

TCU (Trusted Computing Unit) – Teorija i Simulacija PC, Raspberry Pi / Jetson Nano i PYNQ-Z2

Aleksandar Peulić

Tema

- Upoznajemo se sa konceptom TCU i simulacijom na različitim platformama.
- TCU je ključna komponenta za bezbednost u modernim uređajima, a cilj ove prezentacije je da pokaže kako se TCU funkcionalnosti mogu demonstrirati kroz softversku i hardversku simulaciju.

Šta je TCU?

- TCU je modul koji omogućava sigurno izvršavanje operacija u mikroprocesorima.
- Štiti poverljive podatke poput kriptografskih ključeva i sertifikata.
- Zašto je važan? -
- U IoT i automobilskim sistemima, bezbednost podataka je kritična.
- TCU sprečava neautorizovani pristup i manipulaciju podacima.

Osnove TCU

- TCU pruža enkripciju, digitalni potpis i verifikaciju podataka. - Softverski TCU omogućava testiranje i učenje osnovnih funkcionalnosti bez specijalizovanog hardvera.
- Hardverski TCU (kao TPM) čuva ključeve u izolovanoj memoriji i pruža visoki nivo sigurnosti.
- TPM znači Trusted Platform Module.
- To je specijalizovani hardverski čip koji se koristi za bezbedno čuvanje kriptografskih ključeva, certifikata i drugih sigurnih podataka.

Osnove TCU Dijagram

CPU šalje podatke TCU-u → TCU obrađuje podatke → vraća rezultat aplikaciji

Zašto koristiti TCU?

- Poverljivost: bez TCU-a, ključevi se mogu čitati iz memorije.
- Autentikacija: TCU generiše digitalne potpise koji garantuju identitet pošiljaoca.
- Integritet: verifikacija potpisa osigurava da podaci nisu menjani.
- Zaštita od napada: fizički napadi i malware ne mogu lako pristupiti ključevima.

Osnove kriptografije

- Simetrična enkripcija koristi isti ključ za šifrovanje i dešifrovanje.
- Asimetrična enkripcija koristi par ključeva (javni i privatni).
- Digitalni potpis koristi privatni ključ za autentikaciju i javni za verifikaciju.
- Hash funkcije stvaraju jedinstveni otisak podataka.
- Fernet koristi AES enkripciju i HMAC za sigurnu i jednostavnu implementaciju. -

Osnove kriptografije

Dijagram:

plain text → encrypt → cipher text → decrypt → plain text.

TCU tok operacija

1. CPU šalje podatke TCU-u.
2. TCU aktivira LED da signalizira rad.
3. TCU enkriptuje podatke i stvara tajni tekst.
4. TCU generiše digitalni potpis za autentikaciju.
5. TCU verifikuje potpise da bi osigurao integritet.
6. TCU dekriptuje podatke i vraća originalne informacije.
7. LED se isključuje kada je operacija završena. - Ovo pomaže da se vizualno prati tok operacija.

Softverska simulacija na PC-u

- Softverski mock TCU(simulacija“ ili „lažni objekat“ koji imitira ponašanje pravog sistema ili komponente, ali nije pravi hardver ili stvarna implementacija) omogućava testiranje funkcionalnosti bez fizičkog hardvera.
- Funkcije: encrypt, decrypt, sign, verify.
- Konzolni izlaz prikazuje svaki korak operacije.

Praktičan primer PC

- Python kod implementira MockTCU klasu.
- Demonstracija: enkripcija, dekripcija, digitalni potpis, verifikacija.
- Vežbe: promene ulaznog teksta, testiranje grešaka u dekripciji i verifikaciji.
-

