LUFFA
Jednokierunkowa Funkcja Skrótu - Implementacja Referencyjna

Aleksander Wiącek

3 października 2023

1 Wstęp

Luffa jest funkcją skrótu, stworzoną na potrzeby konkursu SHA-3, w którym zakwalifikowała się do drugiego etapu. W trakcie trwania konkursu funkcja ta została zaktualizowana do wersji Luffav2, w której zmieniono postać skrzynki podstawieniowej, kolejność wejścia i wyjścia bitów z poszczególnych słów do S-skrzynki oraz w rundzie finalizacyjnej - runda z pustą wiadomością wykonywana jest zawsze niezależnie od długości wiadomości do skrócenia.
Zaimplementowana została funkcja w wersji drugiej, generująca 256 bitową długość skrótu. Dopełnienie wiadomości następuje poprzez dopisanie do ostatniego bloku 256b sekwencji bitów 100...00 tak, aby spełniona została kongruencja:

$$L + 1 + K \equiv 0 \ (mod 256),\ L \text{ - długość wiadomości, } K \text{ - ilość zer.}$$

Długość dopełnionej wiadomości powinna być zatem wielokrotnością liczby 256.
Funkcja rundy algorytmu, przyjmuje 256 bitową długość bloku wiadomości oraz trzy bloki wartości pośrednich również po 256 bitów każdy. W pierwszej rundzie bloki te są wartościami inicjalizującymi 'IV' postaci:

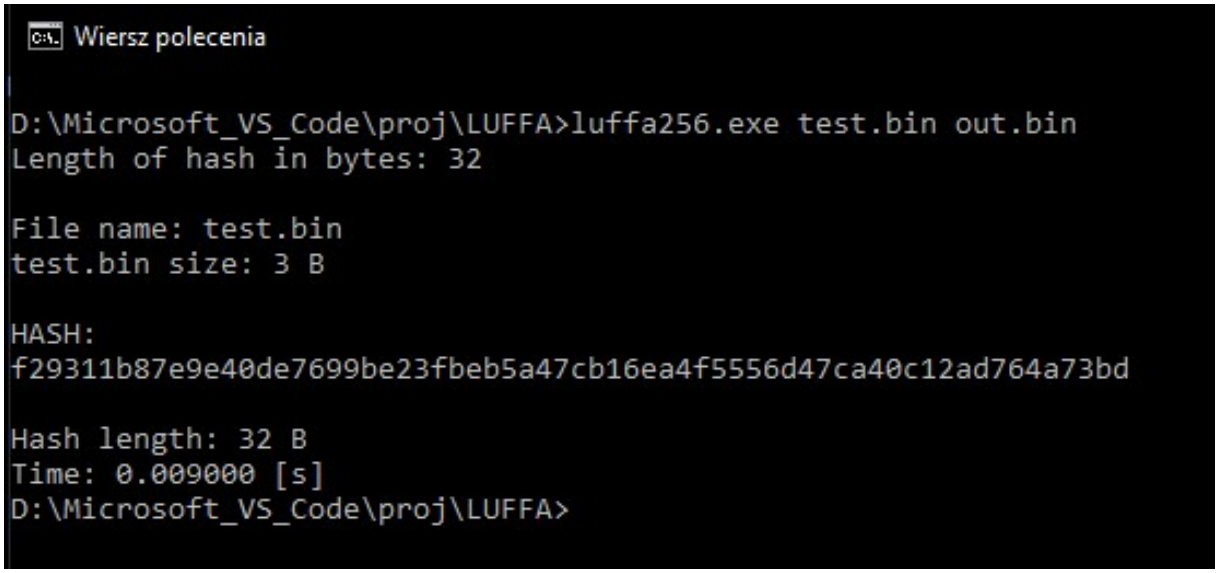
$V_{0,0} = 0x6d251e69$	$V_{0,1} = 0x44b051e0$	$V_{0,2} = 0x4eaa6fb4$	$V_{0,3} = 0xdbf78465$
$V_{0,4} = 0x6e292011$	$V_{0,5} = 0x90152df4$	$V_{0,6} = 0xee058139$	$V_{0,7} = 0xdef610bb$
$V_{1,0} = 0xc3b44b95$	$V_{1,1} = 0xd9d2f256$	$V_{1,2} = 0x70eee9a0$	$V_{1,3} = 0xde099fa3$
$V_{1,4} = 0x5d9b0557$	$V_{1,5} = 0x8fc944b3$	$V_{1,6} = 0xcf1ccf0e$	$V_{1,7} = 0x746cd581$
$V_{2,0} = 0xf7efc89d$	$V_{2,1} = 0x5dba5781$	$V_{2,2} = 0x04016ce5$	$V_{2,3} = 0xad659c05$
$V_{2,4} = 0x0306194f$	$V_{2,5} = 0x666d1836$	$V_{2,6} = 0x24aa230a$	$V_{2,7} = 0x8b264ae7$

