanie, obliczanie skrótów, MAC). Dodatkowo TPM oferuje możliwość bezpiecznego magazynowania danych, raportowania i zarządzania integralnością platformy w oparciu o rejestry konfiguracji platformy (PCR), możliwość zdalnej atestacji, kryptograficznego wiązania i pieczętowania.

Architekturę TPM przedstawiono na rys. 1. TPM działa jako "Główny Punkt Zaufania" (ang. Root of Trust) platformy. Każdy użytkownik platformy musi ufać, że TPM funkcjonuje zgodnie ze specyfikacją i nie został skompromitowany. Dokładny opis możliwości TPM można znaleźć w materiałach wykładowych oraz w specyfikacji TCG.



Architektura TPM

TPM może być używane do autoryzacji sprzętu, a tym samym do określenia czy urządzenie próbujące uzyskać dostęp jest bezpieczne. Jeśli urządzenie zostanie skradziony, dane zostaną zabezpieczone do momentu podania praktycznie niemożliwego do podrobienia klucza, który zabezpiecza dane i czyni je niezdatnymi do użytku.

## Konfiguracja podstawowych narzędzi

Na podstawę wirtualizacji wybrano dystrybucję KaliLinux, w oparciu o oprogramowanie VirtualBox. Dystrybucja w formie prekonfigurowanej dostępna jest pod adresem:



po dekompresji konieczne jest zaimportowanie jej do programu VirtualBox. Po zakończeniu importu wymagane jest zaktualizowanie wszystkich pakietów poprzez uruchomienie poleceń:

```
sudo apt update  
sudo apt upgrade
```

Lista pakietów dodatkowych, niezbędnych do prawidłowego funkcjonowania TPM2 i TSS:

```
sudo apt install autoconf-archive dh-autoreconf libcmocka0 libcmocka-dev  
procps iproute2 build-essential pkg-config libtool automake  
libssl-dev uthash-dev autoconf doxygen libjson-c-dev libini-config-dev  
libcurl4-openssl-dev libgcrypt20-dev libglib2.0-0 libglib2.0-dev m4 pandoc uuid  
uuid-dev python python3 libltdl-dev  
sudo apt autoclean  
sudo apt autoremove  
sudo apt clean
```

Celem instalacji emulatora TPM należy pobrać archiwum z strony:



<https://sourceforge.net/projects/ibmswtpm2/files/latest/downloadls>

Archiwium rozpakować poleceniem:

```
tar -xf ibmtpm1682.tar.gz
```

Kolejno wywołać:

```
cd /src  
make
```



Instalacja dodatkowych pakietów niezbędnych do wykonania może być wykonana poprzez menedżer pakietów lub skompilowane bezpośrednio ze źródeł. Poniżej zaproponowano instalację z kodu źródłowego gdyż ten sposób pozwala na zachowanie wyższej stabilności instalowanych narzędzi.

## Instalacja TPM2-TSS

Celem instalacji konieczne jest wykonanie poniższych poleceń:

```
git clone https://github.com/tpm2-software/tpm2-tss.git  
cd /tpm2-tss  
./bootstrap  
./configure --with-udevrulesdir=/etc/udev/rules.d  
make  
sudo make install  
sudo ldconfig
```

## TPM2-ABRMD

Moduł TPM2-ABRMD powinien być uruchomiony jako użytkownik **tss** albo **root**. Najczęstszą praktyką w tym przypadku jest uruchomienie ABRMD z wykorzystaniem nieuprawnionego sudo użytkownika, którego można utworzyć za pomocą poniższej komendy z kroku 7.

