

**Wrocław, 06.11.2016**

**Bezpieczeństwo Usług Sieciowych**

**Sprawozdanie nr 1**

**Komunikator Klient-Server wykorzystujący algorytm Diffie-Hellman**

**Prowadzący : mgr inż. Przemysław Świercz**

**Autor : Jędrzej Mirowski 191370**

## **1. Cel zadania**

Celem zadania było stworzenie komunikatora umożliwiającego wymianę wiadomości pomiędzy klientem a serwerem. Wymiana wiadomości powinna opierać się w formie zaszyfrowanej, opierając się o protokół Diffie-Hellmana. Użytkownik powinien mieć także wybór czy chce prowadzoną komunikację szyfrować, opierając się o szyfr Cezara lub XOR. Wiadomości przed wysyłaniem powinny być także kodowane z wykorzystaniem kodu base64 oraz wysyłana w oparciu o format JSON.

## **2. Wykorzystane technologie i sposób wykonania zadania**

Aplikacja została napisana w języku Java SE (IDE - Eclipse ). Zarówno część kliencka aplikacji (pakiet client) jak i ta serwerowa (server) posiadają klasy zawierające metodę main, co pozwala uruchomić je niezależnie od siebie. Do budowania aplikacji wykorzystano narzędzie Gradle. Wiadomości szyfrowane są za pomocą szyfru Cezara.

Działanie programu:

- a) Wykorzystując terminal użytkownik uruchamia serwer, podając również w komendzie pożądany port na którym otwarta zostanie komunikacja. Następnie klient może połączyć się z serwerem wykorzystując wybrany adres IP i port. Klient , poprzez dopisek „encrypt” na końcu komendy może włączyć także szyfrowanie wiadomości.
- b) Z powodu problemów implementacyjnych nie wykonano wymiany danych opartych o standard JSON. Wymiana danych odbywa się w z wykorzystaniem metody processData() , poprzez wysyłania pomiędzy klientem i serwerem pakietu danych, zawierających odpowiednie prefixy wiadomości. Na podstawie tych prefixów obie strony podejmują odpowiednie działania. Np. wiadomość z prefixem „\wrong” otrzymana przez Klienta oznacza, że nazwa klienta na którą się zdecydował jest już zajęta i musi wybrać nową.

Właśnie poprzez odpowiednie prefixy odbywa się cała procedura wymiany kluczy w algorytmie Diffie'ego-Hellmana:

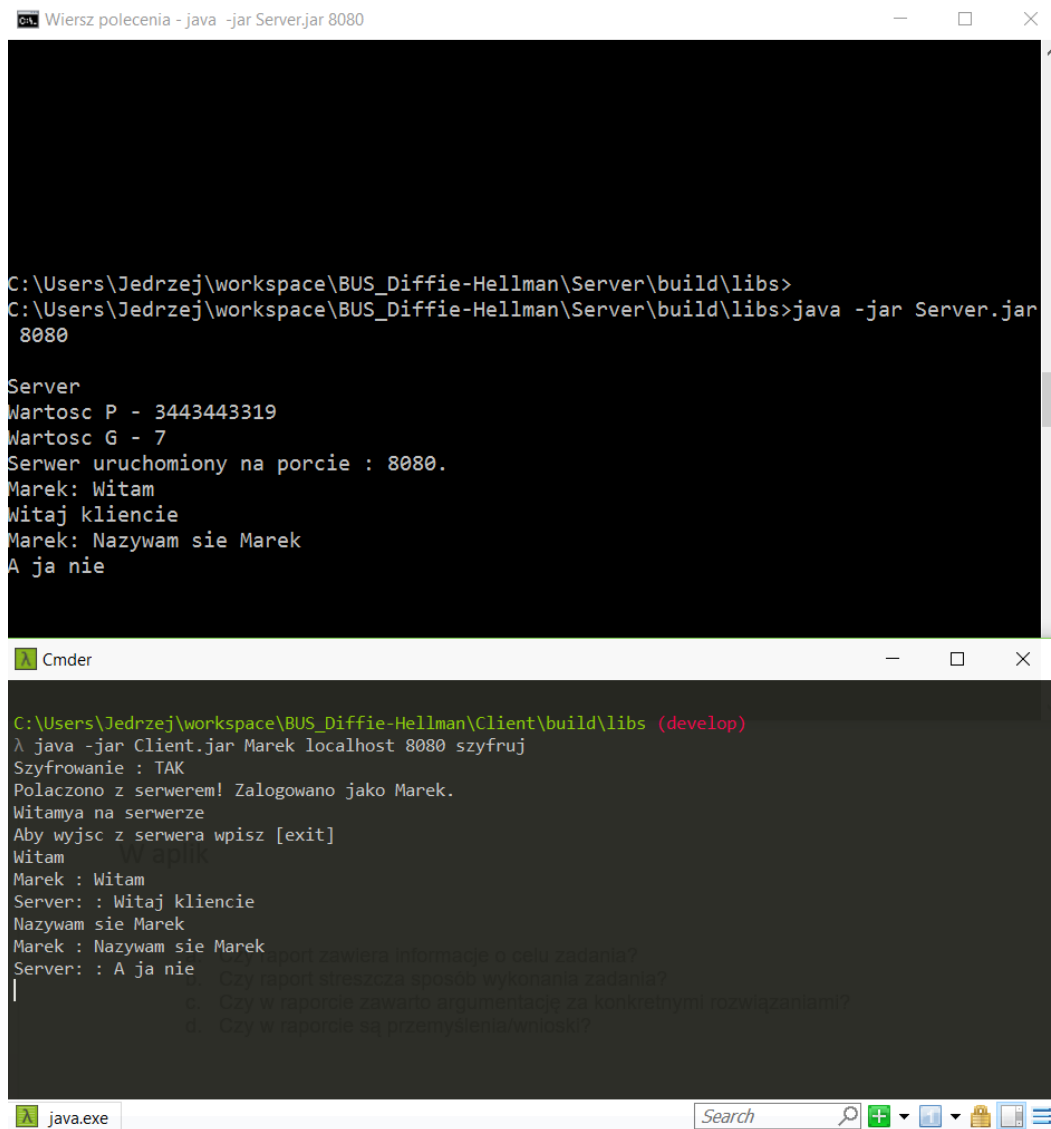
/pgValues (KeyExchangeStep1) – serwer wysyła wiadomość z wartościami P i G.