Wartości pośrednie w funkcji rundy przechodzą przez dwie warstwy: Message Injection oraz przez warstwę nieliniowej permutacji. W wersji 256 bitowej algorytmu warstwa nieliniowej permutacji składa się z 3 bloków Qj, po jednym dla każdego bloku stanu pośredniego. Ostatnią operacją w Qj jest dodanie stałych do zerowego i czwartego słowa bloku stanu pośredniego. W implementacji stałe te są obliczane za pomocą funkcji przedstawionej w specyfikacji, jednak równie dobrze, wszystkie stałe mogłyby być przechowywane w pamięci.

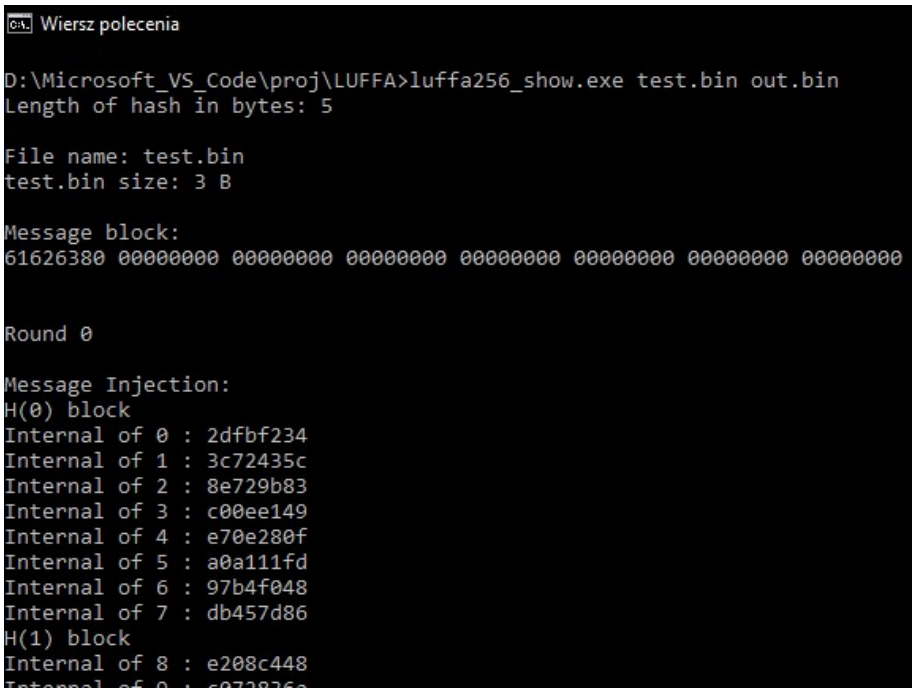
2 Wyniki implementacji

Stworzono dwie wersje programu. Pierwszy plik wykonywalny luffa256.exe wypisuje tylko nazwę skracanego pliku, jego rozmiar, wartość heksadecymalną skrótu, jego długość oraz czas działania w sekundach.
Natomiast plik luffa256_show.exe to ten sam algorytm z tym wyjątkiem, że wypisuje wszystkie ważne wartości pośrednie.

