Stanisław Ciszkiewicz

Jakub Kusznier

**Projekt 1 Cryptool**

Spis treści

**Zadanie 12**

**Zadanie 22**

**Zadanie 33**

**Zadanie 44**

1. ECB4
2. CBC4
3. Porównanie szyfrogramów5

**Zadanie 55**

**Zadanie 66**

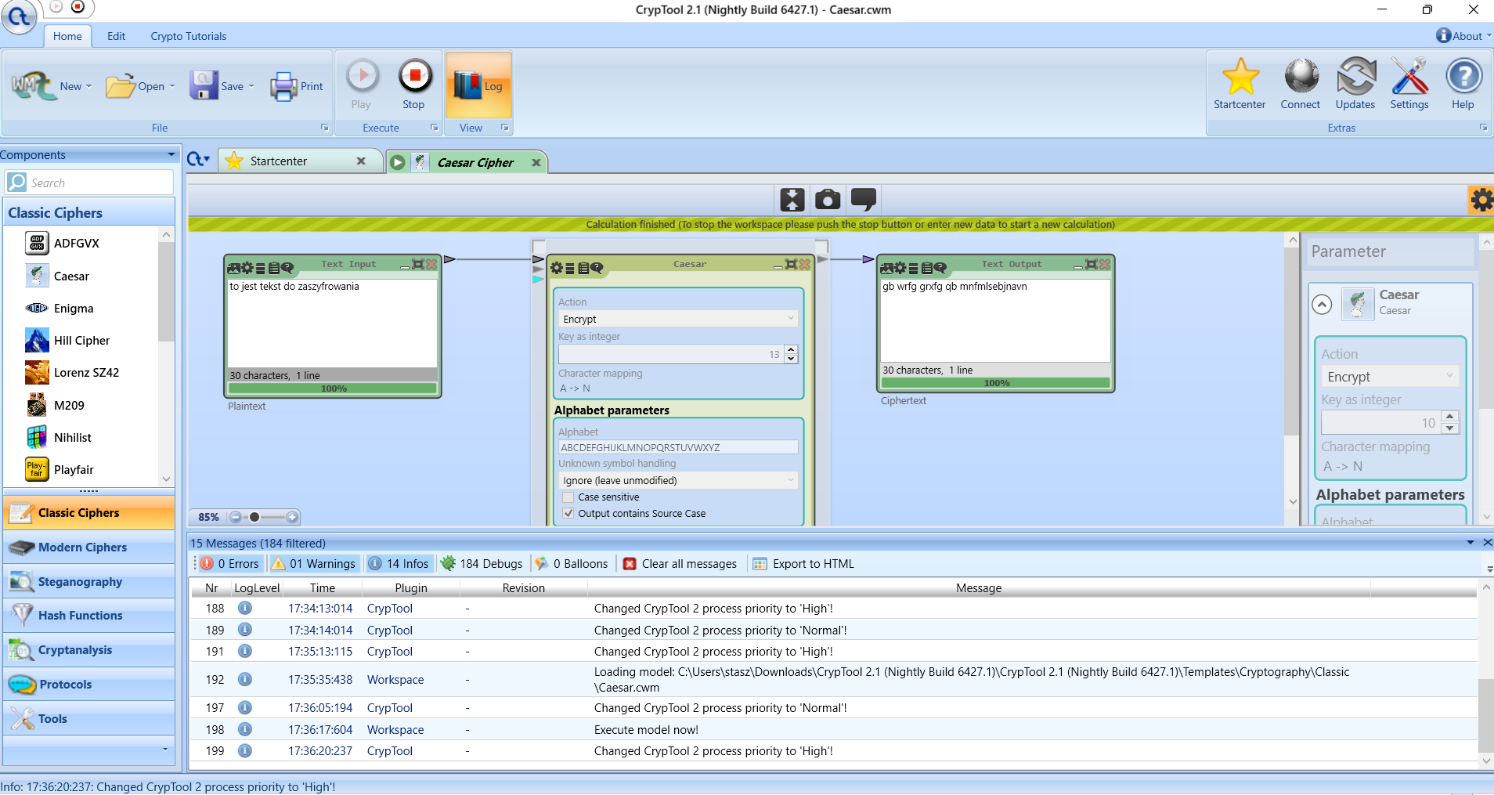
**Zadanie 77**

**Wnioski7**

**Zadanie 1**

**Zaszyfrować tekst „to jest tekst do zaszyfrowania” za pomocą szyfru Cezara z przesunięciem równym 13.**

Z zakładki Templates wybieramy narzędzie Ceasar Cipher **(**ścieżka: Templates/Cryptography/Classical).

