**Salsa20**

Salsa20 – symetryczny szyfr strumieniowy.

**Opis szyfru**

Salsa20 jest szyfrem strumieniowym, który operuje na blokach danych o wielkości 64 bajtów.

Dla każdego 64-bajtowego bloku danych, algorytm wywołuje funkcję rozszerzającą Salsa20. Danymi wejściowymi do funkcji są: sekretny klucz (który może mieć 32 lub 16 bajtów) oraz 8-bajtowy nonce, połączony dodatkowo z numerem aktualnie szyfrowanego bloku.

Wartości numerów bloków zmieniają się od 0 do 264-1 (jest więc zapisany na 8 bajtach). Każde wywołanie funkcji rozszerzającej zwiększa wartość numeru bloku o jeden.

Rdzeniem algorytmu Salsa20 jest funkcja haszująca, która otrzymuje 64 bajty z funkcji rozszerzającej, następnie miesza je, a na koniec zwraca inne 64 bajty danych, z powrotem do funkcji rozszerzającej. Funkcja haszująca jako dane wejściowe przyjmuje ciąg bajtów, na który składają się:

* sekretny klucz.
* nonce wraz z dodatkowym numerem bloku.
* cztery stałe wektory, każdy składający się z 4 bajty określone przez funkcję rozszerzającą, o wartościach zależących od długości używanego sekretnego klucza.

Funkcja haszująca operuje na danych podzielonych na słowa (słowa maszynowe, ang. words). Każde słowo składa się z 4 bajtów i może zawierać liczbę z przedziału od 0 do 232-1. Wynika z tego, że blok danych wejściowych ma 16 słów długości, sekretny klucz zawiera 8 lub 4 słowa, natomiast nonce ma 2 słowa.

Wyjście z funkcji rozszerzającej szyfru Salsa20 jest dodawane XOR do 64-bajtowego bloku danych. Wynikiem tego działania jest 64-bajtowy blok zaszyfrowanego tekstu.

Dodawanie dwóch 4-bajtowych słów jest przedstawione w opisie algorytmu jako a + b. Wynik dodawania jest dzielony modulo przez 2^32 (czyli przez maksymalną wartość słowa 4-bajtowego).

Suma dwóch słów a i b jest równa a+b mod 2^32. Wynik ma 4 bajty długości.

Funkcja Rundy Wierszy pobiera 16 słów wejściowych, przekształca je i zwraca ciąg o długości również 16 słów.

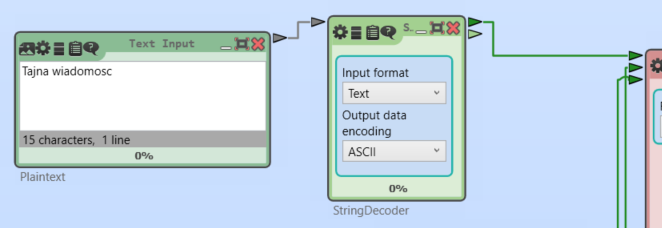
Funkcja ta jest bardzo podobna do Funkcji Rundy Kolumn, ale operuje na słowach w innej kolejności.

**Deszyfrowanie**

W celu odszyfrowania otrzymanego szyfrogramu, należy wykorzystać ten sam algorytm. Dane powinny zostać podzielone w ten sam sposób.

Wyjście z funkcji rozszerzającej szyfru Salsa20 powinno być dodane XOR do 64-bajtowego bloku zaszyfrowanego tekstu. Wynikiem tego sumowania jest 64-bajtowy blok tekstu jawnego.

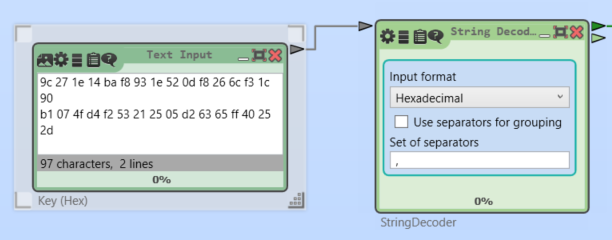
**Aby zaszyfrować tekst “Tajna wiadomosc” dodajemy block z tekstem:**



