



Laboratorium 3 - Testy platformy TPM2- Engine & OpenSSL



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.

Kontynuacja poprzedniego laboratorium

Celem uruchomienia powyższych programów konieczne jest poprawa konfiguracji dostępnej w pliku **`/usr/local/etc/tpm2-tss`**. Niezbędne jest dodanie dyrektywy:

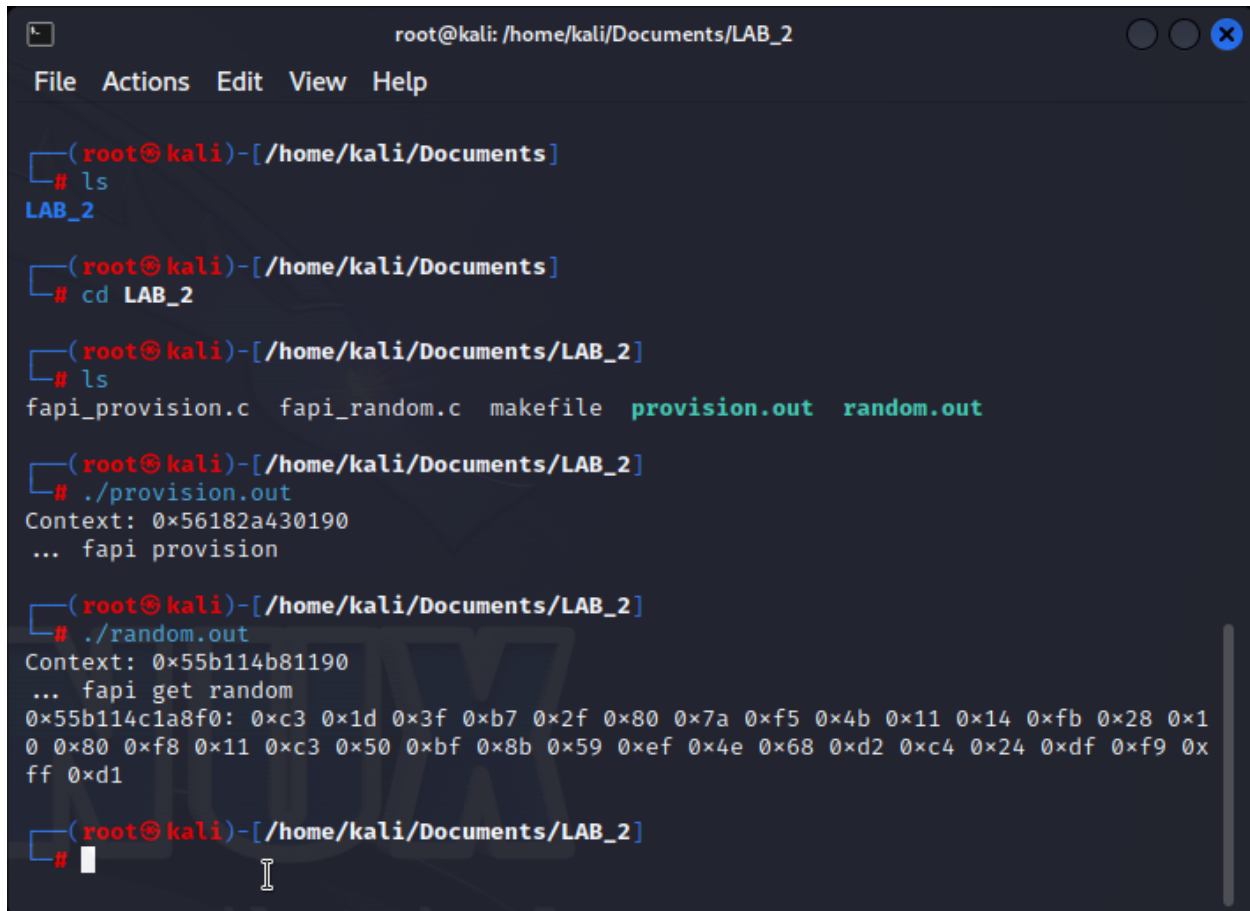
```
"ek_cert_less": "yes"
```

a finalny plik konfiguracyjny powinien wyglądać tak:

```
{
  "profile_name": "P_ECCP256SHA256",
  "profile_dir": "/usr/local/etc/tpm2-tss/fapi-profiles/",
  "user_dir": "~/.local/share/tpm2-tss/user/keystore",
  "system_dir": "/usr/local/var/lib/tpm2-tss/system/keystore",
  "tcti": "",
  "system_pcrs" : [],
```

```
"log_dir" : "/usr/local/var/run/tpm2-tss/eventlog/",
"firmware_log_file": "/dev/null",
"ima_log_file": "/sys/kernel/security/ima/binary_runtime_measurements",
"ek_cert_less": "yes"
}
```