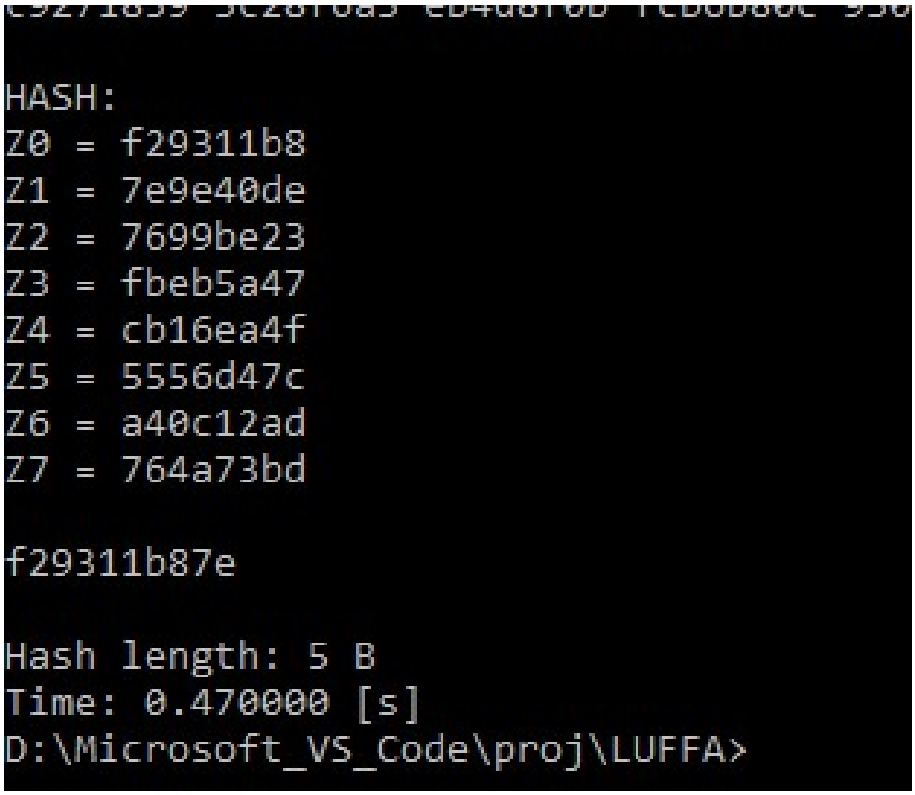
Wyniki dla pliku binarnego zawierającego ciąg znaków ’abc’:



Rys. 1. Wynik działania luffa256.exe



Rys. 2. Wynik działania luffa256_show.exe cz.1



Rys. 3. Wynik działania luffa256_show.exe cz.2

Wartości wektora testowego:

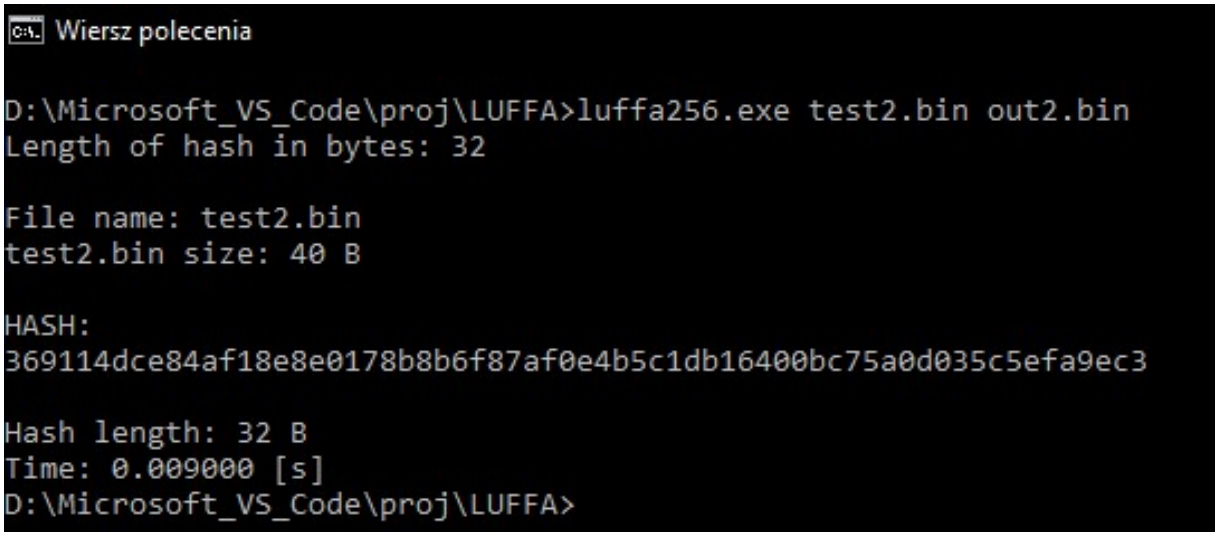
C–2 *Luffa*-256

The message digest of the message "abc" is

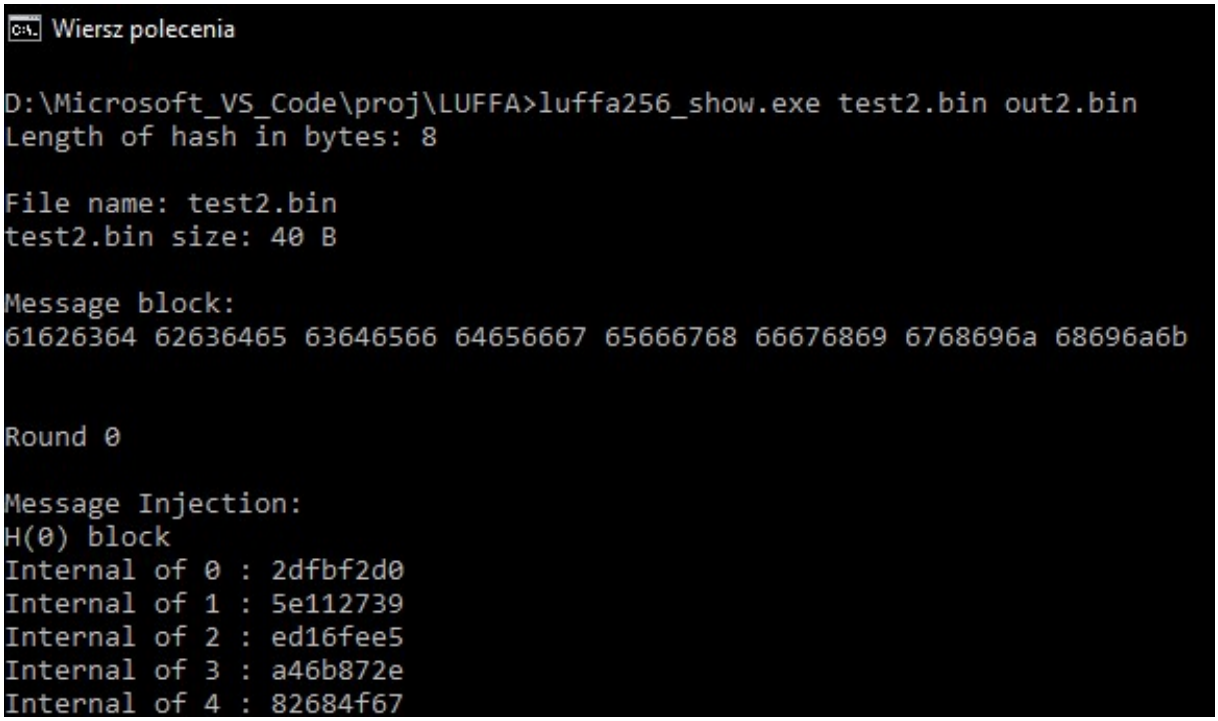
$$Z_{0,0} = \texttt{0xf29311b8}, \quad Z_{0,1} = \texttt{0x7e9e40de},$$
$$Z_{0,2} = \texttt{0x7699be23}, \quad Z_{0,3} = \texttt{0xfbeb5a47},$$
$$Z_{0,4} = \texttt{0xcb16ea4f}, \quad Z_{0,5} = \texttt{0x5556d47c},$$
$$Z_{0,6} = \texttt{0xa40c12ad}, \quad Z_{0,7} = \texttt{0x764a73bd}.$$