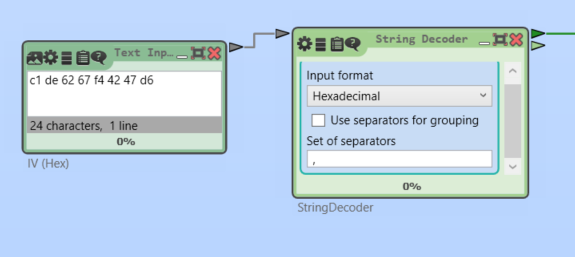
**Potem dodajemy nasz klucz w HEX przedstawieniu:**

9c 27 1e 14 ba f8 93 1e 52 0d f8 26 6c f3 1c 90

b1 07 4f d4 f2 53 21 25 05 d2 63 65 ff 40 25 2d

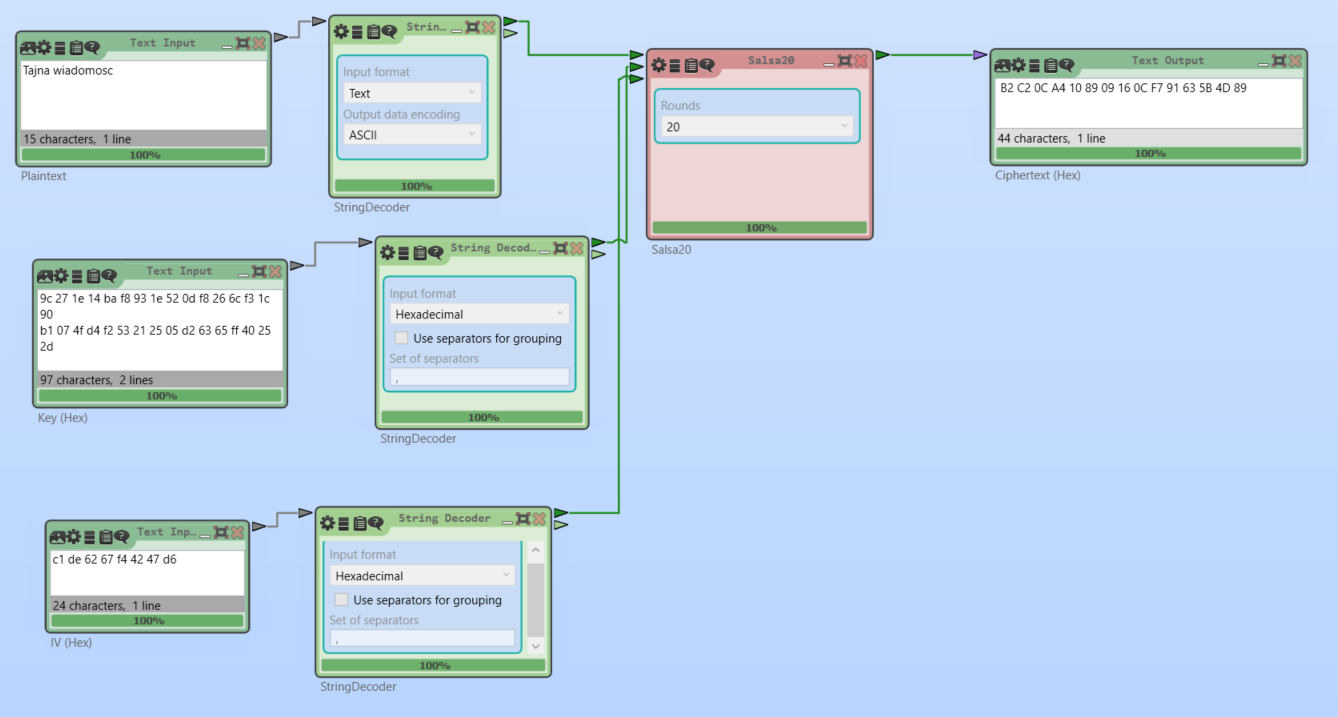


I ostatnim krokiem dodajemy 8 bajtowy nonce (c1 de 62 67 f4 42 47 d6):



Po skończeniu szyfrowania otrzymujemy B2 C2 0C A4 10 89 09 16 0C F7 91 63 5B 4D 89 nasz szyfrowany tekst w przedstawieniu HEX.

**Cały wygląd projektu (szyfrowanie)**



Do odszyfrowania potrzebujemy klucza I szyfrogramu umieszczonych w operatorie XOR.