Security of Computer Systems

Project Report

Authors:
Łukasz Kawa 184948

Version: 2.0

Versions

Version	Date	Description of changes
1.0	20.04.2023	Podsumowanie wykonanej pracy
2.0	20.06.2023	Podsumowanie skończonej pracy

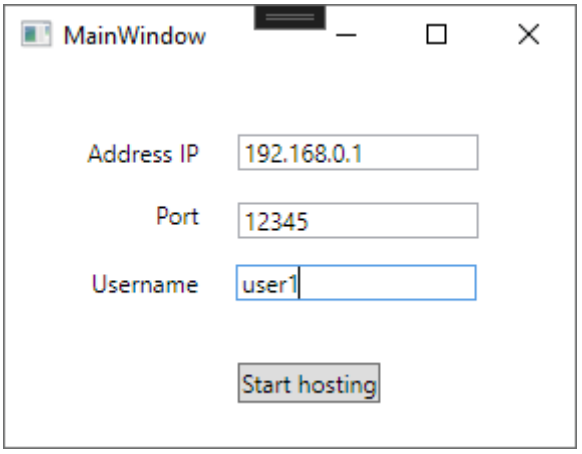
1. Project – control term

1.1 Description

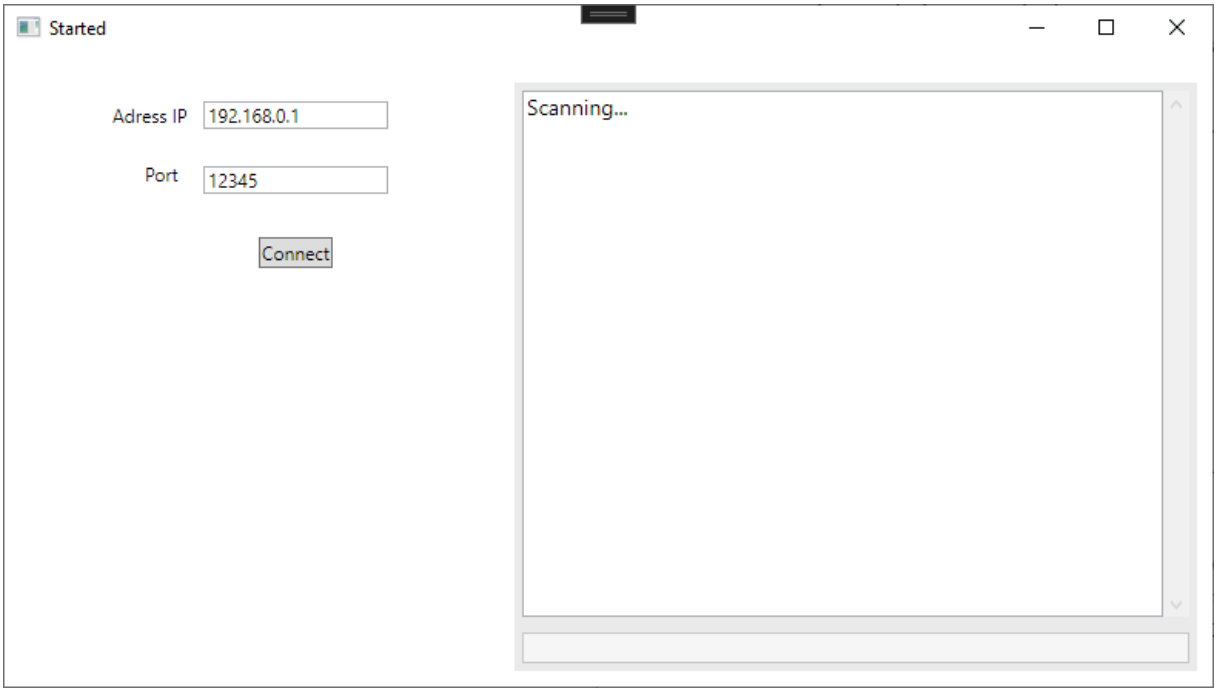
Stworzenie aplikacji, umożliwiającej bezpieczne komunikowanie się dwóch osób. Po nawiązaniu połączenia, wysyłane wiadomości są szyfrowane za pomocą kluczy publicznych, które zostają przesłane przy tworzeniu połączenia. Poza wiadomościami tekstowymi, aplikacja będzie umożliwiała wysyłanie plików.