Rys. 4. Wektor testowy dla luffa256

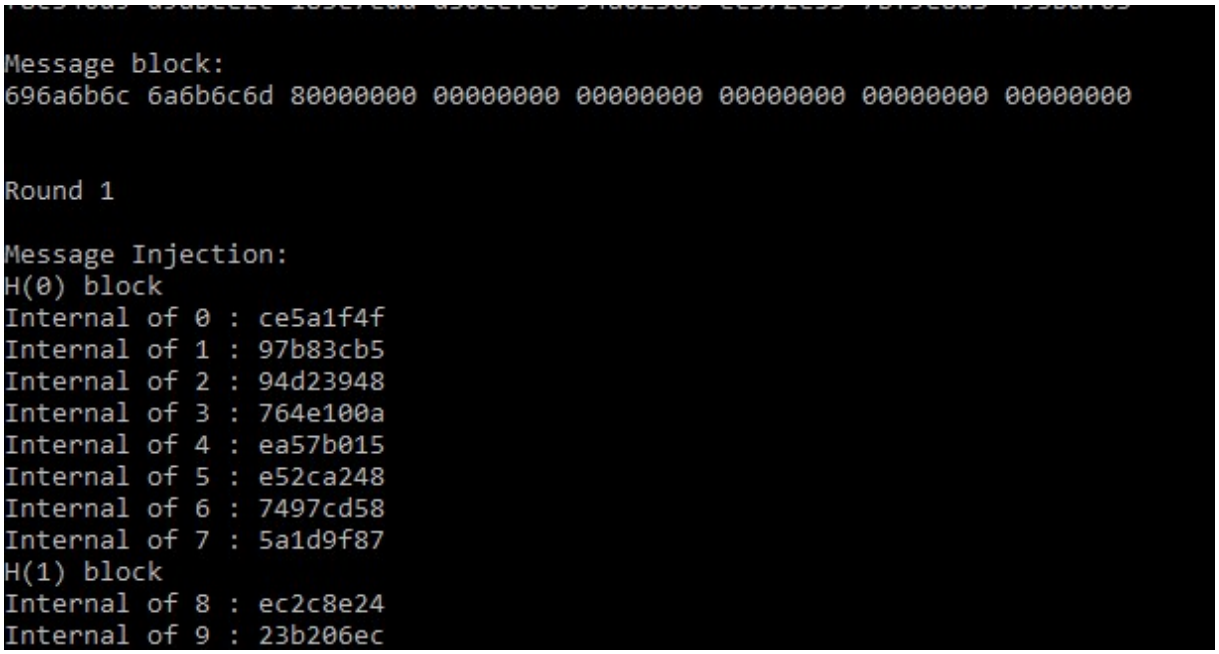
Test dla wiadomości: ’abcdbcdecdefdefgefghfghighijhijkijkljklm’:



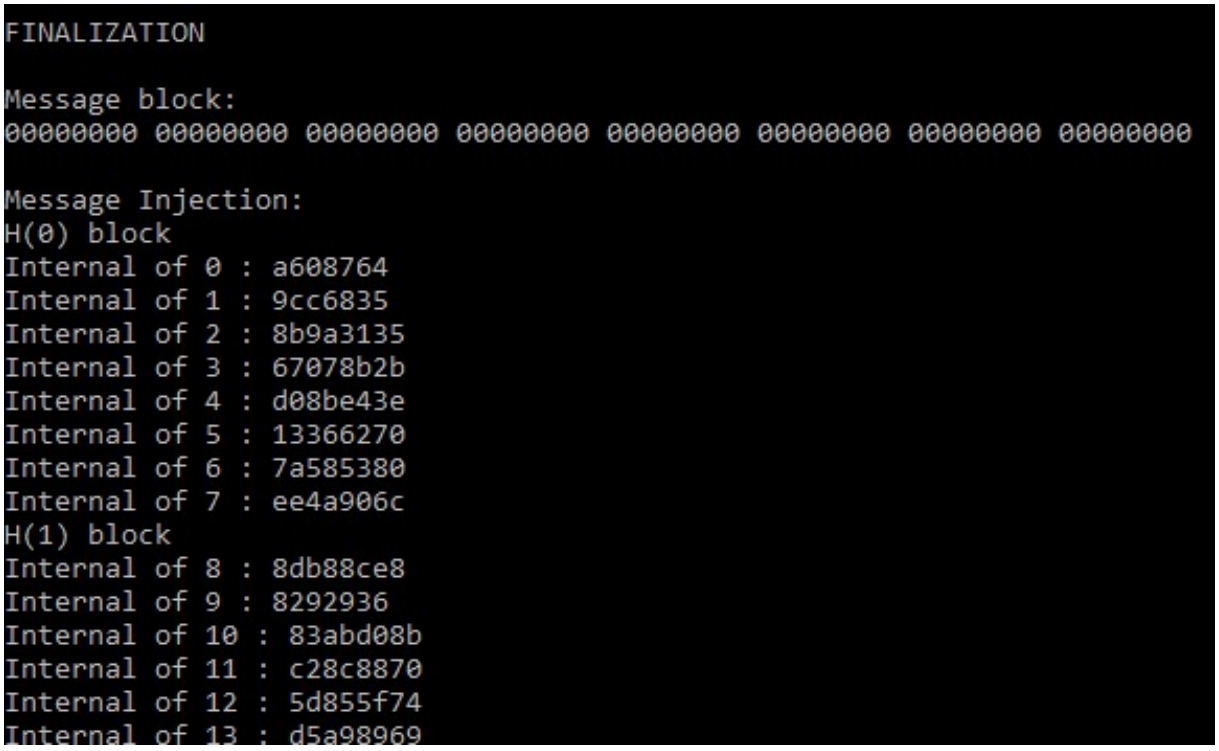
Rys. 5. Wynik działania luffa256.exe dla testu 2



