Team:

Weronika Buras

Radosław Motyka

Mateusz Konar

Opis protokołu SRP

Protokół SRP to tak zwany zero-knowledge proof protocol co w dosłownym tłumaczeniu oznacza “zero wiedzący”. O co więc chodzi? Mianowicie serwer nie przechowuje hasła i klient może pewnie i śmiało się uwierzytelnić. Ataki typu man-in-the-middle czy eavesdropper nie mogą uzyskać istotnych wiadomości do przeprowadzeni ataku. Zalety SRP:

* Uwierzytelniania bez wysyłania hasła do sieci
* Chroni przed atakami man-in-the-middle czy eavesdroppe
* Daje pewność identyfikacji serwera do klienta i na odwrót, daje pewność, że nie grozi nam towarzystwo osoby niepożądanej
* Dodatkowo generuje zaszyfrowany klucz, tak więc klient &serwer mogą przesyłać zaszyfrowane dane na poziomie bezpieczeństwa TLS

Podsumowując SRP jednym zdaniem “Nie wykradniesz hasła, jeśli go nie przechowujesz”

# Opis działania

Rejestracja:

1. Klient wybiera losową wartość soli **s** i oblicza **x=H(s,p), v= gx**
2. Klient->Serwer: **v,s** indeksowane przez nazwę użytkownika **I**

Uwierzytelnienie:

1. Klient->Serwer: generuje losową wartość **a**, wysyła **I** (nazwę użytkownika) oraz **A = ga**
2. Serwer -> Klient: generuje losową wartość **b**, wysyła **s** i **B = kv + gb**
3. Obydwoje: **u=H(A,B)**
4. Klient oblicza **SKlient = (*B* − *kg*x)(*a* + *ux*) = (*kv* + *g*b − *kg*x)(*a* + *ux*) = (*kg*x − *kg*x + *g*b)(a + ux) = (*g*b)(*a* + *ux***)
5. Klient oblicza **KKlient=H(SKlient**)
6. Serwer oblicza ***S*Serwer = (*Av*u)b = (*g*a*v*u)b = [*g*a(*g*x)u]b = (*g*a + ux)b = (*g*b)(a + ux)**
7. Serwer oblicza  **KSerwer= H(*S*Serwer)= KKlient**

Następnie następuje proces udowodnienia pasowania kluczy. Jedną z możliwości jest:

1. Klient→ Serwer: ***M*1 = *H*[*H*(*N*)** [**XOR**](https://en.wikipedia.org/wiki/XOR) ***H*(*g*) | *H*(*I*) | *s* | *A* | *B* | *K*Klient].** Serwer weryfikuje ***M*1.**
2. Serwer → Klient: ***M*2 = *H*(*A* | *M*1 | *K*Serwer)**. Klient weryfikuje ***M*2**.

Alternatywa dla niej:

1. Klient→ Serwer: ***M*1 = *H(* | *A* | *B* | SKlient ).** Serwer weryfikuje ***M*1.**
2. Serwer → Klient: ***M*2 = *H*(*A* | *M*1 | SSerwer)**. Klient weryfikuje ***M*2**

Używając SRP do wyznaczenia dzielonego klucza K, który zostanie użyty natychmiast po wyznaczeniu weryfikacja *M*1  oraz *M2* może zostać pominięta.

Zabezpieczenia:

1. Klient przerwie transakcję, jeśli otrzyma **B=0(modN)** lub **u=0**
2. Serwer przerwie transakcję jeśli otrzyma **0=A(modN)**
3. Klient musi okazać **KKlient** lub **SKlient** **najpierw**. Jeśli serwer otrzyma niepoprawne dane przerywa transakcję bez wysyłania jego  **KSerwer** lub ***S*Serwe**

# 

# Oznaczenia

Poniższe oznaczenia i założenia będą realizować podczas tej realizacji protokołu SRP

* q - duża liczba pierwsza
* N = 2q + 1 - duża liczba pierwsza (co najmniej 1000 bajtów)
* Wszystkie obliczenia są wykonywane w obrębie pierścienia całkowitego modulo N
* g - generator grupy
* H() - funkcja haszująca, np. SHA-256
* k – parametr pochodzący z N i g: k = H(N, g). Używany do zapobiegania “2-for-1 guess”, gdy atakujący podaje się za serwer
* s - sól (losowy ciąg znaków)
* I - nazwa użytkownika
* p - hasło użytkownika
* v = gx - weryfikator hasła
* x = H(s, p)
* A i B – losowe, jednorazowe, nietrwałe klucze odpowiednio użytkownika i serwera

# Diagram UML