1.2 Results

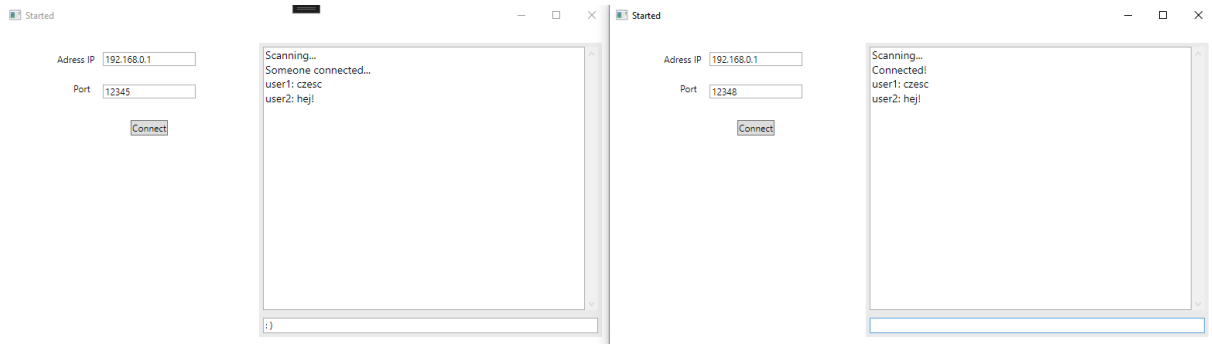
Stworzona została aplikacja służąca do komunikacji pomiędzy dwoma użytkownikami. Do stworzenia aplikacji zostały użyte narzędzia WPF w języku C#. Komunikacja pomiędzy dwoma aplikacjami odbywa się poprzez stosowanie klasy Socket i protokołu TCP. Użytkownicy aplikacji podają pod jakim adresem IP chcą być widoczni i mają możliwość połączenia się z innym użytkownikiem, który jest w trybie nasłuchiwania na połączenie. Każdy użytkownik, przed rozpoczęciem nasłuchiwania musi podać pod jaką nazwą chce być widziany. Po nawiązaniu połączenia użytkownicy mogą ze sobą konwersować. Dostępne są tylko wiadomości tekstowe.



Zdjęcie przedstawia okno inicjalizacji hosta



Host jest w trybie nasłuchiwania, czeka na nadchodzące połączenia



Użytkownik połączył się do hosta, połączenie zostało nawiązane, użytkownicy mogą się ze sobą komunikować

1.3 Summary

Aplikacja umożliwia komunikację między dwoma użytkownikami. Wiadomości wysyłane nie są zaszyfrowane, a połączenie zostaje nawiązane bez potrzeby akceptacji przez któregoś z użytkowników (brak możliwości odrzucenia połączenia).

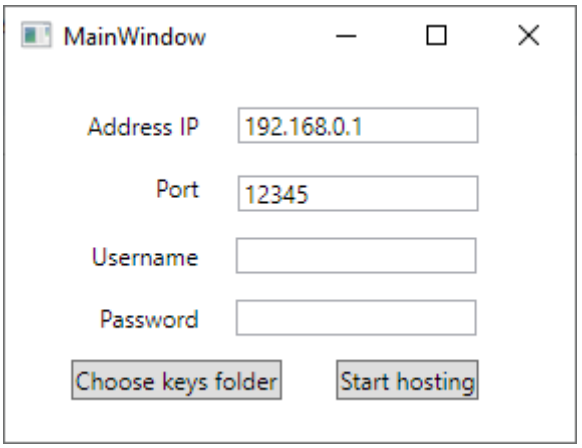
2. Project – Final term

2.1 Description

Zastosowanie algorytmów szyfrujących, pozwalających na bezpieczne wysyłanie danych pomiędzy użytkownikami, w tym plików oraz dodanie progress bara pokazującego aktualny stan pobierania.

