

# Sprawozdanie wstępne

---

## Wykorzystane algorytmy

Do wymiany kluczy użyty zostanie **algorytm Diffiego-Hellmana**:

- Każda ze stron na podstawie własnego klucza prywatnego oblicza klucz publiczny przy pomocy parametrów - podstawy i modułu (takich samych dla obu stron i znanych z góry)
- Wymiana kluczy publicznych następuje na etapie wysłania wiadomości **ClientHello** i **ServerHello**
- Na podstawie uzyskanych kluczy publicznych i posiadanych kluczy prywatnych obie strony obliczają klucz wspólny, który będzie wykorzystany do szyfrowania i odszyfrowywania

Dodatkowo zostanie wykorzystany mechanizm **encrypt-then-mac** - najpierw wiadomość będzie szyfrowana, następnie na jej podstawie zostanie wygenerowany kod MAC, który zostanie dołączony do wiadomości.

Wykorzystany zostanie algorytm HMAC-SHA256

Szyfrowanie i odszyfrowywanie zrealizowane będzie przy pomocy AES z rozmiarem bloku wynoszącym 128 bitów (16 bajtów). Trybem szyfrowania będzie Cipher Block Chaining

Jako że korzystamy z Pythona generowanie kodu MAC oraz obsługa szyfrowania zostanie zrealizowana za pomocą odpowiednich bibliotek.

## Struktura wiadomości

**ClientHello**:

```
{
  "type": "Hello message",
  "public_key": 123456789,
  "base": 5,
  "modulus": 23
}
```

**ServerHello**:

```
{
  "type": "Hello message",
  "public_key": 123456789
}
```

**Szyfrowane wiadomości** - przed odszyfrowaniem będą miały strukturę ciągu bajtów, którego ostatnie 32 zostną poświęcone na kod MAC. Po odszyfrowaniu wiadomości uzyskujemy jej zawartość

**EndSession** - tak samo jak zwykła szyfrowana wiadomość tylko zawiera specjalną zawartość przez którą można ją zidentyfikować jako EndSession. Może to być po prostu "EndSession".

## Przykładowy scenariusz działania

1. Klient łączy się z serwerem i wysyła **ClientHello** z kluczem publicznym klienta
2. Serwer oczekuje na wiadomość **ClientHello** i tylko w odpowiedzi na nią wysyła **ServerHello** z kluczem publicznym serwera
3. Po wymianie kluczy obie strony obliczają klucz wspólny
4. Obie strony mogą wysyłać wiadomość zaszyfrowaną przy pomocy AES i z dołączonym kodem MAC
5. Odbiorca dostaje wiadomość, weryfikuje kod MAC i ją odszyfrowuje
6. Przy odszyfrowaniu wiadomości **EndSession** wracamy do początku i czekamy aż klient znowu wyśle **ClientHello**