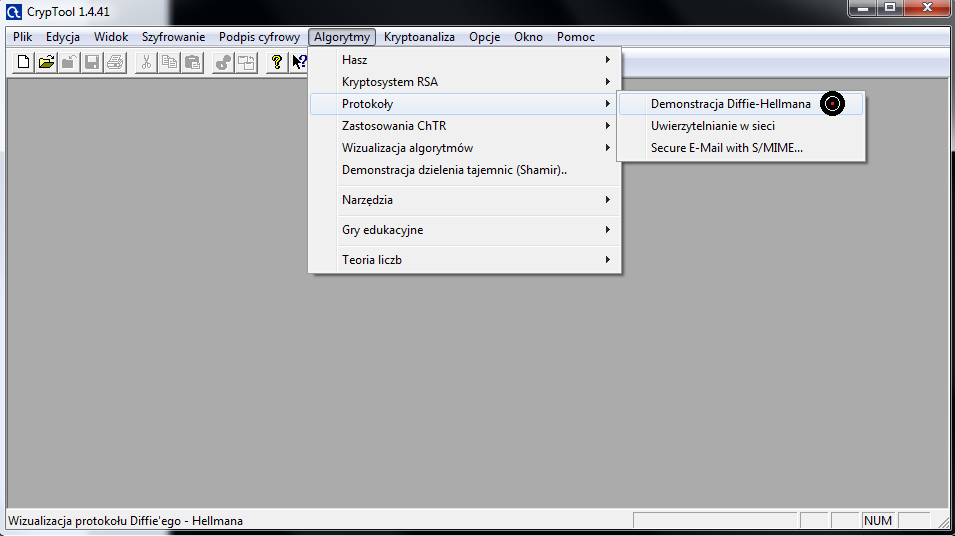
**CrypTool**

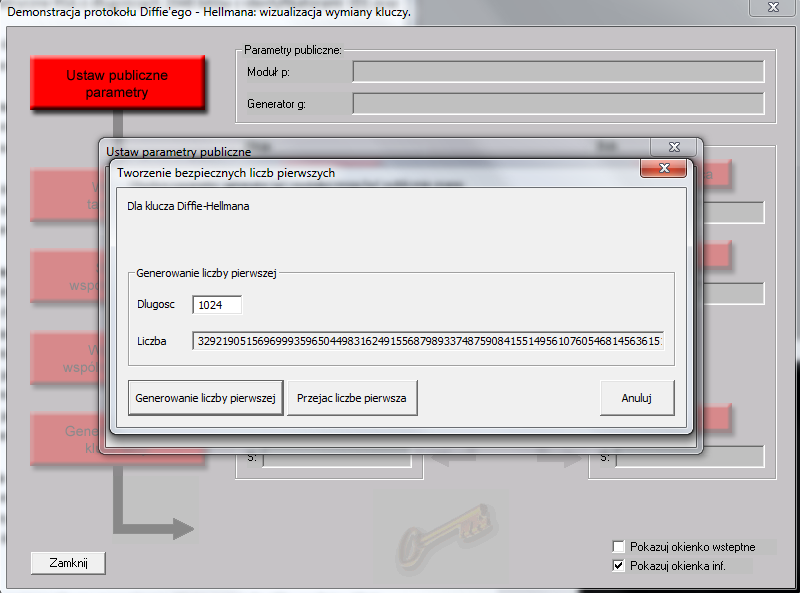
**Zadanie 1. Protokół Diffiego-Hellmana (DH)**

Zostało przeprowadzono obliczenia algorytmu DH dla 1024 bitowej liczby pierwszej p:

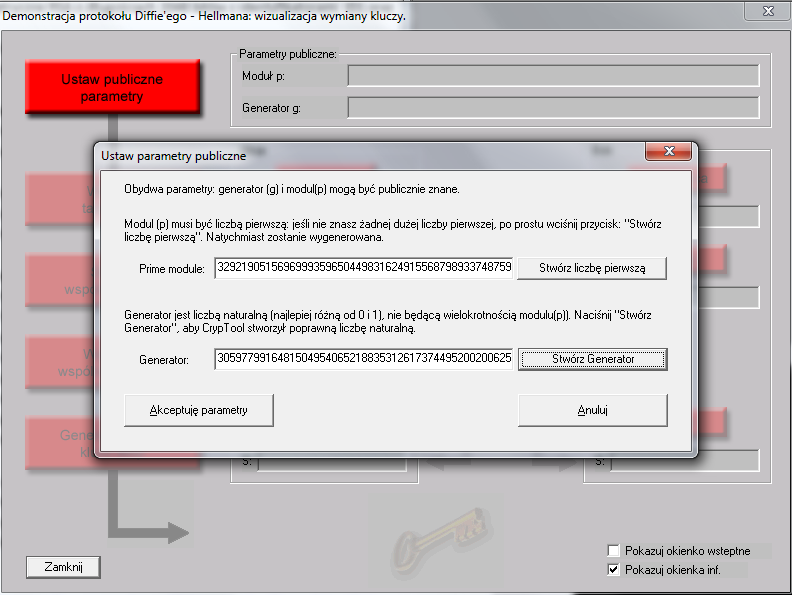
1. Otworzyłem okno Algorytmy->Protokoły->Demonstracja Diffie-Hellmana
2. Najpierw wygenerowałem liczbę pierwszą na podstawie jej rozmiaru
3. Na jej podstawie stworzyłem generator
4. Po czym akceptowałem parametry i przeszedłem do tworzenia sekretów Alicji i Boba (zostały wygenerowane losowo)
5. Dalej zachodzi tworzenie i wymiana wspólnych kluczy na podstawie określonych funkcji
6. Wynik obliczeń widać na „Rysunek 6”