Wynikiem uruchomienia wcześniej skompilowanych programów powinno być:



```
root@kali: /home/kali/Documents/LAB_2
File Actions Edit View Help

(root@kali)-[/home/kali/Documents]
# ls
LAB_2

(root@kali)-[/home/kali/Documents]
# cd LAB_2

(root@kali)-[/home/kali/Documents/LAB_2]
# ls
fapi_provision.c fapi_random.c makefile provision.out random.out

(root@kali)-[/home/kali/Documents/LAB_2]
# ./provision.out
Context: 0x56182a430190
... fapi provision

(root@kali)-[/home/kali/Documents/LAB_2]
# ./random.out
Context: 0x55b114b81190
... fapi get random
0x55b114c1a8f0: 0xc3 0x1d 0x3f 0xb7 0x2f 0x80 0x7a 0xf5 0x4b 0x11 0x14 0xfb 0x28 0x1
0 0x80 0xf8 0x11 0xc3 0x50 0xbf 0x8b 0x59 0xef 0x4e 0x68 0xd2 0xc4 0x24 0xdf 0xf9 0x
ff 0xd1

(root@kali)-[/home/kali/Documents/LAB_2]
#
```

TPM2-ENGINE

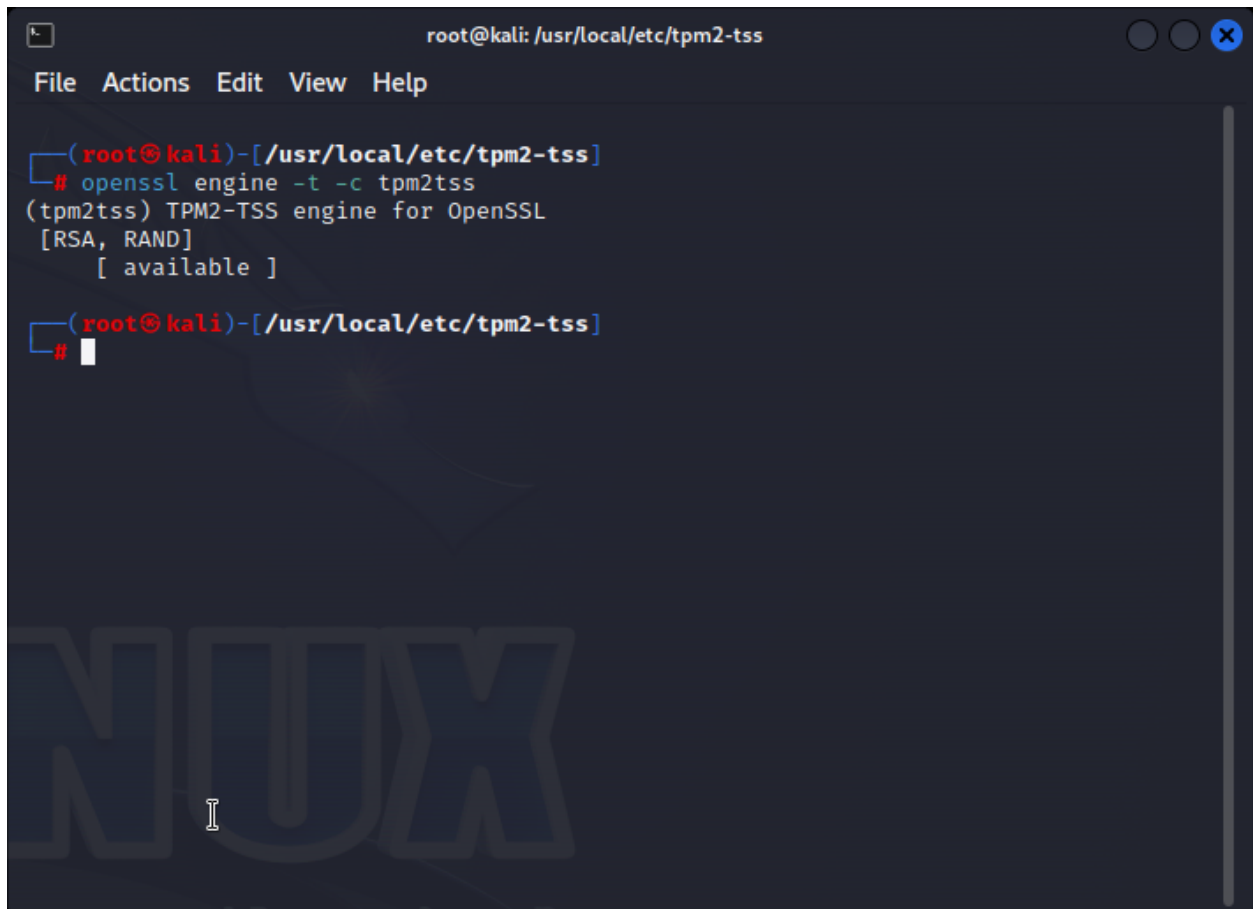
Pakiet tpm2-engine jest wrażliwy na zmianę istotnych parametrów tpm2/tss. W reakcji na te zmiany często przestaje działać (przy przejęciu tpm na własność i przy ręcznym przydzieleniu rejestrów pamięci PCR). Ze względu na fakt, że wykorzystujemy narzędzie na wirtualnej maszynie i sytuacja próby dostępu do modułu przez wielu użytkowników na raz nie wystąpi, można pominąć te polecenia przy inicjalizacji.

```
git clone https://github.com/tpm2-software/tpm2-tss-engine
cd /tpm2-tss-engine
./bootstrap
./configure
make
sudo make install
```