PC (Ubuntu 22) – softverski TCU simulacija

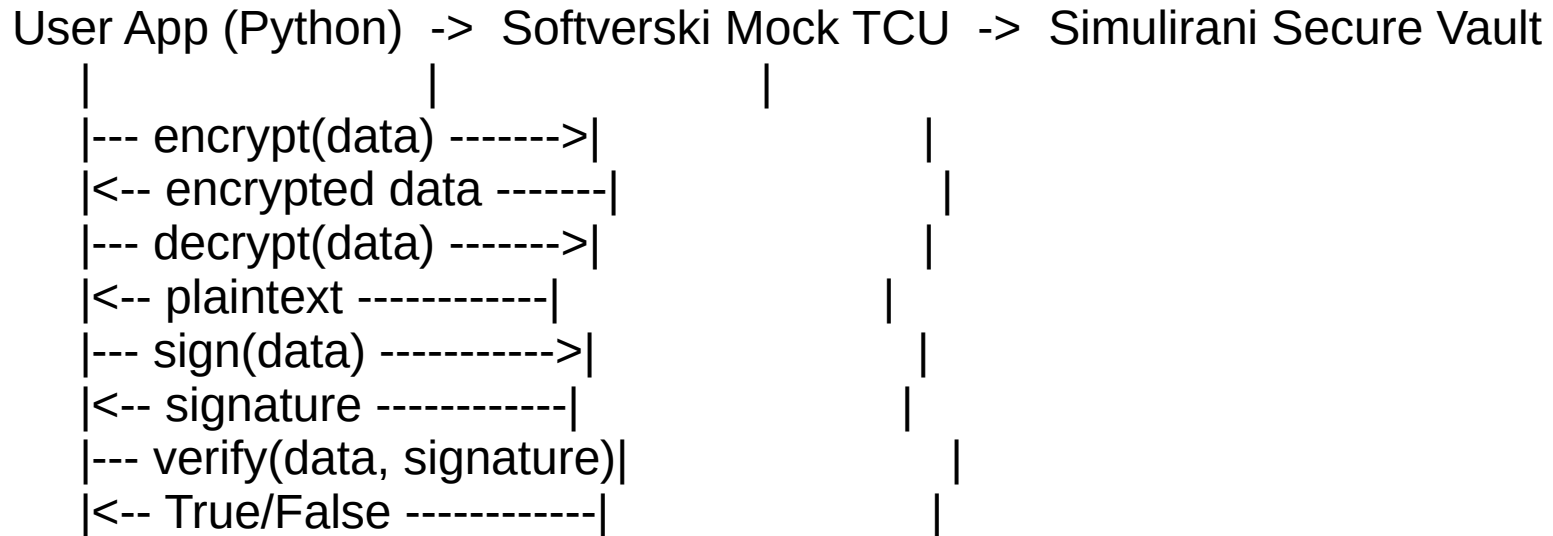
- Python kod implementira MockTCU klasu.
- Demonstracija: enkripcija, dekripcija, digitalni potpis, verifikacija.
- Vežbe: promene ulaznog teksta, testiranje grešaka u dekripciji i verifikaciji.
- Koncept TCU na PC:
- PC omogućava simulaciju TCU funkcija pomoću softverskih alata.

PC (Ubuntu 22) – softverski TCU simulacija

- Softverski mock TCU:
- Funkcije: encrypt(data), decrypt(data), sign(data), verify(data, signature)
- Ključ se generiše i čuva u memoriji
- Ograničenja:
- Ključevi nisu izolovani
- Nema otpornosti na fizičke napade
- Simulira samo logiku TCU-a

PC (Ubuntu 22) – softverski TCU simulacija

- Dijagram toka operacija:



Raspberry Pi / Jetson Nano

- ARM platforma omogućava bližu simulaciju IoT uređaja.
- GPIO LED blink pokazuje kada TCU obrađuje podatke.

Dijagram toka:

aplikacija → TCU → LED → Secure Vault → povratak rezultata

Secure Vault-sigurno skladište podataka ili ključeva

Praktičan primer RPi / Jetson

- Python kod sa gpiozero omogućava kontrolu LED diode.
- Svaka funkcija TCU-a aktivira LED, demonstrirajući obradu podataka

PYNQ-Z2 platforma

- Kombinacija ARM CPU i FPGA pruža mogućnost HW simulacije.
- LED0 blink demonstrira aktivaciju TCU-a u FPGA logici.
- Dijagram toka: CPU → Mock TCU FPGA → LED0 → Secure Vault.
- Prednosti: hardverska izolacija ključeva, brže operacije i vizualna potvrda rada.

Praktičan primer PYNQ-Z2

Python kod koristi pynq.lib za kontrolu LED0.

Demonstracija enkripcije, dekripcije, potpisa i verifikacije

Scenario Full TCU Demo

Cilj: demonstrirati tok operacija TCU-a u celini.

LED vizualizacija pokazuje aktivaciju za svaku funkciju.

Namerno menjanje podataka ili potpisa pokazuje kako TCU detektuje greške.

Greške i testiranje

Dekripcija sa pogrešnim ključem vodi do nečitljivih podataka.

Verifikacija sa pogrešnim potpisom vraća False, što pokazuje integritet podataka.

Namerno menjanje podataka demonstrira sigurnosne mehanizme.

Zaključak

Objašnjena uloga TCU-a u zaštiti podataka.

Razlika softverski vs hardverski TCU.

LED vizualizacija pomaže u razumevanju toka operacija.

Primena: IoT uređaji, embedded sistemi, FPGA simulacije