Celem instalacji konieczne jest wykonanie poniższych poleceń:

```
git clone https://github.com/tpm2-software/tpm2-abrmd.git  
cd /tpm2-abrmd
```

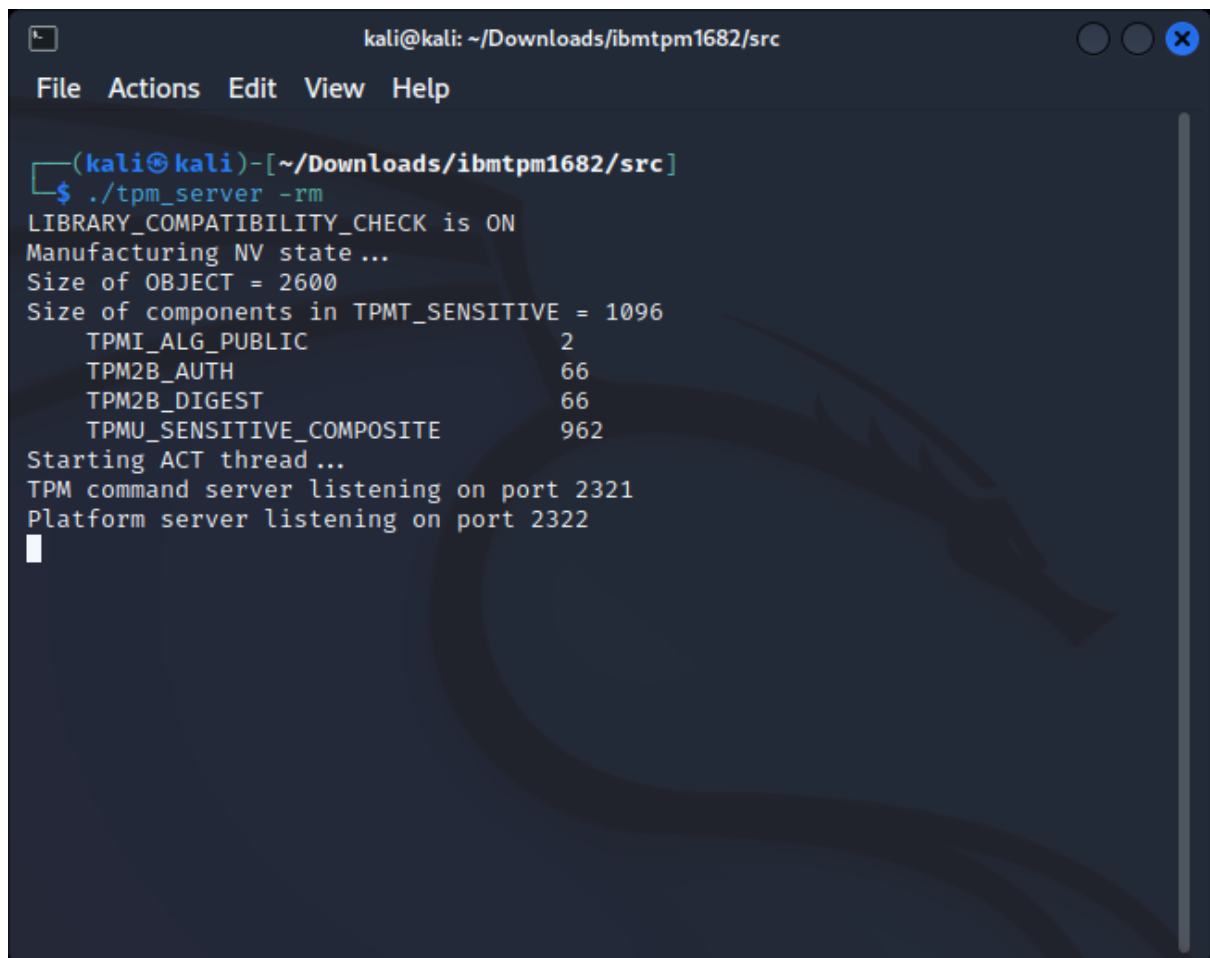
```
./bootstrap
./configure --with-dbuspolicydir=/etc/dbus-1/system.d
make
sudo make install
sudo ldconfig
sudo useradd --system --user-group tss
sudo init 6
```