Prawidłowo zainstalowany TPM2-ENGINE po wykonaniu komendy:

```
openssl engine -t -c tpm2tss
```

powinien nam się ukazać taki oto rezultat:



```
root@kali: /usr/local/etc/tpm2-tss
File Actions Edit View Help

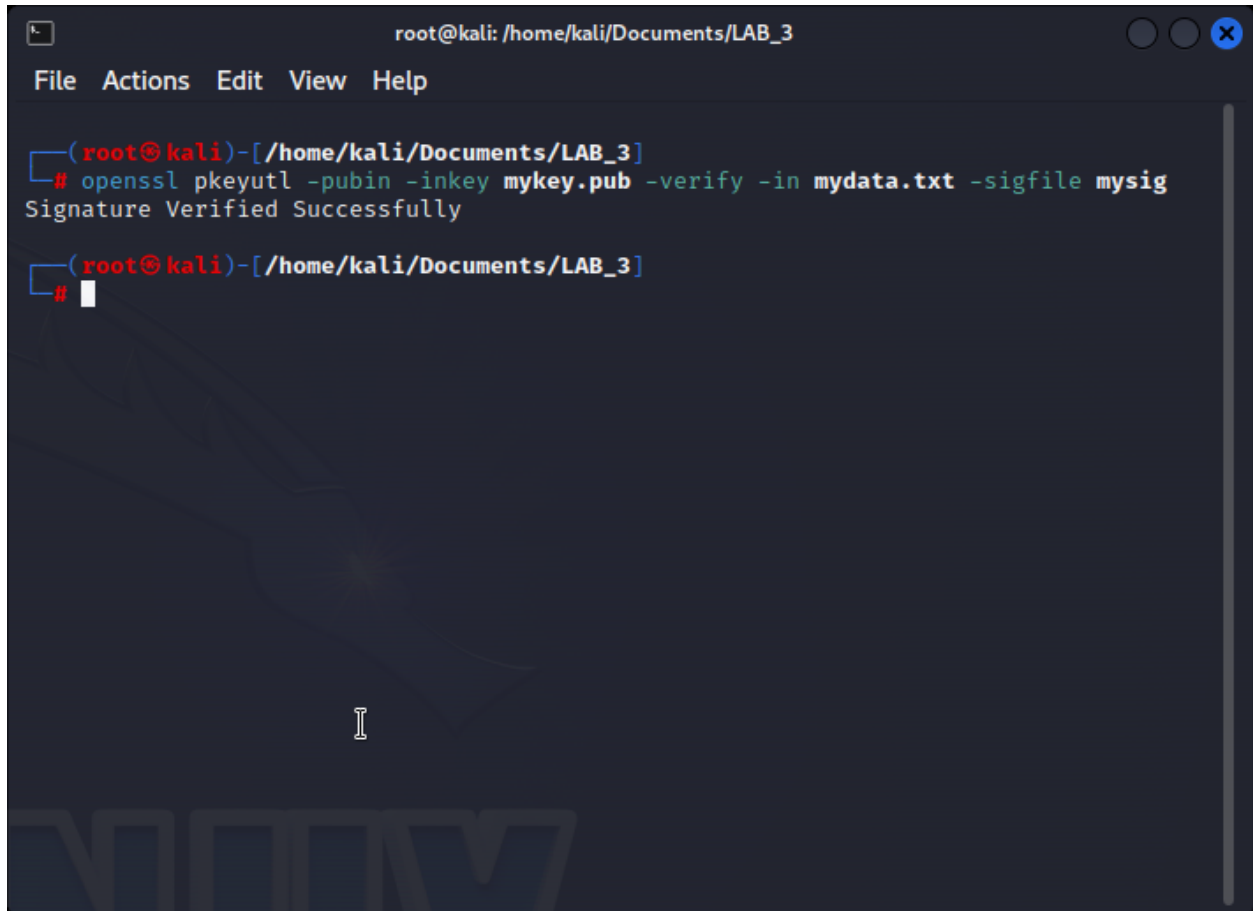
(root@kali)-[/usr/local/etc/tpm2-tss]
# openssl engine -t -c tpm2tss
(tpm2tss) TPM2-TSS engine for OpenSSL
[RSA, RAND]
[ available ]

(root@kali)-[/usr/local/etc/tpm2-tss]
#
```