/aValue (KeyExchangeStep2) – klient wysyła informację o wartości A

/bValue(KeyExchangeStep3) – serwer wysyła wartość B.

/encrypt – klient wysyła informację czy ma być szyfrowanie czy nie.

c) Następnie prowadzona jest wymiana wiadomości pomiędzy klientem a serwerem.



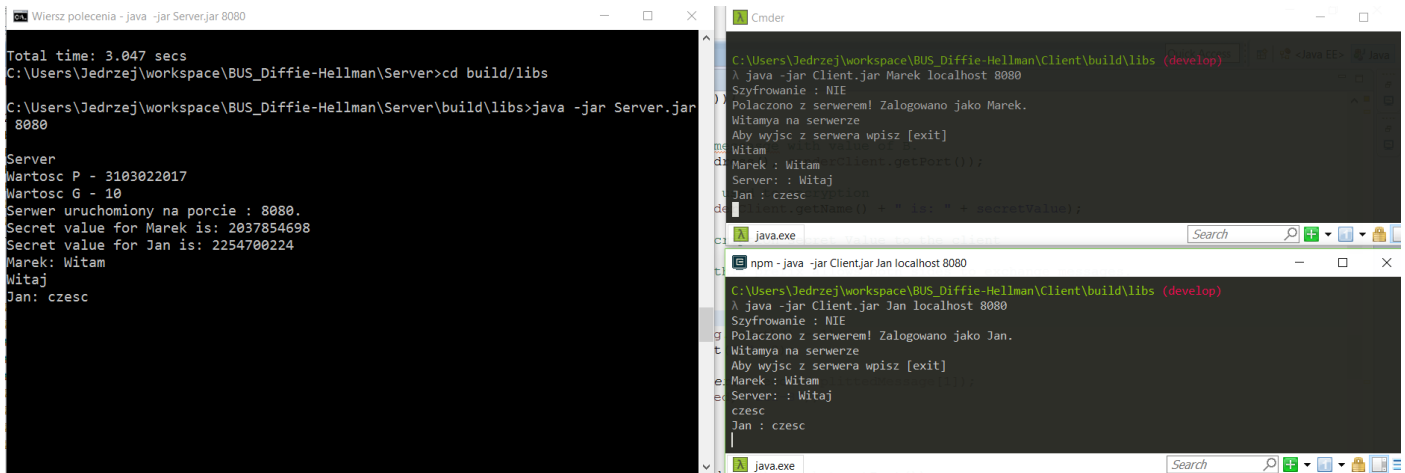
```
Wiersz polecenia - java -jar Server.jar 8080

C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Server\build\libs>
C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Server\build\libs>java -jar Server.jar 8080

Server
Wartosc P - 3443443319
Wartosc G - 7
Serwer uruchomiony na porcie : 8080.
Marek: Witam
Witaj kliencie
Marek: Nazywam sie Marek
A ja nie

Cmder
C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Client\build\libs (develop)
λ java -jar Client.jar Marek localhost 8080 szyfruj
Szyfrowanie : TAK
Polaczono z serwerem! Zalogowano jako Marek.
Witamy na serwerze
Aby wyjsc z serwera wpisz [exit]
Witam
Marek : Witam
Server: : Witaj kliencie
Nazywam sie Marek
Marek : Nazywam sie Marek
Server: : A ja nie
raport zawiera informacje o celu zadania?
c. Czy w raporcie zawarto argumentacje za konkretnymi rozwiazaniami?
d. Czy w raporcie sa przemyślenia/wnioski?
```