****

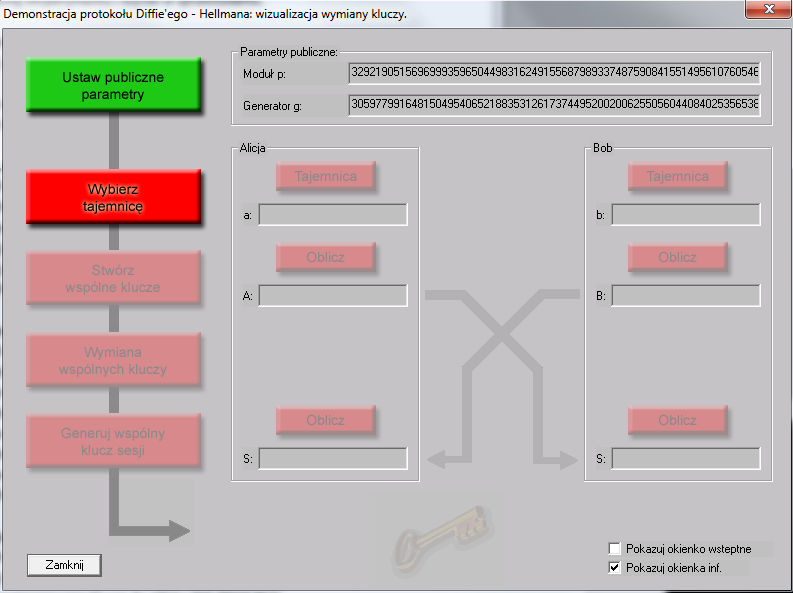
Rysunek

****

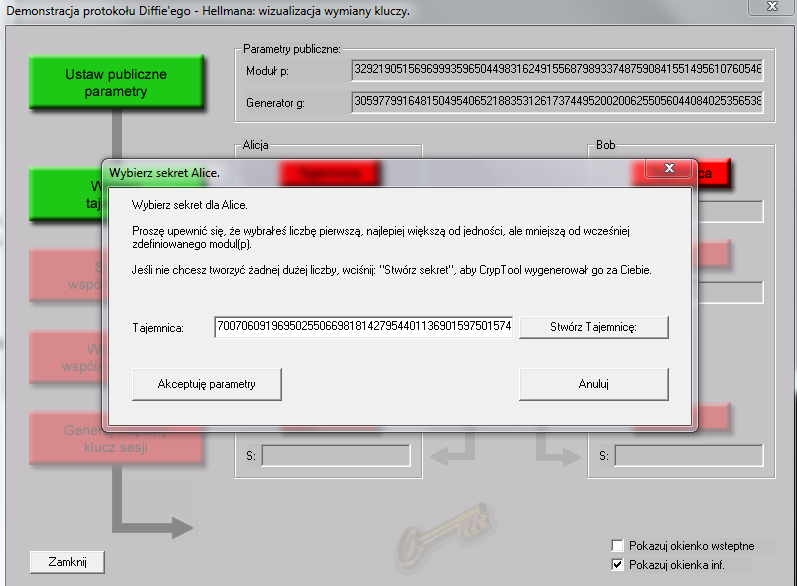
Rysunek

****

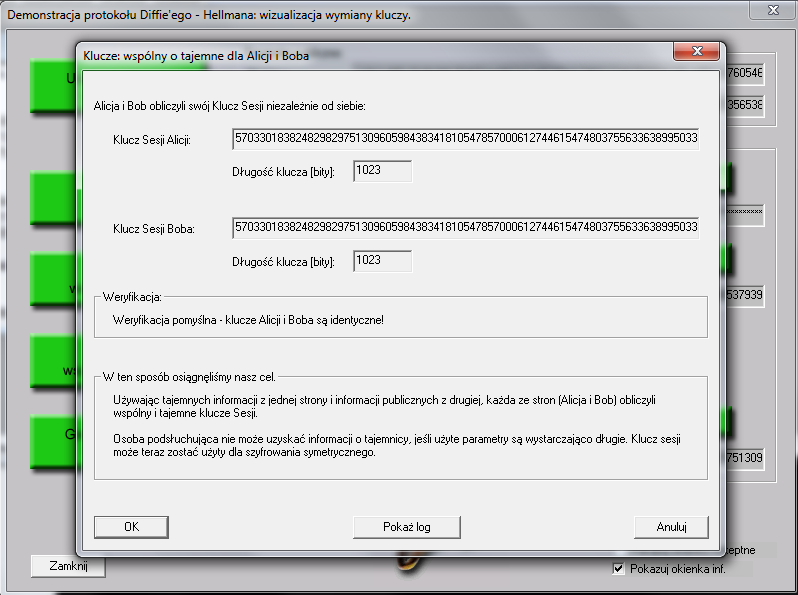
Rysunek

****

Rysunek

****

Rysunek

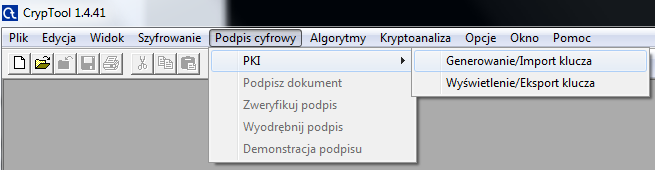
****

Rysunek

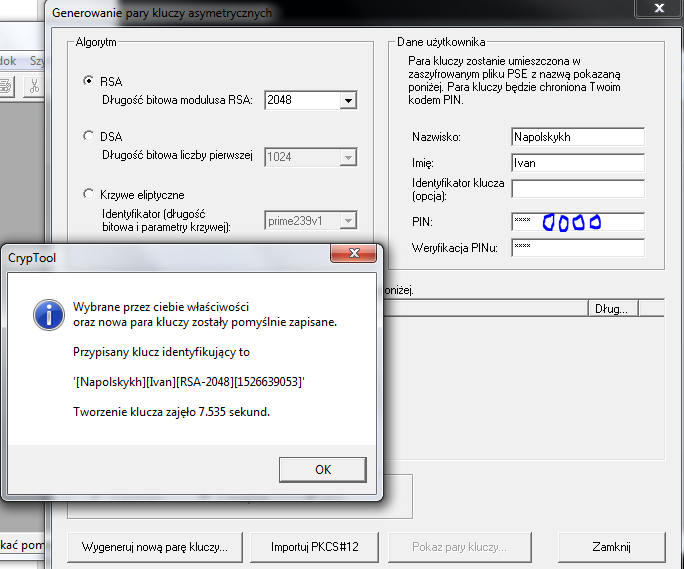
**Zadanie 2. Generacja kluczy RSA.**

1. Wygenerowałem 2 klucze asymetryczne RSA o długościach 2048 bitów z identyfikatorami ID1 oraz ID2. (Rysunek 7 i 8)
2. Eksportowałem wygenerowane klucze do plików PSE (PKCS #12). (Rysunek 9)
3. Dostałem w wyniku dwa pliki o nazwach:

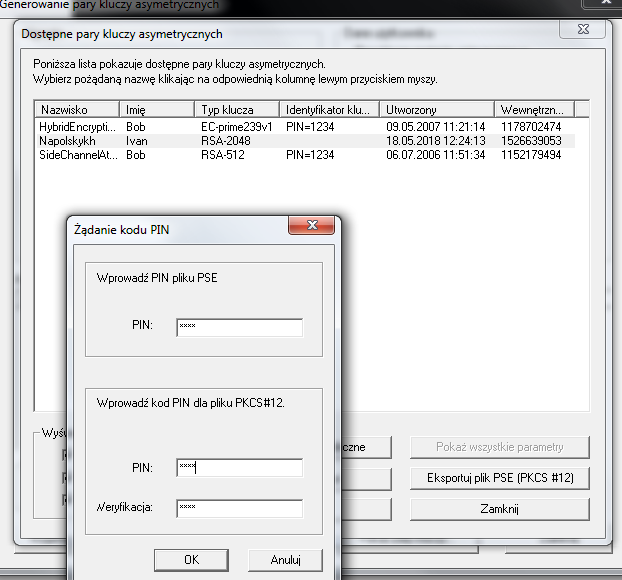
* [Napolskykh][Ivan][RSA-2048][1526639053][D1].p12
* [Napolskykh][Ivan][RSA-2048][1526639432][D2].p12

