Protokoły kryptograficzne
Uwierzytelnienie
Zadanie laboratoryjne 2a.
Zadanie laboratoryjne 2b.

12 kwietnia 2021

1 Zadania

Zadania:

1. Zaimplementować w środowisku *sagemath* protokół SKID3 (zadanie 2a.).
2. Zaimplementować w środowisku *sagemath* protokół Lamporta (zadanie 2b.).

2 Opis protokołów

2.1 Protokół SKID3

Założenia:

- Protokół wykorzystuje funkcję MAC ($HMAC$);
- Użytkownicy **A** i **B** korzystają wspólnie z tego samego klucza tajnego K.

Kroki protokołu:

1. użytkownik **A** wybiera liczbę losową $Rand_A$ i wysyła ją użytkownikowi **B**,
2. użytkownik **B** wybiera liczbę losową $Rand_B$ i wysyła ją do użytkownika **A** wraz z danymi:

$$Rand_B, HMAC_K(Rand_A, Rand_B, B) \ \mapsto \ \mathbf{A}$$

3. użytkownik **A** oblicza $HMAC_K(Rand_A, Rand_B, B)$ i porównuje wynik z wartością otrzymaną od **B**, jeśli wartości są identyczne, to użytkownik **A** ma pewność, że komunikuje się z użytkownikiem **B**, i wysyła do niego wartość $H_K(Rand_B, A)$,
4. użytkownik **B** oblicza $HMAC_K(Rand_B, A)$ i porównuje wynik z wartością otrzymaną od **A**, jeśli wartości są identyczne, użytkownik **B** ma pewność, że komunikuje się z **A**.

2.2 Protokół Lamporta

Założenia:

- Protokół wykorzystuje funkcję jednokierunkową do identyfikacji użytkownika (np. funkcję skrótu).
- Użytkownik musi zapamiętać i zachować w tajemnicy pewną wartość początkową x_0.
- Kolejne wartości x_k w kolejnych krokach będą powstawać jako wyniki działania funkcji jednokierunkowej z argumentem x_{k-1}.
- Użytkownik na początku musi wygenerować n iteracji (n – z ustalonego wcześniej zakresu) funkcji f.
- Następnie podaje wartość x_n systemowi.
- Przy pierwszym logowaniu wartość ta będzie funkcjonować jako identyfikator użytkownika, natomiast x_{n-1} jako hasło.

Kroki protokołu:

1. użytkownik przesyła aktualny identyfikator x_k,
2. system sprawdza istnienie użytkownika o otrzymanym identyfikatorze, po czym żąda podania hasła czyli x_{k-1},
3. użytkownik podaje hasło x_{k-1},
4. system weryfikuje poprawność hasła, sprawdzając, czy $f(x_{k-1}) = x_k$, jeśli tak, to zapamiętuje x_{k-1} jako identyfikator przy następnym logowaniu.

3 Wytyczne

3.1 Protokół SKID3

W ramach zadania należy zaimplementować procedury realizujące następujące elementy:

1. funkcję $HMAC$. W tym celu można wykorzystać dowolną z dostępnych funkcji *hmac*.
2. realizacja kolejnych kroków protokołu.

Parametrem wejściowym jest wspólny klucz tajny użytkowników **A** i **B**.

3.2 Protokół Lamporta

W ramach zadania należy zaimplementować procedury realizujące następujące elementy:

1. funkcję jednokierunkową. W tym celu można wykorzystać dowolną z dostępnych funkcji skrótu.
2. realizacja kolejnych kroków protokołu.

Parametrem wejściowym jest liczba iteracji n.

3.3 Polecenia środowiska *sagemath*

Biblioteki i polecenia środowiska *sagemath*, które można wykorzystać przy realizacji zadania:

- biblioteka *hashlib*;
- biblioteka *hmac*;
- biblioteka *binasci*.

W poniższych przykładach wykorzystana została funkcja skrótu $SHA3-512$. W implementacjach zadań można wykorzystać inną dostępną funkcję skrótu.
Przykład użycia:

```
sage: import hashlib
sage: import hmac
sage: import binascii
sage: # Generacja skrótu wiadomości tekstowej a
sage: d = hashlib.sha3_512()
sage: a = "String wejsciowy"
sage: d.update(a.encode())
sage: hash = d.hexdigest()
sage: print("Skrót:",hash)
Skrót: 1c8419d6fd9b6138a64469f44ba26269e494f9c8e24f4401108c0b871d9027ca956da0
       1d0c0eb31e5baefa01b9e454411aad6219aea6a170fd08e7074d3d264f
sage:
sage: # Generacja skrótu z liczby całkowitej a
sage: d = hashlib.sha3_512()
sage: a = 1234567890
sage: d.update(a.str().encode())
sage: hash = d.hexdigest()
sage: print("Skrót:",hash)
Skrót: 36dde7d288a2166a651d51ec6ded9e70e72cf6b366293d6f513c75393c57d6f33b9498
       79b9d5e7f7c21cd8c02ede75e74fc54ea15bd043b4df008533fc68ae69
sage:
sage: # Generacja wartości funkcji HMAC dla klucza key z liczby całkowitej a
sage: key = "1234567890ABCDEF"
sage: keyh = binascii.unhexlify(key)
sage: a = 1234567890
sage: hhmac= hmac.new(keyh,a.str().encode(),hashlib.sha3_512)
sage: hhmacout = hhmac.hexdigest()
sage: print("HMAC:",hhmacout)
HMAC: 0ed9b0160921fcab689a6c04118610629e569725e9ddb72ababfe004b245ab90cd36570
      101e7b97c755aeb42256b40f50b6be0f0330bc12b1e7126a44f572a2f
```

4 Rozliczenie zadań

W celu rozliczenia zadań należy przesłać w ramach narzędzia $MicrosoftTeams$ dla każdego z zaimplementowanych protokołów jeden plik tekstowy (koniecznie format txt) zawierający:

- kody zaimplementowanych funkcji;
- sekwencję wywołań funkcji realizujących kroki protokołu:

  - numer kroku z informacją, która strona go realizuje
  - jakie wartości zostały w danym kroku wyznaczone (z wypisaniem tych wartości)
  - co jest wysyłane do innych uczestników protokołu (również z wypisaniem wartości)

Na początku każdego z plików w komentarzu proszę umieścić własne dane (imię, nazwisko, grupa szkoleniowa). Zaimplementowane procedury jak i wykonywane kroki muszą byś opatrzone komentarzami (w kodzie źródłowym).

Wymagania dotyczące przesyłania rozwiązań

- Realizacja protokołu SKID3 rozliczana jest w ramach „Zadania 2a".
- Realizacja protokołu Lamporta rozliczana jest w ramach „Zadania 2b".