Serwer obsługuje wielu klientów jednocześnie i dla każdego z nich generowana jest inna wartość klucza. (Na poniższym screenie wartości te zostały podejrżane jednorazowo za pomocą metody println()).



```
Wiersz polecenia - java -jar Server.jar 8080
Total time: 3.047 secs
C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Server>cd build\libs
C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Server\build\libs>java -jar Server.jar
8080

Server
Wartosc P - 3103022017
Wartosc G - 10
Serwer uruchomiony na porcie : 8080.
Secret value for Marek is: 2037854698
Secret value for Jan is: 2254700224
Marek: Witam
Witaj
Jan: czesc

C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Client\build\libs (develop)
A java -jar Client.jar Marek localhost 8080
Szyfrowanie : NIE
Polaczono z serwerem! Zalogowano jako Marek.
Witamy na serwerze
Aby wyjść z serwera wpisz [exit]
Marek : Witam
Server: : Witaj
Jan : czesc
C:\Users\Jedrzej\workspace\BUS_Diffie-Hellman\Client\build\libs (develop)
A java -jar Client.jar Jan localhost 8080
Szyfrowanie : NIE
Polaczono z serwerem! Zalogowano jako Jan.
Witamy na serwerze
Aby wyjść z serwera wpisz [exit]
Marek : Witam
Server: : Witaj
czesc
Jan : czesc
```

### 3. Wnioski i podsumowanie

*Aplikacja spełnia większość założeń projektowych:*

- + Zaimplementowano aplikację Serwer –Klient, z wymianą klucza według protokołu Diffie-Hellmana.
- + Obie strony mogą wymieniać się wiadomościami. Istnieje także możliwość obsługi wielu klientów jednocześnie.
- + Możliwe jest szyfrowanie z wykorzystaniem szyfru Cezara. Napisano także proste testy sprawdzające poprawność metody szyfrującej.
- + zaimplementowano kodowanie base64.

*Aplikacja posiada jednak także wady:*

- nie udało się zaimplementować wymiany danych opartej o standard json. Jest to wynikiem problemów, jakie pojawiały się przy przesyłaniu obiektu json pomiędzy klientem i serwerem, tzn. :

Klient tworzył obiekt JSON --> umieszczał w nim dane --> obiekt parsowany był do typu bytes i wysyłany do serwera --> serwer odbierał dane i zapisywał je do formatu String. Następnie na jego podstawie tworzył obiekt JSON z którego docelowo powinny być wyciągane dane „klucz” – „wartość”. Tu pojawiał się problem, ponieważ pomimo iż dane były poprawnie zapisywane do stringa, to kompilator nie był w stanie poprawnie ich zinterpretować.

Pomimo iż String zaczynał się od zgodnego ze standardem JSON Object znaku { to kompilator wciąż wyrzucał błąd *A JSONObject text must begin with '{'*. Nie pomogło usuwanie spacji, funkcje z zastępowaniem znaków i różne metody obróbki przesyłanych danych.

- Z powodu braku wymiany danych w formacie JSON, nie ma możliwości współpracy z innymi aplikacjami, wykorzystującymi JSONa. Wartości prefixów zostały z góry narzucone i strona kliencka lub serwerowa jest gotowa do współpracy jedynie w przypadku gdyby druga strona również zdecydowała się na wykorzystanie takich samych prefixów.