****

Rysunek

****

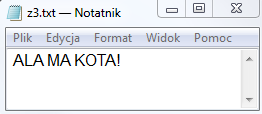
Rysunek

****

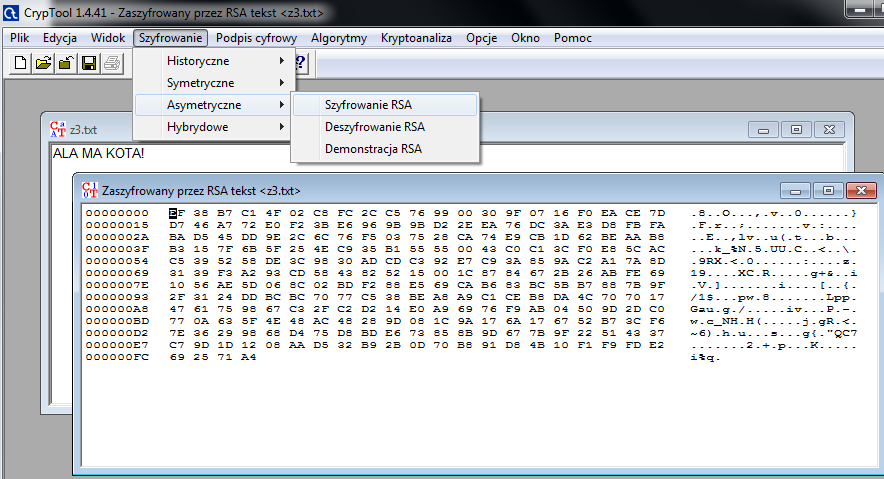
Rysunek

**Zadanie 3. Szyfrowanie/deszyfrowanie RSA.**

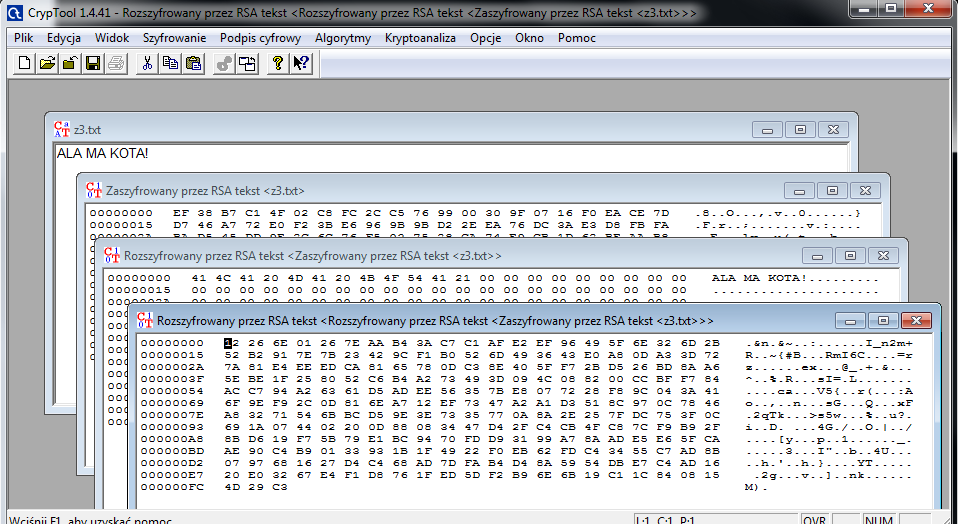
1. Utwórzyłem plik z tekstem jawnym do zaszyfrowania (Rysunek 10)
2. Zaszyfrowałem plik z tekstem jawnym za pomocą klucza ID1 i zapisałem wynik w pliku z tekstem zaszyfrowanym. (Rysunek 11)
3. Zdeszyfrowałem plik z tekstem zaszyfrowanym za pomocą klucza ID1 i porównałem z tekstem jawnym. Są jednakowe (Rysunek 12)
4. Zdeszyfrowałem plik z tekstem zaszyfrowanym za pomocą klucza ID2 i porównałem z tekstem jawnym. Zupełnie się różnią (Rysunek 12)



