|  |  |
| --- | --- |
| **LABORATORIUM BEZPIECZEŃSTWA SYSTEMÓW TELEINFORMATYCZNYCH** | |
| **Temat ćwiczenia:** Podpis Cyfrowy | |
| **Wykonał:** Bartosz Gabruk 157546 | **Data oddania:** 28.05.2025 r. |

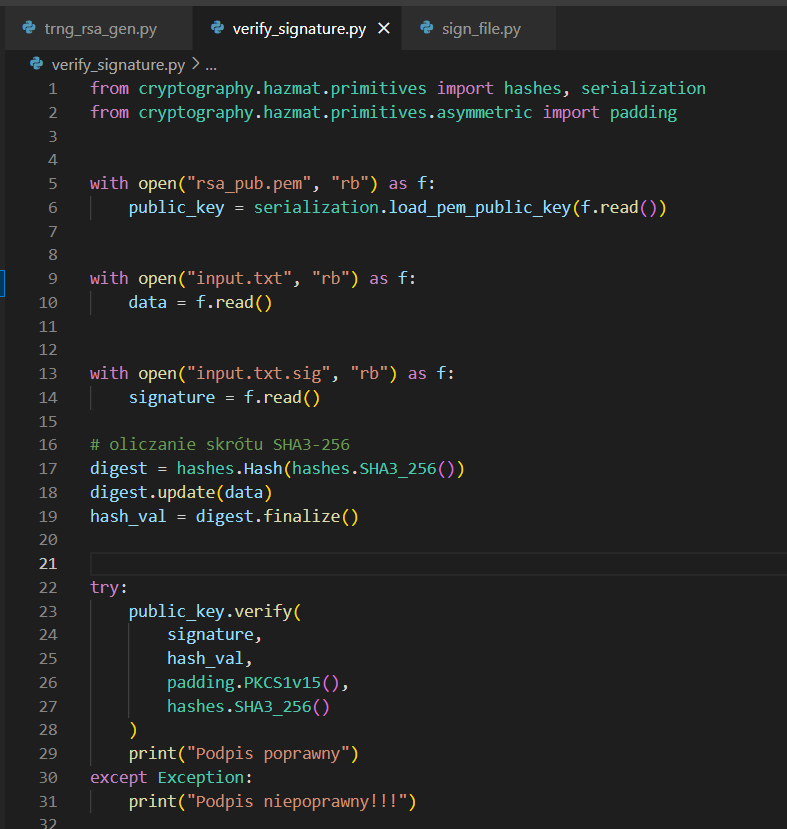
### **Wprowadzenie**

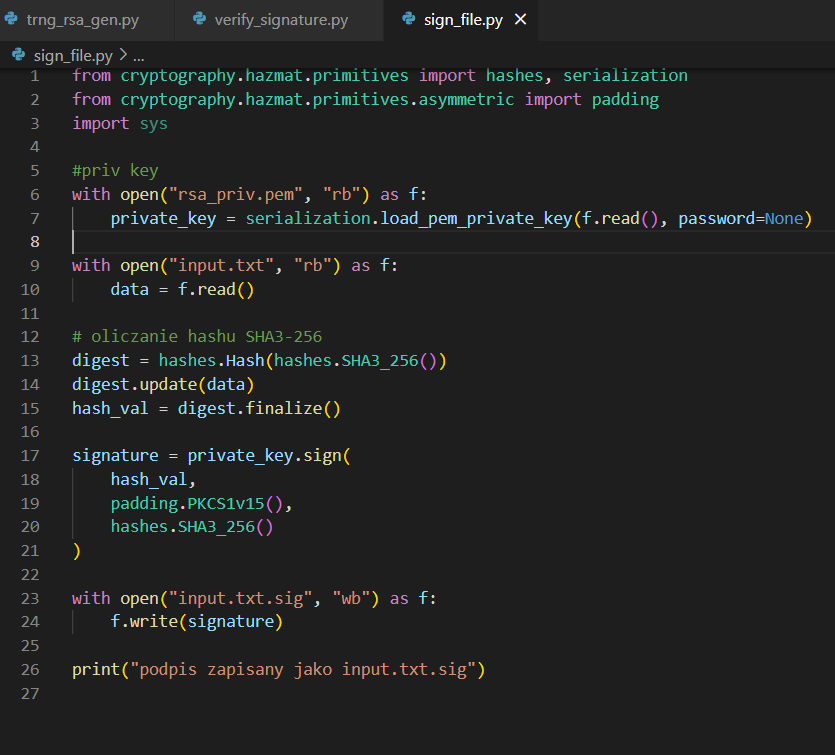
Celem niniejszego projektu jest implementacja systemu podpisu cyfrowego w języku Python, opartego na asymetrycznym algorytmie RSA oraz funkcji skrótu SHA3-256. Projekt demonstruje proces generowania kluczy przy użyciu zewnętrznego źródła losowości (TRNG z pliku aes.bin), podpisywania plików tekstowych oraz weryfikacji ich integralności i autentyczności. Rozwiązanie sprawdza się w scenariuszach, gdzie wymagane jest zapewnienie nienaruszalności i potwierdzenie autorstwa dokumentów.

### **Kody źródłowe**

Trng\_rsa\_get.py



Verify\_signature.py  


sign\_file.py  


### **Kody źródłowe**

Program rozpoczyna pracę od odczytania entropii z pliku aes.bin, co pozwala na deterministyczne zainicjowanie generatora losowości używanego przy tworzeniu pary kluczy RSA (2048 bitów). Wygenerowane klucze zapisywane są w formatach PEM jako rsa\_priv.pem oraz rsa\_pub.pem. Kolejnym krokiem jest wczytanie wybranego pliku tekstowego (input.txt), z którego obliczany jest skrót SHA3-256. Skrót ten jest następnie podpisywany kluczem prywatnym RSA, a wynikowy podpis zapisywany jest w pliku input.txt.sig. Podczas weryfikacji program ładuje klucz publiczny, ponownie oblicza hash z pliku, odszyfrowuje podpis i porównuje oba skróty, wyświetlając informację o poprawności podpisu lub wykryciu modyfikacji.