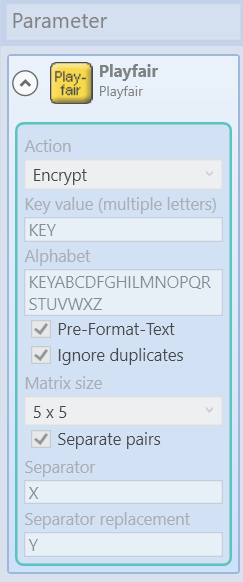
Następnie w polu *Text Input* wklejamy wiadomość do zaszyfrowania („to jest tekst do zaszyfrowania”), natomiast w oknie *Key as Integer* ustawiamy zadaną wartość 13. Aby zaobserwować wynik operacji uruchamiamy narzędzie przyciskiem *Play* (zaszyfrowana wiadomość ukazuje się nam w oknie *Text Output*).

**Wynik:** gb wrfg grxfg qb mnfmlsebjnavn

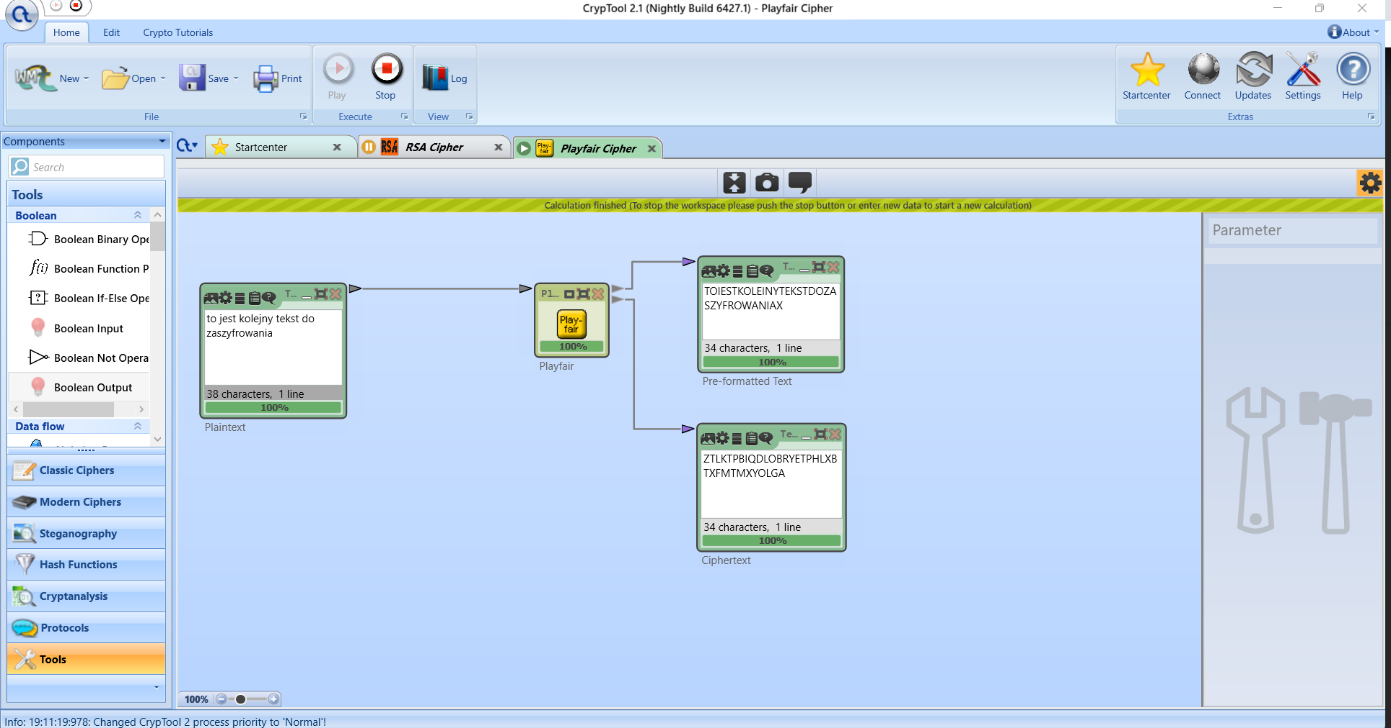
**Zadanie 2**

**Zaszyfrować tekst „to jest kolejny tekst do zaszyfrowania” za pomocą szyfru Playfair z kluczem „key”.**

Z zakładki *Templates* wybieramy narzędzie *Playfair Cipher*   
(ścieżka: Templates/Cryptography/Classical).

Skopiowany z polecenia tekst wklejamy w polu Plaintext. Teraz pozostaje nam zmienienie klucza na „key” – naciskamy na żołtą ikonkę Playfair i w polu Key value wpisujemy podany w zadaniu klucza.

Na koniec po naciśnięciu przycisku Play odczytujemy zaszyfrowaną wiadomość z okna CipherText.