Rysunek 10



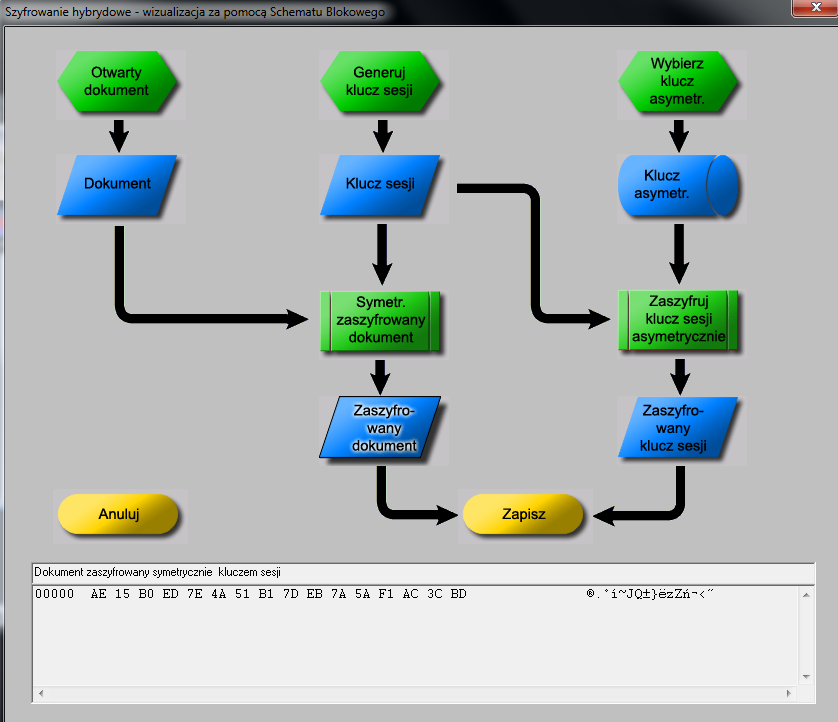
Rysunek



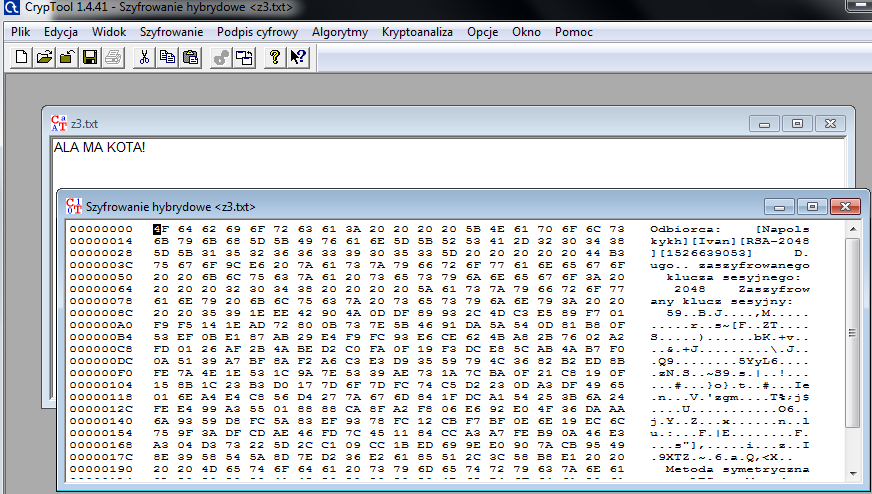
Rysunek

**Zadanie 4. Szyfrowanie/deszyfrowanie hybrydowe RSA-AES.**

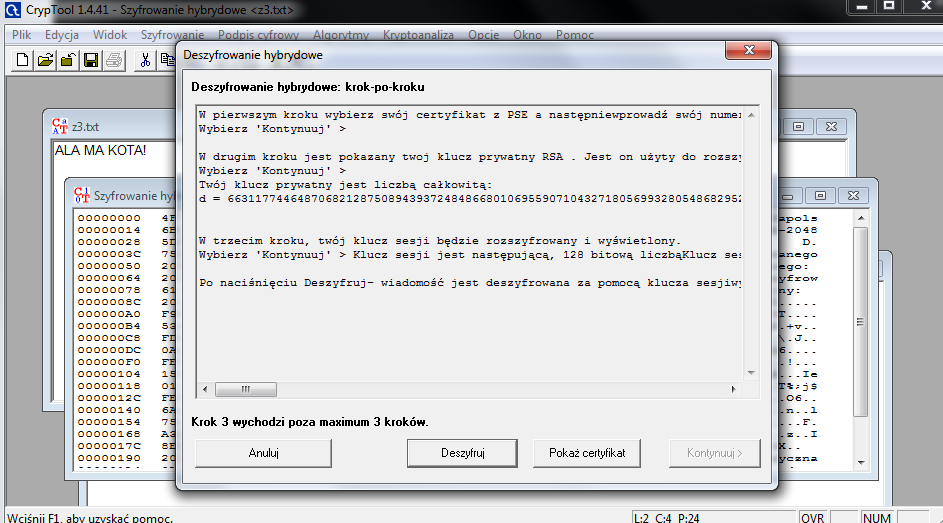
1. Użyłem poprzednio stworzony plik z tekstem jawnym do zaszyfrowania.
2. Przeprowadziłem szyfrowanie hybrydowe RSA-AES przy wykorzystaniu klucza asymetrycznego ID1 i zapisałem wynik z tekstem zaszyfrowanym do pliku. (Rysunek 13 i 14)
3. Przeprowadziłem deszyfrowanie hybrydowe RSA-AES przy wykorzystaniu klucza asymetrycznego ID1 i porównałem z tekstem jawnym. Są jednakowe (Rysunek 15 i 16)
4. Przeprowadziłem deszyfrowanie hybrydowe RSA-AES przy wykorzystaniu klucza asymetrycznego ID2 i porównałem z tekstem jawnym. Nie dało się dokonać tych działań, bo pojawił się komunikat o błędzie podczas sprawdzeniu Certyfikatu (Rysunek 17)