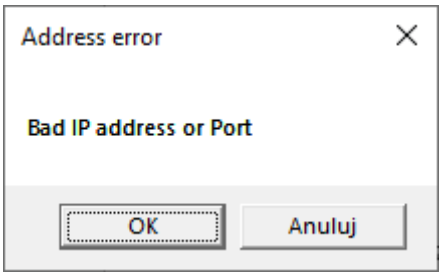
2.2 Results

Aplikacja umożliwia połączenie się dwóch użytkowników ze sobą za pomocą socketa i przy użyciu protokołu TCP. Przed nawiązaniem połączenia, użytkownicy wybierają miejsce przechowywania kluczy publicznych i prywatnych a następnie ustalają swoje hasło, które będzie używane do zabezpieczenia klucza prywatnego.



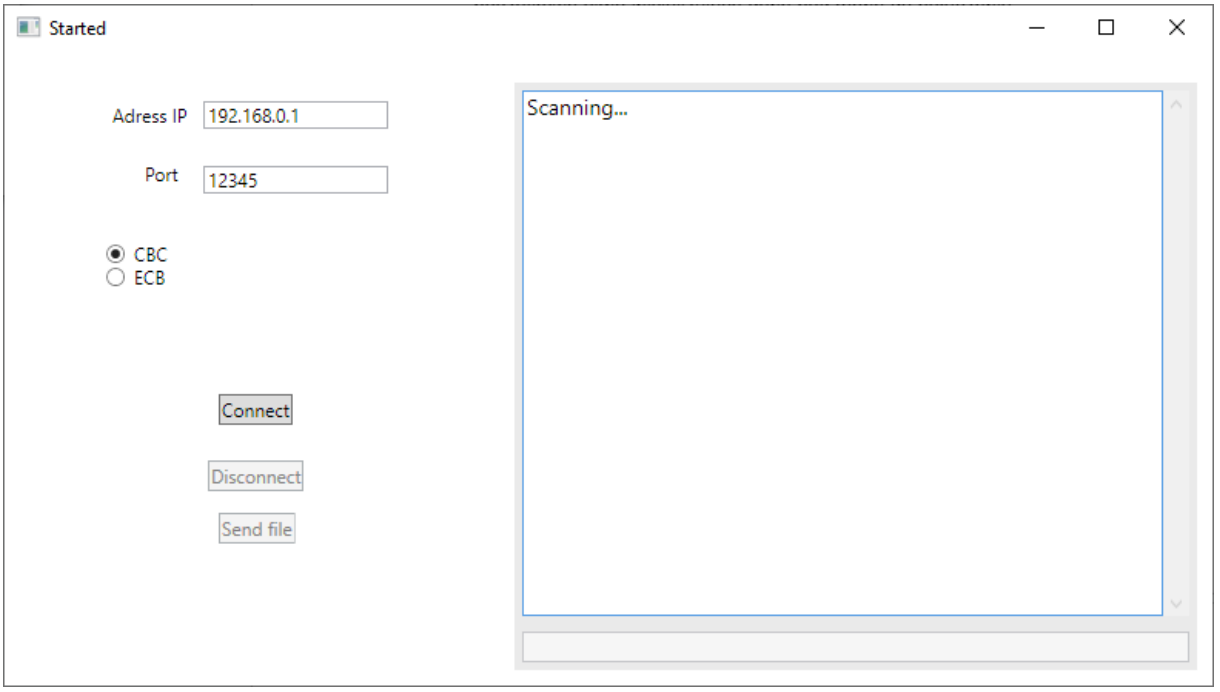
początkowe okno inicjalizujące dane potrzebne do połączenia

Jeżeli chcemy użyć portu bądź folderu z czyimiś kluczami, zostajemy o tym odpowiednio poinformowani.



