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
- Moduł i podstawa zostaną wysłane w wiadomości ClientHello - nie będą zahardkodowane.
- 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.

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 i kluczem o długości 256 bitów. Trybem szyfrowania będzie Cipher Block Chaining

Klucz do szyfrowania i generowania kodu MAC uzyskamy przy wykorzystaniu algorytmu PBKDF2. Wartości takie jak ilość iteracji w tym algorytmie i długość klucza są znane obu stronom jeszcze przed wymianą wiadomości `Hello`.

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 stukturę 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 zawierające klucz publiczny klienta oraz parametry podstawy i modułu
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. Klucz wspólny jest przekształcany na klucz AES (256-bitowy) przy użyciu funkcji PBKDF2
5. Obie strony mogą wysyłać wiadomość zaszyfrowaną przy pomocy AES i z dołączonym kodem MAC
6. Odbiorca dostaje wiadomość, weryfikuje kod MAC i ją odszyfrowuje
7. Przy odszyfrowaniu wiadomości `EndSession` wracamy do początku i czekamy aż klient znowu wyśle `ClientHello`