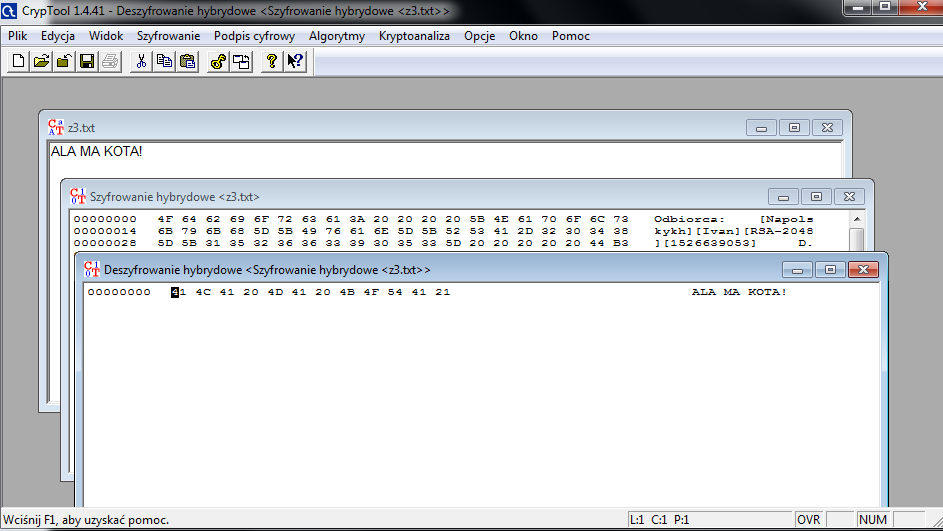
Rysunek



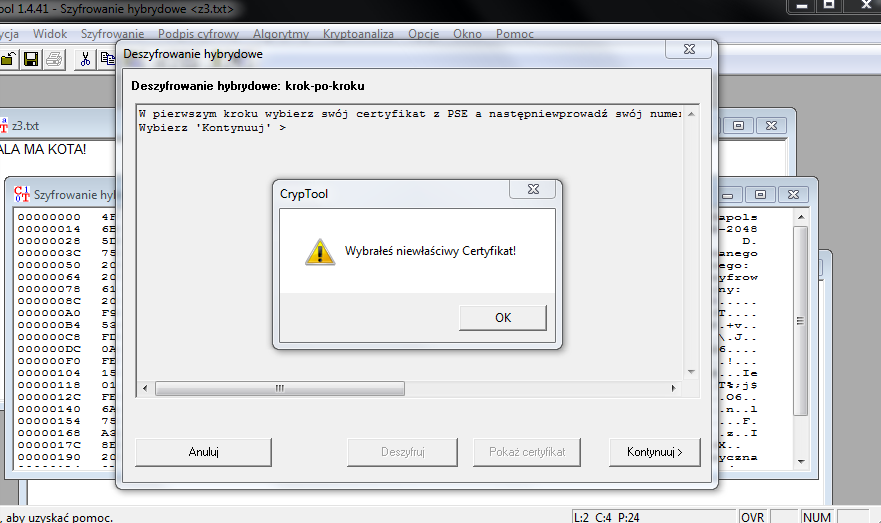
Rysunek



Rysunek



Rysunek



Rysunek

**Zadanie 5. Generacja/weryfikacja podpisu cyfrowego RSA.**

1. Utworzyłem plik z tekstem jawnym do podpisania. (Rysunek 18)
2. Wygenerowałem klucze asymetryczne RSA o długościach 2048 bitów dla Alicji i Boba.
3. Eksportowałem wygenerowane klucze do plików PSE (PKCS #12):

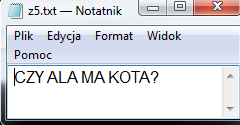
* [Napolskykh][Ivan][RSA-2048][1526639960][Klucz Alicji (0000)].p12
* [Napolskykh][Ivan][RSA-2048][1526639991][Klucz Boba (0000)].p12

**Zadanie 5a.**

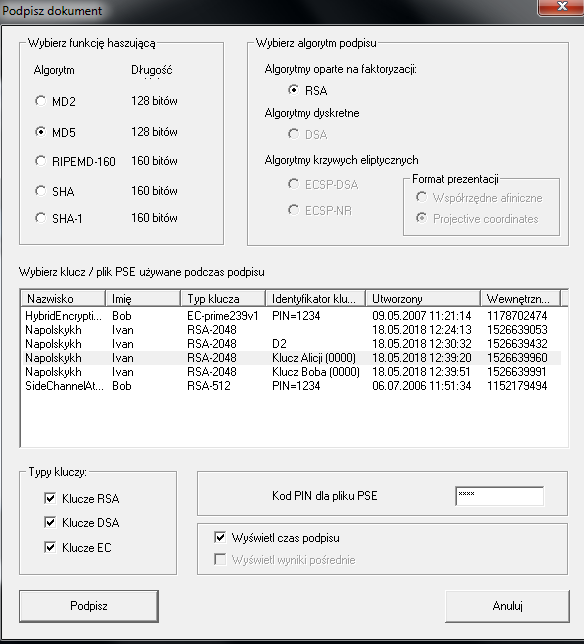
1. Podpisałem utworzony plik za pomocą klucza prywatnego Alicji i zapisałem do pliku podpisu Alicji. (Rysunek 19 i 20)
2. Zweryfikowałem podpis Alicji za pomocą klucza publicznego Alicji. (Rysunek 21 i 22)
3. Zweryfikowałem podpis Alicji za pomocą klucza publicznego Boba. (Rysunek 21 i 23)