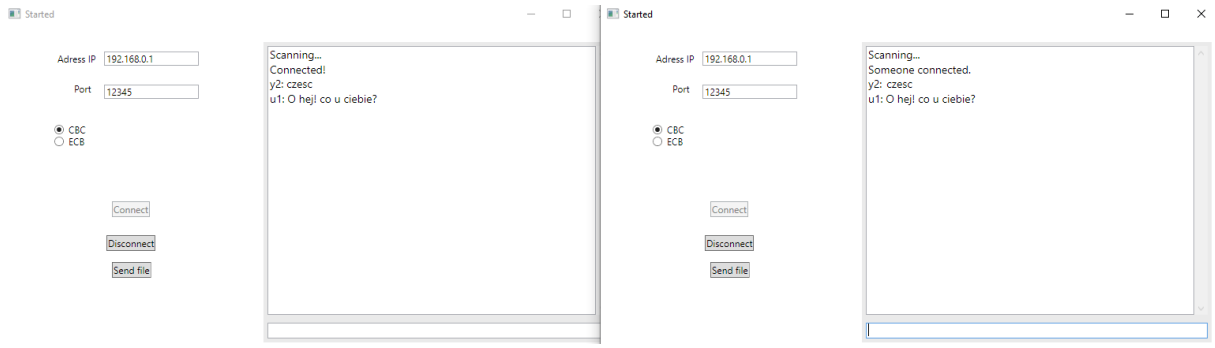
Po poprawnie uzupełnionych danych, możemy rozpocząć nasłuchiwanie (oczekiwanie) na przychodzące połączenia. Po przyjściu połączenia, zostajemy o

tym poinformowani, i zaczyna się wymiana kluczy publicznych.

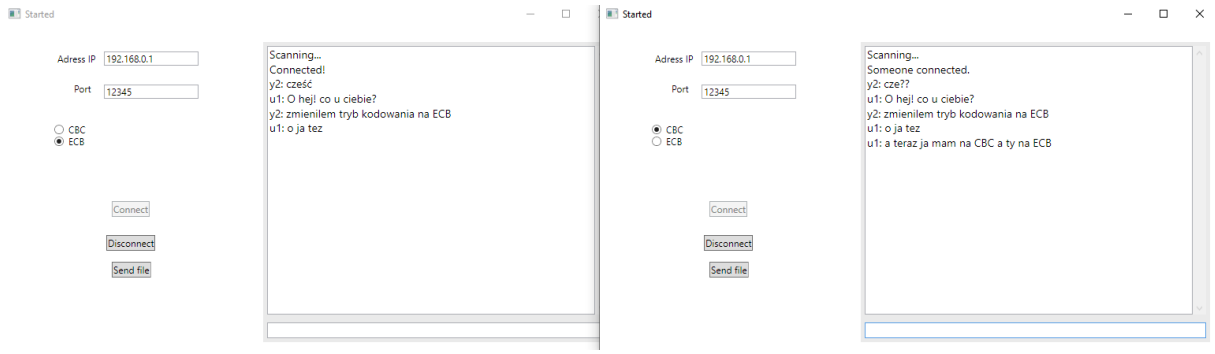


okno czatu w stanie skanowania

W danym oknie, po połączeniu drugiego użytkownika, mamy opcję wyboru modelu zabezpieczania naszych wiadomości - CBC i ECB. Jeśli nasze kodowania różnią się między użytkownikami, to wiadomości takie mogą zostać wyświetlone w postaci znaczków bądź wcale, w zależności od wysłanych bajtów.