## Uruchomienie symulatora

Celem uruchomienia emulatora, konieczne jest przejście do katalogu `/src` gdzie wcześniej był instalowany symulator (domyślnie katalog `ibmtpm1682`) oraz wywołać polecenie:

```
./tpm_server -rm
```

Po poprawnej instalacji powinniśmy uzyskać poniższy wynik po uruchomieniu powyższego skryptu:



```
kali@kali: ~/Downloads/ibmtpm1682/src
File Actions Edit View Help

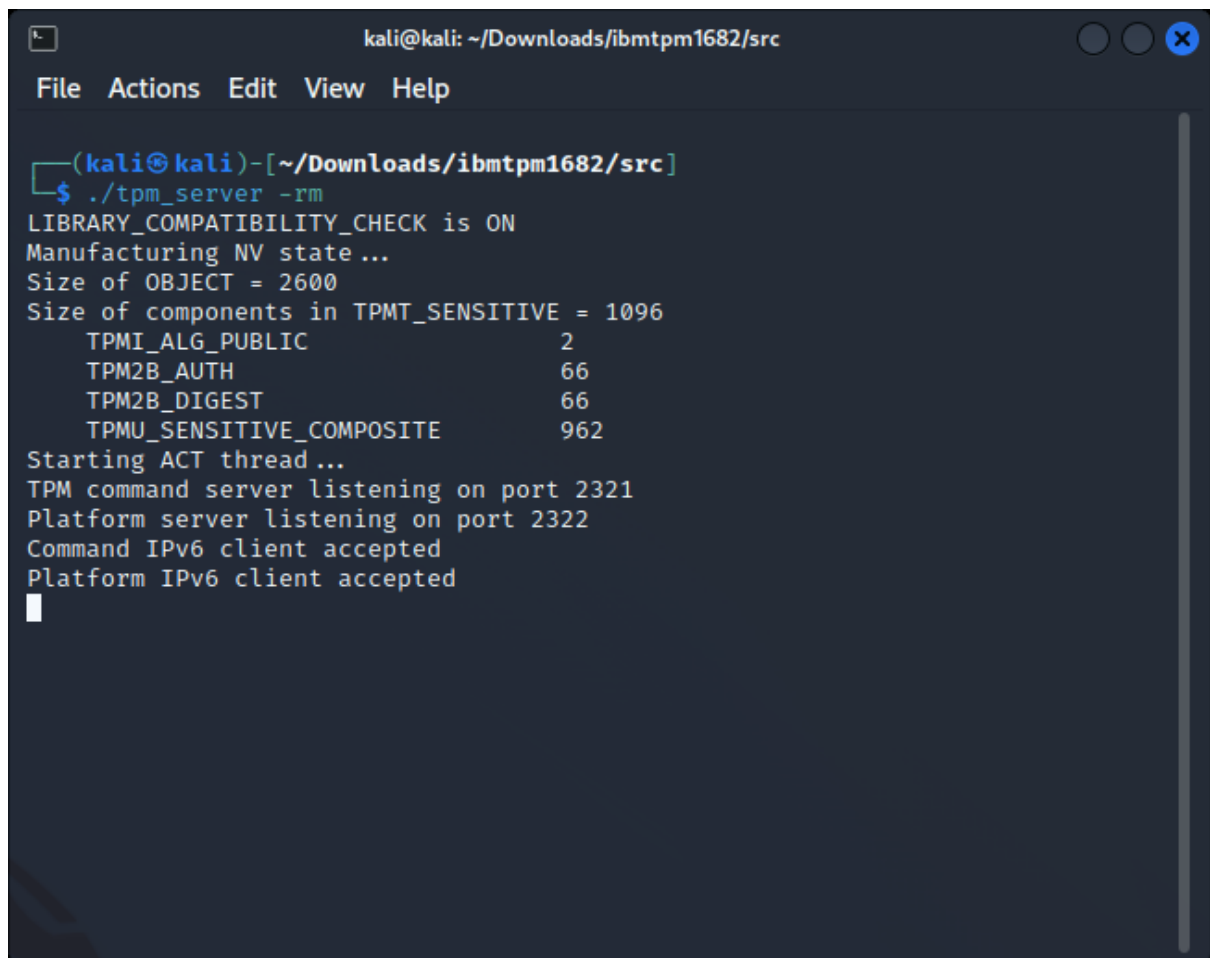
(kali@kali)-[~/Downloads/ibmtpm1682/src]
$ ./tpm_server -rm
LIBRARY_COMPATIBILITY_CHECK is ON
Manufacturing NV state ...
Size of OBJECT = 2600
Size of components in TPMT_SENSITIVE = 1096
  TPMI_ALG_PUBLIC          2
  TPM2B_AUTH               66
  TPM2B_DIGEST             66
  TPMU_SENSITIVE_COMPOSITE 962
Starting ACT thread ...
TPM command server listening on port 2321
Platform server listening on port 2322
█
```