**Zadanie 5b.**

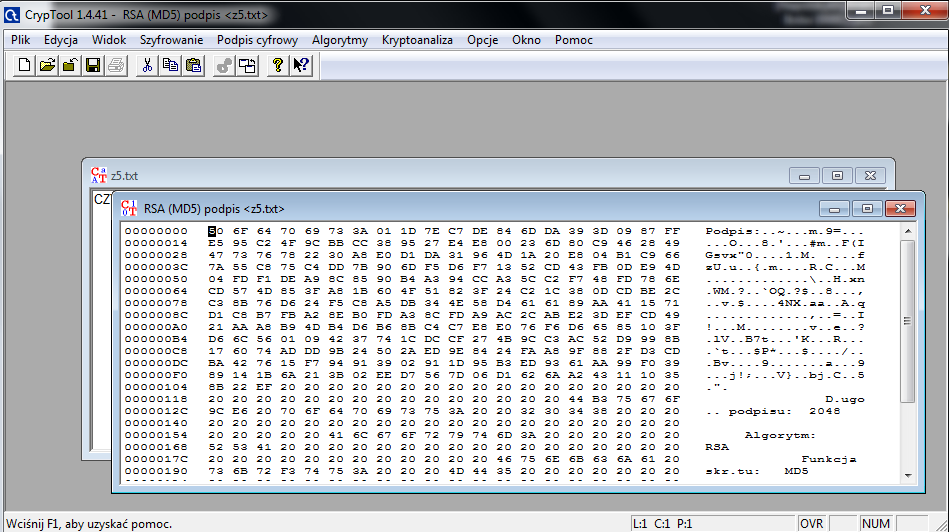
1. Zmodyfikowałem podpisany tekst jawny w pliku podpisu Alicji (Rysunek 24)
2. Zweryfikowałem podpis Alicji za pomocą klucza publicznego Alicji. (Rysunek 25)