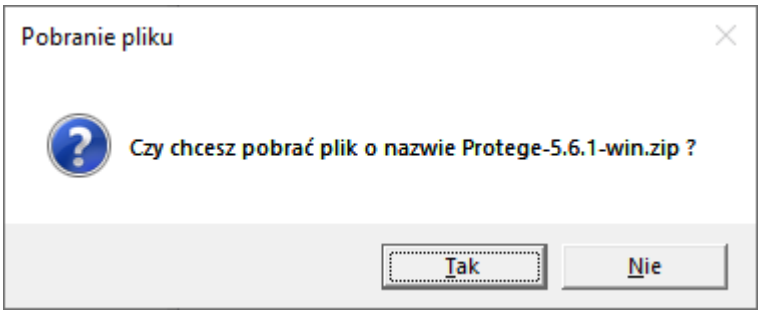
widok trwającej konwersacji



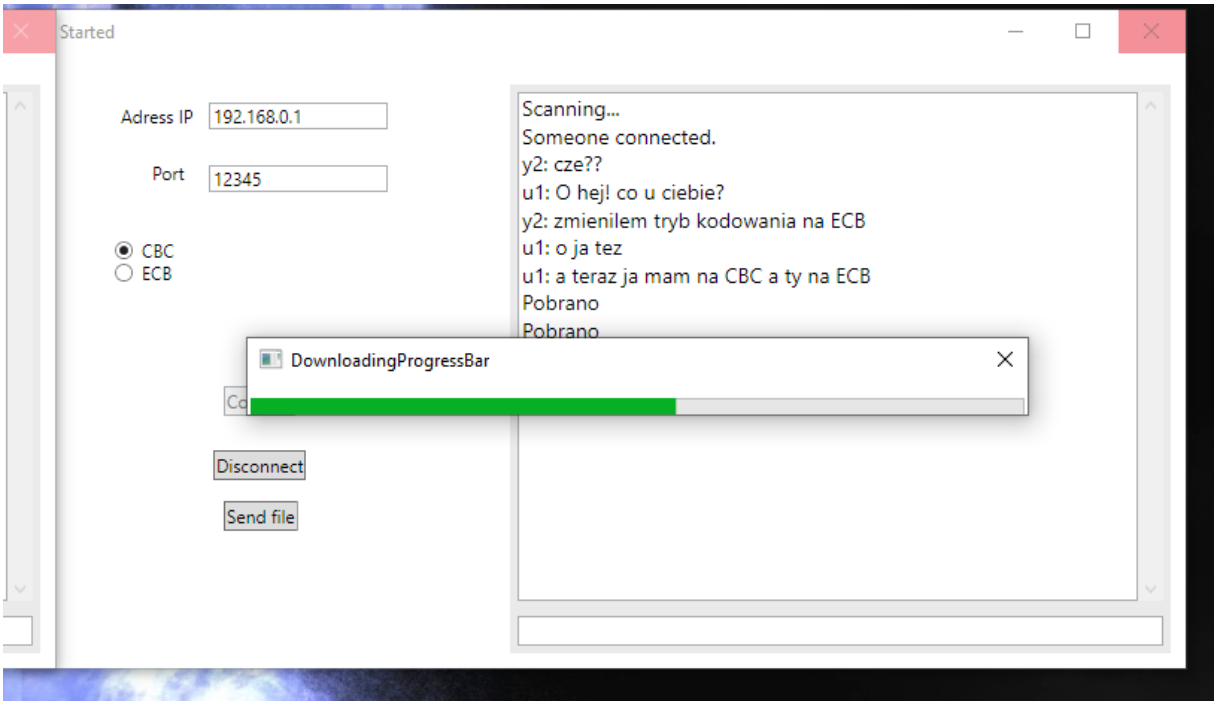
konwersacja po zmianie na różne kodowania i oba na ECB

Do szyfrowania danych został użyty algorytm AES z 256 bitowym kluczem i 128 bitowym wektorem inicjalizującym. Przed każdym wysłaniem “paczki” danych, jest ona szyfrowana, i za każdym razem zostaje wygenerowany nowy klucz, który jest szyfrowany kluczem publicznym odbiorcy, a następnie wysyłany razem z zaszyfrowanymi danymi.

Przy wysyłaniu plików, dzielimy je na mniejsze części, a następnie każdą paczkę kodujemy i wysyłamy. Informacje o tym, ile potrzeba pobrać bajtów, wysyłamy na początku, w pierwszej wiadomości informującej o tym, że będziemy przesyłać plik. Wiadomość ta zawiera informacje o nazwie pliku oraz jego rozmiarze.



Komunikat o tym czy chcemy pobrać dany plik



Progres pobierania pliku o rozmiarze 120 MB

2.3 Summary

Aplikacja do komunikacji w C# WPF na socket, wykorzystująca protokół TCP, umożliwiająca szyfrowanie wiadomości za pomocą algorytmów AES w trybach ECB i CBC, RSA, oraz uwzględniająca funkcje skrótu SHA256 i MD5, a także umożliwiająca przesyłanie plików. Metody umożliwiające bezpieczną komunikację

(szyfrowanie i deszyfrowanie wiadomości) zostało zaimplementowane w bibliotekach Encryption i EncryptionCommunication. Same algorytmy szyfrowania, RSA i AES, zostały użyte już istniejące implementacje w technologii .NET.

3. Literature

[1] https://learn.microsoft.com/pl-pl/dotnet/api/system.net.sockets.socket?view=net-7.0

[2] https://pl.wikipedia.org/wiki/Gniazdo_(telekomunikacja)