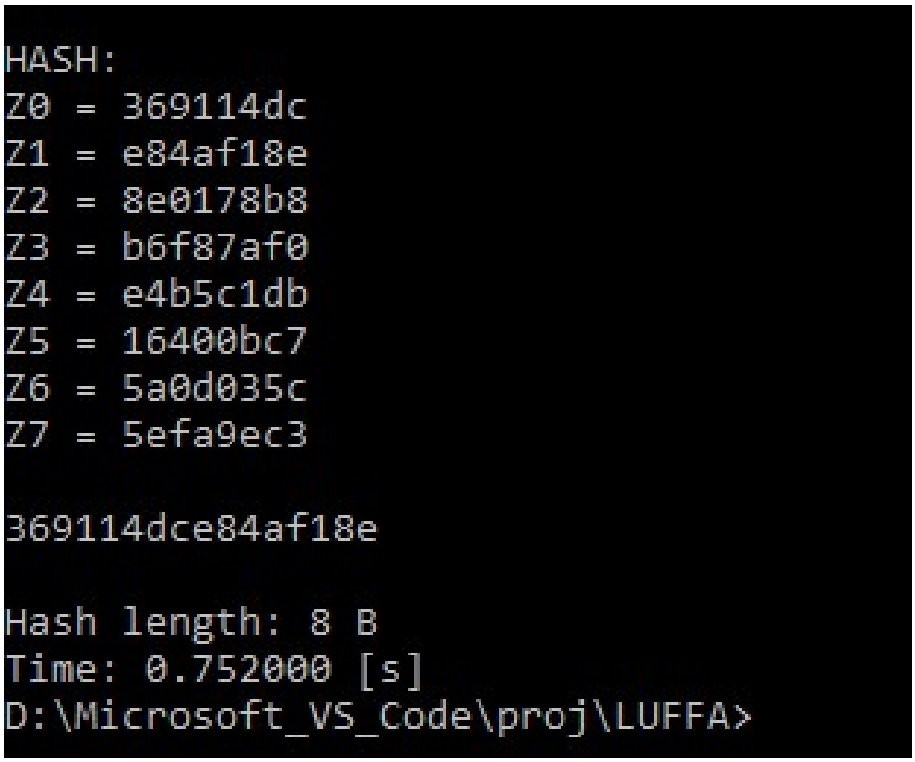
Rys. 6. Wynik działania luffa256.exe_show dla testu 2 - blok 1



Rys. 7. Wynik działania luffa256_show.exe dla testu 2 - blok 2

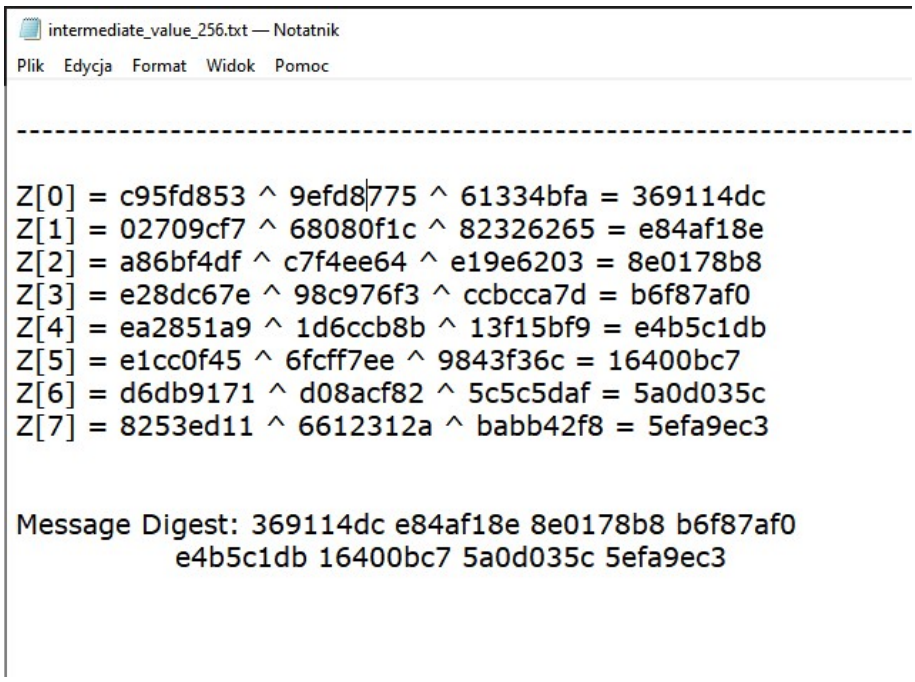


Rys. 8. Wynik działania luffa256_show.exe dla testu 2 - finalizacja



Rys. 9. Wynik działania luffa256_show.exe dla testu 2 - rezultat

Wynik podany przez autorów funkcji dla skracanej wiadomości:



Rys. 10. Rezultat dla ’abcdbcdecdefdefgefghfghighijhijkijkljklm’