Rysunek: Wynik uruchomienia symulatora układu TPM

Aby móc pracować z symulatorem TPM, konieczne jest jeszcze uruchomienie poprzednio zainstalowanego narzędzia `tpm2-abrmd` poprzez uruchomienie polecenia:

```
sudo -u tss /usr/local/sbin/tpm2-abrmd --tcti=mssim
```

Poprawnym wynikiem powinno być wyświetlenie poniższych wyników w oknie terminala uruchomionego symulatora:



```
kali@kali: ~/Downloads/ibmtpm1682/src
File Actions Edit View Help

(kali@kali)-[~/Downloads/ibmtpm1682/src]
$ ./tpm_server -rm
LIBRARY_COMPATIBILITY_CHECK is ON
Manufacturing NV state ...
Size of OBJECT = 2600
Size of components in TPMT_SENSITIVE = 1096
  TPMI_ALG_PUBLIC          2
  TPM2B_AUTH               66
  TPM2B_DIGEST             66
  TPMU_SENSITIVE_COMPOSITE 962
Starting ACT thread ...
TPM command server listening on port 2321
Platform server listening on port 2322
Command IPv6 client accepted
Platform IPv6 client accepted
█
```

## Testowanie zainstalowanych narzędzi

Najpierw przetestujemy połączenie zainstalowanego wcześniej symulatora z narzędziem ABRMD poprzez wykonanie komend:

```
tpm2_selftest; echo $?          #Prawidłowa wartość 0
tpm2_selftest --tcti=tabrmd; echo $? #Prawidłowa wartość 0
tpm2_selftest --tcti=device; echo $? #Prawidłowa wartość 1 (Brak fizycznego urządzenia)
tpm2_selftest --tcti=mssim      #Prawidłowa wartość - zawieszenie, zamknięcie ^C
```

wyniki powyższych komend powinny prezentować się nas

Kolejno testowane będą parametry zainstalowanych narzędzi poprzez wywołanie komendy:

```
tpm2_testparms AES RSA ECC; echo $? #Prawidłowa wartość 0
```

## Odpytanie układu o udostępniane funkcje

Komendą pozwalającą na odpytanie układu o udostępniane funkcje jest:

```
tpm2_getcap -l
```

Ważne aby przeprowadzić również test wbudowanego zegara poprzez wywołanie komendy:

```
tpm2_readclock
```

## Generowanie losowych bajtów

Proces generowania bajtów losowych odbywa się poprzez komendę:

```
tpm2_getrandom -o randomtpm.out 32
```

## Wyświetlanie i ustawianie rejestrów PCR

Wyświetlanie rejestru PCR z wykorzystaniem TPM może być zrealizowane za pomocą poleceń:

- tpm2\_pcrread