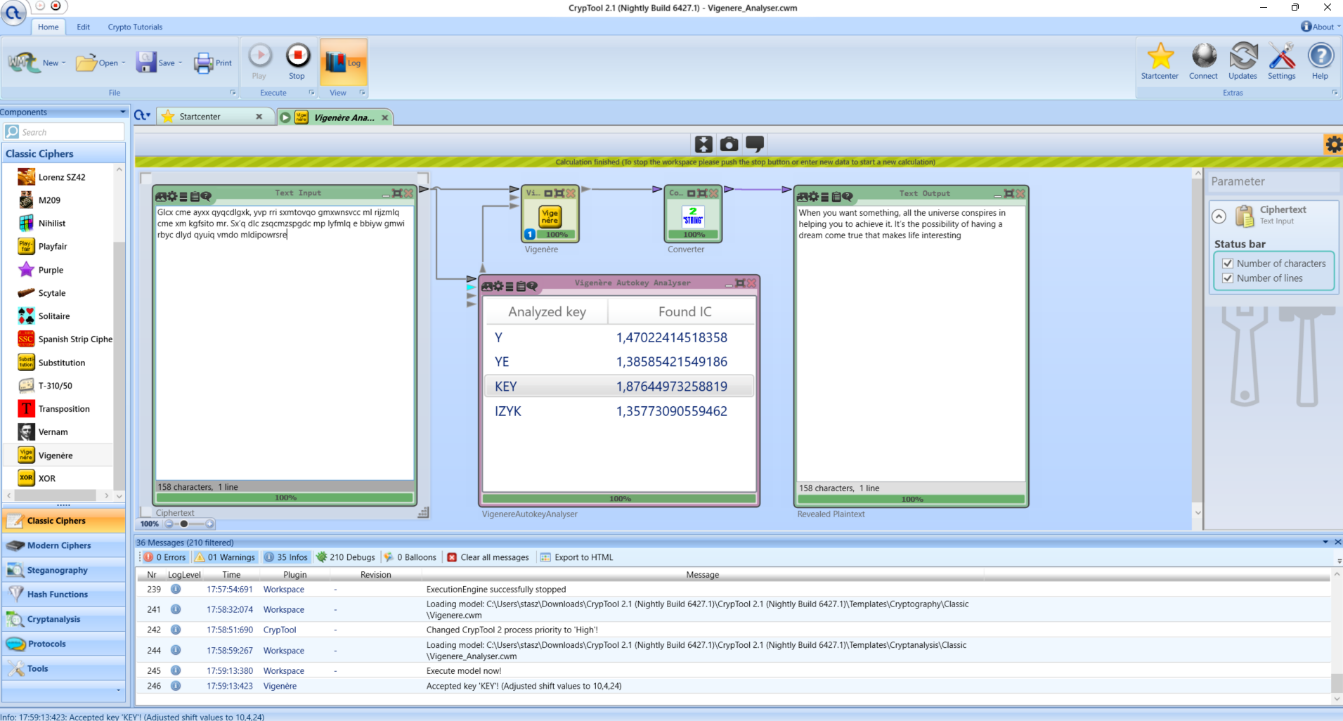
****

**Wynik:** ZTLKTPBIQDLOBRYETPHLXBTXFMTMXYOLGA

**Zadanie 3**

**Dokonać kryptoanalizy tekstu zaszyfrowanego szyfrem Vigenère: „Glcx cme ayxx qyqcdlgxk, yvp rri sxmtovqo gmxwnsvcc ml rijzmlq cme xm kgfsito mr. Sx’q dlc zsqcmzspgdc mp lyfmlq e bbiyw gmwi rbyc dlyd qyuiq vmdo mldipowrsre.”**

Z zakładki *Templates* wybieramy narzędzie *Vigenere Analysis*   
(ścieżka: Templates/Cryptanalysis/Classical).

W oknie *Text Input* wklejamy tekst, który będziemy chcieli odszyfrować. Po uruchomieniu programu dostajemy wynik w oknie *Text Output*. Co więcej, w środkowej części ekranu możemy zauważyć klucz, jakim zakodowana została wiadomość – KEY (narzędzie szukało odpowiedniego klucza).