Podpis RSA

```
echo "my data" > mydata.txt
tpm2tss-genkey -a rsa -s 2048 mykey.key
openssl rsa -engine tpm2tss -inform engine -in mykey.key -pubout -outform pem -out mykey.pub
openssl pkeyutl -engine tpm2tss -keyform engine -inkey mykey.key -sign -in mydata.txt -out mysig
openssl pkeyutl -pubin -inkey mykey.pub -verify -in mydata.txt -sigfile mysig
```

Wynik ostatniej komendy to:



The screenshot shows a terminal window titled 'root@kali: /home/kali/Documents/LAB_3'. The terminal output shows the execution of the command: `openssl pkeyutl -pubin -inkey mykey.pub -verify -in mydata.txt -sigfile mysig`. The output of the command is 'Signature Verified Successfully'. The prompt is `(root@kali)-[/home/kali/Documents/LAB_3]` and the cursor is on a new line.

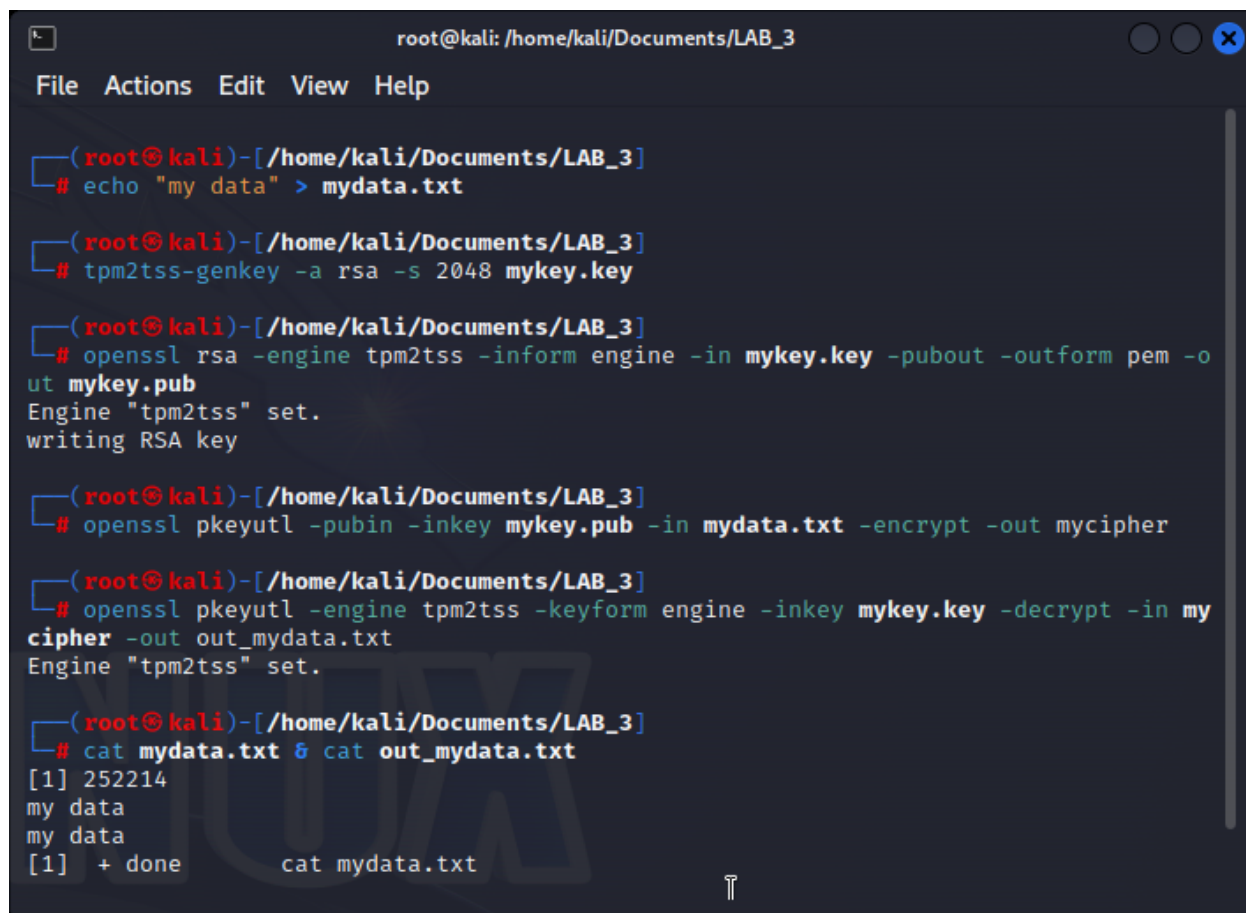
Szyfrowanie i deszyfrowanie RSA

Następująca sekwencja poleceń tworzy klucz RSA, eksportuje klucz publiczny, szyfruje i deszyfruje plik wykorzystując do tego TPM:

```
echo "my data" > mydata.txt
tpm2tss-genkey -a rsa -s 2048 mykey.key
openssl rsa -engine tpm2tss -inform engine -in mykey.key -pubout -outform pem -out mykey.pub
openssl pkeyutl -pubin -inkey mykey.pub -in mydata.txt -encrypt -out mycipher
```

```
openssl pkeyutl -engine tpm2tss -keyform engine -inkey mykey.key -decrypt -in mycipher -out out_mydata.txt
cat mydata.txt & cat out_mydata.txt
```

Wynik to:



```
root@kali: /home/kali/Documents/LAB_3
File Actions Edit View Help

(root@kali)-[/home/kali/Documents/LAB_3]
# echo "my data" > mydata.txt

(root@kali)-[/home/kali/Documents/LAB_3]
# tpm2tss-genkey -a rsa -s 2048 mykey.key

(root@kali)-[/home/kali/Documents/LAB_3]
# openssl rsa -engine tpm2tss -inform engine -in mykey.key -pubout -outform pem -out mykey.pub
Engine "tpm2tss" set.
writing RSA key

(root@kali)-[/home/kali/Documents/LAB_3]