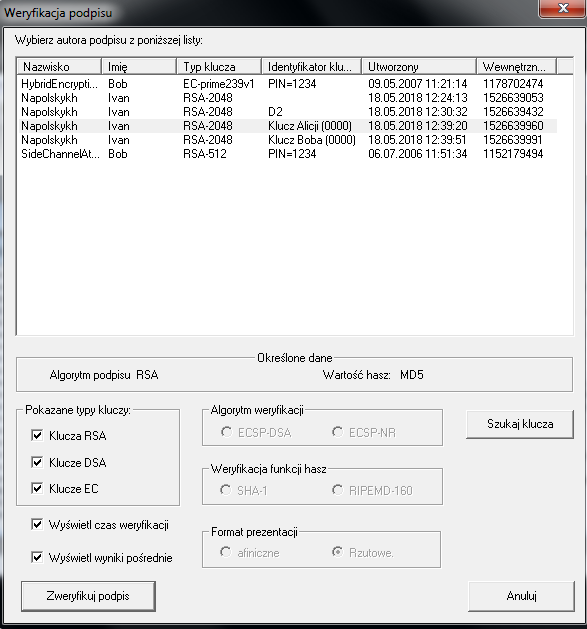
Rysunek



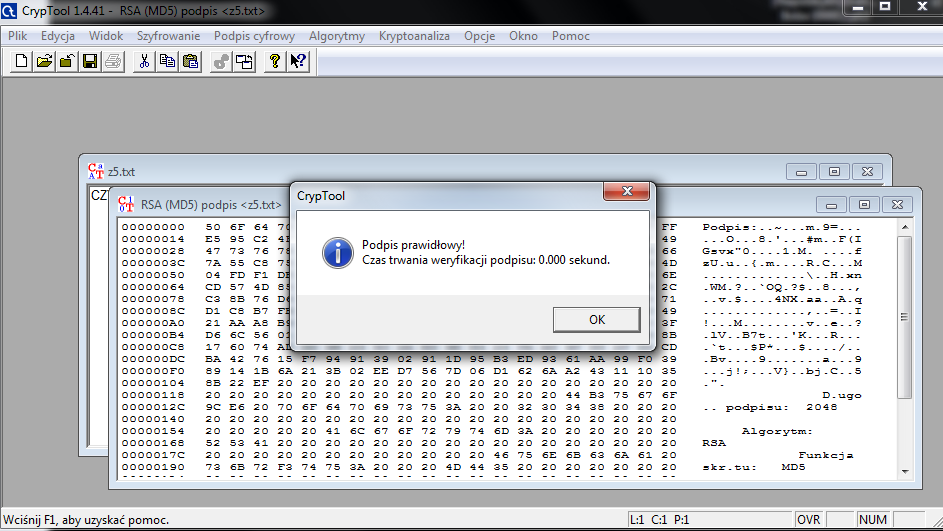
Rysunek



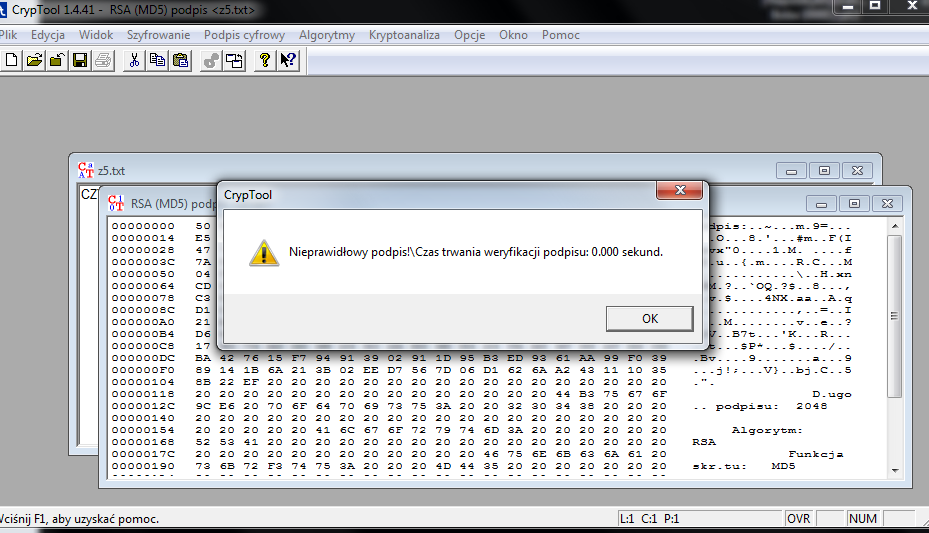
Rysunek



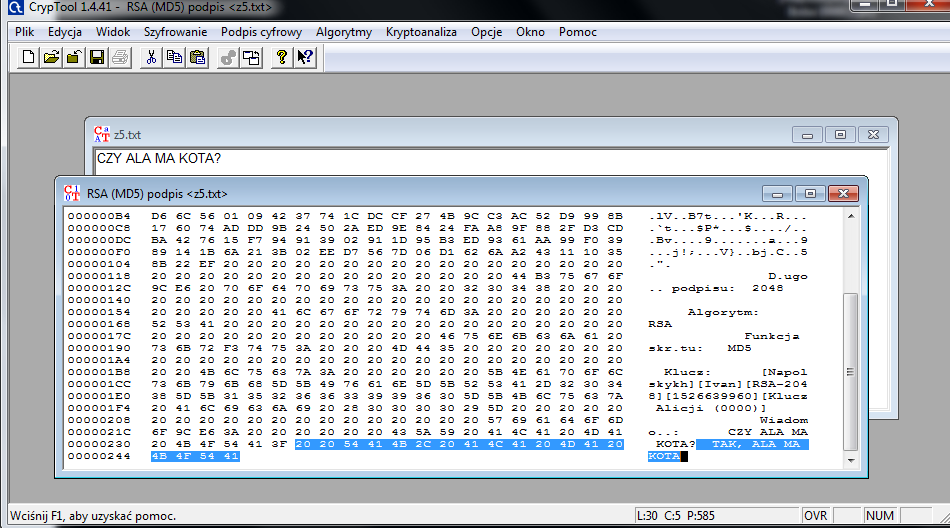
Rysunek



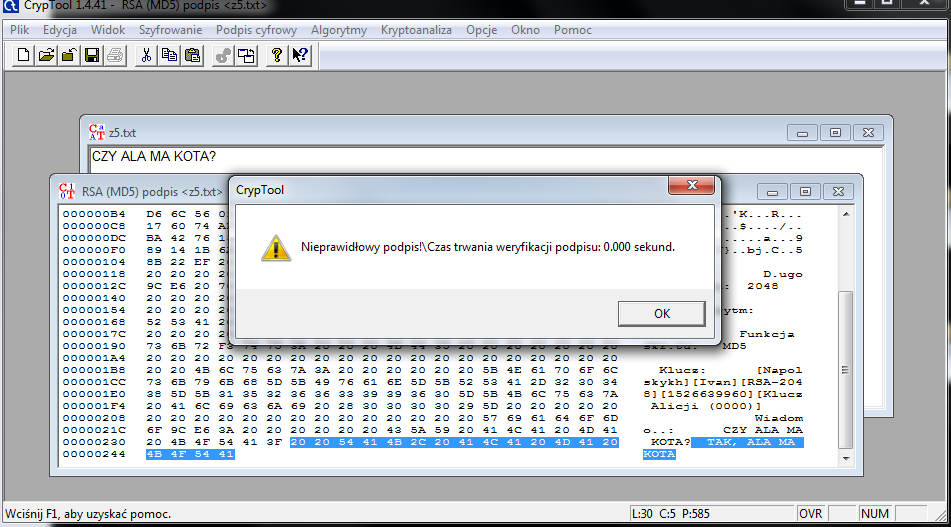
Rysunek



Rysunek



Rysunek



Rysunek

**Sage**

Do zrobienia dla każdego:

1. zadanie 1 lub zadanie 2 oraz
2. zadanie 3.

**Zadanie 1.**

Implementacja algorytmów szyfrowania/deszyfrowania RSA w Sage:

1. napisz funkcję generującą parę kluczy RSA,
2. napisz funkcję szyfrującą RSA,
3. napisz funkcję deszyfrującą RSA,
4. sprawdź poprawność implementacji z pkt. a-c na przykładowych danych,
5. wnioski opisz w sprawozdaniu.

**Zadanie 2.**

Implementacja algorytmów generacji/weryfikacji podpisu cyfrowego w Sage:

1. napisz funkcję generującą parę kluczy RSA,
2. napisz funkcję generującą podpis cyfrowy dla podanej liczby całkowitej (skrót z wiadomości) i podanego klucza prywatnego,
3. napisz funkcję weryfikującą podpis cyfrowy dla podanej liczby całkowitej (skrót z wiadomości) oraz podanego klucza publicznego,
4. przetestuj funkcje utworzone w pkt. a) i b) w celu sprawdzenia, czy istotnie poprawne podpisy weryfikowane są poprawnie, a niepoprawne kwestionowane,
5. wnioski opisz w sprawozdaniu.

**Zadanie 3. dla wszystkich**

Zaimplementuj protokół Diffiego-Hellmana w Sage, wnioski opisz w sprawozdaniu