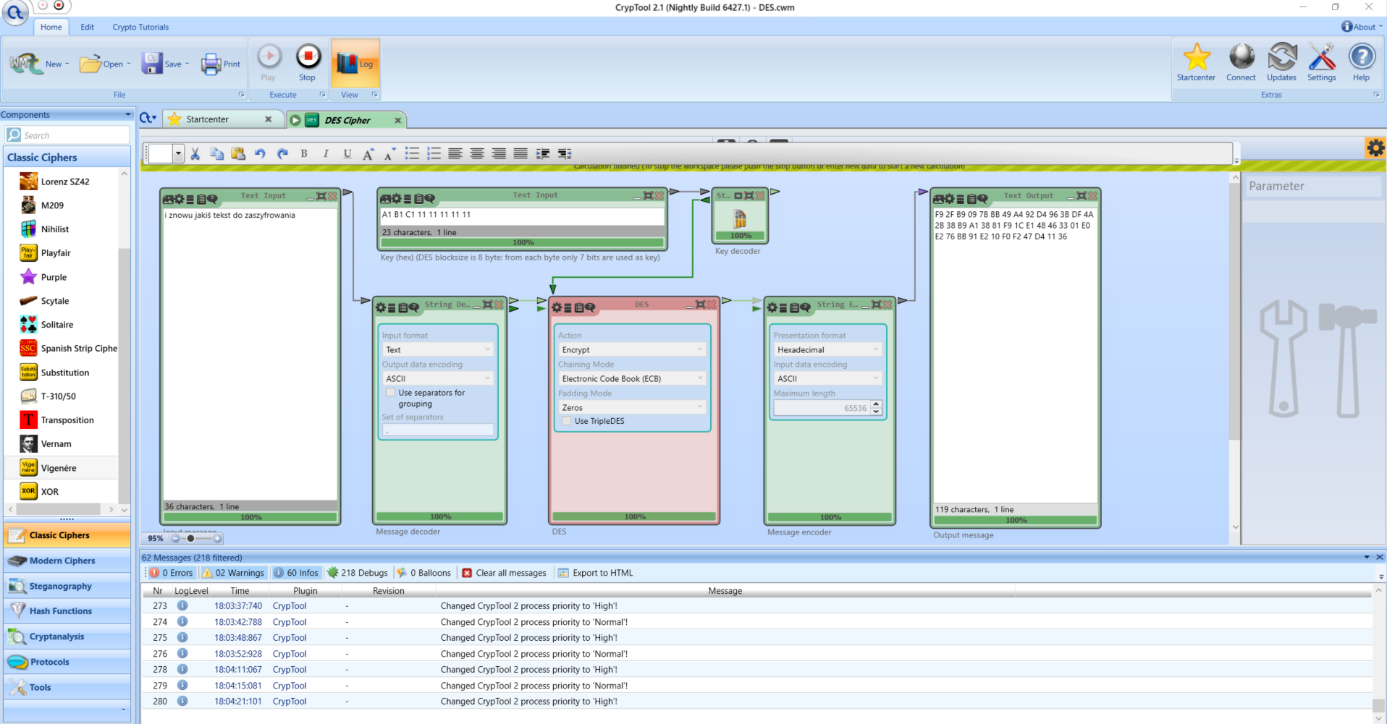
**Wynik:** When you want something, all the universe conspires in helping you to achieve it. It’s the possibility of having a dream come true that makes life interesting.

**Zadanie 4**

**Zaszyfrować tekst „i znowu jakiś tekst do zaszyfrowania” używając DES w trybie ECB i CBC. Porównać szyfrogramy.**

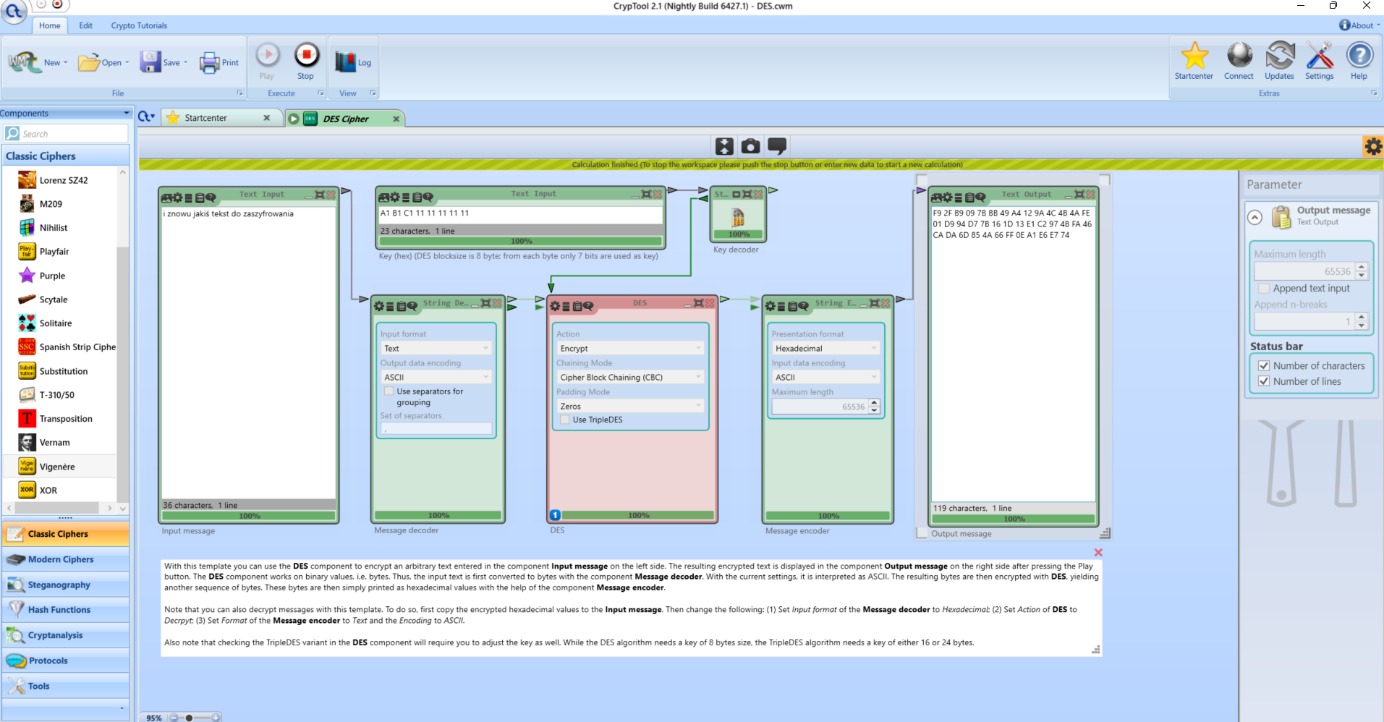
Z zakładki *Templates* wybieramy narzędzie *DNS Cipher*   
(ścieżka: Templates/Cryptography/Modern/Symmetric).