# openssl pkeyutl -pubin -inkey mykey.pub -in mydata.txt -encrypt -out mycipher

(root@kali)-[/home/kali/Documents/LAB_3]
# openssl pkeyutl -engine tpm2tss -keyform engine -inkey mykey.key -decrypt -in mycipher -out out_mydata.txt
Engine "tpm2tss" set.

(root@kali)-[/home/kali/Documents/LAB_3]
# cat mydata.txt & cat out_mydata.txt
[1] 252214
my data
my data
[1] + done      cat mydata.txt
```

Podpis danych z wykorzystaniem klucza TPM2 / weryfikacja podpisu z OpenSSL

```
echo "message" > msg.txt
tpm2_createprimary -c primary.ctx
tpm2_create -C primary.ctx -u key.pub -r key.priv
tpm2_load -C primary.ctx -u key.pub -r key.priv -c key.ctx
openssl dgst -sha1 -binary -out hash.bin msg.txt
tpm2_sign -c key.ctx -g sha1 -f plain -d hash.bin -o hash.plain
tpm2_readpublic -c key.ctx -f der -o sub-pub.der
openssl dgst -verify sub-pub.der -keyform der -sha1 -signature hash.plain msg.txt
```

gdzie wynikiem ostatniej operacji powinno być:



```
root@kali: /home/kali/Documents/LAB_3
File Actions Edit View Help

(root@kali)-[/home/kali/Documents/LAB_3]
# openssl dgst -verify sub-pub.der -keyform der -sha1 -signature hash.plain msg.tx
t
Verified OK

(root@kali)-[/home/kali/Documents/LAB_3]
#
```

The image shows a terminal window with a dark background. At the top, the title bar reads 'root@kali: /home/kali/Documents/LAB_3'. Below the title bar is a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal content shows a prompt '(root@kali)-[/home/kali/Documents/LAB_3]' followed by the command '# openssl dgst -verify sub-pub.der -keyform der -sha1 -signature hash.plain msg.tx'. The output is 't' on the next line and 'Verified OK' on the following line. A second prompt '(root@kali)-[/home/kali/Documents/LAB_3]' is shown with a '#' and a cursor. A large, semi-transparent 'NUTX' watermark is visible in the background of the terminal window.