1. **ECB**

W polu *Text Input* wklejamy tekst „i znowu jakiś tekst do zaszyfrowania”, a następnie w sekcji DES ustawiamy *Chaining Mode* na *Electronic Code Book (ECB)*. Po uruchomieniu programu w oknie *Text Output* ukazuje się zaszyfrowana wiadomość.

**Wynik:** F9 2F B9 09 7B BB 49 A4 92 D4 96 3B DF 4A 2B 38 B9 A1 38 81 F9 1C E1 48 46 33 01 E0 E2 76 BB 91 E2 10 F0 F2 47 D4 11 36

1. **CBC**

W tym przypadku robimy dokładnie to samo co w poprzednim podpunkcie, jednak zmieniamy *Chaining Mode* na *Cipher Block Chaining (CBC)*. Po uruchomieniu programu w oknie *Text Output* ukazuje się zaszyfrowana wiadomość.

**Wynik:** F9 2F B9 09 7B BB 49 A4 12 9A 4C 4B 4A FE 01 D9 94 D7 7B 16 1D 13 E1 C2 97 4B FA 46 CA DA 6D 85 4A 66 FF 0E A1 E6 E7 74

1. **Porównanie szyfrogramów**

**Wynik ECB:** F9 2F B9 09 7B BB 49 A4 92 D4 96 3B DF 4A 2B 38 B9 A1 38 81 F9 1C E1 48 46 33 01 E0 E2 76 BB 91 E2 10 F0 F2 47 D4 11 36

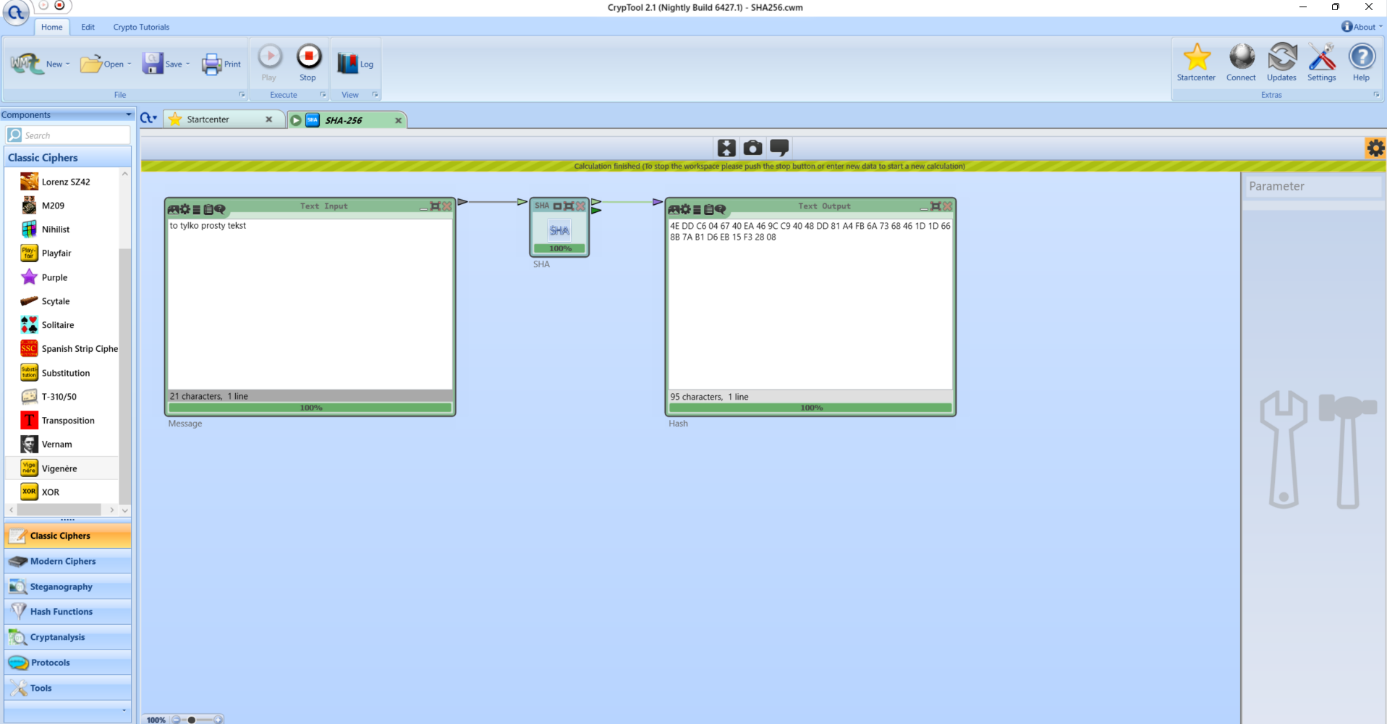
**Wynik CBC:** F9 2F B9 09 7B BB 49 A4 12 9A 4C 4B 4A FE 01 D9 94 D7 7B 16 1D 13 E1 C2 97 4B FA 46 CA DA 6D 85 4A 66 FF 0E A1 E6 E7 74

Zauważamy, że szyfrogramy mają te same znaki tylko dla pierwszych 8 bajtów, reszta zakodowanej wiadomości jest całkowicie inna. Wynika to ze specyfiki działania obydwu szyfrów: w obydwu przypadkach dzielimy szyfrowany tekst na bloki (po 8 bajtów), jednak dla ECB szyfrowanie każdego bloku odbywa się niezależnie, podczas gdy w CBC szyfrowanie każdego kolejnego bloku zależy od poprzedniego.

**Zadanie 5**

**Obliczyć wartość funkcji skrótu SHA-256 dla tekstu „to tylko prosty tekst”**

Z zakładki *Templates* wybieramy narzędzie *SHA-256* (ścieżka: Templates/Hash Functions).



W oknie *Text Input* wklejamy tekst, dla którego chcemy obliczyć wartość funkcji skrótu SHA-256. Po uruchomieniu programu, w oknie *Text Output* otrzymujemy wartość funkcji skrótu.

**Wynik:** 4E DD C6 04 67 40 EA 46 9C C9 40 48 DD 81 A4 FB 6A 73 68 46 1D 1D 66 8B 7A B1 D6 EB 15 F3 28 08

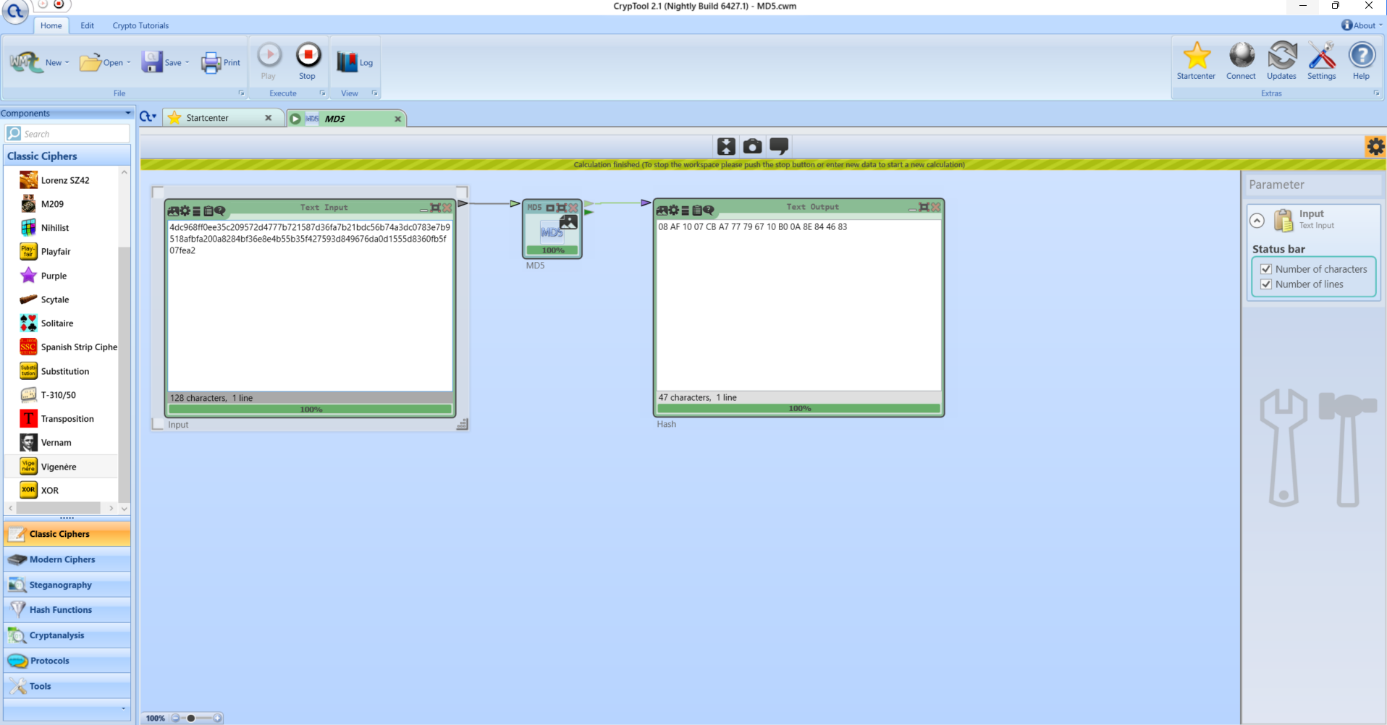
**Zadanie 6**

**Obliczyć skrót MD5 dla dwóch ciągów znakowych zapisanych heksadecymalnie i skomentować wynik:**

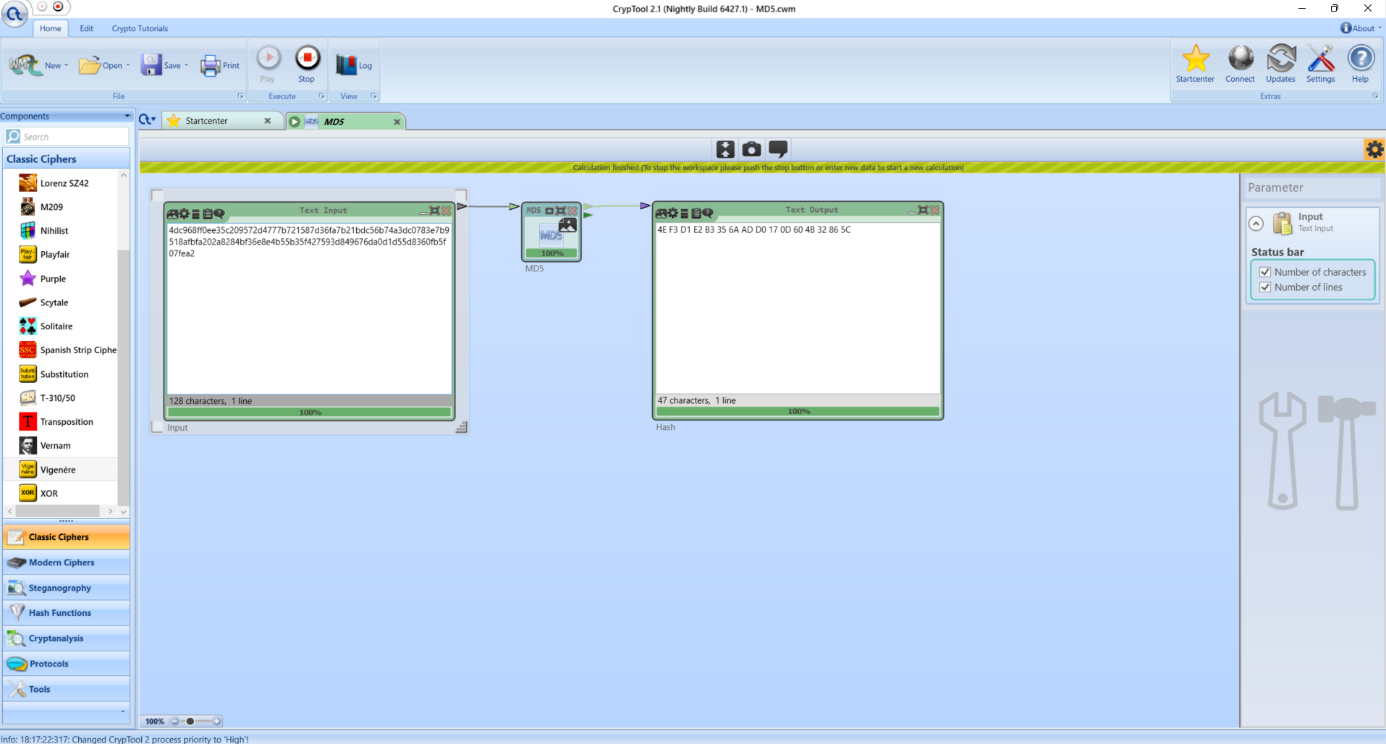
**•4dc968ff0ee35c209572d4777b721587d36fa7b21bdc56b74a3dc0783e7b9518afbfa200a8284bf36e8e4b55b35f427593d849676da0d1555d8360fb5f07fea2**

**•4dc968ff0ee35c209572d4777b721587d36fa7b21bdc56b74a3dc0783e7b9518afbfa202a8284bf36e8e4b55b35f427593d849676da0d1d55d8360fb5f07fea2**

Z zakładki *Templates* wybieramy narzędzie *MD5* (ścieżka: Templates/Hash Functions ).

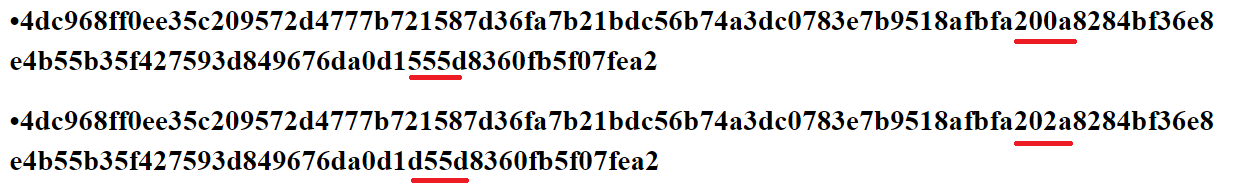


**Wynik:** 08 AF 10 07 CB A7 77 79 67 10 B0 0A 8E 84 46 83



**Wynik:** 4E F3 D1 E2 B3 35 6A AD D0 17 0D 60 4B 32 86 5C

W obu przypadkach dany ciąg znaków wklejamy do pola *Text Input.* Następnie uruchamiamy program i otrzymujemy wyniki.



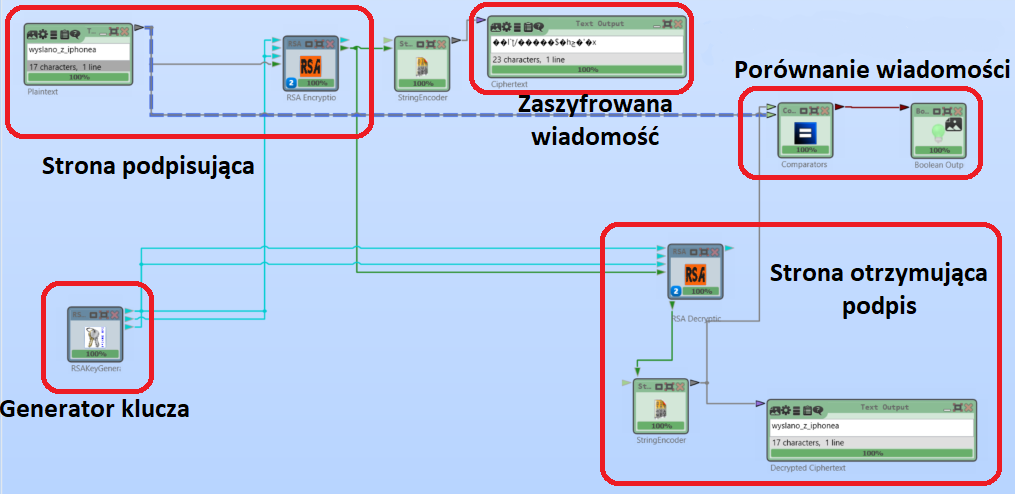
Ciągi znaków różnią się tylko w dwóch miejscach (zaznaczone na czerwono powyżej), jednak różnica dla funkcji skrótu MD5 jest znacząca. Jest to oczekiwany wynik, który pokazuje, że nawet ledwo różniące się, długie ciągi znaków dają zupełnie inne skróty.

**Zadanie 7**

**Stworzyć schemat podpisu wiadomości RSA. Wybrać bezpieczne parametry podpisu. Zweryfikować działanie podpisu na wybranej wiadomości.**

Z zakładki *Templates* wybieramy narzędzie *RSA Cipher*

(ścieżka: Templates/Cryptography/Modern/Asymetric).



Wykorzystujemy gotowy schemat *RSA* Cipher, do którego dodajmey *Comparators* oraz *Boolean Output.* Wejście *Comparators* łączymy z tekstem strony podpisującej oraz z tekstem strony otrzymującej podpis, a na jego wyjściu dodajemy *Boolean Output* (lampka świecąca się w kolorze zielonym świadczy o zgodności tekstu szyfrowanego z rozszyfrowanym.

Strona podpisująca przekazuje tekst, który chcemy zaszyfrować do *RSA Encryption*, które wykorzystując klucz prywatny szyfruje wiadomość, która jest widoczna w oknie *Ciphertext*. *RSA Decryption* korzystając z klucza publicznego rozszyfrowuje wiadomość, którą widzimy w oknie *Decrypted Ciphertext.*

**Wnioski**

Powyższe zadania nauczyły nas wiele na temat szyfrów i funkcji skrótu. Aplikacja Cryptool w łatwy i czytelny sposób przedstawiła postawy szyfrowania (min. szyfr Cezara, Playfair, DES), co pozwoliło na przyswojenie i zrozumienie tematu kryptografii. Ponadto, Cryptool posiada wiele innych narzędzi, w które nie zdążyliśmy się zagłębić podczas wykonywania zadań. Uważamy, że znajomość Cryptoola może okazać się pomocne w przyszłości (chociażby odszyfrowanie funkcji skrótu).